LÊ MINH TRÍ

TỦ SÁCH TIN HỌC
PHƯƠNG NAM

Oracle 8 Database

Oracle 8.05 Enterprise Edition For Windows NT

Oracle8 Database For Windows NT

Release 8.05
Enterprise
Edition

NHÀ XUẤT BẢN TRẺ &
CÔNG TY VĂN HÓA PHƯƠNG NAM
PHỐI HỢP THỰC HIỆN

LÊ MINH TRÍ

Oracle 8 Database

For Windows NT

Release 8.05
Enterprise
Edition

NHÀ XUẤT BẢN TRẺ

Lời Tựa

Tài liệu này mô tả tất cả các đặc trưng của Oracle server, một hệ thống quản lý cơ sở dữ liệu quan hệ đối tượng. Nó mô tả cách Oracle server thực hiện chức năng và đặt nền tảng khái niệm về nhiều thông tin thực tế trong các tài liệu hướng dẫn Oracle server khác.

Oracle8 và Oracle8 Enterprise Edition

Oracle8 Concepts gồm những thông tin mô tả các đặc trưng và tính năng của các sản phẩm Oracle8 và Oracle8 Enterprise Edition. Oracle8 và Oracle8 Enterprise Edition có các đặc trưng cơ bản. Tuy nhiên, nhiều đặc trưng cao cấp chỉ có thể có trong Enterprise Edition, và một số trong các đặc trưng này được tùy chọn. Thí dụ, dùng tính năng đối tượng, bạn phải có Enterprise Edition và Objects Option.

Để có những thông tin về những khác biệt giữa Oracle8 và Oracle8 Enterprise Edition, các đặc trưng và các tùy chọn bạn có thể sử dụng, xem Getting to know Oracle và Oracle8 Enterprise Edition.

Oracle8 và Oracle8 Enterprise Edition

Server cơ sở dữ liệu Oracle đã thiết kế để đáp ứng các đòi hỏi của kỷ nguyên mạng máy tính. Để chắc chắn là Oracle server cơ sở dữ liệu thích hợp với cả các ứng dụng nhỏ, cục bộ và ở quy mô công ty, nó được đề nghị bằng hai cấu hình:

- Oracle8
- Oracle8 Enterprise Edition

Cả Oracle8 và Oracle8 Enterprise Edition đều cung cấp quản lý dữ liệu bảo mật và đáng tin cậy cho các ứng dụng ở quy mô từ các ứng dụng cục bộ nhỏ tới các hệ thống giao dịch trực tuyến dung lượng lớn hoặc các ứng dụng kho dữ liệu mở rộng (truy vấn mở rộng - query-intensive). Chúng còn cung cấp các công cụ cho

quản lý các hệ thống, tính năng động phân phối dữ liệu một cách hiệu quả và khả năng vô hướng cho hiệu suất tối ưu từ các tài nguyên máy tính.
Các hệ thống thông tin liên quan đến kiến trúc dựa trên mạng mới, chúng phải điều hoạt trong các môi trường đa nền, đa tầng. Kiến trúc máy tính mạng dựa trên các chuẩn mở của Oracle (NCA) cho phép các công ty mất ít thời gian để giải quyết công việc quản trị và giải pháp khai triển. Tại trung tâm NCA là Oracle8 và Oracle8 Enterprise Edition, thế hệ thứ hai của server dữ liệu Oracle.

Oracle8

Oracle8 có ý định bổ sung ít hơn và cung cấp tính năng dễ dùng và kinh tế hơn. Quan trọng nhất vì Oracle8 và Oracle8 Enterprise Edition dựa trên cùng mã, nó dễ di trú ứng dụng từ Oracle8 tới Oracle8 Enterprise Edition nếu bạn cần sản phẩm được xác lập đặc trưng có nhiều năng lực hơn và và tính năng vô hướng lớn hơn. Hơn nữa, Oracle8 và Oracle8 Enterprise Edition có thể cùng tồn tại trong cùng môi trường được phân phối và truyền thông thông suốt.

Oracle8 Enterprise Edition

Với nhiều đặc trưng đầu cuối cao, Oracle8 Enterprise Edition có ý định cho các hệ thống lớn hơn. Nó có thể hỗ trợ hàng ngàn người dùng, hầu như không có giới hạn nào về kiểu hoặc số lượng dữ liệu được lưu trữ. Oracle8 Enterprise Edition có thể được khai triển cho các ứng dụng phạm vi doanh nghiệp và cục bộ hoặc các đặc trưng mới có hiệu lực cao có thể được dùng để hỗ trợ các hoạt động 7x24x52. Nó còn hỗ trợ các ứng dụng dựa trên cả các mô hình quan hệ và quan hệ đối tượng.

Phần I 5
TỔNG QUAN VÀ CÀI ĐẶT

Chương 1 Tổng quan 9

Phần II 66
CÁC CẤU TRÚC CƠ SỞ DỮ LIỆU

Chương 2 Các khối dữ liệu, khối lưu riêng và các đoạn 67
Chương 3 Các không gian bảng và các tập tin dữ liệu 105
Chương 4 Từ điển dữ liệu 121

Phần III 131
MÔI TRƯỜNG LÀM VIỆC CỦA ORACLE

Chương 5 Cơ sở dữ liệu, khởi động
và đóng môi trường làm việc 132
Chương 6 Các cấu trúc bộ nhớ 145
Chương 7 Cấu trúc xử lý 170

Phần IV 209
DBMS QUAN HỆ ĐỐI TƯỢNG

Chương 8 Các đối tượng giản đồ 210
Chương 9 Các bảng phân hoạch và các chỉ mục 269
Chương 10 Các kiểu dữ liệu lập sẵn 332
Chương 11 Các kiểu dữ liệu người dùng xác định (tùy chọn các
đối tượng) 359
Chương 12 Sử dụng các kiểu dữ liệu người dùng xác định 378
Chương 13 Các khung nhìn đối tượng 399

Phần V 406
TRUY XUẤT DỮ LIỆU

Chương 14 SQL và PL/SQL 407

Chương 15 Quản lý giao dịch 430
Chương 16 Sắp hàng trước 440
Chương 17 Các thủ tục và các bộ phần mềm 453
Chương 18 Các bộ kích khởi cơ sở dữ liệu 473
Chương 19 Quản lý sự phụ thuộc Oracle 496
Chương 20 Bộ tối ưu hóa 511

Phần VI 615
PARALLEL SQL
VÀ DIRECT-LOAD INSERT

Chương 21 Direct-Load INSERT 615
Chương 22 Tiến hành song song 629

Phần VII 686
BẢO VỆ DỮ LIỆU

Chương 23 Tính đồng thời và nhất quán dữ liệu 686
Chương 24 Tính nguyên vẹn dữ liệu 738
Chương 25 Điều khiển truy xuất cơ sở dữ liệu 765
Chương 26 Các đặc quyền và các vai trò 786
Chương 27 Kiểm soát 807
Chương 28 Phục hồi cơ sở dữ liệu 824

Phần VIII 858
XỬ LÝ ĐƯỢC PHÂN PHỐI VÀ CÁC CƠ SỞ DỮ LIỆU
ĐƯỢC PHÂN PHỐI

Chương 29 Xử lý được phân phối 859
Chương 30 Các cơ sở dữ liệu được phân phối 866
Chương 31 Sao chép cơ sở dữ liệu 891

Phần I

TỔNG QUAN VÀ CÀI ĐẶT

Chương 1

Tổng quan

THUẬT NGỮ ĐƯỢC SỬ DỤNG TRONG SÁCH		
ARCHIVELOG		*Tập tin nhật ký tàng trữ*
Archiver (ARCH)	***Bộ tàng trữ***	
Archived Redo Log	***Sổ nhật ký tàng trữ***	*Các tập tin redo trực tuyến có thể được tàng trữ trước khi được sử dụng lại tùy chọn để tạo sổ nhật ký redo tàng trữ*
Auditing	***Kiểm soát***	*Trợ giúp kiểm tra sử dụng cơ sở dữ liệu nghi ngờ*
Automatic Locking	***Khóa tự động***	*Không đòi hỏi điều tác của người dùng*
Automatic Recovery	***Phục hồi tự động***	*Từ sự cố mạng hoặc hệ thống*
Background Process	***Xử lý nền***	

Basic Recovery Steps	*Các bước phục hồi căn bản*	
CHECK		*Vô hiệu hoá các giá trị không thoả mãn biểu thị logic của hạn định*
Checkpoint (CKPT)	*Điểm kiểm*	*Ghi các vùng đệm cơ sở dữ liệu bị sửa đổi vào các tập tin dữ liệu bằng DBWn*
Client		*Phần client là ứng dụng CSDL front-end (tiền tiêu) và tương tác với người dùng bằng bàn phím, màn hình và thiết bị trỏ, chẳng hạn như chuột*
Client/Server Architecture	Kiến trúc Client/Server	
Commit	Cam kết	
Compatibility	Tính tương thích	
Connectibility	Tính kết nối	Cho phép các kiểu máy tính và các hệ điều hành dùng chung thông tin trên mạng
Consistency	Nhất quán/nhất quán	Dữ liệu người dùng hiển thị hoặc người dùng không thay đổi được dữ liệu ấy cho tới khi người dùng được kết thúc với dữ liệu đó
Control File	Tập tin điều khiển	
Controlled availability	Tính khả dụng có điều khiển	
Data Access	Truy xuất dữ liệu	
Data Blocks	Khối dữ liệu	
Data Definition Statement (DDL)	Câu lệnh định nghĩa dữ liệu	Xác định, duy trì và bỏ các đối tượng giản đồ khi chúng không cần nữa và cho phép người dùng các đặc quyền hoặc quyền truy xuất cơ sở dữ liệu và các đối tượng cá biệt trong cơ sở dữ liệu
Data Dictionary	Từ điển dữ	Tập hợp các bảng và khung

	liệu	nhưn được sử dụng làm tham chiếu chỉ đọc về CSDL

Data Integrity	**Tính nguyên vẹn dữ liệu**	
Data Manipulation Statement (DML)	**Câu lệnh điều tác dữ liệu**	Câu lệnh điều tác dữ liệu của cơ sở dữ liệu (truy vấn, chèn, cập nhật và xóa các hàng của bảng)
Data Segment	**Đoạn dữ liệu**	
Database Backup	**Sao chép cơ sở dữ liệu**	
Database Buffer Cache	**Cache vùng cơ sở dữ liệu/vùng nhớ cung cấp dữ liệu**	
Database Link	**Liên kết cơ sở dữ liệu**	
Database name	**Tên cơ sở dữ liệu**	
Database Security	**Bảo mật cơ sở dữ liệu**	
Database Structure	**Cấu trúc cơ sở dữ liệu**	
Database Trigger	**Bộ kích khởi cơ sở dữ liệu**	Ghi các thủ tục được tiến hành tự động do chèn, cập nhật hoặc xóa khỏi một bảng
Database Writer (DBWn)	**Bộ ghi cơ sở dữ liệu**	Ghi các khối dữ liệu bị sửa đổi từ cache vùng đệm cơ sở dữ liệu vào các tập tin dữ liệu
Datafile	**Tập tin cơ sở dữ liệu**	Một trong những tập tin dữ liệu vật lý của cơ sở dữ liệu và các cấu trúc cơ sở dữ liệu, như các bảng và các chỉ mục là lưu trữ vật lý trong các tập tin dữ liệu được phân bổ cho cơ sở dữ liệu.
Default Tablespace	**Không gian bảng mặc định**	Một đặc trưng cung cấp cho Oracle những thông tin dữ định hướng sử dụng không gian trong các trường hợp mà

		đối tượng gian đồ không được ấn định

Dispatcher (Dnnn)	**Bộ phân công**	Tiến trình của bộ phân công được tạo cho mọi giao thức truyền thông sử dụng (D000, . . ., Dnnn)
Distributed Data Manipulation	**Điều tác dữ liệu được phân phối**	
Distributed Database	**Cơ sở dữ liệu được phân phối**	Mạng của những cơ sở dữ liệu được quản lý bởi nhiều server cơ sở dữ liệu thể hiện đối với người dùng như một CSDL logic đơn giản
Distributed system	**Hệ thống phân phối**	Đối với các môi trường được phân phối và nối mạng, Oracle kết hợp dữ liệu được định vị về mặt vật lý trên các máy tính khác nhau vào một cơ sở dữ liệu logic để tất cả những người dùng mạng có thể truy cập
Embedded SQL Statement	**Các câu lệnh SQL nhúng**	Phối hợp với DDL, DML và các câu lệnh điều khiển giao dịch trong chương trình ngôn ngữ thủ tục (chẳng hạn như được sử dụng với các Oracle Pecompiler)
Enterprise		Quy mô lớn
Extents	**Khối lưu riêng**	Số hiệu xác định của các khối dữ liệu liên tục; cấp kế tiếp của không gian cơ sở dữ liệu logic, được sử dụng để lưu trữ những thông tin riêng biệt
FOREIGN KEY		Đòi hỏi mỗi giá trị trong cột hoặc tập hợp các cột so khớp giá trị trong các UNIQUE hoặc FOREIGN KEY có liên quan
Foreign key	**Khóa ngoại**	Cột hoặc tập hợp các cột trong định nghĩa của hạn định tính nguyên vẹn tham

		chiều

Hash clusters	**Các cụm liên kết trị số**	Một trong các cấu trúc tùy chọn được kết hợp với các bảng, có thể được tạo để tăng hiệu suất truy hồi dữ liệu
High availability	**Tính khả dụng cao**	Ở một số vùng, Oracle làm việc 24/24 mà không bị giới hạn vào lưu lượng truyền cơ sở dữ liệu. Các hoạt động bình thường như sao lưu CSDL và các sự cố hệ thống máy tính cục bộ không gây cản trở sử dụng cơ sở dữ liệu
Index Segment	**Đoạn chỉ mục**	Mỗi chỉ mục có một đoạn chỉ mục để lưu trữ tất cả dữ liệu của nó
Instance failure	**Sự cố môi trường làm việc**	
Integrity	**Tính nguyên vẹn dữ liệu**	Bảo đảm dữ liệu của CSDL và các cấu trúc phản ánh tất cả các thay đổi xảy ra đối với chúng theo chuỗi chính xác
Integrity Constraints	**Các hạn định tính nguyên vẹn dữ liệu**	Cách khai báo để xác định nguyên tắc hoạt động đối với một cột của bảng
integrity rules	**Các nguyên tắc tính nguyên vẹn**	Các luật chi phối những điều hoạt được cho phép về dữ liệu và các cấu trúc của CSDL và bảo vệ chúng
Job Queue (SNPn)	**Hàng đợi tác vụ**	Trong cấu hình CSDL được phân phối, tối đa 36 tiến trình hàng đợi tác vụ (SNP0, ..., SNP9, SNPA, ..., SNPZ) có thể tự động tô bổ các ảnh chụp nhanh bảng
Key	**Khóa**	Cột hoặc tập hợp các cột trong định nghĩa về các kiểu hạn định tính nguyên vẹn nào đó. Khóa mô tả những mối quan hệ giữa các bảng khác

		nhau và các cột của CSDL quan hệ
Listener		Tiến trình chờ yêu cầu dữ liệu Oracle đến từ mạng
Lock (LCKn)	**Khóa**	Các tiến trình khóa (LCK0, ..LCK9) được sử dụng để khóa liên môi trường làm việc trong Oracle Parallel Server
Log Writer (LGWR)	**Bộ ghi sổ nhật ký**	Ghi các mục nhập sổ nhật ký hoạt động vào đĩa
Logical Database Structure	**Cấu trúc CSDL logic**	
Manageable security	**Bảo mật quản lý được**	
Manual Locking	**Khóa thủ công**	Người dùng có thể chạy để khóa mặc định
Media (disk) failure	**Sự cố môi trường (đĩa)**	
Memory Structure	**Cấu trúc bộ nhớ**	
Method	**Phương pháp**	Thủ tục hoặc hàm là thành phần của định nghĩa của kiểu dữ liệu người dùng xác định (kiểu đối tượng, bảng lồng nhau hoặc mảng khả biến)
Multiplexed Control File	**Tập tin điều khiển đa cộng**	
NOARCHIVELOG		Các tập tin sổ nhật ký hoạt động đầy chưa được tàng trữ
NOT NULL		Các mục nhập rỗng trong một cột của bảng
Online Redo Log	**Sổ nhật ký hoạt động trực tuyến**	Tập hợp gồm hai hoặc nhiều tập tin sổ nhật ký hoạt động trực tuyến ghi tất cả những thay đổi đã cam kết xảy ra đối với CSDL
Open and Closed Databases	**Các CSDL mở và đóng**	CSDL mở (truy xuất được) và CSDL đóng (không truy xuất được)
Operation	**Điều hoạt**	Các hoạt động được xác định

		rõ ràng cho phép người dùng điều tác dữ liệu và các cấu trúc CSDL

Optional Manual Override Capability	**Khả năng chạy đè thủ công tùy chọn**	Tùy chọn cho phép người điều hành CSDL giải phóng nhưng tài nguyên bị khóa có thể bị lưu giữ không rõ ràng do sai sót trong thời gian dài
Oracle Data Block	**Khối dữ liệu Oracle**	Ở mức độ chi tiết nhất, một dữ liệu của CSDL Oracle được lưu trữ trong các khối dữ liệu. Kích thước khối dữ liệu được ấn định cho mỗi CSDL Oracle khi CSDL ấy được tạo
Oracle Databases	**CSDL Oracle**	Tập hợp dữ liệu được coi là một đơn vị; mục đích chung của CSDL là lưu trữ và truy hồi hệ thống liên quan
Oracle Instance	**Môi trường làm việc Oracle**	
Oracle Process Architecture	**Kiến trúc xử lý Oracle**	
Oracle Server		Hệ thống quản lý CSDL quan hệ đối tượng cung cấp cách giải quyết tích hợp và toàn diện có thể truy xuất, gồm CSDL Oracle và môi trường làm việc Oracle server
Package	**Bộ phần mềm/bao gói**	Bộ phần mềm, bao gói là phương pháp gói riêng và lưu trữ các biến, các hàm, các thủ tục có liên quan và các cấu trúc bao gói khác cấu tạo cùng với nhau thành một đơn vị trong cơ sở dữ liệu
Partial Backup	**Sao lưu từng phần**	
Physical Database Structure	**Cấu trúc CSDL vật lý**	
PL/SQL		Phần mở rộng ngôn ngữ thủ tục của Oracle với SQL, kết

		hợp tính linh hoạt và thuận tiện của SQL với tính năng thủ tục của ngôn ngữ lập trình có cấu trúc, chẳng hạn như IF ... THEN, WHILE và LOOP
Portability	**Tính khả chuyển**	
PRIMARY KEY		Vô hiệu hóa những giá trị sao chép và các giá trị rỗng trong một cột hoặc tập hợp các cột
Primary key	**Khóa chính**	Cột hoặc tập hợp các cột có trong định nghĩa của hạn định PRIMARY KEY của bảng. Chỉ một khóa chính có thể xác định trên mỗi bảng
Privilege auditing	**Kiểm soát đặc quyền**	Kiểm soát sử dụng các đặc quyền hệ thống đầy năng lực bất kể các đối tượng giản đồ có tên cụ thể
Privilege	**Đặc quyền**	Quyền tiến hành một kiểu câu lệnh SQL cá biệt
Procedures and Functions	**Các thủ tục và các hàm**	Gồm tập hợp các câu lệnh SQL và PL/SQL được tạo nhóm với nhau bằng một đơn vị để giải quyết một vấn đề cá biệt hoặc thực hiện một tập hợp những tác vụ có liên quan
Process Architecture	**Kiến trúc xử lý**	Tiến trình là "xâu điều khiển" hoặc cơ chế trong một hệ điều hành có thể tiến hành một loạt các bước, nó thường có vùng nhớ riêng để tiến hành
Process Monitor (PMON)	**Bộ giám sát tiến trình**	Thực hiện phục hồi khi tiến trình người dùng không thực hiện được. PMON chịu trách nhiệm làm sạch cache và giải phóng những tài nguyên mà tiến trình ấy đang sử dụng. Nó cũng kiểm tra bộ phận công và các tiến trình server để khởi động lại nếu chúng không thực hiện được

Profiles and Resource Limit	**Các lược trình và giao diện tài nguyên**	
Program Global Area (PGA)	**Vùng toàn cục chương trình**	Vùng đệm bộ nhớ chứa dữ liệu và thông tin điều khiển cho một tiến trình server. PGA được Oracle tạo khi tiến trình server khởi động.
Program Interface	**Giao diện chương trình**	Cơ chế tiến trình người dùng truyền thông với tiến trình server, đáp ứng một phương pháp truyền thông tiêu chuẩn giữa công cụ client hoặc ứng dụng bất kỳ (như các Oracle Form) và phần mềm Oracle
Queue Monitor (QMNn)	**Bộ giám sát hàng đợi**	Xử lý nền tùy chọn giám sát các hàng đợi thông điệp cho Oracle Advanced Queuing; có thể được cấu hình tối đa 10 tiến trình bộ giám sát hàng đợi
Read Consistency	**Đọc nhất quán/đọc nhất quán**	
Recoverer (RECO)	**Bộ phục hồi**	Được sử dụng để giải quyết các giao dịch được phân phối phải chờ do mạng hoặc hệ thống bị sự cố
Redo Log	**Sổ nhật ký hoạt động**	Được mô tả trong "Tập tin sổ nhật ký hoạt động" trên trang 1-12, sổ nhật ký hoạt động là tập hợp những tập tin bảo vệ dữ liệu của CSDL bị thay đổi trong bộ nhớ chưa được ghi vào những tập tin dữ liệu (gồm sổ nhật ký hoạt động trực tuyến và sổ nhật ký tàng trữ)
Redo Log Buffer	**Vùng đệm nhật ký hoạt động**	Thuộc vùng toàn cục hệ thống lưu trữ các mục nhập nhật ký hoạt động - nhật ký ghi những thay đổi xảy ra đối với

		CSDL. Các mục nhập nhật ký được lưu trữ trong các vùng đệm sơ nhật ký hoạt động được ghi vào một tập tin sơ nhật ký hoạt động trực tuyến, được sử dụng nếu cần phục hồi CSDL. Nó có kích thước tĩnh
Redo Log File	**Tập tin sổ nhật ký hoạt động**	Ghi lại tất cả những thay đổi xảy ra đối với dữ liệu
Referenced key	**Khóa tham chiếu**	Khoá duy nhất hoặc khóa chính của cùng bảng hoặc khác bảng được khóa ngoại tham chiếu
Replicated environments	**Môi trường sao chép**	
Resource limits	**Các giao diện tài nguyên**	
Role	**Vai trò**	Các nhóm có tên gồm các đặc quyền liên quan được trao cho người dùng hoặc vai trò khác, để dễ dàng quản lý đặc quyền có kiểm soát
Rollback Segment	**Đoạn hồi phục về trạng thái cũ**	Đoạn tạm thời lưu trữ những thông tin "hoàn đổi" được người điều hành CSDL tạo
Rolling Back	**Hồi phục về trạng thái cũ (trước)**	Hồi phục về sau chỉ được một nửa phục hồi dữ liệu. Sau khi hồi phục về sau, bất kỳ thay đổi không được cam kết phải được hoàn đổi lại. Sau khi các tập tin sơ nhật ký hoạt động được áp dụng, thì các đoạn hồi phục được sử dụng để nhận dạng và hoàn đổi các giao dịch đã chưa được cam kết, nhưng đã được ghi vào sổ nhật ký hoạt động. Tiến trình này được gọi là hồi phục về trước (về trạng thái cũ) Oracle thực hiện bước này tự động

Rolling Forward	**Hồi phục về sau**	Bước đầu tiên của phục hồi dữ liệu là hồi phục về sau, nghĩa là áp dụng lại tất cả những thay đổi được ghi trong sổ nhật ký hoạt động vào những tập tin dữ liệu. Hồi phục về sau tiến hành tối đa những tập tin sổ nhật ký hoạt động cần thiết để đưa các tập tin dữ liệu tới thời điểm đòi hỏi
Savepoint	**Điểm lưu**	Có thể được sử dụng để chia một giao dịch thành các phần nhỏ.
Schema object auditing	**Kiểm soát đối tượng giản đồ**	Kiểm soát các truy xuất những đối tượng giản đồ riêng biệt bất kể người dùng nào
Schema Object Privileges	**Đặc quyền đối tượng giản đồ**	Cho phép người dùng thực hiện một hoạt động cá biệt về một đối tượng giản đồ riêng biệt
Security Domain	**Vùng bảo mật**	Mỗi người dùng có một vùng bảo mật - một tập hợp những đặc tính xác định các vấn đề, chẳng hạn như các đặc quyền và các vai trò, các chỉ tiêu không gian bảng, các giới hạn tài nguyên hệ thống
Segment	**Đoạn**	Mức lưu trữ CSDL logic trên một khối lưu riêng
Sequences	**Chuỗi**	Phát sinh một danh sách nối tiếp các số hiệu duy nhất cho các cột bằng số của những bảng CSDL
Server Processes	**Xử lý/tiến trình server**	Xử lý những yêu cầu từ các tiến trình người dùng được kết nối
Session Control Statement	**Câu lệnh điều khiển phiên**	Cho phép người dùng điều khiển các đặc tính của phiên hiện hành của người dùng ấy, gồm cho phép và vô hiệu hóa các vai trò và thay đổi các xác lập ngôn ngữ. Hai câu lệnh hoạt động phiên là ALTER SESSION và SET ROLE

Shared Pool	**Vùng chia dùng chung**	Một phần của vùng toàn cục hệ thống có các cấu trúc bộ nhớ dùng chung, chẳng hạn như các vùng SQL dùng chung
Site Autonomy	**Tự trị vùng**	Mỗi CSDL tham gia vào một CSDL phân phối được điều hành tách riêng và độc lập với các CSDL khác
SQL Statement	**Câu lệnh SQL**	Tất cả các điều hoạt về thông tin trong CSDL Oracle được thực hiện bằng các câu lệnh SQL
Statement auditing	**Kiểm soát câu lệnh**	Kiểm soát các câu lệnh SQL riêng biệt bất kể các đối tượng giản đồ có tên cụ thể
Statement Handle or Cursor	**Trị điều quản câu lệnh hoặc con trỏ**	Con trỏ là một trị điều quản (một tên hoặc một điểm trỏ) đối với bộ nhớ được kết hợp với một câu lệnh riêng biệt
Storage Settings and Quotas	**Các xác lập bộ nhớ và các chỉ tiêu**	Định hướng và giới hạn sử dụng không gian đĩa được phân bổ cho CSDL trên mỗi phương thức người dùng gồm các không gian bảng, các chỉ tiêu không gian bảng tạm thời và mặc định
Structured Query Language (SQL)	**Ngôn ngữ truy vấn có cấu trúc**	Ngôn ngữ lập trình xác định và điều tác CSDL, các cơ sở dữ liệu SQL là các CSDL quan hệ
Structures	**Cấu trúc**	Đối tượng được xác định rõ ràng (chẳng hạn như các bảng, các khung nhìn, các chỉ mục, vân vân)
Synonyms	**Từ đồng nghĩa**	Một bí danh cho một bảng, khung nhìn, chuỗi hoặc một đơn vị chương trình
System Control Statement	**Câu lệnh điều khiển hệ thống**	Các lệnh điều khiển hệ thống thay đổi các tính chất của môi trường làm việc Oracle server. Lệnh điều khiển hệ thống duy nhất là ATLTER SYSTEM
System Global Area	**Vùng toàn**	Vùng nhớ dùng chung có dữ liệu và thông tin điều khiển

(SGA)	cục hệ thống	cho một môi trường làm việc Oracle

System Monitor (SMON)	**Bộ giám sát hệ thống**	
System Privileges	**Đặc quyền hệ thống**	Cho người dùng thực hiện một hoạt động cá biệt trên toàn bộ hệ thống hoặc hoạt động cá biệt trên một kiểu đối tượng giản đồ cá biệt
Table Replication	**Sao chép bảng**	
Tables	**Bảng**	Đơn vị cơ bản của bộ nhớ dữ liệu trong CSDL Oracle, các bảng CSDL lưu trữ tất cả dữ liệu người dùng truy xuất được
Tablespace Quota	**Chỉ tiêu không gian bảng**	Mức không gian bảng có thể sử dụng đối với các đối tượng trong một giản đồ
Tablespaces	**Không gian bảng**	CSDL được chia thành các đơn vị bộ nhớ logic được gọi là các không gian bảng, nó được sử dụng để tạo nhóm các cấu trúc logic có liên quan với nhau
Temporary Segment	**Đoạn tạm thời**	Đoạn được Oracle tạo khi câu lệnh SQL cần vùng hoạt động tạm thời để hoàn tất tiến trình
Temporary Tablespace	**Không gian bảng tạm thời**	
The Relational Model	**Mô hình quan hệ**	Gồm ba phương diện chính: cấu trúc, điều hoạt và các nguyên tắc tính nguyên vẹn dữ liệu
The Server		Phần máy chủ tiến hành phần mềm Oracle và điều quản các chức năng cần thiết cho truy xuất dữ liệu dùng chung và đồng thời
Transaction Control Statement	**Câu lệnh điều khiển**	Quản lý các thay đổi bởi các câu lệnh DML thực hiện

	giao dịch	
Transaction	**Giao dịch**	Đơn vị logic về một hoạt động gồm một hoặc nhiều câu lệnh SQL được một người dùng tiến hành
Trusted Oracle		Sản phẩm hệ thống quản lý CSDL bảo mật đa mức của công ty Oracle
Two-Phase Commit	**Cam kết hai pha**	Sự bảo đảm tính nhất quán dữ liệu ở môi trường được phân phối cũng như ở môi trường không được phân phối
Types of Failures	**Các kiểu sai sót**	Nhiều trường hợp có thể tạm ngưng điều hoạt của CSDL Oracle (lỗi người dùng, sai sót tiến trình và câu lệnh, sự cố môi trường làm việc, sự cố môi trường (đĩa))
UNIQUE		Vô hiệu hóa những giá trị sao chép trong một cột hoặc một tập hợp các cột
Unique key	**Khóa duy nhất**	Cột hoặc tập hợp các cột có trong định nghĩa của hạn định UNIQUE
User (Client) Processes	**Tiến trình người dùng (Client)**	Được tạo và được duy trì để tiến hành mã phần mềm của trình ứng dụng (chẳng hạn như chương trình Pro*C/C++) hoặc công cụ Oracle (như Oracle Enterprise Manager)
User error	**Lỗi người dùng**	Lỗi có thể đòi hỏi CSDL phải được phục hồi về thời điểm trước khi lỗi ấy xảy ra.
Valid username/password combinations	**Tổ hợp tên người dùng/mật khẩu hợp lệ**	Số lượng không gian đĩa các đối tượng giản đồ của người dùng có thể sử dụng, các giới hạn tài nguyên đối với người dùng
View	**Khung nhìn**	Sự trình bày đáp ứng tùy biến dữ liệu trong một hoặc nhiều bảng. Một khung nhìn cũng được coi là một "truy vấn định sẵn".

Whole Database Backup	**Sao chép CSDL tổng thể**	Sao chép tất cả các tập tin dữ liệu, các tập tin sổ nhật ký hoạt động và tập tin điều khiển cấu tạo một CSDL Oracle

Những đặc trưng và tùy chọn mới

Các chủ đề sau:

- **Các chỉ mục và các bảng phân hoạch.**
- **Hiệu suất kho dữ liệu cải tiến.**
- **Các cải tiến đối với các ứng dụng OLTP**
- **Sao chép**
- **Công nghệ quan hệ đối tượng**
- **Di trú và liên tính năng điều hoạt**
- **Các cải tiến khác.**

Các chỉ mục và các bảng phân hoạch

Các cơ sở dữ liệu rất lớn thể hiện các thử thách quan trọng cho các người điều hành và các Developer ứng dụng. Khối lượng dữ liệu lớn làm phức tạp các tác vụ quản trị và ảnh hưởng đến hiệu lực của cơ sở dữ liệu.

Để cải tiến hiệu lực, tạo thuận tiện cho việc quản trị và tăng cường truy vấn và hiệu suất DML, Oracle8 Enterprise Edition cho phép các bảng và các chỉ mục được phân hoạch hoặc tách ra thành các phần nhỏ hơn trên một phạm vi của các giá trị quan trọng. Bởi vì các phân hoạch điều hoạt động lập với nhau, dữ liệu trong bảng phân hoạch có thể sử dụng dù một hoặc nhiều phân hoạch không thể sử dụng. Các phân hoạch còn lập các bảng lớn để quản lý bằng cách tách các điều hoạt quản trị thành các tác vụ nhỏ hơn mà nó có thể được thực hiện song song. Cuối cùng, phân hoạch một bảng hoặc chỉ mục có thể tăng hiệu suất

các điều hoạt dữ liệu bằng cách loại trừ các phân hoạch không cần thiết từ kế hoạch điều hoạt khả thi.
Phân hoạch thông suốt tới cả các ứng dụng và người dùng, vì vậy các câu lệnh SQL chuẩn trong các ứng dụng có sẵn có thể tiến hành dựa trên các bảng đã phân hoạch. Tác nhân tối ưu hóa Oracle là nhận biết phân hoạch và các phân hoạch không chứa bất kỳ dữ liệu nào được một truy vấn đòi hỏi loại trừ khỏi tìm kiếm, thường làm gia tăng hiệu suất đáng kể.

Phân hoạch như thế nào ?

Một bảng hoặc chỉ mục có thể được phân hoạch và chia thành các phần nhỏ hơn. Bạn xác định một chiến lược phân hoạch bảng hoặc chỉ mục khi tạo cấu trúc ấy. Lựa cột hoặc tập hợp các cột đóng vai trò như một khóa phân hoạch và khóa này sẽ quyết định dữ liệu nào được sắp xếp vào mỗi phân hoạch. Dữ liệu từ các điều hoạt chèn vào tự động sắp xếp vào các phân hoạch thích hợp, vì vậy các ứng dụng không cần ghi đè để khai thác phân hoạch.

Tất cả các điều hoạt bạn thực hiện trên các bảng thường xuyên có thể được thực hiện trên các phân hoạch cá nhân của một bảng đã phân hoạch. Thí dụ, bạn có thể Export hoặc sao lưu các phân hoạch đơn giản của một bảng để tránh Export hoặc sao lưu toàn bộ bảng trong một điều hoạt. Bạn cũng có thể thực hiện lệnh ANALYZE trên mỗi phân hoạch hiện hành để tính toán các thống kê cần thiết để tối ưu hóa dựa trên chi phí nhanh chóng hơn.

Mỗi phân hoạch của một bảng đáng phân hoạch hoặc chỉ mục điều tác độc lập. Do đó, các điều hoạt trên một phân hoạch không bị ảnh hưởng bởi tính hiệu lực của các phân hoạch khác. Nếu một phân hoạch trở thành không còn hiệu lực vì một đĩa bị hư hoặc các điều hoạt quản trị, cả các hoạt động truy vấn và DML trên dữ liệu trong các phân hoạch khác có thể vẫn tiếp tục.

Điều khiển dữ liệu nhiều hơn

Người điều hành có thể xác định các thuộc tính bộ nhớ cho mỗi phân hoạch và bố trí phân hoạch trong hệ thống tập tin server để tăng mức độ chi tiết điều khiển đối với các cơ sở dữ liệu rất lớn. Các phân hoạch có thể được thực hiện ngoại tuyến riêng biệt, sao lưu, phục hồi, Export và Import, nạp theo thời gian giới hạn cần thiết đối với các điều hoạt quản lý. Phân hoạch chỉ mục riêng biệt có thể được thiết lập cho một phân hoạch bảng để làm giảm thời gian cần thiết đối với các điều hoạt bảo trì chỉ mục. Các điều hoạt phân hoạch có thể được thực hiện song song. Các phân hoạch làm tăng hiệu lực bằng cách cô lập môi trường và các ứng dụng bị sai sót không đòi hỏi dữ liệu từ phân hoạch có thể truy xuất để chạy dữ liệu không nén.

Quản trị thuận lợi

Oracle8 Enterprise Edition cung cấp một tập hợp các lệnh quản lý dễ dùng cho các phân hoạch. Thí dụ, bạn có thể bổ sung bảng chứa cửa sổ thời gian trôi qua của dữ liệu với một bảng phân hoạch để chỉ dữ liệu có giá trị một năm được chứa trong bảng đó. Để thực hiện điều này, chỉ cần bổ sung các phân hoạch mới vào cuối bảng và loại các phân hoạch chứa dữ liệu hơn một năm. Bạn còn có thể tách và hợp nhất các phân hoạch để cân bằng dữ liệu trong các phân hoạch. Ngoài ra, các bảng độc lập sử dụng phiên bản Oracle7 7.3 UNION ALL để xem có thể được chuyên đổi thành các phân hoạch của một bảng đơn giản. Điều này cho phép người dùng các cơ sở dữ liệu Oracle7 databases rất lớn nhanh chóng hơn so với phân hoạch Oracle8 Enterprise Edition mà không phải thiết lập lại các bảng lớn.

Hiệu suất kho dữ liệu cải tiến

Oracle8 Enterprise Edition đưa ra các đặc trưng mới cải tiến hiệu suất kho dữ liệu:

- Xử lý Star-Query tăng cường
- Các điều hoạt song song mới.
- Kích thước dữ liệu gia tăng

Xử lý Star-Query tăng cường

Oracle8 Enterprise Edition đưa ra các cải tiến hiệu suất đối với việc xử lý các star queries (các truy vấn sao) thường trong các ứng dụng kho dữ liệu. Truy vấn sao hoặc giản đồ sao xảy ra khi một hoặc nhiều bảng rất lớn thường được gọi là các bảng sự thật có các quan hệ với nhiều bảng nhỏ hơn, được gọi là các bảng kích thước. Oracle7 đã đưa ra tính năng tối ưu hóa truy vấn sao để cung cấp các cải tiến hiệu suất cho các loại truy vấn này. Tuy nhiên, trong Oracle8 Enterprise Edition, xử lý truy vấn sao đã được cải tiến để cung cấp sự tối ưu hóa hơn cả đối với các truy vấn sao.

Trong Oracle8 Enterprise Edition, một phương pháp mới đối với các truy vấn sao khả thi đã được đưa ra. Bằng sử dụng thuật toán có hiệu quả hơn và sử dụng các chỉ mục bitmap, xử lý truy vấn sao mới cung cấp hiệu suất tăng đáng kể so với các ứng dụng kho dữ liệu.
Oracle8 Enterprise Edition có hiệu suất cao nhất với nhiều loại truy vấn sao, gồm các giản đồ sao với bảng sự thật "rải rác" nơi tiêu chuẩn loại trừ một số lớn các hàng của bảng sự thật. Ngoài ra, khi một giản đồ có nhiều bảng sự thật, tác nhân tối ưu hóa xử lý truy vấn có hiệu quả. Cuối cùng, Oracle8 Enterprise Edition có thể xử lý các truy vấn sao một cách hiệu quả với các bảng nhiều kích thước hoặc lớn, các bảng không bị hạn định kích thước và các bảng có kích thước với kiểu giản đồ "bông tuyết".

Thuật toán tối ưu hóa truy vấn sao của Oracle8 Enterprise Edition không giống như Oracle7, không tạo ra bất kỳ liên kết tích Đề-các. Bây giờ các truy vấn sao được xử lý trong hai giai đoạn cơ bản. Thứ nhất, Oracle8 Enterprise Edition truy

hồi các hàng cần thiết chính xác từ bảng sự thật. Sự truy hồi này được thực hiện bằng các chỉ mục bitmap rất có hiệu quả. Giai đoạn hai liên kết tập hợp kết quả này từ bảng sự thật ấy với các bảng có kích thước thích hợp. Điều này cho phép các sự tối ưu hóa tốt hơn gồm nhiều truy vấn sao phức tạp, chẳng hạn như nhiều bảng sự thật. Thuật toán mới đó sử dụng các chỉ mục bitmap để tiết kiệm bộ nhớ đáng kể hơn các phương pháp trước đây cần phải có các chỉ mục B-tree ràng buộc với nhau. Thuật toán mới còn thực hiện song song hoàn toàn, kể cả tác vụ duyệt chỉ mục song song trên cả các bảng đã phân hoạch hoặc chưa phân hoạch.

Các điều hoạt song song mới

Các điều tác chèn, cập nhật và xóa bây giờ có thể chạy song song trong Oracle8 Enterprise Edition. Các điều tác này được coi là DML song song, được tiến hành song song qua đa xử lý. Nhờ có những điều tác song song này, câu lệnh sẽ được hoàn tất nhanh hơn nhiều hơn nếu cùng câu lệnh ấy được thực hiện theo kiểu nối tiếp. DML song song bổ sung truy vấn song song bằng cách cung cấp sự tiến hành giao dịch song song cũng như các truy vấn. DML song song hữu dụng cho hỗ trợ quyết định (DSS) hoặc môi trường kho dữ liệu, nơi các điều tác DML khối thông thường. Tuy nhiên, các điều tác DML song song cũng có thể tăng tốc độ các tác vụ lô trong cơ sở dữ liệu OLTP.

Oracle8 Enterprise Edition hỗ trợ các điều tác chèn, cập nhật và xóa vào các bảng phân hoạch. Nó còn hỗ trợ các điều tác chèn song song vào các bảng chưa phân hoạch. Điều tác chèn song song trên bảng chưa phân hoạc giống với điều tác nạp đường dẫn trực tiếp có thể sử dụng trong Oracle7. Điều đó làm cải tiến hiệu suất bằng cách định dạng và ghi các khối đĩa trực tiếp vào các tập tin dữ liệu, bỏ qua cache bộ đệm và các tắc nghẽn cổ chai quản lý không gian. Trong trường hợp này, mỗi xử lý chèn dữ

liệu song song vào một đoạn trên mức cao của bảng. Sau khi giao dịch cam kết, mức cao được chuyển ra ngoài các đoạn mới.

Để sử dụng DML song song, phải cho phép thực hiện các điều tác chèn, cập nhật hoặc xóa trước. Thông thường, các điều tác DML song song được thực hiện trong các chương trình lô hoặc trong một ứng dụng thực hiện điều tác, cập nhật, chèn hoặc xóa khối. Các hướng dẫn mới có thể sử dụng để xác định quan hệ song song của các câu lệnh DML.

Để có thêm thông tin, xem Oracle8 Parallel Server Concepts và Administration.

Kích thước cơ sơ dữ liệu gia tăng

Oracle8 Enterprise Edition có thể quản lý các cơ sở dữ liệu của hàng trăm terabyte về kích thước nhờ phân hoạch, các cải tiến quản trị và các tăng cường bên trong. Nhiều giới hạn kích thước trong các phiên bản ban đầu đã được tăng lên, chẳng hạn như số cột trong mỗi bảng, kích thước cơ sở dữ liệu tối đa và số tập tin trong mỗi cơ sở dữ liệu.

Các cải tiến cho các ứng dụng OLTP

Đòi hỏi lợi ích của các ứng dụng OLTP từ một số các đặc trưng mới cải tiến tính năng vô hướng, hiệu suất và tính năng quản lý.

Sao lưu mở rộng / hệ thống con phục hồi

Phục hồi và sao lưu được server quản lý của Oracle8 và Oracle8 Enterprise Edition cung cấp tính năng phục hồi và sao lưu được tích hợp trong Oracle server. Thông tin chi tiết được duy trì khi các sao lưu được thực hiện, chính xác những phần nào của cơ sở dữ liệu được sao lưu và nơi các tập tin được lưu trữ. Nếu cần phục hồi, Recovery Manager phân tích trạng thái cơ sở dữ liệu và xác định các điều tác cần thiết để sửa chữa cơ sở dữ liệu. Sau đó nó

tự động thực hiện các điều tác này, đơn giản hóa việc phục hồi cho các người điều hành và giảm lỗi có thể do con người gây ra. Giao diện GUI đơn giản trong Oracle Enterprise Manager điều khiển sao lưu và phục hồi. Một API còn có thể dùng cho các bên thứ ba có thể muốn cung cấp giao diện thay thế. Các giao diện tầng quản lý môi trường đối với các sản phẩm quản lý băng từ bên thứ ba có thể sử dụng. Legato Storage Manager được cung cấp với cả Oracle8 và Oracle8 Enterprise Edition.

Oracle8 Enterprise Edition cung cấp nhiều mức, các sao lưu gia tăng làm giảm kích thước sao lưu rất nhiều, chính vì làm thay đổi các khối được sao lưu. Điều này làm giảm đáng kể thời gian cần thiết để sao lưu một tập tin dữ liệu.

Phục hồi điểm kịp thời không gian bảng cho phép một hoặc nhiều không gian bảng được phục hồi so với thời trước đây, trong khi phần còn lại của cơ sở dữ liệu tiếp tục chạy. Điều này cho phép nhiều kiểu lỗi người dùng được điều chỉnh dễ dàng. Thí dụ, nếu một người dùng chạy một tác vụ lô cập nhật không chính xác nhiều bản ghi trong một bảng, thì bảng ấy có thể được phục hồi thành tác vụ lô thời gian trước. Ngoài ra, nếu một bảng đột ngột giảm xuống hoặc bị xóa bỏ, nó có thể được phục hồi về thời điểm trước điều hoạt này.

Xem thêm Sao lưu Oracle8 và Hướng dẫn phục hồi.

Các ứng dụng đáp ứng số lượng người dùng rất lớn

Rất nhiều cải tiến trong các Oracle8 server, Oracle8 Enterprise Edition và Net8 tăng khả năng sử dụng hệ điều hành và các tài nguyên mạng. Phân chia kết nối tạm thời giảm kết nối vật lý cho những người dùng rảnh rỗi (và thiết lập lại toàn bộ kết nối khi cần), do đó làm tăng số người dùng có thể được hỗ trợ. Oracle Connection Manager có thể được dùng để cấu hình một tầng trung gian để quản lý các kết nối có số lượng người dùng rất lớn.

Bằng cách cấu hình nhiều trình quản lý kết nối, Oracle8 Enterprise Edition có thể hỗ trợ nhiều người dùng hiện hành. Cơ sở dữ liệu dùng chung liên kết những người dùng đa công vào server cơ sở dữ liệu bằng một kết nối đơn, làm giảm đòi hỏi tài nguyên, nhất là đối với các kiến trúc ứng dụng đa tầng.

Sắp hàng nâng cao

Đặc trưng sắp hàng nâng cao của Oracle8 Enterprise Edition hỗ trợ gửi thông điệp cơ sở dữ liệu hoặc sắp hàng qua một tập hợp các bảng sắp hàng và các chức năng hàng đợi.

Sắp hàng nâng cao bổ sung hỗ trợ trực tiếp trong cơ sở dữ liệu để trì hoãn sự thực hiện giao dịch vào thời gian sau và thực hiện các giao dịch theo thứ tự đặc biệt. Khả năng này cho phép bạn tăng gấp đôi các ứng dụng phân phối và loại trừ sự phụ thuộc vào các hệ thống nội trú đối với các ứng dụng đòi hỏi tính năng vô hướng cao. Các điều hoạt sắp hàng đợi và tháo hàng đợi có thể được dùng để chuyển đổi xử lý từ trong một giao dịch thành một xử lý nền, do đó làm tăng thời gian trả lời giao dịch. Ngoài ra, sắp hàng đợi có thể được dùng để bổ sung các ứng dụng luồng công việc di chuyển dữ liệu tới một hệ thống khi trạng thái dữ liệu thay đổi. Thí dụ, các thứ tự di chuyển từ một ứng dụng mục nhập có thứ tự tới một ứng dụng truyền tải và sau đó tới hệ thống tính phí trong thời gian hoạt động của một thứ tự. Sắp hàng đợi nâng cao còn hoạt động cùng với các hệ thống sắp hàng giám sát TP chung.

Những thông điệp có thể được sắp hàng đợi và tháo hàng đợi bằng cách ứng dụng hoặc các hàng đợi khác. Đặc trưng truyền đó cho phép truyền thông với nhau mà không phải kết nối với cùng cơ sở dữ liệu hoặc cùng hàng đợi. Sử dụng các liên kết cơ sở dữ liệu tương tự và Net8, các thông điệp có thể được truyền từ một hàng đợi tới một hàng đợi khác, bất kể các thông điệp này là cục bộ hoặc từ xa.

Xem thêm Hướng Dẫn Cua Developer Ứng Dụng Oracle8

Các cải tiến server song song

Rất nhiều cải tiến trong Oracle Parallel Server tăng hiệu suất, tính năng vô hướng, cách dùng bộ nhớ và tính hiệu lực. Trình quản lý khóa phân phối tích hợp và chúng thay thế các trình quản lý khóa được các nhà cung cấp hệ điều hành khác nhau cung cấp để cải tiến hiệu suất và tính khả chuyển trên hầu hết các nền. Nhiều cải tiến hiệu suất tạo cho Oracle Parallel Server nhanh hơn đáng kể. Hơn nữa bảng V$ toàn cục mới cải tiến tính năng quản lý.

Hiện này Oracle Parallel Server dùng trình quản lý khóa phân phối tích hợp riêng (DLM) để xử lý các yêu cầu xử lý các tài nguyên liên nút. Trước đây, Oracle dựa vào các nhà cung cấp hệ điều hành để cung cấp khả năng DLM cho Parallel Sever chạy trên cho trước. Tích hợp DLM trong Oracle8 Enterprise Edition cho phép Oracle Parallel Server có thể dùng trên các nền trước đây không hỗ trợ dùng các tài nguyên nút chéo.

Oracle8 Enterprise Edition đưa ra nhiều cải tiến về hiệu suất Oracle Parallel Server. Thế hệ số hiệu thay đổi hệ thống (SCN) hiện này được tối ưu hóa để cung cấp các SCN bằng cửa sổ môi trường làm việc có hiệu quả. Riêng cải tiến này có thể tăng lên 10 tới 15 phần trăm về hiệu suất của Oracle Parallel Server. Ngoài ra, hiện này DLM trữ các khóa để tránh truyền thông nút chéo về thông tin khóa. Dụng chạm hoặc sự tranh chấp đối với các khối qua các nút còn bị giảm do cải tiến thuật toán điều khiển truy cập các khối có tranh chấp. Các chỉ mục khóa đảo ngược làm giảm các chỉ mục "điểm nóng", nhất là các chỉ mục chủ yếu bằng cách đảo ngược các byte của các khối nhánh và do đó loại trừ sự tranh chấp đối với các khối nhánh trong các môi trường làm việc. Các phân hoạch còn trợ giúp hiệu suất của Parallel Server bằng cách cho phép các người điều hành cơ sở dữ

liệu sắp xếp các phân hoạch tới các nút cá biệt, có thể làm giảm sự đụng chạm đáng kể. Cuối cùng Oracle8 Enterprise Edition làm giảm phần điều hành chung bộ nhớ được kết hợp với khóa hạt mịn trong Oracle Parallel Server.

Oracle8 Enterprise Edition cho phép thiết kế các nhóm môi trường làm việc cho các truy vấn song song hoặc xử lý DML song song. Bạn có thể xác định nhóm nào mà một môi trường làm việc thuốc về và sau đó dùng nhóm ấy để xử lý các câu lệnh từ các ứng dụng nào đó. Điều này rất hữu dụng để tách OLTP từ xử lý kho dữ liệu trong các server liên kết của bạn. Sử dụng các môi trường làm việc trong một Oracle Parallel Server bảo đảm là các truy vấn kho dữ liệu không ảnh hưởng đến hiệu suất của các ứng dụng OLTP.

Oracle8 Enterprise Edition đưa ra các khung nhìn cố định toàn cục (GV$) cho Oracle Parallel Server. Điều này cho phép các người điều hành khởi nhập vào một môi trường làm việc của một Parallel Server và khung nhìn toàn cục chứa dữ liệu từ tất cả các khung nhìn V$ trong cụm liên kết ấy. Điều này tạo cho các điều hoạt quản trị đơn giản đáng kể và cải tiến hiệu suất của các người điều hành cơ sở dữ liệu.

Giải quyết thông.suốt ứng dụng

Nếu một nút trong Oracle Parallel Server sai sót, giải quyết ứng dụng thông suốt sẽ di trú các kết nối và tự động thiết lập lại các phiên của chúng trên một nút khác. Các ứng dụng của bạn sẽ tiếp tục chạy và có thể bạn không biết được sai sót đó. Điều này cung cấp hiệu lực trong trường hợp ngừng chạy có lập lịch trình hoặc chưa lập lịch trình.

Tùy theo số lượng phần điều hành chung mà bạn sẵn sàng chịu trên các máy client hoặc sao lưu, sai sót môi trường làm việc có thể dẫn đến sự di trú hoàn toàn rõ ràng của các kết nối người

dùng với một nút bị sai sót. Bạn có thể xác định muốn tất cả các truy vấn để được trữ trên máy client để chúng có thể được môi trường làm việc lại trên nút bị sai sót. Ngoài ra, bạn có thể có một phiên được kết nối trước với nút bị sai sót để loại trừ thời gian cần thiết để kết nối lại với môi trường làm việc bị sai sót.

Giải ứng dụng thông suốt hữu ích không những có thể sẵn sàng sử dụng mà còn để cân bằng nạp thủ công hoặc ngưng chạy có thứ tự của hệ thống. Nếu có quá nhiều người dùng kết nối với một môi trường làm việc, bạn có thể kết thúc các phiên của họ và di trú họ sang một nút khác. Bạn cũng có thể đóng một nút và di trú các người dùng sang một nút giải quyết sai sót khác sau khi giao dịch của họ hoàn tất. Các lần đóng máy không còn làm gián đoạn công việc của những người dùng.

Lưu ý: để khai thác tính năng này, các ứng dụng phải được ghi đặc biệt bằng cách dùng các lệnh gọi Oracle8 OCI mới.

Xem thêm Oracle8 Parallel Server Concepts và Administration.

Hỗ trợ trình giám sát TP cải tiến

Các hỗ trợ XA động cải tiến hiệu suất cho nhiều ứng dụng hơn với các trình giám sát XA-compliant Transaction Processing (TP) tiêu chuẩn công nghệ. Oracle8 và Oracle8 Enterprise Edition cải tiến hỗ trợ giao diện XA với hỗ trợ để đăng ký động và các nhánh giao dịch kép rộng. Chúng cũng cung ứng hiệu suất và phục hồi tốt hơn khi được dùng với Oracle Parallel Server.

Xem thêm Hướng Dẫn Của Developer Ứng Dụng Oracle8 và Oracle8 Parallel Server Concepts và Administration.

Quản trị bảo mật tốt hơn

Oracle8 và Oracle8 Enterprise Edition kể cả Security Server cung cấp môi trường đăng ký đơn giản đối với những người dùng và các vai trò quản trị trung tâm. Oracle Security Server theo chuẩn bảo mật dựa trên chứng chỉ X.509 đối với sự thẩm định khóa chung/riêng. Sự thẩm định hỗ tương giữa client và server còn được hỗ trợ để bảo vệ tránh cơ sở dữ liệu "xấu" được thiết kế để đón bắt truyền thông của client. Hơn nữa, bộ công cụ những chữ ký số hóa được cung cấp để tạo các ứng dụng được thiết kế để nhận dạng sự can thiệp trái phép dữ liệu.

Oracle8 và Oracle8 Enterprise Edition còn có bảo quản và quản trị mật khẩu cải tiến. Bây giờ bạn có thể xác định một lược trình mật khẩu để tăng cương giản đồ bảo mật. Các mật khẩu có thể hết thời hạn sau một thời gian nào đó hoặc được kiểm tra tính phức tạp (thí dụ, chiều dài tối thiểu). Bạn có thể xác định chính sách riêng của mình hoặc dùng chức năng lưu trữ chuẩn cho chức năng kiểm chiều dài, nội đung hoặc dùng lại các mật khẩu mới. Ngoài ra, bạn có thể tạo các tài khoản người dùng để người dùng ấy phải thay đổi mật khẩu ngay từ lần truy cập hệ thống đầu tiên.

Oracle8 và Oracle8 Enterprise Edition còn bổ sung hỗ trợ cho các liên kết cơ sở dữ liệu có đặc quyền để tạo cho nó không cần lồng mật khẩu trong một liên kết cơ sở dữ liệu và các dịch vụ mật mã dữ liệu có thể gọi từ OCI và PL/SQL.

Sao chép

Sao chép của Oracle8 và Oracle8 Enterprise Edition cung cấp hỗ trợ cải tiến cho những hệ thống phân phối khai triển hàng loạt, những cấu hình giải quyết sai sót lưu lượng truyền cao hơn và các hệ thống kho dữ liệu chuyên môn.

Hiện này hiệu suất sao chép có thể gia tăng. Các thay đổi sao chép các vùng có thể được truyền và áp dụng song song, xóa bỏ

giới hạn lưu lượng truyền về sao chép một cách có hiệu quả, trong khi duy trì tính toàn diện giao dịch. Ngoài ra, nhiều tính năng sao chép được ghi lại và di chuyển từ các kích khởi PL/SQL thành mã C bên trong động cơ Oracle để cung cấp hiệu suất tăng lên đáng kể. Cuối cùng, một số lượng dữ liệu được gửi trên mạng cho mỗi giao dịch được sao chép giảm xuống và các cuộc hành trình trên mạng được duy trì ở mức tối thiểu.

Sự tự động hóa và các hệ thống khai triển hàng loạt khác hiện nay dễ dàng quản lý và thiết kế với các khả năng lập tập hợp con của cả Oracle8 và Oracle8 Enterprise Edition. Các ảnh chụp nhanh truy vấn con phức tạp hiện nay có thể có cả mệnh đề tô bổ nhanh để cho phép bạn thiết kế một ảnh chụp nhanh các hàng lựa dựa trên một truy vấn đối với một bảng khác. Thí dụ, bạn có thể thiết kế một ảnh chụp nhanh cho mỗi người kinh doanh dựa trên bảng ORDERS. Ảnh chụp nhanh đó có thể chỉ chứa các đơn đặt hàng bằng cách lựa các đơn đặt hàng dựa trên tiêu chuẩn của một bảng ASSIGNMENTS. Mỗi người kinh doanh chỉ thấy các đơn đặt hàng của mình và chỉ phải tương tác với bảng gốc khi tải lên các đơn đặt hàng mới.

Sao chép của Oracle8 Enterprise Edition còn cung cấp nhiều cải tiến về tính năng quản lý và dễ dùng. Các cải tiến quan trọng về Oracle Replication Manager gồm các Wizard để tạo các môi trường sao chép dễ hơn so với thiết lập và duy trì. Các cải tiến bổ sung đối với tính năng quản lý sao chép lại gồm đăng ký ảnh chụp nhanh để cung cấp những thông tin có giá trị về những khu vùng nào kết hợp với các ảnh chụp nhanh; các ảnh chụp nhanh chủ yếu nhất để cho phép ảnh chụp nhanh hơn tô bổ sau khi tái tổ chức một bảng; một thuật toán bảo mật mới dễ quản lý hơn và trở thành không hoạt động hạt mịn (fine-grain) để cho phép người điều hành đưa ra các thay đổi mức giản đồ đối với một nhóm sao chép trong khi các nhóm khác vẫn tiếp tục hoạt động.

Xem thêm Sao chép Oracle8.

Công nghệ quan hệ đối tượng

Oracle8 Enterprise Edition tạo được một bước nhảy quan trọng về công nghệ quản lý dữ liệu với mô hình quan hệ đối tượng. Ngày nay các giản đồ cơ sở dữ liệu và các ứng dụng đang ngày càng phức tạp hơn. Thường nhiều ứng dụng riêng biệt với cùng dữ liệu, chẳng hạn như những thông tin về khách hàng, lập hóa đơn và chuyển hàng hóa tồn tại trong các giản đồ cơ sở dữ liệu khác nhau và một bộ phận MIS phải quản lý điều hợp. Quản lý thông tin của từng bộ phận trong tập đoàn càng trở nên tác vụ khó khăn của các đối tượng quan hệ tích hợp và các ứng dụng khác nhau, có thể từ các nhà kinh doanh khác nhau thành một mô hình dữ liệu người dùng cuối có liên quan chặt chẽ hơn. Bằng cách tăng cường cơ sở dữ liệu quan hệ với khối lưu riêng đối tượng, Oracle lập địa chỉ nhu cầu ấy để đơn giản hóa mô hình dữ liệu và mở rộng cơ sở dữ liệu với các kiểu dữ liệu mới.

Các đặc trưng quan hệ đối tượng gồm:

- Các hiển thị và các kiểu đối tượng
- Gọi thủ tục bên ngoài từ trong một cơ sở dữ liệu
- Hỗ trợ bên client hàng đợi với các đối tượng
- Sự khai triển các môi trường quan hệ
- Các công cụ phát triển đối với mô hình đối tượng
- Dữ liệu đa phương tiện
- Java
- Tính năng mở rộng

Hiển thị và các kiểu đối tượng

Các kiểu đối tượng cung cấp cách để mở rộng các hệ thống kiểu dữ liệu quan hệ của Oracle. Các cơ sở dữ liệu quan hệ hỗ trợ ba kiểu dữ liệu: các ký tự, các số và ngày tháng. Các kiểu đối tượng cho phép bạn xác định các kiểu dữ liệu mới và dùng chúng khi bạn điều tiết các kiểu dữ liệu quan hệ. Thí dụ, bạn có thể tạo một Address kiểu gọi mới. Kiểu đối tượng này có thể có dữ liệu,

các thuộc tính được gọi, chẳng hạn như đường phố, thành phố và mã bưu điện. Kiểu đối tượng còn có thể có các phương pháp hoặc các thủ tục định sẵn, chẳng hạn như Distance để tính khoảng cách giữa các địa chỉ. Các phương pháp này có thể được ghi bằng ngôn ngữ PL/SQL hoặc C. Một địa chỉ có thể sau được dùng ở bất kỳ nơi nào có thể là kiểu dữ liệu thường xuyên, trong các định nghĩa cột bằng biến PL/SQL hoặc ngay cả là định nghĩa cho một bảng đối tượng.

Các kiểu đối tượng của Oracle có thể dùng các kỹ thuật lập mô hình đầy năng lực cho các đối tượng phức tạp. Thí dụ, bạn có thể biểu thị các tập hợp của các đối tượng tương tự trong các cấu trúc mảng hoặc các bảng được lồng nhau. Bạn cũng có thể lưu trữ các tham chiếu các đối tượng để đi qua nhanh mà không phải liên kết các bảng.

Các kiểu đối tượng cho phép các Developer ứng dụng viết mã logic ứng dụng trong cơ sở dữ liệu hoặc server ứng dụng tầng ở giữa trái lại với sử dụng mã bên client. Tất cả những ứng dụng sau đó có thể dùng chung logic của các kiểu dữ liệu mới để các Developer không cần phải viết lại mã đó. Đặc trưng này cung cấp các ưu điểm tạo ra các thành phần dùng lại được và phân hoạch ứng dụng rõ ràng để mã đó có thể cư trú và thực hiện trên tầng đạt được hiệu suất tốt nhất: client, server ứng dụng hoặc server cơ sở dữ liệu.

Oracle8 Enterprise Edition theo sau chuẩn SQL3 hợp nhất cho định nghĩa kiểu và các kỹ thuật lập mô hình đối tượng. SQL3 xác định cú pháp để tạo và sửa đổi cửa sổ kiểu đối tượng, phát sinh và lưu trữ các dấu định danh (OIDs), tạo các tham chiếu hoặc các điểm trỏ các đối tượng và lập mô hình các tập hợp của những đối tượng giống nhau.

Gọi các thủ tục bên ngoài từ trong cơ sở dữ liệu

Oracle8 và Oracle8 Enterprise Edition cung cấp phương pháp an toàn, nhanh đối với cơ sở dữ liệu để tạo lệnh gọi một chương trình bên ngoài. Lệnh gọi ấy còn có thể được tạo ra bằng các giao thức mở giống như HTTP hoặc IIOP (chuẩn CORBA). Các thủ tục bên ngoài cho phép bạn dùng mã ứng dụng có sẵn hoặc viết mã tối ưu rất cao đối với các mục đích riêng biệt, chẳng hạn như thuật toán tính toán phức tạp giống như Fast Fourrier Transform (FFT). Ngoài ra, bạn có thể dùng các thủ tục bên ngoài để giao diện với các ứng dụng khác hoặc với các thiết bị chuyên môn như các hệ thống được nhúng.

Xem thêm PL/SQL, Hướng Dẫn Của Người Dùng và Hướng Dẫn Của Tham Chiếu Và Người Điều Hành Oracle.

Hỗ trợ bên client đối với các đối tượng

Lưu trữ đối tượng bên client cho phép các ứng dụng người dùng truy hồi phân cấp các đối tượng phức tạp thành lưu trữ ứng dụng. Sau đó ứng dụng ấy có thể qua các đối tượng mà không phải thực hiện các truy hồi mạng bổ sung. Điều này cung cấp sự thuận tiện và cách nhanh nhất để dùng các đối tượng trong một ứng dụng client và viết mã rất giống mã hướng đối tượng ban đầu.

Một tiện ích mới, Object Type Translator đã được giới thiệu là cách nhanh chóng để phát sinh đầu mục và các tập tin bổ sung cho các ứng dụng dựa trên các giản đồ đối tượng.

Sự khai triển của các môi trường quan hệ

Oracle8 Enterprise Edition được thiết kế để cho phép các người dùng khai triển dễ dàng việc sử dụng tính năng hướng đối tượng, như tất cả các ứng dụng có sẵn tương thích trở về trước. Các mở rộng quan hệ đối tượng mới được thiết lập trên cùng nền tảng như tính năng quan hệ, nghĩa là các người dùng không phải hủy

bỏ hoặc viết lại các ứng dụng quan hệ có sẵn trước khi di trú vào Oracle8 Enterprise Edition. Không giống như các cơ sở dữ liệu quan hệ khác, thiết kế này cho phép các ứng dụng quan hệ cũ hơn vẫn đọc và viết các hàng và các cột cùng tồn tại với các ứng dụng hướng đối tượng mới để đọc và viết các đối tượng. Oracle8 Enterprise Edition cung cấp các hiển thị đối tượng để truy hồi dữ liệu quan hệ và biểu thị dữ liệu với client như nó là một đối tượng hoặc ngược lại.

Thí dụ, một hệ thống nhập đơn đặt hàng quan hệ có sẵn có thể cần một front-end mới cho World Wide Web. Các ứng dụng có sẵn truy cập giản đồ quan hệ có thể vẫn hoạt động và một tập hợp các khung nhìn đối tượng mới có thể được phát triển bằng sự biểu thị đối tượng cho client. Các ứng dụng cũ và mới có thể dựa trên cùng dữ liệu, nhưng mỗi ứng dụng có sự biểu thị riêng.

Xem thêm Oracle Concepts.

Các công cụ phát triển để lập mô hình đối tượng

Object Database Designer cung ứng hỗ trợ đối tượng bằng một công cụ thiết kế. Nó giúp bạn thiết kế, tạo và truy xuất các hệ thống Oracle8 Enterprise Edition dựa trên đối tượng.

Xem thêm Nhà Thiết Kế Cơ Sở Dữ Liệu Đối Tượng.

Dữ liệu đa phương tiện

Các đối tượng lớn Large Objects (LOBs) xử lý dữ liệu không có cấu trúc, chẳng hạn như các ảnh, âm thanh, video và văn bản, có nhiều tính năng phong phú hơn các bộ xử lý trước của chúng, LONG và LONG RAW. Character LOBs (CLOB hoặc NCLOB), Binary LOBs và BFILES (LOBs được lưu trữ bên ngoài) có thể được sao chép và có thể là một thuộc tính của một đối tượng. Bạn cũng có thể có nhiều LOB trên mỗi bảng. Các LOB cũng có

kích thước tối đa lớn hơn LONGs và có cơ chế khác nhau để duy trì và đọc nhất quán và truy xuất ngẫu nhiên.

Dữ liệu LOB có chỉ mục để khởi động truy xuất nhanh ở một byte riêng biệt. Thí dụ, bạn có thể đọc/ghi ở các độ lệch byte riêng biệt. Bạn cũng có thể đọc/ghi các LOB bằng Cache vùng đệm Oracle8 hoặc truy xuất chúng trực tiếp từ đĩa.

Tính năng LOB có thể dùng với cả Oracle8 và Oracle8 Enterprise Edition và không đòi hỏi tùy chọn Objects để được phép cài đặt.

Xem thêm Oracle Concepts.

Java

Oracle có lịch sử dài hỗ trợ các môi trường nền chéo. Oracle8 và Oracle8 Enterprise Edition thiết lập dựa trên sự hỗ trợ mạnh cho Java.

Oracle hiện hành có hai phương pháp truy xuất dữ liệu Oracle từ các chương trình Java: một trình điều khiển Oracle cung cấp JDBC tích hợp với các kiểu đối tượng của Oracle và JSQL để nhúng các câu lệnh SQL thành mã Java. Oracle cung cấp các trình điều khiển JDBC của nó để có hiệu suất tốt hơn. JSQL cho phép bạn có cả các câu lệnh SQL trong một ứng dụng Java. Trình biên dịch trước của JSQL sau đó chuyển đổi SQL thành các lệnh gọi JDBC để cho phép bạn dùng mã SQL có sẵn trong các ứng dụng Java mới.

Xem thêm Các Trình Điều Khiển Oracle8 JDBC.

Tính năng mở rộng

Tính năng mở rộng cho phép bạn xác định các kiểu dữ liệu riêng của mình. Bạn có thể mở rộng các khả năng của cơ sở dữ liệu hiện hành của bạn bằng cách tạo các kiểu cơ sở dữ liệu mới cho các ứng dụng riêng biệt. Các kiểu dữ liệu mới này có thể được dùng bằng cách tương tự như các kiểu dữ liệu bạn hiện có; các ứng dụng tương tự có thể được thực hiện. Các hộp Image, Spatial, Time Series, Visual, and Context bây giờ có thể dùng là các phần mở rộng đối với server dữ liệu.

Di trú và tính năng điều hợp

Một tiện ích đơn giản và nhanh thiết lập lại từ điển dữ liệu và chuyển đổi các tập tin, các tập tin ghi sổ nhật ký và các khối dữ liệu. Tiện ích di trú chuyển đổi bất kỳ cơ sở dữ liệu của Oracle 7.1, 7.2, hoặc 7.3 thành cơ sở dữ liệu của Oracle8 hoặc Oracle8 Enterprise Edition. Các ứng dụng Oracle7 chạy không thay đổi dựa trên một trong hai sản phẩm của Oracle8. Các lệnh phân phối từ một trong các sản phẩm của Oracle8 chạy dựa trên Oracle7 và ngược lại.

Xem thêm Di Trú Của Oracle8.

Các tăng cường khác các bảng có tổ chức chỉ mục

Các bảng có tổ chức chỉ mục lưu trữ các cột dữ liệu của một bảng trong các nút nhánh của cấu trúc chỉ mục câu Oracle B. Điều này làm giảm các đòi hỏi tổng thể bộ nhớ khi phần lớn các cột có chỉ mục bằng cách chỉ lưu trữ các cột một lần. Thay vì cả trong chỉ mục và bảng tách riêng. Các bảng có tổ chức chỉ mục còn làm giảm thời gian truy xuất do truy hồi tất cả các cột từ một vị trí thay vì hai.

Các chỉ mục phím đảo ngược

Các chỉ mục phím đảo ngược làm giảm "các điểm nóng" trong các chỉ mục, nhất là các chỉ mục tăng dần. Các chỉ mục không cân bằng có thể làm cho chỉ mục ấy tăng lên sâu hơn khi bảng cơ sở phát triển. Các chỉ mục phím đảo ngược các byte của các mục nhập khối nhánh, do đó ngăn cản "các chỉ mục lướt".

Xử lý hạn định cải tiến

Xử lý hạn định đã cải tiết rất nhiều. Hiện này bạn có thể dùng những chỉ mục không duy nhất để tăng cường các hạn định UNIQUE và PRIMARY KEY. Điều này loại trừ các chỉ mục không cần thiết và cho phép các chỉ mục vẫn duy trì giá trị trong khi các hạn định được vô hiệu hóa. Kiểm tra hạn định bị trì hoãn chuyển đổi kiểm tra hạn định toàn diện từ cuối sự tiến hành của câu lệnh khi một cam kết (commit) được đưa ra. Điều này đơn giản hóa mã hóa các điều hoạt nào đó liên quan đến các hạn định. Ngoài ra, tất cả các hạn định có thể được cho phép đồng thời và song song với các hạn định khác, và cho phép DML đồng thời khi có sự cho phép tiếp tục.

Bộ ký tự trong một cơ sở dữ liệu

Trong cả Oracle8 và Oracle8 Enterprise Edition, một kiểu dữ liệu NCHAR mới cho phép một bộ ký tự thứ hai trong một cơ sở dữ liệu. Điều này làm cải tiến hiệu suất và khả năng tiên đoán của bộ nhớ đối với một ngôn ngữ Châu Á nào đó, các cơ sở dữ liệu bộ ký tự nhiều byte. Ngoài ra, Unicode 2.0 ban đầu có chiều dài cố định và hỗ trợ Chinese GBK mới.

Các cải tiến hỗn hợp

Khung nhìn không cập nhập được (một khung nhìn được liên kết) có thể được cập nhật bằng cách dùng một kiểu bộ kích khởi mới, bộ kích khởi INSTEAD OF. Bộ kích khởi này cho phép bạn thay thế các điều hoạt INSERT, UPDATE và DELETE trên các

khung nhìn logic của riêng bạn để ngay cả các khung nhìn dựa trên các kết nối phức tạp cũng có thể được hiệu chỉnh. Ngoài ra, bạn có thể đưa các truy vấn con vào danh sách lựa của các truy vấn khác để cho trở lại các tập hợp kết quả từ một danh sách tùy ý gồm các bảng chi tiết mà không phải thực hiện các điều hoạt kết nối. Bằng cách đưa biểu thức con trỏ vào danh sách lựa, bạn có thể truy nạp các hàng từ các bảng chi tiết trong một chương trình 3GL không có điều kiện kết nối phức tạp riêng biệt trong mệnh đề FROM.

CÀI ĐẶT

Cài đặt Oracle8 Enterprise Edition Phiên bản 8.05 cho Windows NT

Cài đặt Oracle8 Enterprise Edition cho Windows NT

Tài liệu này mô tả cách cài đặt Oracle8 Enterprise Edition cho Windows NT. Nó bao gồm những chủ đề theo sau:

Các đòi hỏi hệ thống cài đặt CD Pre-installation

Các sản phẩm đòi hỏi bổ sung có thể sử dụng để cài đặt

Cài đặt các sản phẩm trên Windows NT và Windows 95/98, cài đặt thủ công Legato Storage Manager

Tháo bỏ cài đặt Legato Storage Manager Oracle7 trên Windows 3.1x.

Tháo bỏ các sản phẩm Oracle

1. Trước khi bạn cài đặt Oracle8 Enterprise Edition cho Windows NT xem lại bản sao cứng *Release Notes*.
2. Xem lại chương 2 Oracle8 Enterprise Getting Started cho Windows NT về thông tin sản phẩm.
3. Cài đặt và kiểm tra phần cứng và phần mềm mạng của bạn, nếu có thể dùng được.

Quyết định xem Kiểu Cài Đặt nào bạn muốn thực hiện:

Đối với kiểu cài đặt này...	Xét...
Cài đặt Oracle8 Enterprise Edition trên hệ thống không có phiên bản trước.	Tài liệu này

Nâng cấp Oracle8 EnterpriseEdition từ một phiên bản trước của bản Oracle8 Enterprise.	"Database Coexistense and Migration" của bản Oracle8 EnterpriseGetting Started for Windows NT và tài liệu này.
"Database Coexistense and Migration" của bản Oracle8 Enterprise từ mỗi một phiên bản Oracle7 Server trước.	"Database Coexistense and Migration" của bản Oracle8 Enterprise Getting Started for Windows NT và tài liệu này.
Cài đặt Oracle8 Enterprise Edition vào trong nhiều ban đầu Oracle hoặc vào trong cùng phiên bản ban đầu như Oracle7 Server.	"Database Coexistense and Migration" và " Using Multiple Oracle Home" của Oracle8 Enterprise Edition Getting Started for Windows NT và tài liệu này.

Lưu ý:
Hai phiên bản Oracle8 Enterprise Edition cho Windows NT không cùng tồn tại trong cùng thư mục nếu chúng cùng phiên bản chính (thí dụ, phiên bản 8.05 và phiên bản 8.0.4) vì chúng phụ thuộc vào các tập tin chung; một tập tin ghi đè các tập tin kia. Bắt đầu với phiên bản 8.0.4, chúng có thể được cài đặt thành công trong các thư mục ban đầu tách riêng.

QUAN TRỌNG:
Nếu bạn sử dụng Oracle Web Application Server để cài đặt Oracle Web Application Server CD-ROM có trước. Nếu bạn cố cài đặt Oracle Web Application Server CD-ROM sau khi cài đặt Oracle8 Enterprise Edition, hộp thoại xuất hiện hỏi bạn xem có muốn xuống cấp các thành phần của phần mềm đó không. Trả lời No cho những thông báo này. Ngoài ra, Oracle Web

Application Server phải được cài đặt và chạy trước khi cài đặt Oracle8 Context Cartridge Viewer Cartridge và/hoặc Oracle8 Context Configuration Manager.

Các đòi hỏi hệ thống

Trước khi cài đặt, kiểm tra các đòi hỏi phần cứng và phần mềm

Các đòi hỏi phần cứng

Oracle8 Enterprise Edition đòi hỏi phần cứng sau:

Windows NT Server	Windows NT và Windows 95/98 Clients
Máy tính cá nhân dựa trên bộ xử lý Pentium. Các máy tính dùng bộ đa xử lý cũng được hỗ trợ.	Máy tính cá nhân dựa trên bộ xử lý Intel 80486 hoặc cao hơn.
48 MB RAM.	*Windows NT*: 16 MB RAM cho Application User 32 MB RAM cho Database Administrator *Windows 95/98*: 8 MB RAM cho Application User 24 MB RAM cho Database Administrator
Cài đặt và mạng chức năng.	**Cài đặt và mạng chức năng.**
Không gian đĩa cứng: Không gian đĩa cứng tối thiểu 200 MB. Không gian bổ sung:	Không gian đĩa cứng: Đề nghị cho Application User 0155MB đề nghị cho Database Administrator

[1] Để truy xuất cửa sổ Software Asset Manager, khởi động Oracle Installer và lựa Custom Installation. Nếu bạn không muốn thực hiện Custom Installation sau khi xét các đòi hỏi không gian đĩa của các sản phẩm, nhấp Exit và bắt đầu tiến trình cài đặt lại.

60 MB Oracle Context Cartridge 1 MB Oracle8 Spatial Cartridge 6 MB Oracle Time Series Cartridge 1 MB Oracle Visual Information Retrieval Cartridge 1 MB Oracle Image Cartridge 5 MB Oracle Parallel Server Option 1 MB Oracle8 Partitioning Option 1 MB Oracle8 Objects Option Để tìm dược không gian đĩa mà mỗi sản phẩm đòi hỏi, lựa sản phẩm đó trong cửa sổ *Software Asset Manager*[1] và hiển thị các đòi hỏi không gian đĩa trong trường thông tin.	Để biết được sản phẩm đòi hỏi bao nhiêu không gian đĩa, lựa sản phẩm ấy trong *Software Asset Manager*[1] window, và hiển thị không gian đĩa cần thiết trong trường thông tin.
Oracle8 Recovery Manager đòi hỏi tầng quản lý chẳng hạn như Legato Storage Manager trên CD-ROM. Legato Storage Manager đòi hỏi: 32 MB RAM Không gian đĩa cứng 64 MB Bổ sung 5% tổng số dữ liệu dự phòng hoặc 100 MB cho các chỉ mục trực tuyến. Tài liệu Legato trên CD-ROM trong LEGATO/thư mục DOCS	

Các đòi hỏi phần mềm

Oracle8 Enterprise Edition đòi hỏi phần mềm sau:

Windows NT Server	Windows NT và Windows 95/98 Client
Windows NT Workstation 4.0 Windows NT Server 4.0 hoặc Windows NT Server, Enterprise Edition 4.0	Windows NT Workstation 4.0 Windows NT Server 4.0 Windows NT Server, Enterprise Edition 4.0 hoặc Windows 95/98
Trình duyệt web cho phép các khung và Java	Trình duyệt web cho phép các khung và Java
Bạn phải cài đặt Oracle8 Objects Option, nếu bạn muốn dùng Oracle Image Cartridge, Oracle Time Series Cartridge, hoặc Oracle Visual Information Retrieval Cartridge để thực hiện chức năng chính xác.	
Oracle WebServer 2.1*x* hoặc Web Application Server 3.0 phải được cài đặt và chạy trước khi *cài đặt* các thành phần của Oracle Context Option Workbench, the Viewer Cartridge và Configuration Manager (chỉ có thể sử dụng bằng cài đặt Custom).	

Các đòi hỏi bổ sung

Trước khi cài đặt, kiểm tra các đòi hỏi cài đặt trước của Oracle Enterprise Manager:

Oracle Enterprise Manager

Trước khi cài đặt phiên bản mới của Oracle Enterprise Manager:

1. Chắc chắn bạn đã đóng các phiên bản cũ hơn của Oracle Enterprise Manager Administrator Toolbar (nếu chúng tồn tại trên hệ thống của bạn).
2. Thôi cài đặt bất kỳ phiên bản nào của Oracle Enterprise Manager để bỏ các tập tin không cần thiết.

Các sản phẩm được dùng để cài đặt

Oracle Installer cho bạn cài đặt, cập nhật và xóa các sản phẩm của Oracle8 Enterprise Edition nhanh. Khi bạn chạy Oracle Installer, một hộp thoại hiển thị các tùy chọn cài đặt. Khi bạn lựa một tùy chọn, một nhóm các sản phẩm tự động được cài đặt. Các tùy chọn hiển thị tùy theo hệ điều hành bạn đang dùng.

Nếu đang dùng...	Các tùy chọn có thể sử dụng...
Windows NT	Oracle8 Enterprise Edition Oracle8 Client Custom Installation
Windows 95/98	Oracle8 Client Custom Installation
Windows 3.1x	Oracle7 Client Custom Installation

Các bảng sau liệt kê các thành phần có trong mỗi tùy chọn. Tùy chọn Custom Installation cho bạn lựa cài đặt từ một danh sách của tất cả các sản phẩm có thể sử dụng cho hệ điều hành của

mình. Xem Oracle8 Enterprise Edition Getting Started for Windows NT về các mô tả của mỗi sản phẩm.

Các sản phẩm Oracle8 Enterprise Edition

Các sản phẩm sau được cài đặt nếu bạn lựa tùy chọn Oracle8 Enterprise Edition:

Oracle8 Enterprise Edition	Oracle Intelligent Agent
Oracle8 Utilities	Oracle Database Assistant
Oracle8 Enterprise Edition Cartridges[1] và Options	Oracle Migration Assistant for Microsoft Access
Oracle Call Interface	Oracle Data Migration Assistant
Oracle Enterprise Manager Products và Administrator Toolbar	Net8 Client and Net8 Server (kể cả Net8 Assistant, Net8 Easy Config, Oracle Names Server, Oracle Connection Manager, và Oracle Protocol Adapters)
SQL*Plus	Oracle Patch Assistant
Oracle Objects for OLE	Oracle Web Publishing Assistant

[1] Oracle8 Cartridges và Options không được cài đặt tự động. Bạn phải lựa cài đặt Oracle8 Cartridges and Options được nhắc khi cài đặt.

Oracle8 ODBC Driver	Oracle8 Performance Utility
Oracle8 Enterprise Edition Release Notes	Oracle Documentation
Assistant Common Files	Oracle Installer
Oracle Replication Manager	Java Runtime Environment

Các sản phẩm Oracle8 Client

Khi bạn lựa tùy chọn Oracle8 Client, bạn được nhắc cài đặt một trong hai tùy chọn phụ:

- Application User
- Database Administrator

Các sản phẩm được cài đặt tự động với từng tùy chọn phụ được liệt kê bên dưới:

Sản phẩm	Ứng dụng người dùng ?	Người điều hành CSDL?
Oracle8 Utilities	KHÔNG	CÓ
Oracle Object Type Translator	KHÔNG	CÓ

Oracle Call Interface	KHÔNG	CÓ
Oracle Enterprise Manager Products and Administrator Toolbar	KHÔNG	CÓ
Net8 Client (including Net8 Assistant and Oracle Protocol Adapters)	CÓ	CÓ
SQL*Plus	CÓ	CÓ
Oracle Objects for OLE	CÓ	CÓ
Oracle8 ODBC Driver	CÓ	CÓ
Oracle Documentation	CÓ	CÓ
Assistant Common Files	CÓ	CÓ
Oracle Trace Collection Services	CÓ	CÓ
Java Runtime Environment	CÓ	CÓ
Oracle Installer	CÓ	CÓ
Oracle8 Client Release Notes	CÓ	CÓ
Required Support Files	CÓ	CÓ
Oracle Web Publishing Assistant	KHÔNG	CÓ[1]

Oracle Migration Assistant for Microsoft Access	KHÔNG	CÓ
Oracle Replication Manager	KHÔNG	CÓ

Lưu ý:
Dù CD-ROM trong bộ phần mềm của bạn chứa nhiều sản phẩm của Oracle, bạn chỉ được cài đặt những sản phẩm này khi đã đăng ký.
Nếu bạn nhấp vào Cancel trong hộp thoại Oracle Installer, sẽ thoát khỏi Oracle Installer.

Cài đặt các sản phẩm trên Windows NT và Windows 95/98

Để cài đặt các sản phẩm Oracle:

1. Khởi động hệ điều hành của bạn. Nếu bạn đang chạy Windows NT, khởi nhập vào server Windows NT là thành viên của nhóm Administrators.
2. Đóng tất cả các ứng dụng của Oracle (nếu bạn đang chạy).
3. Nếu cài đặt trên Windows NT, dừng tất cả các dịch vụ của Oracle (nếu đang chạy). Để dừng các dịch vụ:
   a. Từ Control Panel, lựa biểu tượng Services.
   b. Nếu bất kỳ dịch vụ Oracle nào tồn tại và có trạng thái *Started*, lựa dịch vụ ấy và nhấp Stop.
   c. Nhấp Close để thoát khỏi cửa sổ *Services*.
4. Đưa đĩa CD-ROM Oracle8 Enterprise Edition for Windows NT.

   Hệ điều hành tự động phát hiện Oracle8 Enterprise Edition for Windows NT CD-ROM của bạn. Hộp thoại *Oracle8 Enterprise Edition -Autorun* xuất hiện và nhắc bạn lựa:

   - Begin Installation

- Browse Information
- Explore CD

5. Chọn Begin Installation.

Lưu ý:
Mỗi lần bạn đưa đĩa CD-ROM vào, tiến trình được mô tả trong Step 4 xảy ra. Để khởi động lại Oracle Installer không phải bỏ và đưa CD-ROM vào lại, lựa ổ đĩa mà CD-ROM sắp xếp, và nhấp đôi vào SETUP.EXE.

Hộp thoại *Oracle Installation Settings* xuất hiện.

6. Xác định tên công ty, tên ban đầu Oracle, vị trí ban đầu Oracle, và ngôn ngữ ban đầu Oracle nơi:

Field	Mô tả
Tên Công ty	Tên công ty mà bản sao phần mềm Oracle software này được cho phép.
Tên viết tắt Oracle	Tên bạn muốn dùng cho Oracle Home. Nếu bản sao này là lần đầu bạn cài đặt các sản phẩm Oracle trên máy này, Oracle Corporation đề nghị bạn chấp nhận tên ban đầu mặc định, DEFAULT_HOME. DEFAULT_HOME cũng là tên Oracle được dùng cho các cài đặt trước của phần mềm Oracle trên máy của mình. Tên Oracle Home phải dài từ 1 tới 16 ký tự, và chỉ có thể gồm các ký tự chữ-số hoặc gạch dưới.
Vị trí Oracle Home	Thư mục con mà bạn muốn có các sản phẩm đối với Oracle ban đầu này được cài đặt. Giá trị mặc định cho vị trí ban đầu trên máy mà các sản phẩm Oracle khác không được cài đặt là ổ đĩa cục bộ lớn nhất có thể dùng.

Ngôn ngữ Oracle Home	Ngôn ngữ bạn muốn dùng để cài đặt và chạy các sản phẩm Oracle. Ngôn ngữ mặc định là tiếng Anh được cài đặt bổ sung vào bất kỳ ngôn ngữ khác bạn lựa.

7. Nhấp OK:

Hộp thoại *Select Installation Options* xuất hiện.

8. Lựa tùy chọn cài đặt, rồi nhấp OK.
9.

Tùy chọn cài đặt	**Xem phần...**
Oracle8 Enterprise Edition (*Windows NT only*)	"Oracle8 Enterprise Edition"
Oracle8 Client (*Windows NT or Windows 95/98*)	"Oracle8 Client"
Custom Installation	"Custom Installation"

Oracle8 Enterprise Edition

Hộp thoại *Select Cartridges and Options* hiển thị các tùy chọn và Oracle database:

- Oracle Parallel Server Option
- Oracle8 Context Cartridge
- Oracle8 Image Cartridge
- Oracle8 Objects Option
- Oracle8 Partitioning Option
- Oracle8 Spatial Cartridge
- Oracle8 Time Series Cartridge
- Oracle8 Visual Information Retrieval Cartridge

1. Lựa (các) thay đổi mà bạn được cho phép, sau đó nhấp OK.

Cảnh báo:

Nếu bạn lựa Parallel Server Option, một hộp thoại xuất hiện để bạn tiếp tục cài đặt có hoặc không có Parallel Server Option. Bạn sẽ không thể khởi động bất kỳ dịch vụ Oracle services nào cho tới khi tầng Operating System Dependant (OSD) nhà kinh doanh cung cấp, được Oracle xác nhận được cài đặt đầy đủ.

Lưu ý:

Nếu Oracle Installer phát hiện phiên bản Oracle7 về cơ sở dữ liệu trên ổ đĩa cứng của bạn, nó sẽ hỏi xem bạn có muốn di trú cơ sở dữ liệu tới Oracle8 Enterprise Edition for Windows NT.

Nếu bạn lựa YES:

Bạn không được nhắc lựa loại cơ sở dữ liệu cài đặt.

Oracle Data Migration Assistant khởi động vào cuối cài đặt và di trú cơ sở dữ liệu Oracle7 của bạn tới Oracle8 Enterprise Edition for Windows NT.

Hộp thoại *Select a Starter Database Configuration* xuất hiện.

2. Lựa loại cơ sở dữ liệu bạn muốn cài đặt, rồi nhấp OK:
3.

Loại cơ sở dữ liệu	Tùy chọn này...
Cấu hình Typical	Cài đặt cơ sở dữ liệu trình khởi động sẵn sàng sử dụng bằng các xác lập sau: • Tên của ORC*x* • System Identifier (SID) của ORCL cho cơ sở dữ liệu trình khởi động đầu tiên trên máy của bạn. Các cơ ở dữ liệu bổ sung sử dụng quy ước đặt tên ORCx or OR*xx* nơi một số được thêm vào để bảo đảm SID là duy nhất. • INTERNAL mật khẩu của ORACLE • Khả năng thực hiện đối xứng, sao chép cập

	nhật bất kỳ nơi nào. Bất kỳ Oracle Database Cartridges bạn lựa cũng có trong cơ sở dữ liệu của trình khởi động.
Cấu hình Custom	Khởi động Oracle Database Assistant vào cuối tiến trình cài đặt. Trợ giúp này giúp bạn tạo một cơ sở dữ liệu Typical Custom. Tùy chọn Typical tạo một cơ sở dữ liệu chuẩn dùng các xác lập khởi hoạt mặc định. Tùy chọn Custom chỉ dành cho các người dùng quen với các thủ tục tạo cơ sở dữ liệu nâng cao. Xem "Creating a Database", trong *Oracle8 Enterprise Edition Getting Started for Windows NT* về các thông tin bổ sung.
None	Không tạo cơ sở dữ liệu, nhưng bạn có thể tạo cơ sở dữ liệu của riêng mình sau. Tất cả các thành phần của server được cài đặt. Tùy chọn None chỉ dành cho những người dùng quen với các thủ tục tạo cơ sở dữ liệu nâng cao. Nếu bạn định cài đặt tùy chọn Oracle Parallel Server, bạn phải chọn None.

Hộp thoại *Installing Legato Storage Manager* xuất hiện.

3. Nhấp Yes nếu bạn muốn cài đặt Legato Storage Manager.

Hộp thoại *Installing Oracle Documentation* xuất hiện.

4. Lựa cài đặt tài liệu Oracle8 HTML trên ổ đĩa cứng của bạn hoặc để nó trên CD-ROM, sau đó nhấp OK.

Cài đặt Oracle8 Enterprise Edition tiếp tục. Một thanh tiến hành xuất hiện chỉ phần trăm hoàn tất cài đặt. Hộp thoại *Installation Completed* dialog xuất hiện.

5. Nhấp Yes để xem tài liệu ấy hoặc nhấp No để thoát khỏi Oracle Installer.

Nếu bạn lựa tùy chọn Oracle8 Cartridge và đã lựa để cài đặt cơ sở dữ liệu của trình khởi động, một hộp thoại xuất hiện thông báo cho bạn biết cách dùng tùy chọn này phải nạp dữ liệu vào cơ sở dữ liệu ấy. Để thực hiện điều này, chạy Oracle Database Assistant và chọn tùy chọn sử đổi hoặc xem Oracle8 Enterprise Edition Getting Started for Windows NT về những hướng dẫn cấu hình thủ công.

Oracle8 Client

Hộp thoại *Oracle8 Client Configuration* hiển thị Application User và các tùy chọn phụ của Database Administator.

1. Lựa tùy chọn phụ, rồi nhấp OK.

   Hộp thoại *Installing Oracle Documentation* xuất hiện.
2. Lựa cài đặt tài liệu Oracle8 HTML trên ổ đĩa cứng của bạn hoặc để nó trên CD-ROM, rồi nhấp OK.

   Cài đặt tiếp tục. Một thanh tiến hành xuất hiện chỉ số phần trăm hoàn thành của cài đặt. Hộp thoại *Installation Completed* xuất hiện.
3. Nhấp Yes để xem tài liệu ấy hoặc nhấp No để thoát khỏi Oracle Installer.

Cài đặt Custom

Cửa sổ *Software Asset Manager* hiển thị tất cả các sản phẩm và các bộ phần mềm có thể sử dụng để cài đặt:

1. Lựa các sản phẩm/các bộ phần mềm để cài đặt.

   Các sản phẩm/các bộ phần mềm có thể sử dụng để cài đặt được liệt kê ở mé trái của cửa sổ. Các sản phẩm Oracle mà bạn đã cài đặt rồi xuất hiện ở mé phải của cửa sổ. Để xem mô tả tóm tắt một thành phần, nhấp vào tên của thành phần đó.

Lựa nhiều sản phẩm để cài đặt, ấn phím CTRL khi bạn lựa.

2. Nhấp Install để cài đặt (các) sản phẩm/(các) bộ phần mềm bạn đã lựa.
3. Cung cấp các trả lời thích hợp cho bất kỳ Hộp thoại nào xuất hiện.

   Custom Installation tiếp tục. Một thanh tiến hành xuất hiện chỉ số phần trăm hoàn thành của cài đặt đó.
4. Để thoát khỏi Oracle Installer, nhấp Exit từ cửa sổ *Software Asset Manager*, rồi nhấp Yes.

Cài đặt Legato Storage Manager thủ công

Nếu bạn quyết định không cài đặt Legato Storage Manager khi cài đặt Oracle8 Enterprise Edition, bạn có thể cài đặt thủ công nó sau.

Để cài đặt Legato Storage Manager thủ công:

1. Khởi nhập vào máy tính là thành viên của nhóm Administrator của Windows NT.
2. Ngưng dịch vụ Oracle, OracleService*SID*:
   a. Từ thanh tác vụ, lựa Start > Settings > Control Panel.
   b. Nhấp đôi vào Services.
   c. Lựa OracleService*SID*, và nhấp Stop.
   d. Nhấp Close để thoát khỏi cửa sổ *Services*.
3. Đưa đĩa CD Oracle8 Enterprise Edition for Windows NT vào ổ CD-ROM.

   Định vị thư mục *ORACLE_HOME*\LEGATO và nhấp đôi vào biểu tượng tập tin LSMINST.EXE. Cài đặt Legato Storage Manager ngầm định (không có Hộp thoại xuất hiện). Khi cài đặt hoàn tất theo mặc định Legato Storage Manager được cài đặt trong thư mục WIN32APP\NSR\BIN. Tập tin giao diện ORASBT.DLL được cài đặt trong thư mục WINNT\SYSTEM32.
4. Xác định các dịch vụ Legato services đã khởi động:

a. Từ thanh tác vụ, lựa Start > Settings > Control Panel.
b. Nhắp đôi Services.
c. Kiểm Status của các dịch vụ được đánh dấu "Started" sau:
```
NetWorker Backup and Recover Server
NetWorker Portmapper Service
NetWorker Remote Exec Service
```

5. Bổ sung cài đặt thư mục Legato Storage Manager vào đường tìm hệ thống của bạn:
   a. Mở Control Panel.
   b. Nhắp đôi System.
   c. Nhắp tab Environment.
   d. Lựa Path từ hộp danh sách System Variables.
   e. Bổ sung vào trường Value:
   ```
   ;C:\WIN32APP\NSR\BIN
   ```
   nơi có dấu chấm phẩy (;) tách mục nhập mới ra khỏi các mục hiện có..
   f. Nhắp Set.
   g. Nhắp Apply.
   h. Nhắp OK.
   i. Khởi động lại máy tính của bạn để thay đổi kết quả.
   j. Thông tin bổ sung:
   Xem, "Media Management" trong *Legato Storage Manager Administrator's Guide* về thông tin cấu hình thiết bị lưu trữ.

Tháo bỏ Legato Storage Manager

Quan trọng:

Không xóa bỏ các chỉ mục nếu bạn đang cập nhật từ Legato Storage Manager vào Legato BusinesSuite Module for Oracle. Lựa Partial Removal. Chắn chắn bạn có bản

sao lưu mới nhất của toàn bộ các chỉ mục (*\NSR\INDEX*), các tập tin tài nguyên server (*\NSR\RES*), và cơ sở dữ liệu môi trường (*\NSR\MM*) hiện có. Xem *Quick Start Guide for the Legato NetWorker BusinesSuite Module for Oracle* bạn đang cập nhật.

Để xóa Legato Storage Manager:

1. Ngưng các dịch vụ của Legato Storage Manager:
   a. Mở Control Panel.
   b. Nhấp đôi Services.
   c. Lựa các dịch vụ và nhấp Stop cho mỗi dịch vụ:
      `NetWorker Backup and Recover Server`
      `NetWorker Portmapper Service`
      `NetWorker Remote Exec Service`
   d. Nhấp Close để đóng Services control panel.
2. Từ thanh tác vụ chọn Start > Programs > NetWorker Group > Uninstall NetWorker.
3. Lựa Complete Removal hoặc Partial Removal.

Lưu ý:

Chỉ lựa Complete Uninstall nếu bạn muốn xóa tất cả cơ sở dữ liệu, các chỉ mục và các tập tin tài nguyên của mình.

NetWorker Uninstaller xóa Legato Storage Manager khỏi hệ thống của bạn.

Xóa tập tin giao diện ORASBT.DLL từ thư mục C:\WINNT\SYSTEM32, nơi WINNT là thư mục cài đặt cho Windows NT.

Cài đặt Oracle7 Client trên Windows 3.1x

Để cài đặt phần mềm Oracle7 Client trên Windows:

1. Khởi động Windows 3.1x.
2. Ngưng tất cả các ứng dụng đang chạy.

3. Đưa CD Oracle8 Enterprise Edition for Windows NT vào ổ CD-ROM của bạn.
4. Khởi động Oracle Installer:
    a. Mở File Manager.
    b. Lựa ổ đĩa CD-ROM.
    c. Định vị thư mục ban đầu của CD-ROM ấy và nhấp đôi SETUP.EXE.

    Hộp thoại *Language* dialog xuất hiện.
5. Lựa ngôn ngữ hoặc nhấp OK để chấp nhận ngôn ngữ mặc định, tiếng Anh là tiếng được cài đặt bổ sung vào bất kỳ ngôn ngữ khác mà bạn lựa. Để bổ sung một ngôn ngữ sau khi cài đặt, "National Language Support" trong *Oracle8 Enterprise Edition Getting Started for Windows NT*.

    Hộp thoại *Oracle Installation Settings* xuất hiện.
6. Xác định tên công ty và thư mục ban đầu Oracle để cài đặt các sản phẩm, rồi nhấp OK.

    Hộp thoại *Oracle7 Client* xuất hiện.
7. Lựa một trong các tùy chọn client sau, sau đó nhấp OK.
    - Oracle7 Client
    - Custom Installation

Nếu bạn lựa tùy chọn Oracle7 Client, hộp thoại *Oracle7 Client* hiển thị các tùy chọn phụ của Application User and Network Manager.

a. Lựa một tùy chọn phụ, rồi nhấp OK.

    Hộp thoại *Installing Oracle Documentation* xuất hiện.

b. Lựa cài đặt tài liệu Oracle HTML trên ổ đĩa cứng hoặc để nó trên CD-ROM, rồi nhấp OK.

c. Cài đặt tiếp tục. Một thanh tiến hành xuất hiện chỉ số phần trăm hoàn thành cài đặt. Hộp thoại *Installation Completed* xuất hiện.
d. Nhấp Yes để xem tài liệu đó hoặc nhấp No để thoát khỏi Oracle Installer.

Nếu bạn đã chọn tùy chọn Custom Installation, thực hiện theo các hướng dẫn trong phần "Custom Installation".

Tháo bỏ các sản phẩm Oracle

Tháo bỏ các sản phẩm Oracle:

1. Ngưng tất cả các dịch vụ Oracle.
2. Khởi động Oracle Installer.
3. Trường Installed Products hiển thị ở phần bên phải của cửa sổ *Software Asset Manager*.
4. Lựa (các) sản phẩm bạn muốn xóa khỏi trường Installed Products.
5. Nhấp Remove.

   Oracle Installer cảnh báo bạn về bất kỳ các phụ thuộc có thể gây ra các vấn đề nếu các sản phẩm đặc biệt nào bị xóa và nhắc bạn xác định thôi cài đặt.

Oracle8 Database

Trước khi cài đặt

Những điều kiện tiên quyết

Giả sử bạn đã biết:

- Windows NT và đã được cài đặt và trác nghiệm nó trên máy tính của mình và phần cứng mạng
- Các khái niệm quản lý cơ sở dữ liệu liên quan

Mục đích

Hướng dẫn này cần thiết cho bất kỳ ai cấu hình hoặc điều hành Oracle8 Enterprise Edition.

Lưu ý:

Hướng dẫn này chỉ mô tả các đặc trưng của phần mềm Oracle8 Enterprise Edition áp dụng cho các hệ điều hành Windows NT, Windows 95, và Windows 98. Để có thêm thông tin về Oracle8 Enterprise Edition có thể áp dụng tất cả các hệ điều hành, xem tài liệu khác trong bộ phần mềm của bạn được liệt kê ở "Bắt đầu với tài liệu của bạn". Các quy ước sau được dùng trong hướng dẫn này:

Các quy ước tài liệu và mã

Quy ước	Thí dụ	Ý nghĩa
	C:\ORANT\DATABASE	Chỉ các tên lệnh, SQL, và các từ khóa, như trong ALTER DATABASE. Tất cả chữ hoa cũng được sử dụng cho các tên thư mục và các tên tập tin.
Chữ nghiêng		Chỉ giá trị bạn cung cấp. Thí dụ, nếu một lệnh yêu cầu bạn gõ *filename*, bạn phải gõ tên thực của tập tin ấy.

		Chữ nghiêng cũng được dùng để nhấn mạnh trong văn bản và để chỉ các tên của các hướng dẫn khác.
Ngoặc vuông [ ]	Start > Programs > Oracle for Windows NT - [*HOME_NAME*] > Oracle Database Assistant.	Các mục tùy chọn ở trong ngoặc vuông, thí dụ, bạn có thể khởi động Oracle Installer từ home Oracle mặc định hoặc từ một home khác của Oracle được chỉ bằng [*HOME_NAME*] nếu bạn sử dụng tên chủ của Oracle. Các ngoặc vuông còn chỉ một phím chức năng, thí dụ [Enter].
Chọn Start >	Chọn Start > Programs > Oracle for Windows NT > Oracle Installer	Cách để khởi động một chương trình. Thí dụ, khởi động Oracle Installer, bạn phải nhấp vào nút Start trên thanh tác vụ, sau đó chọn Programs, Oracle for Windows NT, Oracle Installer.
C:\>	C:\ORANT\DATABASE	Biểu thị dấu nhắc lệnh Windows NT của ổ đĩa hiện hành. Dấu nhắc của bạn phản ánh thư mục con trong đó bạn đang làm việc. Tham chiếu như dấu nhắc lệnh *MS-DOS* trong hướng dẫn này.
Dấu gạch chéo ngược (\) trước tên thư mục	\DATABASE	Chỉ thư mục đó là một thư mục con của thư mục gốc.
ORACLE_HOME	Đi tới thư mục *ORACLE_HOME*\DATABASE SVRMGR> @%ORACLE_HOME%\RDBMS80\ADMIN\CATALOG.SQL	*ORACLE_HOME* biểu thị là chữ ổ đĩa và thư mục mức trên cùng nơi phần mềm Oracle được cài đặt có thể là: • C:\ORANT for Windows NT • C:\ORAWIN95 for Windows 95 • C:\ORAWIN98 for Windows 98 • C:\ORAWIN for Windows 3.1 Hoặc bất kỳ điều gì bạn có thể gọi là ban đầu Oracle

		Trong các lệnh Server Manager, bạn có thể thấy: %ORACLE_HOME%. Server Manager có thể định vị thư mục Oracle Home bằng cách sử dụng %ORACLE_HOME% khác nhau. Quy ước này có thể được dùng trong Server Manager, SQL*Plus, Export Utility, và Import Utility.
HOME NAME	Oracle*HOME NAME*TNSListener80	Biểu thị tên chủ của Oracle nếu bạn dùng nhiều home Oracle. Quy ước này không thể áp dụng cho chỉ một home *Oracle*. Tên chủ có thể tối đa 16 ký tự chữ-số. Chỉ ký tự đặc biệt trong tên ban đầu được gạch dưới.
HOME*ID*	HOME0, HOME1, HOME2	Biểu thị một khóa phụ duy nhất đăng ký cho mỗi thư mục gốc Oracle trong đó bạn cài đặt các sản phẩm. Một HOME*ID* mới được tạo và được gia số mỗi lần bạn cài đặt các sản phẩm vào thư mục gốc Oracle khác trên một máy tính. Mỗi HOME*ID* có các xác lập tham số cấu hình riêng đối với các sản phẩm Oracle cài đặt.
Ký hiệu	Dấu chấm . Dấu phẩy , Gạch nối - Chấm phẩy ; Hai chấm : Dấu bằng = Gạch chéo ngược \ Dấu trích dẫn đơn '' Dấu trích dẫn kép "" Dấu ngoặc đơn ()	Các ký hiệu khác hơn các ngoặc vuông và các thanh dọc phải được nhập bằng các lệnh chính xác như được trình bày.

Phần II

CÁC CẤU TRÚC CƠ SỞ DỮ LIỆU

Phần II mô tả kiến trúc có cấu trúc cơ bản của Oracle server, gồm các cấu trúc bộ nhớ logic và vật lý. Phần II trình bày các chương sau:

Chương 2, "Các khối dữ liệu, khối lưu riêng và các đoạn"
Chương 3, "Các không gian bảng và các tập tin dữ liệu"
Chương 4, "Từ điển dữ liệu"

Lưu ý:
Để có thêm thông tin về các cấu trúc Oracle server, xem thêm:

Chương 6, "Cấu trúc bộ nhớ"
Chương 7, "Cấu trúc xử lý"
Chương 8, "Các đối tượng giản đồ"
Chương 9, "Các bảng phân hoạch và các chỉ mục"

Chương 2

Các khối dữ liệu, khối lưu riêng và các đoạn

Chương này mô tả tính chất của quan hệ giữa các cấu trúc bộ nhớ logic trong Oracle server. Nó gồm:

Các quan hệ giữa các khối dữ liệu, khối lưu riêng và các đoạn
Các khối dữ liệu
Khối lưu riêng
Các đoạn

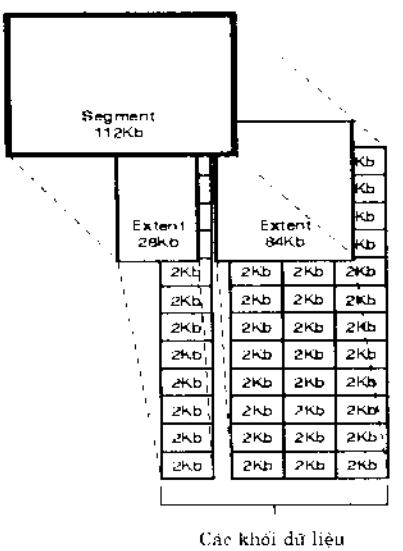
Những thông tin bổ sung:

> Nếu bạn đang dùng Trusted Oracle, xem tài liệu Trusted Oracle của mình để có thêm thông tin về Bộ Nhớ Trong Môi Trường Đó.

Các quan hệ giữa các khối dữ liệu, các phần mở rộng và các đoạn

Oracle phân bổ không gian cơ sở dữ liệu logic cho tất cả dữ liệu trong một cơ sở dữ liệu. Các đơn vị của phân bổ không gian cơ sở dữ liệu là các khối, các khối lưu riêng và các đoạn. Các hình minh họa sau trình bày các quan hệ giữa các cấu trúc dữ liệu này:

Hình 2-1 Các quan hệ giữa các đoạn, các khối lưu riêng và các khối dữ liệu



Ở mức độ chi tiết nhất, Oracle lưu trữ dữ liệu trong các khối dữ liệu (còn được gọi là các khối logic, các khối Oracle hoặc các trang). Một khối dữ liệu tương ứng với một số byte riêng của không gian cơ sở dữ liệu vật lý trên đĩa.

Mức kế tiếp của không gian cơ sở dữ liệu logic được gọi là khối lưu riêng. Một khối lưu riêng là một số các khối dữ liệu liên tiếp riêng biệt được phân bổ để lưu trữ một kiểu thông tin riêng.
Mức của bộ nhớ cơ sở dữ liệu logic trên một khối lưu riêng được gọi là đoạn. Một đoạn là một tập hợp các khối lưu riêng đã được phân bổ cho một kiểu cấu trúc dữ liệu riêng. Thí dụ, dữ liệu của

mỗi bảng được lưu trữ trong một đoạn dữ liệu riêng, trong khi dữ liệu của mỗi chỉ mục được lưu trữ trong đoạn chỉ mục riêng của nó. Nếu bảng ấy hoặc chỉ mục được phân hoạch, mỗi phân hoạch được lưu trữ trong đoạn riêng.

Oracle phân bổ không gian cho các đoạn bằng các đơn vị của một khối lưu riêng. Khi các đoạn có sẵn của một đoạn đầy đủ, thì Oracle phân bổ khối lưu riêng khác cho đoạn đó. Vì các khối lưu riêng được phân bổ theo nhu cầu, các khối lưu riêng có thể hoặc không thể liên tiếp trên đĩa.

Một đoạn (và tất cả các khối lưu riêng của nó) được lưu trong không gian bảng. Trong một không gian bảng, một đoạn có thể mở rộng các tập tin dữ liệu (có các khối lưu riêng với dữ liệu từ nhiều tập tin). Tuy nhiên, mỗi khối lưu riêng có thể chứa dữ liệu từ chỉ một tập tin dữ liệu.

Các khối dữ liệu

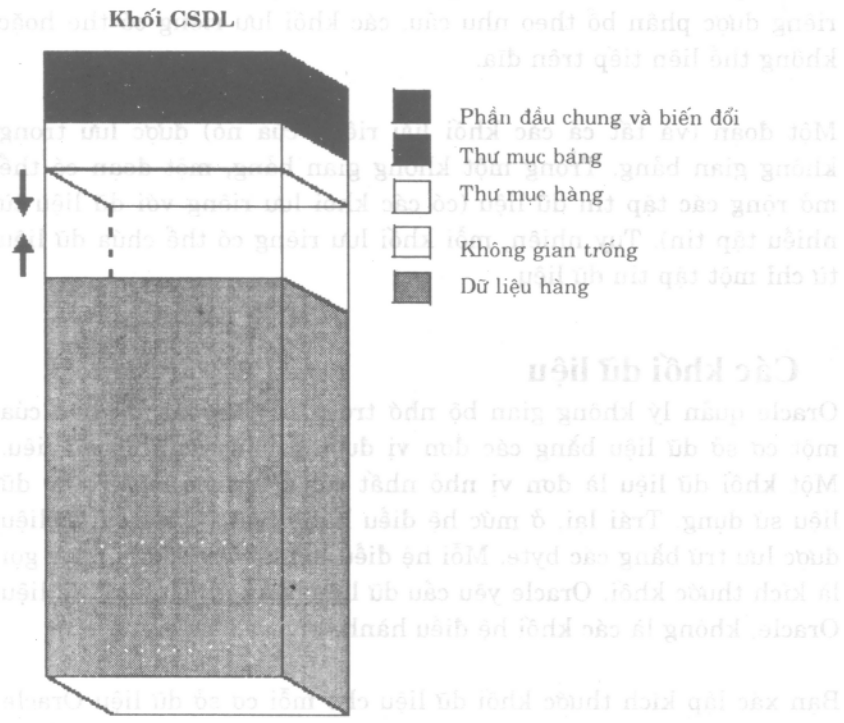
Oracle quản lý không gian bộ nhớ trong các tập tin dữ liệu của một cơ sở dữ liệu bằng các đơn vị được gọi là các khối dữ liệu. Một khối dữ liệu là đơn vị nhỏ nhất của I/O được một cơ sở dữ liệu sử dụng. Trái lại, ở mức hệ điều hành, vật lý, tất cả dữ liệu được lưu trữ bằng các byte. Mỗi hệ điều hành có các khối được gọi là kích thước khối. Oracle yêu cầu dữ liệu bằng nhiều khối dữ liệu Oracle, không là các khối hệ điều hành.

Bạn xác lập kích thước khối dữ liệu cho mỗi cơ sở dữ liệu Oracle khi tạo cơ sở dữ liệu ấy. Kích thước khối dữ liệu này là bội số của kích thước khối của hệ điều hành trong giới hạn tối đa (cổng xác định) để tránh I/O không cần thiết. Các khối dữ liệu Oracle là các đơn vị nhỏ nhất của bộ nhớ mà Oracle có thể sử dụng hoặc phân bổ.

Định dạng khối dữ liệu

Định dạng khối dữ liệu Oracle tương tự bất kể khối dữ liệu ấy chứa bảng, chỉ mục hoặc dữ liệu cụm liên kết. Hình 2.2 minh họa định dạng của một khối dữ liệu.

Hình 2-2 Format khối dữ liệu



Phần đầu (chung và biến đổi)

Phần đầu chứa thông tin khối chung, chẳng hạn như địa chỉ khối và kiểu đoạn (thí dụ, dữ liệu, chỉ mục hoặc truy hồi)

Thư mục bảng

Phần của khối dữ liệu này chứa những thông tin về các bảng có các hàng trong khối này.

Thư mục hàng

Phần của khối dữ liệu này chứa những thông tin về các hàng thực trong khối (kể cả những địa chỉ cho mỗi phân đoạn hàng trong vùng dữ liệu hàng).

Một khi không gian đã được phân bổ trong thư mục hàng của phần điều hành chung của khối dữ liệu, không gian này không được phục hồi khi hàng được xóa. Do đó, một khối hiện trống nhưng có tối đa 50 hàng cùng một lúc tiếp tục có 100 byte được phân bổ trong phần đầu cho thư mục hàng ấy. Oracle dùng lại không gian này chỉ khi các hàng mới được chèn vào khối đó.

Phần điều hành chung

Phần đầu khối dữ liệu, thư mục bảng và thư mục hàng được tham chiếu chung bằng phần đầu. Một số phần điều hành chung của khối có kích thước cố định; tổng kích thước phần điều hành của khối có thể biến đổi. Trung bình, các phần biến đổi và cố định của tổng số phần điều hành chung khối dữ liệu từ 84 tới 107 byte.

Dữ liệu hàng

Phần khối dữ liệu này chứa dữ liệu bảng hoặc chỉ mục. Các hàng có thể mở rộng các khối, xem "Chuỗi Xích Hàng Và Di Trú".

Không gian trống

Không gian trống được phân bổ để chèn các hàng mới và để cập nhật các hàng đòi hỏi không gian bổ sung (thí dụ, khi phần cuối

trống được cập nhật giá trị không rỗng). Nếu các hàng chèn được phát hành thực sự xảy ra trong khối dữ liệu cho trước là một chức năng của không gian trống hiện hành trong khối dữ liệu và giá trị của tham số quản lý không gian PCTFREE. Phần kế tiếp, "Giới Thiệu PCTEREE, PCTUSED và Chuỗi Xích Hàng" chứa nhiều thông tin trên các tham số quản lý không gian.

Trong các khối dữ liệu được phân bổ cho đoạn dữ liệu của một bảng hoặc một cụm liên kết hay cho đoạn chỉ mục, không gian trống còn có thể lưu giữ các mục nhập giao dịch. Một mục nhập giao dịch cần thiết trong một khối cho mỗi câu lệnh INSERT, UPDATE, DELETE, and SELECT...FOR UPDATE truy xuất một hoặc nhiều hàng trong khối ấy. Không gian cần thiết cho các mục nhập giao dịch phụ thuốc vào hệ điều hành; tuy nhiên, các mục nhập giao dịch trong phần lớn các hệ điều hành cần gần 23 byte.

Giới thiệu PCTFREE, PCTUSED và chuỗi xích hàng

Hai tham số quản lý không gian là PCTFREE và PCTUSED cho phép bạn điều khiển sử dụng không gian trống để chèn và cập nhật tất cả hàng trong những khối dữ liệu của một đoạn riêng biệt. Bạn xác định những tham số này khi tạo hoặc thay đổi một bảng hoặc một cụm liên kết (có đoạn dữ liệu riêng). Bạn cũng có thể xác định tham số của bộ nhớ PCTFREE khi tạo hoặc thay đổi một chỉ mục (có đoạn chỉ mục riêng).

Lưu ý:
Điều đề cập ở đây không áp dụng cho các kiểu dữ liệu LOB (BLOB, CLOB, NCLOB và BFILE) - chúng không dùng tham số bộ nhớ PCTFREE hoặc các danh sách trống. Xem thêm "Các Kiểu Dữ Liệu LOB"

Tham số PCTFREE

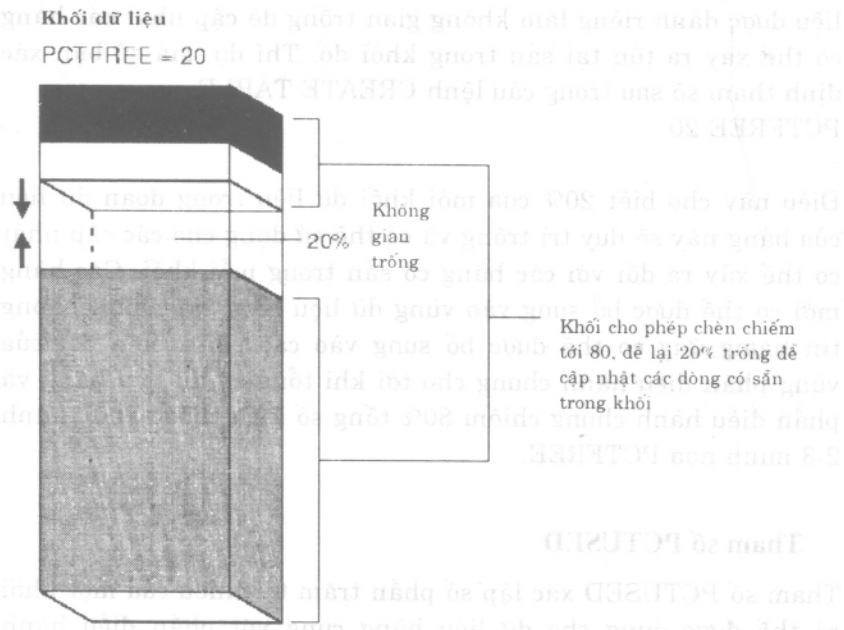
Tham số PCTFREE xác lập số phần trăm tối thiểu của khối dữ liệu được dành riêng làm không gian trống để cập nhật các hàng có thể xảy ra tồn tại sẵn trong khối đó. Thí dụ, giả sử bạn xác định tham số sau trong câu lệnh CREATE TABLE:
PCTFREE 20

Điều này cho biết 20% của mỗi khối dữ liệu trong đoạn dữ liệu của bảng này sẽ duy trì trống và có thể sử dụng cho các cập nhật có thể xảy ra đối với các hàng có sẵn trong mỗi khối. Các hàng mới có thể được bổ sung vào vùng dữ liệu hàng và những thông tin tương ứng có thể được bổ sung vào các phần biến đổi của vùng phần điều hành chung cho tới khi tổng số dữ liệu hàng và phần điều hành chung chiếm 80% tổng số kích thước khối. Hình 2-3 minh họa PCTFREE.

Tham số PCTUSED

Tham số PCTUSED xác lập số phần trăm tối thiểu của một khối có thể được dùng cho dữ liệu hàng cùng với phần điều hành chung trước khi các hàng mới sẽ được bổ sung vào khối ấy. Sau khi một khối dữ liệu được chứa đầy tới mức giới hạn được PCTFREE xác định, thì Oracle xét khối không thể sử dụng để chèn các hàng mới cho tới khi số phần trăm của khối đó giảm xuống dưới tham số PCTUSED. Cho tới khi giá trị này đạt được, thì Oracle dùng không gian trống của khối dữ liệu chỉ cập nhật các hàng được chứa trong khối dữ liệu. Thí dụ, giả sử bạn xác định tham số sau trong câu lệnh CREATE TABLE:
PCTUSED 40

Hình 2-3 PCTFREE

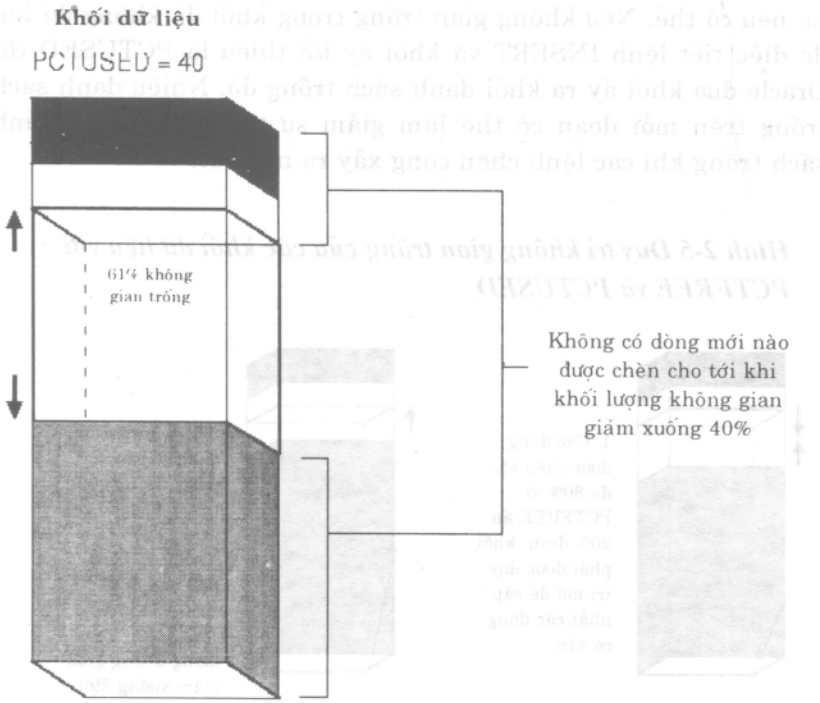


Trong trường hợp này, khối dữ liệu được dùng cho đoạn dữ liệu của bảng này được xét không thể sử dụng để chèn bất kỳ hàng mới nào cho tới khi số lượng không gian đã dùng trong khối ấy giảm xuống 39% hoặc dưới (giả sử khối ấy là không gian đã dùng đạt tới PCTFREE trước đó). Hình 2-4 minh họa điều này.

PCTFREE và PCTUSED hoạt động với nhau như thế nào

PCTFREE và PCTUSED hoạt động với nhau để tối ưu hóa khai thác không gian trong các khối dữ liệu của các khối lưu riêng trong một đoạn dữ liệu. Hình 2-5 minh họa sự tương tác của hai tham số này.

Hình 2-4 PCTUSED

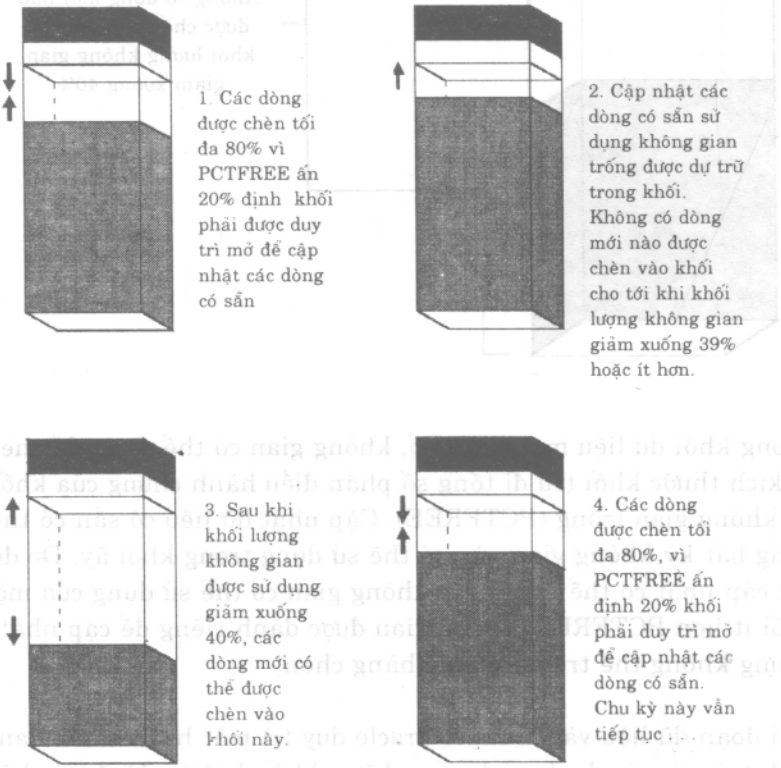


Trong khối dữ liệu mới phân bổ, không gian có thể dùng để chèn là kích thước khối trừ đi tổng số phần điều hành chung của khối và không gian trống (PCTFREE). Cập nhật dữ liệu có sẵn có thể dùng bất kỳ không gian nào có thể sử dụng trong khối ấy. Do đó, các cập nhật có thể làm giảm không gian có thể sử dụng của một khối ít hơn PCTFREE, không gian được dành riêng để cập nhật nhưng không thể truy xuất các hàng chèn.

Mỗi đoạn dữ liệu và chỉ mục, Oracle duy trì một hoặc nhiều danh sách trống - các danh sách của những khối dữ liệu đã được phân bổ cho các khối lưu riêng của đoạn đó và có không gian trống lớn hơn PCTFREE; những khối này có thể sử dụng cho các hàng

chèn. Khi bạn đưa ra lệnh statement, thì Oracle kiểm danh sách trống của bảng cho khối dữ liệu có thể sử dụng đầu tiên và dùng nó nếu có thể. Nếu không gian trống trong khối đó không đủ lớn để điều tiết lệnh INSERT và khối ấy tối thiểu là PCTUSED thì Oracle đưa khối ấy ra khỏi danh sách trống đó. Nhiều danh sách trống trên mỗi đoạn có thể làm giảm sự tranh chấp các danh sách trống khi các lệnh chèn cùng xảy ra một lúc.

Hình 2-5 Duy trì không gian trống của các khối dữ liệu với PCTFREE và PCTUSED



Sau khi bạn đưa ra câu lệnh DELETE hoặc UPDATE, thì Oracle xử lý câu lệnh ấy và kiểm để xem không gian trống được sử dụng trong khối ấy có nhỏ hơn PCTUSED. Nếu nhỏ hơn, khối ấy tiếp cận đầu danh sách giao dịch trống và nó là khối đầu tiên trong các khối có thể sử dụng trong giao dịch đó. Khi giao dịch ấy cam kết, không gian trống trong khối trở nên có thể dùng cho các giao dịch khác.

Tính hiệu lực và nén không gian trống trong khối dữ liệu

Hai kiểu câu lệnh có thể làm tăng không gian trống của một hoặc nhiều khối dữ liệu: các lệnh DELETE và UPDATE cập nhật các giá trị có sẵn thành giá trị nhỏ hơn. Không gian được giải phóng từ những kiểu lệnh này có thể dùng cho các lệnh INSERT kế tiếp dưới các điều khiển sau:

Nếu câu lệnh INSERT trong cùng giao dịch và liên tiếp với câu lệnh giải phóng không gian, thì câu lệnh INSERT có thể sử dụng không gian được tạo ra đó.
Nếu câu lệnh INSERT trong giao dịch tách ra khỏi câu lệnh giải phóng không gian (có lẽ đang được thực hiện bằng một người dùng khác) thì câu lệnh INSERT có thể sử dụng không gian được tạo ra chỉ sau giao dịch khác cam kết và chỉ nếu cần không gian đó.

Không gian được giải phóng có thể hoặc không thể liên tiếp với vùng chính của không gian trống trong một khối dữ liệu. Oracle kết hợp không gian trống của khối dữ liệu chỉ khi (1) lệnh INSERT hoặc UPDATE cố sử dụng một khối chứa không gian trống đủ để chứa được một phần hàng mới và (2) không gian trống được phân đoạn để phần hàng mới không thể được chèn vào một phần liên tiếp của khối ấy. Oracle chỉ nén trong các trường hợp đó, vì nếu không hiệu suất của hệ thống cơ sở dữ liệu sẽ bị giảm do nén liên tiếp không gian trống trong các khối dữ liệu.

Lập chuỗi xích hàng và di trú

Trong hai trường hợp, dữ liệu cho hàng trong một bảng có thể quá lớn để điền một khối dữ liệu đơn. Trong trường hợp thứ nhất, hàng quá lớn để điền một khối dữ liệu khi nó được chèn đầu tiên. Trong trường hợp này, Oracle lưu trữ dữ liệu cho hàng bằng một chuỗi xích các khối dữ liệu được dành riêng cho đoạn đó. Lập chuỗi xích hàng rất thường xảy ra với các hàng lớn, chẳng hạn như các hàng chứa một cột kiểu dữ liệu LONG hoặc LONG RAW. Lập chuỗi xích hàng trong các trường hợp này không thể tránh khỏi.

Lưu ý:

Định dạng hàng và một phần hàng được mô tả ở "Định Dạng Hàng Và Kích Thước".

Tuy nhiên, trong trường hợp hai, hàng ban đầu điền vào dữ liệu được cập nhật để chiều dài hàng tổng quát tăng và không gian trống của khối được điền vào đầy đủ. Trong trường hợp này, Oracle di trú dữ liệu cho hoàn toàn hàng ấy tới khối dữ liệu mới, giả sử toàn bộ hàng ấy có thể điền vào một khối mới. Oracle duy trì phần hàng ban đầu của hàng được di trú để chỉ khối mới chứa hàng được di trú; ROWID của hàng di trú không thay đổi.

Lưu ý:
Xem thêm về ROWID, "Kiểu Dữ Liệu ROWID".

Khi một hàng được lập chuỗi xích hoặc di trú, hiệu suất I/O được kết hợp với hàng này giảm xuống vì Oracle phải thực hiện tác vụ duyệt nhiều khối dữ liệu để truy hồi thông tin cho hàng đó.

Thông tin bổ sung :

Xem Oracle Tunning về Thông Tin Giảm Các Hàng Lập Chuỗi Xích và Di Trú Để Chứng Tỏ Hiệu Suất I/O.

Các khối lưu riêng

Một khối lưu riêng là một đơn vị logic của phân bổ không gian bộ nhớ cơ sở dữ liệu được thiết lập bằng một số khối dữ liệu liên tiếp. Một hoặc nhiều khối lưu riêng lần lượt tạo thành một đoạn. Khi không gian có sẵn trong một đoạn được sử dụng hoàn toàn, Oracle phân bổ khối lưu riêng cho đoạn đó.

Khi các khối lưu riêng được phân bổ

Khi bạn tạo một bảng, Oracle phân bổ đoạn dữ liệu của bạn một khối lưu riêng ban đầu của một số khối dữ liệu riêng. Dù chưa hàng nào được chèn, nhưng các khối dữ liệu Oracle tương ứng với khối lưu riêng ban đầu được dành riêng cho các hàng của bảng đó.

Nếu các khối dữ liệu của khối lưu riêng của một đoạn trở thành đầy và cần nhiều không gian để lưu giữ dữ liệu mới, Oracle tự động phân bổ khối lưu riêng tăng lên cho đoạn đó. Một khối lưu riêng tăng lên là khối lưu riêng kế tiếp của cùng kích thước hoặc lớn hơn khối lưu riêng đã phân bổ trước trong đoạn đó. (Phần kế tiếp giải thích các yếu tố kiểm soát kích thước của các khối lưu riêng gia tăng).

Định vị các mục đích duy trì, khối phần đầu của mỗi đoạn chứa một thư mục của các khối lưu riêng trong đoạn đó.

Các đoạn hồi phục luôn luôn có ít nhất hai khối lưu riêng. Xem thêm, "Các Khối Lưu Riêng Được Sử Dụng Và Phân Bổ Cho Các Đoạn Hồi Phục".

Lưu ý:

Chương này áp dụng cho các điều hoạt liên tiếp, trong đó một xử lý của server phân ngữ và tiến hành câu lệnh SQL. Các khối lưu riêng được phân bổ hơi khác trong các câu lệnh SQL song song

đòi hỏi nhiều xử lý server. Xem "Không Gian Trống Và DDL Song Song".

Xác định số lượng và kích thước của các khối lưu riêng

Các tham số bộ nhớ được biểu thị bằng các thuật ngữ của các khối lưu riêng xác định mỗi đoạn. Các tham số bộ nhớ áp dụng cho tất cả các kiểu đoạn. Chúng điều khiển cách Oracle phân bổ không gian cơ sở dữ liệu trống cho một đoạn định trước như thế nào. Thí dụ, bạn có thể xác định bao nhiêu không gian được dành riêng ban đầu cho một đoạn dữ liệu của bảng hoặc bạn có thể giới hạn số các khối lưu riêng mà bạn có thể phân bổ bằng cách xác định những tham số bộ nhớ của một bảng trong mệnh đề STORAGE của câu lệnh CREATE TABLE.

Các khối lưu riêng được phân bổ như thế nào

Oracle điều khiển sự phân bổ của các khối lưu riêng gia tăng cho một đoạn như sau:

Oracle tìm trong không gian trống (trong không gian bảng chứa đoạn ấy) tập hợp liên tiếp, trống đầu tiên của các khối dữ liệu của kích thước của khối lưu riêng gia tăng hoặc lớn hơn bằng cách sử dụng thuật toán sau:

Oracle tìm tập hợp các khối dữ liệu liên tiếp để so khớp kích thước của khối lưu riêng cùng với một khối để giảm phân đoạn bên trong. (Kích thước được làm tròn thành kích thước của khối lưu riêng tối thiểu cho không gian bảng, nếu cần.) Thí dụ, nếu một khối lưu riêng mới đòi hỏi 19 khối dữ liệu, thì Oracle tìm đúng 20 khối dữ liệu. Nếu các khối lưu riêng mới bằng 5 hoặc ít hơn, thì Oracle không bổ sung thêm khối cho yêu cầu đó.

Nếu so khớp chính xác không tìm thấy, thì sau đó Oracle tìm tập hợp các khối dữ liệu liên tiếp nhiều hơn số lượng cần thiết. Nếu Oracle tìm thấy một

nhóm các khối liên tiếp tối thiểu 5 khối nhiều hơn kích thước của khối lưu riêng cần thiết, nó tách nhóm các khối ấy thành các khối lưu riêng tách ra, một trong các khối lưu riêng có kích thước nó cần. Nếu Oracle tìm thấy một nhóm các khối lớn hơn kích thước nó cần, nhưng nhỏ hơn 5 khối, thì nó phân bổ tất cả các khối liên tiếp thành khối lưu riêng mới.

Trong thí dụ hiện hành, nếu Oracle không tìm thấy một tập hợp gồm đúng 20 khối dữ liệu liên tiếp, thì Oracle tìm một tập hợp các khối dữ liệu liên tiếp lớn hơn 20. Nếu tập hợp thường thấy nhất mà nó tìm thấy chứa 25 khối hoặc hơn, nó tách các khối và phân bổ 20 khối thành khối lưu riêng mới và để 5 khối hoặc hơn duy trì không gian trống. Trái lại, nó phân bổ tất cả các khối (từ 21 tới 24) thành khối lưu riêng.

Nếu Oracle không tìm thấy tập hợp các khối dữ liệu liên tiếp bằng hoặc lớn hơn, thì nó kết hợp bằng bất kỳ khối dữ liệu kế cận và trống vào không gian bảng tương ứng để tạo thành các tập hợp các khối dữ liệu liên tiếp lớn hơn. (Nền SMON còn xử lý không gian kế cận định kỳ). Sau khi kết các khối dữ liệu của không gian bảng lại, Oracle thực hiện những tìm kiếm được mô tả trong 1a và 1b.

Nếu một khối lưu riêng không thể được phân bổ sau lần tìm thứ hai, Oracle cố lập kích thước lại các tập tin bằng phần mở rộng tự động. Nếu Oracle không thể lập kích thước lại các tập tin ấy, nó sẽ tạo ra một lỗi.

Một khi Oracle tìm thấy và phân bổ không gian trống cần thiết trong không gian bảng, nó phân bổ phần không gian trống tương ứng với kích thước của khối lưu riêng tăng thêm. Nếu Oracle tìm thấy số lượng không gian trống lớn hơn đòi hỏi cho khối lưu riêng

ấy, Oracle để phần còn lại làm không gian trống (không dưới 5 khối liên tiếp).
Oracle cập nhật phần đầu đoạn và từ điển dữ liệu để cho thấy một khối lưu riêng đã được phân bổ và không gian đã phân bổ không còn trống nữa.

Các khối của một khối lưu riêng mới được phân bổ, dù chúng trống, nhưng vẫn còn dữ liệu cũ. Thường, Oracle định dạng các khối của một khối lưu riêng mới được phân bổ khi nó bắt đầu sử dụng khối lưu riêng ấy, chỉ bằng yêu cầu (bắt đầu với các khối trên danh sách đoạn trống). Tuy nhiên, trong một vài trường hợp, chẳng hạn như khi một người điều hành buộc phân bổ một khối lưu riêng gia tăng với tùy chọn ALLOCATE EXTENT của một câu lệnh ALTER TABLE hoặc ALTER CLUSTER, Oracle định dạng các khối của khối lưu riêng khi nó phân bổ khối lưu riêng ấy.

Khi các khối lưu riêng được thôi phân bổ

Nói chung, các khối lưu riêng của một đoạn không cho trở lại không gian bảng cho tới khi bạn giảm đối tượng có dữ liệu được lưu trữ trong đoạn ấy (bằng cách sử dụng câu lệnh DROP TABLE hoặc DROP CLUSTER). Các trường hợp ngoại trừ sau:

Chủ nhân của bảng hoặc cụm liên kết hay một người dùng với đặc quyền DELETE ANY, có thể cắt bớt bảng hoặc cụm liên kết bằng câu lệnh TRUNCATE...DROP STORAGE.
Định kỳ, Oracle có thể thôi phân bổ một hoặc nhiều khối lưu riêng của một đoạn hồi phục nếu nó có kích thước OPTIMAL xác định.

Người điều hành cơ sở dữ liệu (DBA) có thể thôi phân bổ những khối lưu riêng chưa sử dụng bằng cách dùng cú pháp SQL sau:

ALTER TABLE table_name DEALLOCATE UNUSED;

Khi các khối lưu riêng trống, Oracle cập nhật từ điển dữ liệu để phản ánh các khối lưu riêng thu hồi làm không gian có thể sử dụng. Dữ liệu bất kỳ trong các khối của các khối lưu riêng trống trở nên không thể truy xuất và Oracle xóa dữ liệu khi các khối được sử dụng tiếp theo cho các khối lưu riêng khác.

Thông tin bổ sung :

Xem Hướng Dẫn Của Nhà Điều Hành Oracle Và Tham Chiếu Oracle SQL về các khối lưu riêng thôi phân bổ.

Các bảng không có cụm liên kết

Bao lâu bảng không có cụm liên kết tồn tại hoặc cho tới khi bạn cắt giảm bảng, bất kỳ khối dữ liệu nào được phân bổ vào đoạn dữ liệu của nó vẫn được phân bổ cho bảng ấy. Oracle chèn các hàng mới vào khối nếu có đủ chỗ trống. Dù bạn xóa tất cả các hàng của một bảng, Oracle không giành lại các khối dữ liệu để các đối tượng khác sử dụng trong không gian bảng.

Sau khi bạn giảm bảng, không gian này có thể được thu hồi khi các khối lưu riêng khác đòi hỏi không gian trống.

Oracle thu hồi tất cả các khối lưu riêng dữ liệu và các đoạn chỉ mục của nó cho các không gian bảng mà chúng tồn tại và tạo ra các khối lưu riêng có thể sử dụng cho các đối tượng khác trong không gian bảng. Tiếp theo, khi

các đoạn khác đòi hỏi các khối lưu riêng lớn hơn, thì Oracle nhận dạng và kết hợp các khối lưu riêng được thu hồi liên tiếp để tạo thành các khối lưu riêng lớn hơn đã yêu cầu.

Các bảng có cụm liên kết

Các bảng có cụm liên kết lưu trữ thông tin của chúng trong đoạn dữ liệu cho cụm liên kết ấy. Do đó, nếu bạn giảm một bảng trong một cụm liên kết, đoạn dữ liệu ấy vẫn duy trì cho các bảng khác trong cụm liên kết ấy và không có khối lưu riêng nào được thôi phân bổ. Bạn còn có thể cắt bớt các khối lưu riêng trống (ngoại trừ các cụm liên kết có trị số).

Các ảnh chụp nhanh và các nhật ký hoạt động ảnh chụp nhanh

Oracle thôi phân bổ các khối lưu riêng của các ảnh chụp nhanh và các nhật ký hoạt động ảnh chụp nhanh theo phương thức như đối với các bảng có cụm liên kết và không có cụm liên kết.

Thông tin bổ sung: .

Xem Oracle8 Replication về Các Ảnh Chụp Nhanh và Các Nhật Ký Hoạt Động Ảnh Chụp Nhanh.

Các chỉ mục

Tất cả các khối lưu riêng được phân bổ vào một đoạn chỉ mục vẫn được phân bổ bao lâu chỉ mục ấy còn tồn tại. Khi bạn giảm chỉ mục ấy hoặc bảng kết hợp hay cụm liên kết, Oracle thu hồi các khối lưu riêng cho các sử dụng khác trong không gian bảng đó.

Các đoạn hồi phục

Oracle định kỳ kiểm xem các đoạn hồi phục của cơ sở dữ liệu có phát triển lớn hơn kích thước tối ưu của chúng không. Nếu một đoạn hồi phục lớn hơn kích thước tối ưu (nghĩa là có quá nhiều khối lưu riêng), thì Oracle tự động thôi phân hoạch một hoặc nhiều khối lưu riêng khỏi đoạn hồi phục. Xem "Các Khối Lưu Riêng Được Thôi phân bổ Khỏi Đoạn Hồi Phục Như Thế Nào".

Các đoạn tạm thời

Khi Oracle thực hiện xong một câu lệnh đòi hỏi một đoạn tạm thời, Oracle tự động giảm đoạn tạm thời và đưa các khối lưu riêng được phân bổ cho đoạn đó trở lại không gian bảng được kết hợp. Một sắp xếp đơn giản phân bổ đoạn tạm thời riêng của nó, trong không gian bảng tạm thời của người dùng đưa lệnh ấy, và sau đó đưa các khối lưu riêng trở lại không gian bảng.

Tuy nhiên, nhiều sắp xếp có thể sử dụng các đoạn sắp xếp trong không gian bảng tạm thời được thiết kế riêng cho các sắp xếp ấy. Những sắp xếp này chỉ được phân bổ một lần cho môi trường làm việc ấy và chúng không trở lại sau sắp xếp ấy nhưng vẫn có thể sử dụng cho các sắp xếp khác. Xem thêm "Các Đoạn Tạm Thời".

Các đoạn

Một đoạn là một tập hợp các khối lưu riêng chứa tất cả dữ liệu cho một cấu trúc bộ nhớ logic trong một không gian bảng. Thí dụ, đối với mỗi bảng, Oracle phân bổ một hoặc nhiều khối lưu riêng để tạo thành đoạn dữ liệu của bảng đó; đối với mỗi chỉ mục, Oracle phân bổ một hoặc nhiều khối lưu riêng để tạo thành đoạn chỉ mục của nó.

Các cơ sở dữ liệu Oracle dùng bốn kiểu đoạn:
Các đoạn dữ liệu
Các đoạn chỉ mục
Các đoạn tạm thời
Các đoạn hồi phục

Các phần sau đề cập mỗi kiểu đoạn.

Các đoạn dữ liệu

Mỗi bảng không có cụm liên kết hoặc phân hoạch và mỗi cụm liên kết trong cơ sở dữ liệu Oracle có một đoạn dữ liệu đơn để lưu giữ tất cả dữ liệu của nó. Oracle tạo đoạn dữ liệu này khi bạn tạo bảng có cụm liên kết hoặc cụm liên kết với lệnh CREATE.

Các tham số bộ nhớ cho bảng không có cụm liên kết hoặc cụm liên kết xác định các khối lưu riêng của đoạn dữ liệu của nó được phân bổ. Bạn có thể xác lập các tham số trực tiếp với lệnh CREATE hoặc ALTER thích hợp. Những tham số bộ nhớ này ảnh hưởng đến hiệu quả của truy hồi dữ liệu và bộ nhớ cho đoạn dữ liệu được kết hợp với đối tượng ấy.

Lưu ý:

Oracle tạo các đoạn cho ảnh chụp nhanh và các nhật ký hoạt động ảnh chụp nhanh theo cách thức như đối với các bảng có cụm liên kết và không có cụm liên kết.

Thông tin bổ sung:
Xem Oracle8 Replication về Những Thông Tin Ảnh Chụp Nhanh và Các Nhật Ký Hoạt Động Ảnh Chụp Nhanh; Oracle8 SQL về thông tin trên Các Lệnh CREATE và ALTER.

Các đoạn chỉ mục

Mọi chỉ mục trong cơ sở dữ liệu Oracle có một đoạn chỉ mục đơn để lưu giữ tất cả dữ liệu của nó. Oracle tạo đoạn chỉ mục ấy cho chỉ mục khi bạn đưa ra lệnh CREATE INDEX. Trong lệnh này, bạn có thể xác định các tham số bộ nhớ cho các khối lưu riêng của đoạn chỉ mục và không gian bảng trong đó để tạo đoạn chỉ mục. (Các đoạn của bảng và một chỉ mục được kết hợp với nó không phải chiếm giữ cùng không gian bảng). Xác lập các tham số bộ nhớ trực tiếp ảnh hưởng đến hiệu quả của truy hồi dữ liệu và bộ nhớ.

Các đoạn tạm thời

Khi xử lý các truy vấn, Oracle thường đòi hỏi không gian làm việc tạm thời cho các giai đoạn trung gian của câu lệnh SQL phân ngữ và thực hiện. Oracle tự động phân bổ không gian đĩa này được gọi là một đoạn tạm thời. Tiêu biểu, Oracle đòi hỏi một đoạn tạm thời bằng một vùng làm việc để sắp xếp. Oracle không tạo ra một đoạn nếu điều hoạt sắp xếp tìm thấy cách khác để thực hiện điều hoạt dùng các chỉ mục.

Các điều hoạt đòi hỏi các đoạn tạm thời

Các lệnh có thể đòi hỏi sử dụng một đoạn tạm thời:
CREATE INDEX
SELECT ... ORDER BY
SELECT DISTINCT ...
SELECT ... GROUP BY
SELECT ... UNION
SELECT ... INTERSECT
SELECT ... MINUS

Một số kết nối không có chỉ mục và các truy vấn con có tương quan cũng có thể đòi hỏi sử dụng một đoạn tạm thời. Thí dụ, nếu một truy vấn chứa mệnh đề DISTINCT, GROUP BY và ORDER BY, Oracle có thể đòi hỏi tối đa hai đoạn tạm thời. Nếu các ứng dụng thường đưa ra các lệnh trong danh sách trên, thì người điều hành có thể cần cải tiến hiệu suất bằng cách điều chỉnh tham số khởi hoạt SORT_AREA_SIZE.

Thông tin bổ sung:
Xem Oracle8 Reference về thông tin SORT_AREA_SIZE và Các Tham Số Khởi Hoạt Khác.

Các đoạn tạm thời được phân bổ như thế nào

Oracle phân bổ các đoạn tạm thời theo nhu cầu trong một phiên người dùng, trong không gian bảng tạm thời của người dùng phát hành lệnh ấy. Bạn xác định không gian bảng này bằng lệnh CREATE USER hoặc lệnh ALTER USER sử dụng tùy chọn TEMPORARY TABLESPACE. Nếu không gian bảng tạm thời không được xác định cho người dùng, thì không gian bảng tạm thời mặc định là không gian bảng SYSTEM. Những đặc điểm của bộ nhớ mặc định chứa không gian bảng xác định các khối lưu riêng của đoạn tạm thời.

Oracle bỏ các đoạn tạm thời khi câu lệnh ấy hoàn tất. Bởi vì phân bổ và thôi phân bổ của các đoạn tạm thời xảy ra thường xuyên, tạo một không gian bảng rộng cho các đoạn tạm thời là hợp lý. Bằng cách thực hiện như vậy, bạn có thể phân phối các I/O qua thiết bị đĩa và bạn có thể tránh sự phân đoạn của SYSTEM và trái lại các không gian bảng khác sẽ lưu giữ các đoạn tạm thời.

Xem thêm thông tin về Lệnh Gán Không Gian Bảng Tạm Thời Của Người Dùng, xem "Điều khiển Truy Xuất Cơ Sở Dữ Liệu".

Các mục nhập cho các thay đổi đối với các đoạn tạm thời được sử dụng cho các điều hoạt sắp xếp không được lưu trữ trong nhật ký Redo Log, ngoại trừ cho các điều hoạt quản lý không gian trên đoạn tạm thời.

Các đoạn hồi phục

Mỗi cơ sở dữ liệu chứa một hoặc nhiều đoạn hồi phục. Một đoạn hồi phục ghi lại các giá trị cũ của dữ liệu đã bị thay đổi bởi mỗi giao dịch (dù được cam kết hay không). Các đoạn hồi phục được sử dụng để cung cấp đọc liên tục để hồi phục các giao dịch và phục hồi cơ sở dữ liệu. Để có thông tin cụ thể về các đoạn hồi phục thực hiện chức năng trong các trường hợp này như thế nào, xem các phần thích hợp của tài liệu này:

Chủ đề	Phần
Đọc nhất quán	Điều khiển đồng thời nhiều phiên bản
Hồi phục giao dịch	Hội phục các giao dịch
Phục hồi cơ sở dữ liệu	Các đoạn hồi phục và hồi phục

Nội dụng của một đoạn hồi phục

Thông tin trong một đoạn hồi phục gồm nhiều mục nhập hồi phục. Trong các thông tin khác, mục nhập hồi phục gồm thông tin khối (số hiệu tập tin và ID khối tương ứng với dữ liệu đã bị thay đổi) và dữ liệu như nó đã tồn tại trước khi một điều hoạt trong một giao dịch. Oracle liên kết các mục nhập hồi phục cho cùng giao dịch, vì

vậy các mục nhập có thể tìm thấy dễ dàng nên cần thiết cho hồi phục giao dịch.

Cả người dùng cơ sở dữ liệu hoặc người điều hành không có thể truy xuất hoặc đọc các đoạn hồi phục; chỉ Oracle có thể ghi và đọc chúng. (Chúng được SYS người dùng sở hữu, bất kể người dùng nào tạo ra chúng).

Ghi sổ các mục nhập hồi phục

Các mục nhập hồi phục thay đổi các khối dữ liệu trong đoạn hồi phục và Oracle ghi tất cả những thay đổi đối với các khối dữ liệu, kể cả các mục nhập hồi phục trong nhật ký Redo Log. Bản ghi thứ hai này của những thông tin hồi phục rất quan trọng đối với các giao dịch hoạt động (chưa cam kết hoặc đã hồi phục) vào thời điểm hệ thống bị sự cố. Nếu sự cố hệ thống xảy ra, Oracle tự động phục hồi thông tin của đoạn hồi phục, kể cả các mục nhập hồi phục cho những giao dịch hoạt động là thành phần của môi trường làm việc hoặc phục hồi môi trường. Một khi phục hồi hoàn tất, Oracle thực hiện các hồi phục thực của những giao dịch chưa được cam kết hoặc chưa được hồi phục vào thời điểm sự cố hệ thống.

Khi cần thông tin hồi phục

Đối với mỗi đoạn hồi phục, Oracle duy trì danh sách bảng giao dịch của tất cả những giao dịch sử dụng đoạn kết hợp và các mục nhập hồi phục cho mỗi thay đổi được những giao dịch này thực hiện. Oracle sử dụng các mục nhập hồi phục trong đoạn hồi phục để thực hiện hồi phục giao dịch và tạo các kết quả đọc nhất quán cho các truy vấn.

Các đoạn hồi phục ghi dữ liệu trước khi thay đổi trên mỗi cơ sở giao dịch. Đối với mỗi giao dịch, Oracle liên kết mỗi thay đổi mới thành thay đổi trước. Nếu bạn phải hồi phục giao dịch ấy, Oracle áp dụng các thay đổi trong một chuỗi xích vào các khối dữ liệu theo thứ tự để phục hồi dữ liệu thành trạng thái trước đây.

Tương tự, khi Oracle cung cấp tập hợp những kết quả đọc nhất quán cho một truy vấn, nó có thể dùng thông tin trong các đoạn hồi phục để tạo một tập hợp dữ liệu nhất quán đối với một điểm đơn đúng lúc.

Các giao dịch và các đoạn hồi phục

Mỗi lần một giao dịch người dùng bắt đầu, giao dịch ấy được gán vào một đoạn hồi phục một trong hai cách sau:
Oracle có thể gán một giao dịch tự động vào một đoạn hồi phục có thể sử dụng. Lệnh gán giao dịch xảy ra khi bạn đưa ra câu lệnh DML đầu tiên hoặc DDL trong giao dịch ấy. Oracle không bao giờ gán các giao dịch chỉ đọc (các giao dịch chỉ chứa các truy vấn) đối với một đoạn hồi phục, bất kể giao dịch bắt đầu bằng câu lệnh SET TRANSACTION READ ONLY hay không.
Ứng dụng có thể gán một giao dịch rõ ràng với một đoạn hồi phục riêng biệt. Vào lúc bắt đầu giao dịch, một Developer ứng dụng hoặc người dùng có thể xác định đoạn hồi phục riêng biệt mà Oracle sẽ dùng khi thực hiện giao dịch ấy. Điều này cho phép Developer ứng dụng hoặc người dùng lựa một đoạn hồi phục lớn hoặc nhỏ phù hợp với giao dịch ấy.

Trong thời gian giao dịch, tiến trình người dùng kết hợp chỉ ghi thông tin hồi phục vào đoạn hồi phục được gán.

Khi bạn cam kết một giao dịch, Oracle giải phóng thông tin hồi phục nhưng không hủy nó ngay. Thông tin còn lại trong đoạn hồi phục để tạo các khung nhìn đọc nhất quán của dữ liệu thích hợp cho các truy vấn đã bắt đầu trước khi giao dịch cam kết. Để bảo đảm dữ liệu hồi phục có thể sử dụng tối đa cho các khung nhìn ấy, Oracle ghi các khối lưu riêng của các đoạn hồi phục liên tiếp. Khi khối lưu riêng cuối cùng của đoạn hồi phục trở thành đầy, khối lưu riêng thứ nhất trong đoạn ấy. Giao dịch chạy lâu (không hoạt động hoặc hoạt động) có thể đòi hỏi một khối lưu riêng mới được phân bổ cho đoạn hồi phục ấy. Xem Hình 2-6, Hình 2-7 và Hình 2-8 thêm các thông tin về những giao dịch dùng các khối lưu riêng của một đoạn hồi phục.

Mỗi đoạn hồi phục có thể xử lý một số giao dịch cố định từ một môi trường làm việc. Nếu bạn không gán các giao dịch rõ ràng thành các đoạn hồi phục cụ thể, Oracle phân phối các giao dịch hoạt động qua các đoạn hồi phục có thể sử dụng để tất cả các đoạn hồi phục được gán gần bằng số giao dịch. Phân phối không phụ thuộc vào kích thước của các đoạn hồi phục có thể sử dụng. Do đó, trong các môi trường nơi tất cả các giao dịch phát sinh cùng số lượng thông tin hồi phục, thì tất cả các đoạn hồi phục có thể có cùng kích thước.

Thông tin bổ sung:

Số giao dịch mà một đoạn hồi phục có thể xử lý là chức năng hệ điều hành xác định của kích thước khối dữ liệu. Xem tài liệu hệ điều hành xác định của bạn để biết thêm thông tin.

Các khối lưu riêng được dùng và phân bổ cho các đoạn hồi phục như thế nào

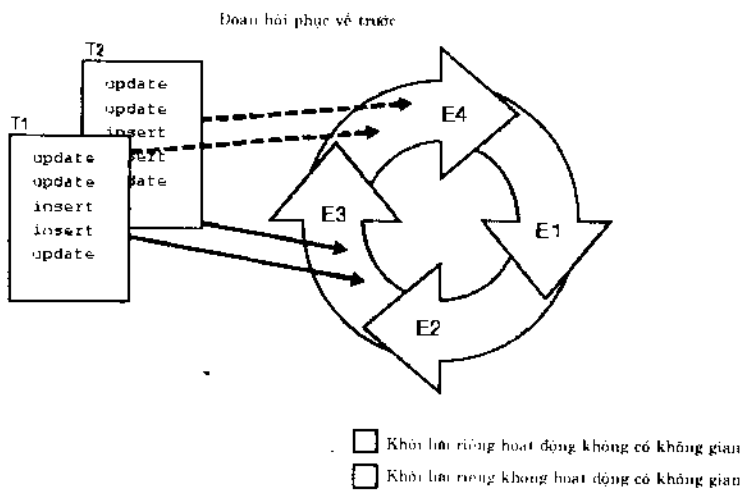
Khi bạn tạo một đoạn hồi phục, bạn có thể xác định những tham số bộ nhớ để điều khiển sự phân bổ của các khối lưu riêng cho đoạn đó. Mỗi đoạn hồi phục phải có ít nhất hai khối lưu riêng được phân bổ.

Một giao dịch ghi liên tiếp một đoạn hồi phục đơn. Mỗi giao dịch ghi chỉ một khối lưu riêng của đoạn hồi phục vào bất kỳ lúc nào. Nhiều giao dịch hoạt động có thể ghi đồng thời một đoạn hồi phục chẳn cùng khối lưu riêng của đoạn hồi phục; tuy nhiên, mỗi khối dữ liệu trong khối lưu riêng của đoạn hồi phục có thể chứa những thông tin chỉ một giao dịch đơn.

Khi giao dịch không còn không gian trong khối lưu riêng hiện hành và cần tiếp tục ghi, Oracle tìm một khối lưu riêng có thể sử dụng của cùng đoạn hồi phục bằng một trong hai cách:
Nó có thể sử dụng lại một khối lưu riêng đã được phân bổ vào đoạn hồi phục ấy.
Nó có thể đạt được (và phân bổ) một khối lưu riêng mới cho đoạn hồi phục ấy.
Giao dịch thứ nhất cần đạt được nhiều không gian hồi phục để kiểm khối lưu riêng kế tiếp của đoạn hồi phục. Nếu khối lưu riêng kế tiếp không chứa thông tin từ một giao dịch hoạt động, thì Oracle tạo cho nó là khối lưu riêng hiện hành và tất cả những giao dịch cần nhiều không gian hơn kể từ đó có thể ghi thông tin hồi phục vào khối lưu riêng hiện hành mới. Hình 2-6 minh họa hai giao dịch, T1 và T2 bắt đầu ghi vào khối lưu riêng thứ ba (E3) và tiếp tục ghi vào khối lưu riêng thứ tư (E4) của đoạn hồi phục.

Khi các giao dịch tiếp tục ghi và điền khối lưu riêng hiện hành, Oracle kiểm khối lưu riêng kế tiếp đã được phân bổ cho đoạn hồi phục ấy để xác định xem nó có thể sử dụng không. Trong Hình 2-7, khi E4 đầy hoàn toàn, T1 và T2 tiếp tục ghi thêm vào khối lưu riêng kế tiếp được phân bổ cho đoạn hồi phục có thể sử dụng; trong hình này, E1 là khối lưu riêng kế tiếp. Hình này cho thấy tính chất chu kỳ của khối lưu riêng sử dụng các đoạn hồi phục.

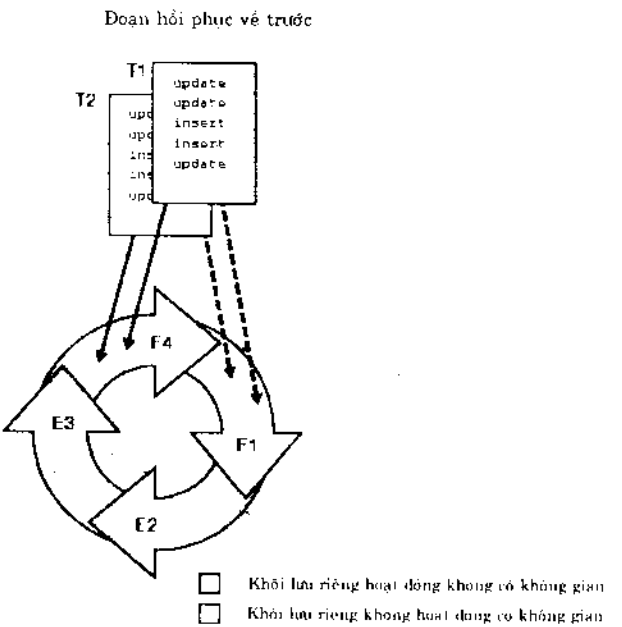
Hình 2-6 Sử dụng các khối lưu riêng được phân bổ trong đoạn hồi phục



Để tiếp tục ghi thông tin hồi phục cho một giao dịch, Oracle luôn luôn cố sử dụng lại khối lưu riêng kế tiếp trong vòng đầu tiên. Tuy nhiên, khối lưu riêng kế tiếp chứa dữ liệu từ một đoạn hồi phục cho tới số khối lưu riêng đạt tới giá trị được xác lập cho tham số MAXEXTENTS bộ nhớ của đoạn hồi phục.

Hình 2-8 trình bày một khối lưu riêng mới được phân bổ cho một đoạn hồi phục. Các giao dịch chưa được cam kết tiến hành thời gian dài (không hoạt động hoặc hoạt động hay các giao dịch phân phối không chắc chắn liên tục). Vào thời điểm này, chúng đang ghi vào khối lưu riêng thứ tư, E4, trong đoạn hồi phục. Tuy nhiên, khi E4 hoàn toàn đầy, các giao dịch không thể tiếp tục ghi thêm vào khối lưu riêng kế tiếp liên tục. Do đó, Oracle phân bổ một khối lưu riêng mới, E5 cho đoạn hồi phục này và các giao dịch tiếp tục ghi vào khối lưu riêng mới này.

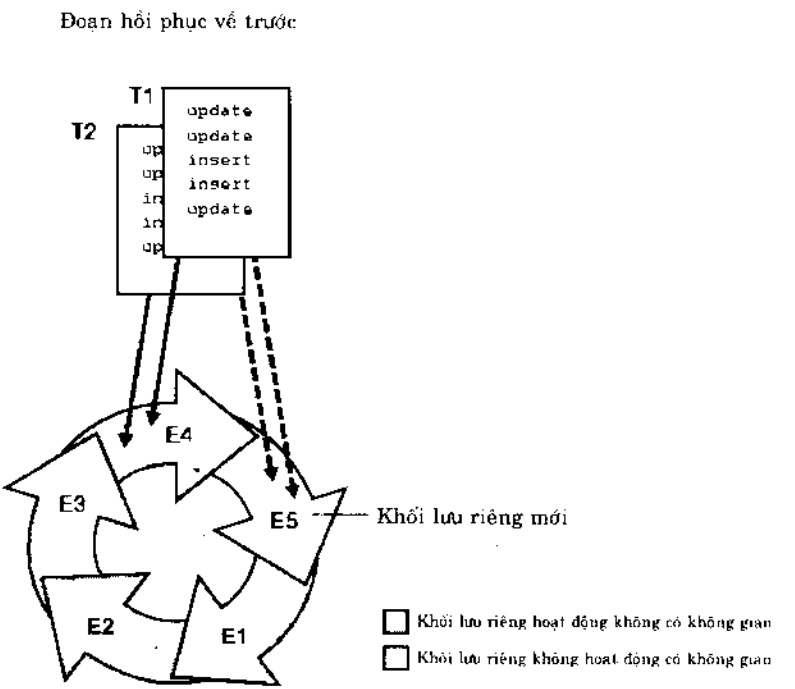
Hình 2-7 Sử dụng chu kỳ của các khối lưu riêng được phân bổ trong một đoạn hồi phục



Các khối lưu riêng được thôi phân bổ từ một đoạn hồi phục như thế nào

Khi bạn giảm một đoạn hồi phục, Oracle đưa tất cả các khối lưu riêng của đoạn hồi phục trở lại không gian bảng của nó. Các khối lưu riêng được cho trở lại sau đó có thể sử dụng cho các đoạn khác trong không gian bảng ấy.

Hình 2-8 Sự phân bổ của khối lưu riêng mới cho một đoạn hồi phục



Khi bạn tạo hoặc thay đổi một đoạn hồi phục, bạn có thể dùng tham số bộ nhớ OPTIMAL (chỉ áp dụng cho các đoạn hồi phục) để xác định kích thước tối ưu của đoạn ấy bằng byte. Nếu một giao dịch cần tiếp tục ghi thông

tin hồi phục từ một khối lưu riêng tới một khối lưu riêng khác trong đoạn hồi phục ấy, Oracle so sánh kích thước hiện hành của đoạn hồi phục với kích thước tối ưu của đoạn ấy. Nếu đoạn hồi phục lớn hơn kích thước tối ưu của nó và các khối lưu riêng theo sau ngay khối lưu riêng vừa được điền không hoạt động, Oracle thôi phân phối các khối lưu riêng không hoạt động liên tiếp từ đoạn hồi phục cho tới khi tổng kích thước của đoạn hồi phục bằng hoặc gần bằng nhưng không nhỏ hơn kích thước tối ưu của nó. Oracle luôn luôn giải phóng các khối lưu riêng cũ nhất khi những khối này ít có thể được sử dụng để đọc liên tiếp.

Xác lập OPTIMAL của đoạn hồi phục không thể nhỏ hơn không gian kết hợp được phân bổ cho số lượng tối thiểu các khối lưu riêng đối với đoạn ấy:

(INITIAL + NEXT + NEXT + ... up to MINEXTENTS) bytes

SYSTEM đoạn hồi phục

Oracle tạo một đoạn hồi phục ban đầu được gọi SYSTEM bất kỳ khi nào cơ sở dữ liệu được tạo. Đoạn này trong không gian bảng SYSTEM và sử dụng những tham số mặc định của không gian báng. Bạn không thể giảm đoạn hồi phục SYSTEM. Một môi trường làm việc luôn luôn chiếm đoạn hồi phục SYSTEM bổ sung vào bất kỳ đoạn hồi phục khác mà nó cần.

Nếu có nhiều đoạn hồi phục, Oracle cố dùng đoạn hồi phục SYSTEM chỉ cho các giao dịch hệ thống đặc biệt và phân phối các giao dịch người dùng trong các đoạn hồi phục khác nhau; nếu có quá nhiều giao dịch cho các đoạn hồi phục không phải SYSTEM, Oracle dùng đoạn SYSTEM nếu cần. Nói chung, sau khi tạo cơ sở dữ liệu, bạn phải

tạo ít nhất một đoạn hồi phục bổ sung trong không gian bảng SYSTEM.

Các môi trường làm việc Oracle và các kiểu đoạn hồi phục

Khi một môi trường làm việc Oracle mở một cơ sở dữ liệu, nó phải giành được một hoặc nhiều các đoạn hồi phục để môi trường làm việc ấy có thể xử lý thông tin hồi phục được các giao dịch liên tiếp tạo ra. Một môi trường làm việc có thể giành được cả các đoạn riêng và chung. Một đoạn hồi phục riêng được một môi trường làm việc giành được rõ ràng khi môi trường làm việc ấy mở một cơ sở dữ liệu. Các đoạn hồi phục về trạng thái cũ chung tạo thành một vùng chia của các đoạn hồi phục mà bất kỳ môi trường làm việc nào chiếm được đoạn hồi phục có thể sử dụng.

Số đoạn riêng và chung bất kỳ có thể tồn tại trong một cơ sở dữ liệu. Một môi trường làm việc mở một cơ sở dữ liệu, môi trường làm việc ấy cố giành được một hoặc nhiều đoạn hồi phục theo đúng những nguyên tắc sau:

Môi trường làm việc phải giành được ít nhất một đoạn hồi phục. Nếu môi trường làm việc ấy chỉ là môi trường làm việc truy xuất cơ sở dữ liệu, thì nó giành được đoạn SYSTEM. Nếu môi trường làm việc ấy là một trong nhiều môi trường làm việc truy xuất cơ sở dữ liệu trong một Oracle Parallel Server, thì nó giành được đoạn hồi phục SYSTEM và ít nhất một đoạn hồi phục khác. Nếu không, Oracle cho trở lại một lỗi và môi trường làm việc ấy không thể mở cơ sở dữ liệu đó.

Môi trường làm việc luôn luôn cố giành được ít nhất số đoạn hồi phục bằng với thương số của các giá trị đối với các tham số khởi hoạt sau:

CEIL(TRANSACTIONS/TRANSACTIONS_PER_ROLLBACK_SEGMENT)

CEIL là một hàm SQL cho trở lại số nguyên nhỏ nhất hoặc bằng với dữ liệu nhập bằng số. Trong thí dụ trên, nếu TRANSACTIONS bằng 155 và TRANSACTIONS_PER_ROLLBACK_SEGMENT bằng 10, thì môi trường làm việc sẽ cố giành được ít nhất 16 đoạn hồi phục. (tuy nhiên, một môi trường làm việc có thể mở cơ sở dữ liệu dù môi trường làm việc ấy không thể giành được số đoạn hồi phục được phép chia trên cho).

Lưu ý:

Tham số TRANSACTIONS_PER_ROLLBACK_SEGMENT không giới hạn số giao dịch có thể dùng đoạn hồi phục. Đúng hơn, nó quyết định số đoạn hồi phục mà một môi trường làm việc cố gắng chiếm khi mở một cơ sở dữ liệu.

Sau khi giành được một đoạn hồi phục SYSTEM, môi trường làm việc kế tiếp cố giành được tất cả các đoạn hồi phục riêng bằng tham số ROLLBACK_SEGMENTS của môi trường làm việc ấy xác định. Nếu một môi trường làm việc trong một Oracle Parallel Server mở một cơ sở dữ liệu và cố giành được một đoạn hồi phục riêng, tiếp theo bằng một môi trường làm việc khác đòi hỏi, thì môi trường làm việc thứ hai cố giành được được đoạn hồi phục bị một lỗi khi khởi động. Một lỗi còn được cho trở lại nếu môi trường làm việc cố giành được một đoạn riêng không tồn tại. Nếu môi trường làm việc ấy đã giành được các đoạn hồi phục đủ bằng số 3, thì không cần hoạt động nào nữa. Tuy nhiên, nếu một môi trường

làm việc hệ điều hành nhiều đoạn hồi phục, môi trường làm việc ấy cố giành được các đoạn hồi phục chung.

Một khi môi trường làm việc đòi hỏi một đoạn hồi phục chung, môi trường làm việc khác không thể dùng đoạn đó cho tới khi đoạn hồi phục ấy được ngoại tuyến hoặc môi trường làm việc ấy đòi hỏi đoạn hồi phục được đóng lại.

Cơ sở dữ liệu được Oracle Parallel Server sử dụng tùy chọn chỉ có thể có các đoạn chung và không có các đoạn riêng, bao lâu số đoạn trong cơ sở dữ liệu đủ cao để bảo đảm mỗi môi trường làm việc mở cơ sở dữ liệu ấy có thể giành được ít nhất hai đoạn hồi phục, trong đó một là đoạn hồi phục SYSTEM. Tuy nhiên, khi dùng Oracle Parallel Server, bạn có thể dùng các đoạn hồi phục riêng.

Các trạng thái đoạn hồi phục

Một đoạn hồi phục luôn luôn ở một trong nhiều trạng thái, tùy theo ngoại tuyến được một môi trường làm việc giành được, liên quan đến một giao dịch chưa được giải quyết, theo nhu cầu phục hồi hoặc bỏ đi. Trạng thái của đoạn hồi phục quyết định xem nó có thể được dùng trong các giao dịch không, cũng như các thủ tục điều hành nào mà một DBA có thể thực hiện dựa vào nó.

Các trạng thái đoạn hồi phục:

Ngoại tuyến	Chứa (trực tuyến) bất kỳ môi trường làm việc nào giành được
Trực tuyến	Được một môi trường làm việc giành được (trực tuyến); có thể chứa dữ liệu từ các giao dịch hoạt động.
Cần phục hồi	Chứa dữ liệu từ các giao dịch chưa được cam kết hoặc bị sự cố.

Có thể sử dụng một phần	Chứa dữ liệu từ một giao dịch không chắc chắn (nghĩa là, giao dịch phân phối chưa được giải quyết)
Không hợp lệ	Đã được bỏ đi (một khi không gian được phân bổ vào đoạn hồi phục sau đó sẽ được dùng khi một đoạn hồi phục mới được tạo).

Bảng từ điển dữ liệu DBA_ROLLBACK_SEGS liệt kê trạng thái của mỗi đoạn hồi phục theo cùng thông tin hồi phục khác. Hình 2-9 trình bày một đoạn hồi phục di chuyển từ một trạng thái này sang một trạng thái khác như thế nào.

PARTLY AVAILABLE và các đoạn hồi phục NEEDS RECOVERY

Các trạng thái PARTLY AVAILABLE và NEEDS RECOVERY rất giống nhau. Một đoạn hồi phục ở trạng thái luôn luôn chứa dữ liệu từ một giao dịch chưa được giải quyết.

Đoạn hồi phục PARTLY AVAILABLE được sử dụng bằng một giao dịch phân phối không chắc chắn không thể giải quyết được vì sự cố mạng. Đoạn hồi phục NEEDS RECOVERY được sử dụng bởi một giao dịch (cục bộ hoặc phân phối) không thể giải quyết vì một sự cố môi trường cục bộ, chẳng hạn như mất tập tin dữ liệu hoặc bị sai sót hay chính nó bị sai sót.

Oracle hoặc DBA có thể đưa đoạn hồi phục PARTLY AVAILABLE trực tuyến. Trái lại, bạn phải lựa một đoạn hồi phục NEEDS RECOVERY là OFFLINE trước khi nó có thể được sang trực tuyến. (Nếu bạn phục hồi cơ sở dữ liệu và do đó giải quyết giao dịch ấy, Oracle

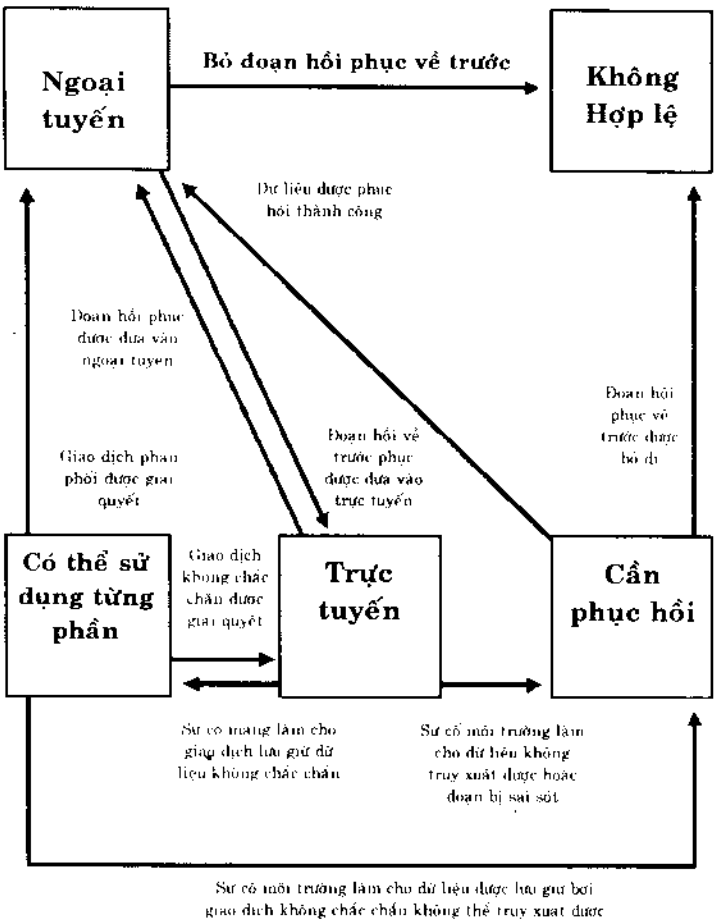
tự động thay đổi trạng thái của đoạn hồi phục NEEDS RECOVERY sang OFFLINE.)
Một DBA có thể bỏ một đoạn hồi phục NEEDS RECOVERY. (Điều này cho phép DBA bỏ các đoạn bị sai sót). Đoạn PARTLY AVAILABLE không thể được bỏ; trước hết bạn phải giải quyết giao dịch không chắc chắn, do tiến trình RECO tự động hoặc bằng thủ công.

Thông tin bổ sung :
Xem Oracle8 Distributed Database Systems về thông tin Các Giao Dịch Phân Phối.

Nếu bạn đưa đoạn hồi phục PARTLY AVAILABLE trực tuyến (bằng một lệnh hoặc khi khởi động môi trường), Oracle có thể dùng nó cho các giao dịch mới. Tuy nhiên, giao dịch không chắc truy vấn lưu giữ một số mục nhập của bảng giao dịch, vì vậy số giao dịch mới có thể dùng đoạn hồi phục bị giới hạn. (Xem "Khi Thông Tin Hồi Phục Cần Thiết" về thông tin Bảng Giao Dịch)

Ngoài ra, cho tới khi bạn giải quyết giao dịch không chắc chắn, giao dịch tiếp tục lưu giữ các khối lưu riêng mà nó cần trong đoạn hồi phục, để tránh các giao dịch khác không sử dụng chúng. Do đó, đoạn hồi phục có thể cần giành các khối lưu riêng cho các giao dịch hoạt động và do đó phát triển. Để tránh đoạn hồi phục phát triển, người điều hành cơ sở dữ liệu có thể thích tạo một đoạn hồi phục mới cho các giao dịch để sử dụng cho tới khi giao dịch không chắc chắn được giải quyết, thay vì đưa đoạn PARTLY AVAILABLE sang trực tuyến.

Hình 2-9 Các trạng thái đoạn hồi phục và các chuyển luồng trạng thái



Các đoạn hồi phục được trì hoãn

Khi một không gian bảng thực hiện ngoại tuyến để các giao dịch không thể được hồi phục ngay, Oracle ghi vào đoạn hồi phục bị trì hoãn. Đoạn hồi phục được trì hoãn có thể chứa các mục nhập hồi phục không thể được áp

dụng đối với không gian bảng ấy, vì vậy chúng có thể được áp dụng khi không gian bảng trở lại. Những đoạn này biến mất ngay khi không gian bảng được đưa trở lại trực tuyến và được phục hồi. Oracle tự động tạo các đoạn hồi phục trong không gian bảng SYSTEM.

Chương 3

Các không gian bảng và các tập tin dữ liệu

Chương này mô tả về những không gian bảng, các cấu trúc cơ sở dữ liệu logic chủ yếu của bất kỳ cơ sở dữ liệu Oracle nào và những tập tin dữ liệu vật lý tương ứng với mỗi không gian bảng. Chương này gồm:

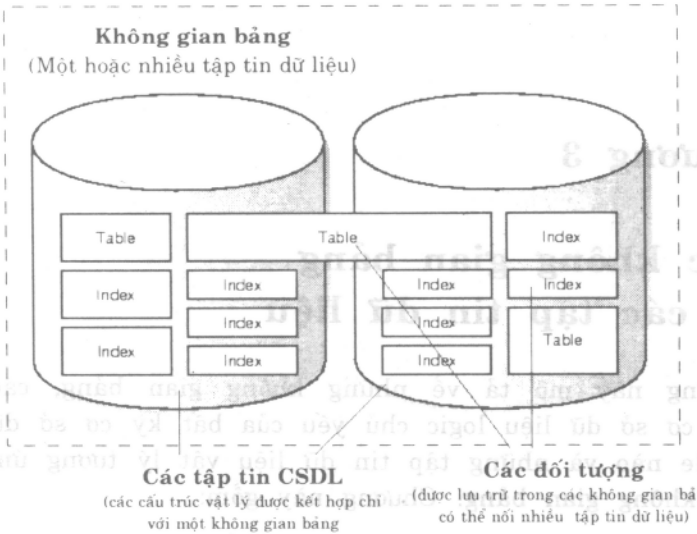
- **Giới thiệu về những không gian bảng và các tập tin dữ liệu**
- **Các không gian bảng**
- **Những tập tin dữ liệu**

Thông tin bổ sung :
Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu của Trusted Oracle về những không gian bảng và các tập tin dữ liệu trong môi trường đó.

Không gian bảng và những tập tin dữ liệu

Oracle lưu trữ dữ liệu một cách logic trong các không gian bảng và vật lý trong các tập tin dữ liệu được kết hợp với không gian bảng tương ứng. Hình 3-1 minh họa mối quan hệ này.

Hình 3-1 Những tập tin dữ liệu và các không gian bảng



Các cơ sở dữ liệu, không gian bảng và những tập tin dữ liệu có quan hệ chặt chẽ, nhưng chúng có những khác biệt quan trọng:

Các cơ sở dữ liệu và những không gian bảng	Cơ sở dữ liệu gồm một hoặc nhiều đơn vị bộ nhớ logic được gọi là các không gian bảng. Dữ liệu của cơ sở dữ liệu được lưu trữ tập hợp trong các không gian bảng của cơ sở dữ liệu.
Các không gian bảng và các tập tin dữ liệu	Mỗi không gian bảng trong một cơ sở dữ liệu Oracle gồm một hoặc nhiều tập tin được gọi là các tập tin dữ liệu. Những cấu trúc vật lý này thích

	hợp với hệ điều hành trong đó Oracle đang chạy.
Các cơ sở dữ liệu và các tập tin dữ liệu	Dữ liệu của cơ sở dữ liệu được lưu trữ tập hợp trong các tập tin dữ liệu cấu tạo thành không gian bảng của cơ sở dữ liệu. Thí dụ, cơ sở dữ liệu đơn giản nhất sẽ có một không gian bảng và một tập tin dữ liệu. Một cơ sở dữ liệu phức tạp hơn có thể có ba không gian bảng, mỗi không gian bảng gồm hai tập tin dữ liệu (thành tổng số 6 tập tin dữ liệu).

Các phần sau giải thích thêm về các không gian bảng và các tập tin dữ liệu.

Các không gian bảng

Cơ sở dữ liệu được chia thành một hoặc nhiều đơn vị bộ nhớ logic được gọi là các không gian bảng. Người điều hành cơ sở dữ liệu (DBA) dùng các không gian bảng để:

Điều khiển phân bổ không gian đĩa cho dữ liệu của cơ sở dữ liệu

Gán chỉ tiêu không gian cụ thể cho những người dùng cơ sở dữ liệu

Điều khiển tính khả dụng dữ liệu bằng cách dùng các không gian cá biệt ngoại tuyến hoặc trực tuyến

Thực hiện sao lưu cơ sở dữ liệu từng phần hoặc các điều hoạt phục hồi

Phân bổ bộ nhớ dữ liệu qua những thiết bị để cải tiến hiệu suất

Một DBA có thể tạo các không gian bảng mới, bổ sung các tập tin dữ liệu vào các không gian bảng, xác lập và thay đổi các xác lập bộ nhớ đoạn mặc định cho các đoạn

được tạo trong không gian bảng, tạo không gian bảng chỉ đọc hoặc chỉ ghi, tạo không gian bảng tạm thời hoặc vĩnh viễn và bỏ các không gian bảng.

Phần này gồm các chủ đề sau:

Không Gian Bảng SYSTEM
Phân Bổ Thêm Không Gian Cho Cơ Sở Dữ Liệu
Đưa Các Không Gian Bảng Trực Tuyến Và Ngoại Tuyến
Các Không Gian Bảng Tạm Thời
Các Không Gian Bảng Chỉ Đọc

Các không gian bảng được chia thành các đơn vị logic của bộ nhớ được gọi là các đoạn, được đề cập chi tiết ở "Các Khối Dữ Liệu, Các Khối Lưu Riêng Và Các Đoạn".

Không gian bảng SYSTEM

Mọi cơ sở dữ liệu Oracle chứa một không gian bảng có tên SYSTEM, mà Oracle tự động tạo khi cơ sở dữ liệu ấy được tạo. Không gian bảng SYSTEM luôn luôn chứa các bảng từ điển dữ liệu cho toàn bộ cơ sở dữ liệu.

Một cơ sở dữ liệu nhỏ chỉ cần không gian bảng SYSTEM; tuy nhiên, Oracle Corporation đề nghị bạn tạo ít nhất một không gian bảng bổ sung để lưu trữ dữ liệu người dùng tách ra khỏi thông tin từ điển dữ liệu. Điều này tạo cho bạn nhiều khả năng linh hoạt hơn trong các điều hoạt quản lý cơ sở dữ liệu khác nhau và giảm tranh chấp giữa các đối tượng từ điển và các đối tượng giản đồ cho các tập tin dữ liệu.

Lưu ý:
Không gian bảng SYSTEM luôn luôn trực tuyến khi cơ sở dữ liệu mở. Xem "Đưa Các Không Gian Bảng Trực

Tuyến Và Ngoại Tuyến".

Tất cả dữ liệu được lưu trữ thay vì các đơn vị chương trình PL/SQL được lưu trữ (các thủ tục, các chức năng, các bộ phần mềm và các bộ kích khởi) cư trú trong không gian bảng SYSTEM. Nếu cơ sở dữ liệu chứa nhiều đơn vị chương trình này, người điều hành cơ sở dữ liệu cần cho phép không gian những đối tượng này dùng trong không gian bảng SYSTEM. Xem thêm về những đối tượng này và không gian mà chúng cần, "Các Thủ Tục Và Các Bộ Phần Mềm"; "Các Bộ kích khởi Cơ Sở Dữ Liệu".

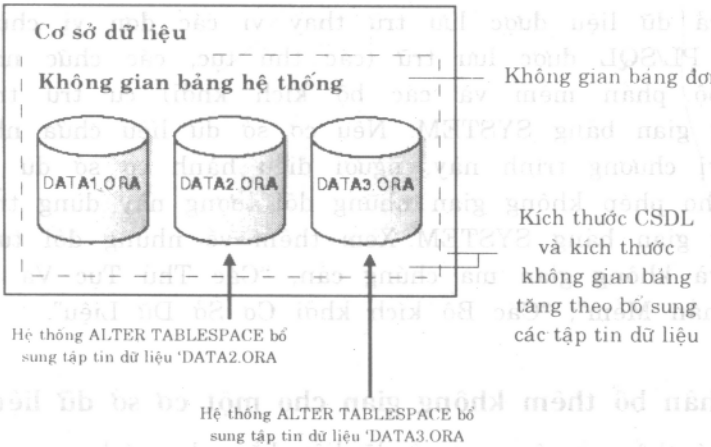
Phân bổ thêm không gian cho một cơ sở dữ liệu

Ban có thể mở rộng cơ sở dữ liệu bằng ba cách:

Bổ sung tập tin dữ liệu vào không gian bảng
Bổ sung không gian bảng mới
Tăng kích thước của tập tin dữ liệu

Khi bạn bổ sung một tập tin dữ liệu khác vào không gian bản có sẵn, bạn tăng số lượng không gian đĩa được phân bổ cho không gian bảng tương ứng. Hình 3-2 minh họa sự gia tăng không gian này.

Hình 3-2 Mở rộng một cơ sở dữ liệu bằng cách bổ sung tập tin dữ liệu vào không gian bảng

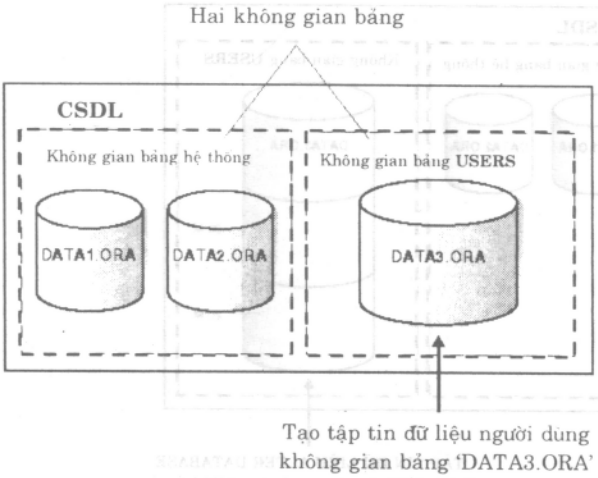


Một cách khác, bạn có thể tạo một không gian bảng mới (chứa ít nhất một tập tin dữ liệu bổ sung) để tăng kích thước của cơ sở dữ liệu. Hình 3-3 minh họa điều này.

Kích thước của không gian bảng là kích thước của (các) tập tin dữ liệu cấu tạo không gian bảng; kích thước của một cơ sở dữ liệu là kích thước chung của những không gian bảng cấu tạo cơ sở dữ liệu ấy.

Tùy chọn thứ ba làm thay đổi kích thước của tập tin dữ liệu hoặc cho phép các tập tin dữ liệu trong các không gian bảng có sẵn tăng nhanh khủng khiếp theo nhu cầu gia tăng không gian. Bạn thực hiện điều này bằng cách thay các tập tin có sẵn hoặc bổ sung các tập tin với các đặc tính mở rộng động. Hình 3-4 minh họa điều này.

Hình 3-3 Mở rộng cơ sở dữ liệu bằng cách bổ sung một không gian bảng mới



Thông tin bổ sung :

Xem Hướng dẫn của Oracle Administrator về cách gia tăng không gian trong cơ sở dữ liệu của bạn.

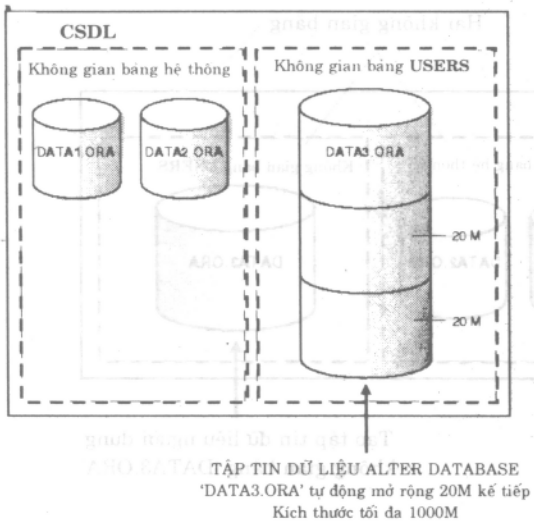
Đưa các không gian bảng vào trực tuyến và ngoại tuyến

Một người dùng điều hành cơ sở dữ liệu có thể đưa bất kỳ không gian bảng (ngoại từ không gian bảng SYSTEM) trong cơ sở dữ liệu Oracle trước (có thể truy xuất) hoặc ngoại tuyến (không thể truy xuất) bất kỳ khi nào mở.

Lưu ý:

Không gian bảng SYSTEM luôn luôn trực tuyến khi cơ sở dữ liệu mở vì từ điển dữ liệu phải luôn luôn mở để có thể sử dụng Oracle.

Hình 3-4 Mở rộng cơ sở dữ liệu bằng cách lập kích thước các tập tin dữ liệu động



Một không gian bảng thường trực tuyến để dữ liệu được chứa trong nó những người dùng có thể sử dụng cơ sở dữ liệu. Tuy nhiên, người điều hành cơ sở dữ liệu có thể dùng không gian bảng ngoại tuyến.

Để tạo một phần cơ sở dữ liệu không thể sử dụng, trong khi cho phép truy xuất bình thường phần còn lại của cơ sở dữ liệu

Để thực hiện sao lưu không gian bảng ngoại tuyến (dù không gian bảng có thể được sao lưu khi trực tuyến và sử dụng)

Bạn không thể dùng không gian bảng ngoại tuyến nếu nó chứa các đoạn hồi phục bất kỳ đang sử dụng. Xem "Các Đoạn Hồi Phục".

Khi một không gian bảng ngoại tuyến

Khi một không gian bảng ngoại tuyến, Oracle không cho phép bất kỳ câu lệnh SQL liên tiếp tham chiếu các đối tượng được chứa trong không gian bảng đó. Các giao dịch hoạt động với các câu lệnh hoàn tất tham chiếu dữ liệu trong không gian bảng đó không bị ảnh hưởng ở mức giao dịch. Oracle lưu dữ liệu hồi phục tương ứng với các câu lệnh đã hoàn tất trong đoạn hồi phục được trì hoãn (trong không gian bảng SYSTEM). Khi không gian bảng được đưa trở lại ngoại tuyến, Oracle đưa dữ liệu hồi phục vào không gian bảng ấy, nếu cần.

Khi một không gian bảng ngoại tuyến hoặc trở lại trực tuyến, điều này được ghi trong từ điển dữ liệu trong không gian bảng SYSTEM. Nếu một không gian bảng đã ngoại tuyến khi bạn đóng cơ sở dữ liệu, thì không gian bảng vẫn ngoại tuyến khi cơ sở dữ liệu gắn liên tiếp hoặc mở lại.

Bạn có thể đưa một không gian bảng trực tuyến chỉ trong cơ sở dữ liệu mà nó đã được tạo vì thông tin từ điển dữ liệu cần thiết trong không gian bảng SYSTEM của cơ sở dữ liệu đó. Một không gian bảng ngoại tuyến không thể được đọc hoặc sửa đổi bằng bất kỳ tiện ích nào khác hơn Oracle. Do đó, các không gian bảng không thể được chuyển giao từ cơ sở dữ liệu này tới cơ sở dữ liệu khác.

Thông tin bổ sung :
Chuyển giao dữ liệu của Oracle có thể đạt được bằng các công cụ được mô tả trong Oracle8 UtilitiesTransfer.

Oracle tự động chuyển không gian bảng từ trực tuyến sang ngoại tuyến khi gặp phải các lỗi nào đó (thí dụ, khi xử lý của trình ghi cơ sở dữ liệu (DBWn) không thực hiện được sau nhiều lần cố ghi tập tin dữ liệu của không

gian bảng). Các người dùng cố truy xuất các bảng trong không gian bảng ngoại tuyến bị một lỗi. Nếu vấn đề ấy gây cho I/O đĩa này bị sự cố môi trường, bạn phải phục hồi không gian bảng sau khi điều chỉnh vấn đề phần cứng.

Sử dụng các không gian bảng cho các thủ tục đặc biệt

Nếu bạn tạo nhiều không gian bảng để tách các kiểu dữ liệu khác nhau, bạn dùng các không gian bảng ngoại tuyến cho các thủ tục khác nhau; các không bảng khác vẫn trực tuyến và thông tin trong chúng vẫn có thể sử dụng. Tuy nhiên, các trường hợp đặc biệt có thể xảy ra khi các không gian bảng được dùng ngoại tuyến. Thí dụ, nếu hai không gian bảng được dùng để tách dữ liệu bảng ra dữ liệu chỉ mục, thực hiện theo sau đúng:

Nếu không gian bảng chứa các chỉ mục ngoại tuyến, các truy vấn vẫn có thể truy xuất dữ liệu bảng vì các truy vấn không đòi hỏi một chỉ mục để truy xuất dữ liệu bảng ấy.

Nếu không gian bảng chứa các bảng ngoại tuyến, dữ liệu bảng trong cơ sở dữ liệu không thể truy xuất vì các bảng cần truy xuất dữ liệu ấy.

Nói tóm lại, nếu Oracle có đủ thông tin trong các không gian bảng trực tuyến để tiến hành một câu lệnh, thì nó sẽ thực hiện như vậy. Nếu nó cần dữ liệu trong một không gian bảng ngoại tuyến, thì nó gây cho câu lệnh không thực hiện được.

Các không gian bảng chỉ đọc

Mục đích chủ yếu của các không gian bảng chỉ đọc để loại trừ nhu cầu thực hiện sao lưu và phục hồi các phần

cơ sở dữ liệu lớn và tĩnh. Oracle không bao giờ cập nhật các tập tin của không gian bảng chỉ đọc và do đó các tập tin có thể cư trú trên môi trường chỉ đọc, chẳng hạn như các CD ROM hoặc các ổ đĩa WORM.

Lưu ý:
Vì bạn có thể đưa không gian bảng trực tuyến trong cơ sở dữ liệu mà nó đã được tạo, các không gian bảng chỉ đọc không có nghĩa thỏa đáng tàng trữ hoặc đòi hỏi dữ liệu công bố.

Bất kỳ khi nào bạn tạo một không gian bảng mới, nó luôn luôn được tạo là đọc-ghi. Bạn có thể thay đổi không gian thành chỉ đọc với tùy chọn READ ONLY của lệnh ALTER TABLESPACE, cũng như tạo tất cả những tập tin dữ liệu chỉ đọc được kết hợp của không gian bảng. Bạn có thể dùng tùy chọn READ WRITE để tạo lại không gian bảng chỉ đọc thành đọc-ghi.

Thông tin bổ sung :
Xem Oracle SQL Reference về lệnh ALTER TABLESPACE.

Tạo một không gian bảng chỉ đọc không làm thay đổi trạng thái ngoại tuyến hoặc trực tuyến của nó. Các tập tin dữ liệu ngoại tuyến không thể được truy xuất. Đưa một tập tin dữ liệu trong không gian bảng chỉ đọc trực tuyến tạo cho tập tin ấy có thể chỉ đọc. Tập tin ấy không thể được ghi vào nếu không gian bảng kết hợp của nó được trở lại trạng thái đọc-ghi. Bạn có thể dùng các tập tin của không gian bảng chỉ đọc trực tuyến hoặc ngoại tuyến độc lập bằng cách sử dụng tùy chọn DATAFILE của lệnh ALTER DATABASE.

Các không gian bảng chỉ đọc không thể được sửa đổi. Để cập không gian bảng chỉ đọc, trước hết bạn phải tạo không gian bảng chỉ ghi. Sau khi cập nhật không gian bảng, bạn có thể xác lập lại nó là chỉ đọc.

Vì các không gian bảng chỉ đọc không thể bị sửa đổi, chúng không cần được sao lưu lặp lại. Ngoài ra, nếu bạn cần phục hồi cơ sở dữ liệu của mình, bạn không cần phục hồi bất kỳ các không gian bảng chỉ đọc, vì chúng không thể bị sửa đổi. Tuy nhiên, các không gian bảng chỉ đọc có thể cần sự chú ý khi môi trường làm việc hoặc phục hồi môi trường, tùy theo chúng được đọc-ghi như thế nào và khi nào.

Thông tin bổ sung :
Xem Hướng dẫn phục hồi và sao lưu của Oracle8.

Bạn có thể bỏ các mục, chẳng hạn như các bảng và các chỉ mục từ một không gian bảng chỉ đọc, ngay khi bạn có thể bỏ các mục từ một không gian bảng ngoại tuyến. Tuy nhiên, bạn không thể tạo hoặc thay đổi các đối tượng trong không gian bảng chỉ đọc.

Bạn không thể bổ sung các tập tin dữ liệu vào không gian bảng chỉ đọc, dù bạn dùng không gian bảng ngoại tuyến. Khi bạn bổ sung một tập tin dữ liệu, Oracle phải cập nhật phần đầu tập tin ấy và hoạt động ghi này không được phép trong không gian bảng chỉ đọc.

Các không gian bảng tạm thời

Bạn có thể quản lý không gian cho các hoạt động sắp xếp hiệu quả hơn bằng cách thiết kế các không gian bảng tạm thời riêng cho các sắp xếp đó. Thực hiện như vậy để loại trừ sự nối tiếp khóa của những điều hoạt

quản lý không gian một cách hiệu quả liên quan đến sự phân bổ và thôi phân bổ của không gian sắp xếp. Tất cả điều hoạt sử dụng các sắp xếp - kể cả các kết nối, các thiết lập chỉ mục, thứ tự (ORDER BY), sự tính toán của các khối kết tạp (GROUP BY) và lệnh ANALYZE để tập hợp những thống kê bộ ưu hóa - lợi ích từ các không gian bảng tạm thời. Hiệu suất có được chủ yếu trong môi trường Oracle Parallel Server.
Không gian bảng tạm thời có thể chỉ được dùng cho các đoạn sắp xếp. (Nó không giống như không gian bảng mà một người dùng thiết kế cho các đoạn tạm thời, có thể là không gian người dùng có thể sử dụng bất kỳ). Không đối tượng vĩnh viễn nào có thể cư trú trong không gian bảng tạm thời. Các đoạn sắp xếp được sử dụng khi một đoạn được dùng chung bởi nhiều điều hoạt sắp xếp. Một đoạn sắp xếp tồn tại trong mỗi môi trường làm việc thực hiện một điều hoạt sắp xếp trong không gian bảng cho trước.

Những không gian bảng tạm thời cung cấp các cải tiến hiệu suất khi bạn có nhiều sắp xếp quá lớn không thích hợp với bộ nhớ. Đoạn sắp xếp của không gian bảng tạm thời cho trước được tạo vào lúc nhiều hoạt sắp xếp đầu tiên. Đoạn sắp xếp mở rộng bằng cách phân bổ các khối lưu riêng cho tới khi kích thước đoạn bằng hoặc lớn hơn tổng số nhu cầu bộ nhớ của các sắp xếp hoạt động trên minh họa đó.

Bạn tạo các không gian bảng tạm thời sử dụng cú pháp SQL sau:
CREATE TABLESPACE tablespace TEMPORARY | PERMANENT;

Bạn cũng có thể thay đổi không gian bảng từ PERMANENT thành TEMPORARY hoặc ngược lại bằng cách sử dụng cú pháp sau:
ALTER TABLESPACE tablespace TEMPORARY;

Thông tin bổ sung :
Xem Oracle8 SQL Reference về các lệnh CREATE TABLESPACE và ALTER TABLESPACE.

Các tập tin dữ liệu

Không gian bảng trong cơ sở dữ liệu Oracle gồm một hoặc nhiều tập tin dữ liệu vật lý. Một tập tin dữ liệu có thể được kết hợp với chỉ một không gian bảng và chỉ một cơ sở dữ liệu.

Oracle tạo một tập tin dữ liệu cho một không gian bảng bằng cách phân bổ số lượng không gian cụ thể cùng với phần điều hành chung cần thiết cho phần mở đầu tập tin ấy. Khi tập tin dữ liệu được tạo, hệ điều hành trong Oracle đang chạy chịu trách nhiệm xóa thông tin cũ và các quyền hạn khỏi tập tin trước khi phân bổ nó vào Oracle. Nếu tập tin ấy lớn, tiến trình này có mất nhiều thời gian.

Thông tin bổ sung :
Để có thêm thông tin về số lượng không gian cần thiết cho phần đầu tập tin của các tập tin dữ liệu trên hệ điều hành của bạn, xem tài liệu xử lý hệ điều hành Oracle.

Không gian bảng đầu tiên trong cơ sở dữ liệu bất kỳ luôn luôn là không gian bảng SYSTEM, vì vậy Oracle tự động phân bổ các tập tin dữ liệu đầu tiên của bất kỳ cơ

sở dữ liệu cho không gian bảng SYSTEM khi tạo cơ sở dữ liệu.

Nội dung tập tin dữ liệu

Khi tập tin dữ liệu được tạo lần đầu, không gian đĩa phân bổ được định dạng nhưng không chứa bất kỳ dữ liệu người dùng nào; tuy nhiên, Oracle dự trữ không gian để lưu trữ dữ liệu cho các đoạn tương lai của không gian bảng kết hợp - nó được Oracle dùng riêng. Khi dữ liệu gia tăng trong không gian bảng, Oracle sử dụng không gian trống trong các tập tin dữ liệu kết hợp để phân bổ các khối lưu riêng cho đoạn ấy. Xem "Các Khối Dữ Liệu, Các Khối Lưu Riêng Và Các Đoạn".

Dữ liệu được kết hợp với các đối tượng giản đồ trong không gian bảng được lưu trữ về mặt vật lý trong một hoặc nhiều tập tin dữ liệu cấu tạo không gian bảng ấy. Lưu ý một đối tượng giản đồ không tương ứng với một tập tin dữ liệu cụ thể; đúng hơn, một tập tin dữ liệu là một tàng thư cho dữ liệu của đối tượng bất kỳ trong không gian bảng cụ thể. Oracle phân bổ không gian cho dữ liệu được kết hợp với một đối tượng trong một hoặc nhiều tập tin dữ liệu của không gian bảng. Do đó, một đối tượng có thể "mở rộng" một hoặc nhiều tập tin dữ liệu. Trừ khi bảng lập dải "striping" được sử dụng (nơi dữ liệu được mở rộng trên nhiều đĩa), người điều hành cơ sở dữ liệu và các người dùng cuối không thể điều khiển tập tin dữ liệu nào lưu trữ đối tượng.

Kích thước của các tập tin dữ liệu

Bạn có thể thay đổi kích thước của tập tin dữ liệu sau khi tạo hoặc bạn có thể xác định một tập tin dữ liệu sẽ tăng lên một cách linh hoạt như các đối tượng giản đồ

trong không gian bảng. Tính năng này cho bạn có ít tập tin dữ liệu hơn trên mỗi không gian bảng và có thể đơn giản hóa việc quản lý các tập tin dữ liệu.

Thông tin bổ sung:
Xem Hướng dẫn của Oracle Administrator về cách lập lại kích thước các tập tin dữ liệu.

Các tập tin dữ liệu ngoại tuyến

Bạn có thể đưa các không gian bảng ngoại tuyến (làm cho không thể sử dụng) hoặc đưa chúng vào trực tuyến (làm cho chúng có thể sử dụng) bằng một đơn vị khi bạn đưa không gian bảng ngoại tuyến hoặc trực tuyến tương ứng. Bạn có thể đưa các tập tin dữ liệu ngoại tuyến hoặc; tuy nhiên, điều này chỉ được thực hiện trong một số thủ tục phục hồi cơ sở dữ liệu.

Chương 4

Từ điển dữ liệu

Chương này mô tả tập trung vào các bảng tham chiếu chỉ đọc và các hiển thị của mỗi cơ sở dữ liệu Oracle được coi là trung tâm của từ điển dữ liệu. Chương này gồm:

- **Giới thiệu từ điển dữ liệu**
- **Cấu trúc của từ điển dữ liệu**
- **Sys, chủ nhân của từ điển dữ liệu**
- **Từ điển dữ liệu được sử dụng như thế nào**
- **Các bảng hiệu suất động**

Giới thiệu từ điển dữ liệu

Một trong những phần quan trọng nhất của một cơ sở dữ liệu Oracle là từ điển dữ liệu của nó, nó là một tập hợp các bảng chỉ đọc để cung cấp thông tin về cơ sở dữ liệu kết hợp của nó. Từ điển dữ liệu chứa:

Định nghĩa của tất cả đối tượng giản đồ trong cơ sở dữ liệu (bảng, các khung nhìn, các chỉ mục, các cụm liên kết, những từ đồng nghĩa, các chuỗi, các thủ tục, các hàm, các bộ phần mềm, các bộ kích khởi, v.v...); không gian được

phân bổ và hiện được sử dụng bao nhiêu, các giá trị mặc định của đối tượng giản đồ cho các cột.
Tính nguyên vẹn hạn định thông tin.
Các tên của những người dùng Oracle
Các đặc quyền và các vai trò mà mỗi người dùng đã được cho
Thông tin kiểm soát, chẳng hạn như ai đã truy xuất hoặc cập nhật các đối tượng giản đồ khác nhau.
Trong Trusted Oracle, các nhãn có thể tất cả đối tượng giản đồ và người dùng (xem tài liệu Trusted Oracle)
Những thông tin cơ sở dữ liệu tổng quát khác

Từ điển dữ liệu được cấu trúc trong các bảng và cách xem, giống như dữ liệu của cơ sở dữ liệu khác. Tất cả các bảng và khung nhìn của từ điển dữ liệu cho một cơ sở dữ liệu cho trước được lưu trữ trong không gian bảng SYSTEM của cơ sở dữ liệu đó.

Từ điển dữ liệu không chỉ tập trung vào mọi cơ sở dữ liệu của Oracle, mà nó còn là công cụ quan trọng cho tất cả người dùng, từ những người dùng cuối tới các nhà thiết kế ứng dụng và các người điều hành cơ sở dữ liệu Để truy xuất từ điển dữ liệu, bạn sử dụng các câu lệnh SQL. Vì từ điển dữ liệu là chỉ đọc, bạn chỉ có thể đưa các truy vấn (các câu lệnh SELECT) dựa trên các bảng và các khung nhìn của từ điển dữ liệu.

Cấu trúc của từ điển dữ liệu

Từ điển dữ liệu của cơ sở dữ liệu gồm:

Các bảng cơ bản	Các bảng cơ bản lưu trữ thông tin về các cơ sở dữ liệu được kết hợp. Chỉ Oracle ghi và đọc các bảng này. Các người dùng ít khi truy cập chúng trực tiếp vì chúng được tiêu chuẩn hóa và phần lớn dữ liệu được lưu trữ bằng định dạng mật mã.
Các khung nhìn người dùng truy xuất được	Các khung nhìn tóm tắt và hiển thị thông tin được lưu trữ trong các bảng cơ bản của từ điển dữ liệu. Những khung nhìn giải mã dữ liệu bảng cơ sở thành hệ thống hữu dụng, chẳng hạn như người dùng hoặc các tên bảng sử dụng các kết nối và các mệnh đề WHERE để đơn giản hóa thông tin. Phần lớn người dùng được cho truy xuất các khung nhìn trước, thay vì các bảng cơ bản.

SYS, chủ nhân của từ điển dữ liệu

SYS người dùng làm chủ tất cả các bảng cơ sở và các khung nhìn người dùng truy xuất được của từ điển dữ liệu. Do đó, người dùng Oracle không thể thay đổi (cập nhật, xóa hoặc chèn) bất kỳ hàng hoặc các đối tượng giản đồ nào được chứa trong giản đồ SYS, vì hoạt động như vậy để duy trì tính nguyên vẹn dữ liệu. Người điều hành bảo mật sẽ theo dõi nghiêm ngặt tài khoản trung tâm này.

Cảnh báo:

Thay đổi hoặc điều tác dữ liệu trong các bảng từ điển

dữ liệu cơ bản có thể ảnh hưởng vĩnh viễn và bất lợi cho cơ sở dữ liệu.

Từ điển dữ liệu được sử dụng như thế nào

Từ điển dữ liệu có ba cách sử dụng chủ yếu:

Oracle truy xuất từ điển dữ liệu để tìm thông tin về các người dùng, các đối tượng giản đồ và các cấu trúc bộ nhớ.
Oracle sửa đổi từ điển dữ liệu mỗi lần câu lệnh người dùng định nghĩa dữ liệu (DDL) được đưa ra.
Bất kỳ người dùng Oracle đều có thể sử dụng từ điển dữ liệu bằng tham chiếu chỉ đọc thông tin về cơ sở dữ liệu.

Oracle sử dụng từ điển dữ liệu như thế nào

Dữ liệu trong các bảng cơ bản của từ điển dữ liệu cần thiết cho Oracle thực hiện chức năng. Do đó, chỉ Oracle sẽ ghi hoặc thay đổi thông tin từ điển dữ liệu.

Trong khi điều hoạt cơ sở dữ liệu, Oracle đọc từ điển dữ liệu để khẳng định các đối tượng giản đồ tồn tại và để người dùng truy xuất chúng chính xác. Oracle còn cập nhật từ điển dữ liệu liên tục để phản ánh các thay đổi trong những cấu trúc cơ sở dữ liệu, kiểm soát, các quyền hạn được cung cấp và dữ liệu.

Thí dụ, nếu người KATHY tạo một bảng có tên PARTS, các hàng mới được bổ sung vào từ điển dữ liệu để phản ánh bảng mới, các cột, đoạn, các khối lưu riêng và các đặc quyền mà KATHY có trên bảng đó. Thông tin mới này được hiển thị sau đó khi các hiển thị từ điển được yêu cầu.

Các từ đồng nghĩa chung cho các hiển thị từ điển dữ liệu

Oracle tạo các từ đồng nghĩa chung trên nhiều hiển thị từ điển dữ liệu để người dùng có thể truy xuất chúng thuận tiện. (Người điều hành bảo mật còn có thể tạo các từ đồng nghĩa chung cho các đối tượng giản đồ được sử dụng trên toàn bộ hệ thống). Người dùng nên tránh đặt tên các đối tượng giản đồ riêng với các tên tương tự được sử dụng cho các từ đồng nghĩa chung.

Trữ từ điển dữ liệu để truy xuất nhanh

Nhiều thông tin từ điển dữ liệu được trữ trong SGA (bộ nhớ cache từ điển), vì Oracle luôn luôn truy xuất từ điển dữ liệu khi điều hoạt cơ sở dữ liệu phẩm định tính hợp lệ truy xuất của người dùng và để xác minh trạng thái của các đối tượng giản đồ. Tất cả thông tin được lưu trữ trong bộ nhớ sử dụng thuật toán LRU (Least Recently Used).

Thông tin được lưu trữ thường trong các bộ nhớ cache cần thiết để phân ngữ. Các cột COMMENTS mô tả các bảng và các cột của chúng không được trữ nếu chúng không thường được truy xuất.

Các chương trình và từ điển dữ liệu khác

Các sản phẩm Oracle khác có thể tham chiếu các hiển thị có sẵn và tạo các bảng từ điển dữ liệu bổ sung hoặc các hiển thị của riêng chúng. Các Developer ứng dụng viết các chương trình để tham chiếu từ điển dữ liệu, họ sẽ tham chiếu các từ đồng nghĩa chung thay vì các bảng cơ bản: các từ đồng nghĩa ít có thể thay đổi giữa các bản phát hành phần mềm.

Bổ sung các mục từ điển dữ liệu mới

Bạn có thể bổ sung các bảng hoặc các hiển thị mới vào từ điển dữ liệu. Nếu bạn bổ sung các đối tượng của từ điển dữ liệu mới, chủ nhân của các đối tượng mới ấy phải là SYSTEM người dùng hoặc người dùng Oracle thứ ba.

Cảnh báo:
Không bao giờ tạo các đối tượng mới thuộc SYS người dùng, ngoại trừ bằng cách chạy ký mã được công ty Oracle cung cấp để tạo các đối tượng từ điển dữ liệu.

Xóa các mục từ điển dữ liệu

Tất cả thay đổi đối với từ điển dữ liệu được Oracle thực hiện để trả lời cho các lệnh DDL, do đó không có dữ liệu nào trong bất kỳ bảng từ điển nào sẽ được người dùng xóa hoặc thay đổi.

Chỉ một ngoại từ đối với nguyên tắc này là SYS.AUD$ bảng. Khi kiểm soát cho phép, bảng này có thể phát triển mà không bị giới hạn. Dù bạn không được bỏ bảng AUDIT_TRAIL, người điều hành bảo mật có thể xóa an toàn dữ liệu khỏi nó vì các hàng chỉ được dùng cho thông tin và không cần cho Oracle chạy.

Những người dùng Oracle sử dụng từ điển dữ liệu như thế nào

Các khung nhìn của từ điển dữ liệu đáp ứng là tham chiếu cho tất cả người dùng cơ sở dữ liệu. Bạn truy xuất các khung nhìn từ điển dữ liệu bằng ngôn ngữ SQL. Một số khung nhìn người dùng Oracle có thể truy xuất; những khung nhìn khác chỉ dành cho người điều hành.

Từ điển dữ liệu luôn luôn có thể sử dụng khi cơ sở dữ liệu được mở. Nó cư trú trong không gian bảng SYSTEM luôn luôn trực tuyến.

Từ điển dữ liệu gồm các tập hợp của ba khung nhìn chứa thông tin tương tự và phân biệt với nhau bằng các tiền tố của chúng:

Bảng 4-1 Các tiền tố khung nhìn từ điển dữ liệu

Tiền tố	Phạm vi
USER	Khung nhìn của người dùng (những gì trong giản đồ của người dùng)
ALL	Khung nhìn của người dùng được mở rộng (những gì người dùng có thể truy xuất)
DBA	Khung nhìn của người điều hành cơ sở dữ liệu (những gì tất cả người dùng có thể truy xuất)

Tập hợp của các cột giống nhau qua các khung nhìn với các ngoại trừ sau:
Các khung nhìn với tiền tố USER thường loại trừ cột OWNER. Cột này được ám chỉ trong các khung nhìn USER để người dùng đưa ra truy vấn ấy.
Một số khung nhìn DBA có các cột bổ sung chứa thông tin hữu dụng cho người điều hành.

Thông tin bổ sung :
Xem Oracle8 Reference Về Danh Sách Toàn Bộ Các Khung Nhìn Từ Điển Dữ Liệu Và Các Cột Của Chúng.

Các khung nhìn với tiền tố USER

Các khung nhìn hầu như có thể liên quan đến những người dùng cơ sở dữ liệu tiêu biểu là các khung nhìn với tiền tố USER. Các khung nhìn này tham chiếu môi trường riêng tư của người dùng trong cơ sở dữ liệu, kể cả những thông tin về các đối tượng giản đồ được người dùng tạo, cho phép, v.v.. chỉ hiển thị các hàng thích hợp cho người dùng có các cột giống với các khung nhìn khác, ngoại trừ cột OWER được ám chỉ (người dùng hiện hành) cho trở lại tập hợp con của những thông tin trong các khung nhìn ALL_ có thể có các từ đồng nghĩa PUBLIC được viết tắt để sử dụng thuận tiện.

Thí dụ, truy vấn sau cho trở lại tất cả đối tượng được chứa trong giản đồ của bạn:

```
SELECT object_name, object type FROM user_objects;
```

Các khung nhìn với tiền tố ALL

Các khung nhìn với tiền tố ALL tham chiếu phối cảnh tổng quát về cơ sở dữ liệu của người dùng. Các khung nhìn cho trở lại những thông tin về các đối tượng giản đồ mà người dùng truy xuất bằng các đặc quyền và các vai trò chung hoặc riêng biệt, để bổ sung vào các đối tượng giản đồ mà người dùng làm chủ. Thí dụ, truy vấn sau cho trở lại thông tin về tất cả các đối tượng mà bạn truy xuất:

```
SELECT owner, object_name, object_type FROM all_objects;
```

Các khung nhìn với tiền tố DBA

Các khung nhìn với tiền tố DBA trình bày khung nhìn toàn cục của toàn bộ cơ sở dữ liệu. Do đó, chúng có nghĩa là chỉ được hỏi bởi các người điều hành cơ sở dữ liệu. Bất kỳ người dùng nào được cho phép đặc quyền hệ

thống SELECT ANY TABLE có thể yêu cầu các khung nhìn có tiền tố DBA-prefixed về từ điển dữ liệu.

Các từ đồng nghĩa không được tạo cho các khung nhìn này, vì các khung nhìn DBA sẽ chỉ được yêu cầu bởi các người điều hành. Do đó để yêu cầu các khung nhìn DBA, các người điều hành phải cho tiền tố tên khung nhìn với chủ nhân của nó, SYS, như trong

```
SELECT owner, object_name, object_type FROM
sys.dba_objects;
```

Các người điều hành có thể chạy tập tin ký mã DBA_SYNONYMS.SQL để tạo các từ đồng nghĩa riêng cho các khung nhìn DBA trong các tài khoản của họ nếu họ có đặc quyền hệ thống SELECT ANY TABLE. Thực hiện ký mã này tạo các từ đồng nghĩa chỉ dành cho người dùng hiện hành.

DUAL

Bảng có tên DUAL là một bảng nhỏ mà Oracle và các chương trình người dùng viết có thể tham chiếu để bảo đảm một kết quả được biết rõ. Bảng này có một cột được gọi là DUMMY và một hàng chứa giá trị "X".

Thông tin bổ sung :
Xem mô tả của lệnh SELECT trong Oracle SQL Reference Về Những Thông Tin Bảng DUAL.

Các bảng hiệu suất động

Trong toàn bộ điều hoạt của nó, Oracle duy trì một tập hợp các bảng "ảo" để ghi lại hoạt động của cơ sở dữ liệu hiện hành. Những bảng này được gọi là các bảng hiệu suất động.

Hiệu suất động không phải thực sự là các bảng và chúng sẽ không được phần lớn người dùng truy xuất. Tuy nhiên, những người điều hành cơ sở dữ liệu có thể hỏi và tạo các khung nhìn về các bảng và cho phép các người dùng truy xuất những khung nhìn đó.

SYS làm chủ các bảng hiệu suất động; các tên của chúng đều bắt đầu bằng V_$. Các khung nhìn được tạo trên các bảng này và sau đó các từ đồng nghĩa được tạo cho các khung nhìn ấy. Các tên từ đồng nghĩa bắt đầu với V$. Thí dụ, V$DATAFILE chứa thông tin về các tập tin dữ liệu của cơ sở dữ liệu và V$FIXED_TABLE chứa những thông tin về tất cả các bảng hiệu suất động và các khung nhìn trong cơ sở dữ liệu.

Thông tin bổ sung :
Xem Oracle8 Reference về danh sách toàn bộ của các từ đồng nghĩa của các khung nhìn hiệu suất động và các cột của chúng.

Phần III

MÔI TRƯỜNG LÀM VIỆC CỦA ORACLE

Phần III mô tả các cấu trúc bộ nhớ và các xử lý thiết lập môi trường làm việc Oracle server và giải thích môi trường làm việc Oracle khởi động và đóng như thế nào.

Phần III gồm các chương sau:

Chương 5 Cơ Sở Dữ Liệu, Khởi Động và Đóng Môi trường làm việc
Chương 6 Cấu Trúc Bộ Nhớ
Chương 7 Cấu Trúc Xử Lý

Chương 5

Cơ sở dữ liệu, khởi động và đóng môi trường làm việc

Chương này giải thích các thủ tục liên quan đến khởi động ngưng môi trường làm việc và cơ sở dữ liệu của Oracle. Nó gồm:

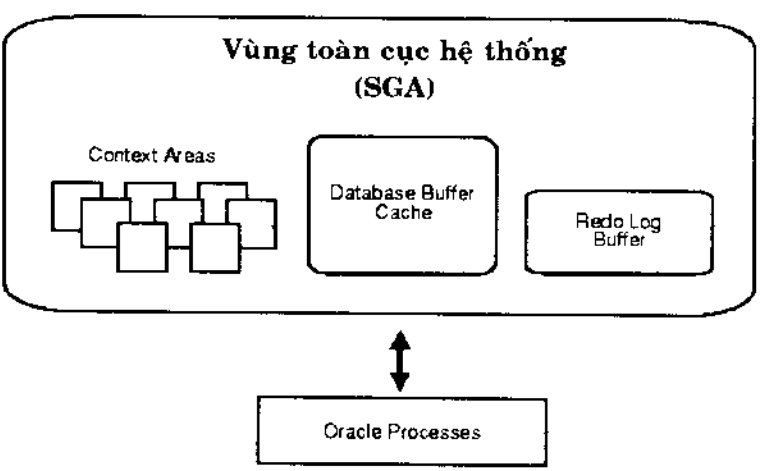
- **Tổng quan về môi trường làm việc của Oracle**
- **Kết nối với các đặc quyền của người điều hành**
- **Các tập tin tham số**
- **Môi trường làm việc và khởi động cơ sở dữ liệu**
- **Cơ sở dữ liệu và đóng môi trường làm việc**

Thông tin bổ sung :
Nếu bạn đang sử dụng Trusted Oracle, tham chiếu với tài liệu Trusted Oracle của bạn về khởi động và đóng trình ứng dụng môi trường đó.

Tổng quan của môi trường làm việc Oracle

Mọi cơ sở dữ liệu Oracle chạy được kết hợp với một môi trường làm việc Oracle. Khi cơ sở dữ liệu được khởi động trên một server cơ sở dữ liệu (bất kể kiểu máy tính), Oracle phân bổ một vùng bộ nhớ được gọi là System Global Area (SGA) và khởi động một hoặc nhiều xử lý Oracle. Kết hợp này của SGA và các tiến trình của Oracle được gọi là môi trường làm việc Oracle. Bộ nhớ và các tiến trình của một môi trường làm việc quản lý dữ liệu của cơ sở dữ liệu được kết hợp một cách có hiệu quả và đáp ứng một hoặc nhiều người dùng của cơ sở dữ liệu. Hình 5-1 trình bày môi trường làm việc Oracle, xem thêm "Cấu Trúc Bộ Nhớ" và "Cấu Trúc Xử Lý" chi tiết về SGA và Các Xử Lý Của Oracle.

Hình 5-1 Môi trường làm việc của Oracle



Môi trường làm việc và cơ sở dữ liệu

Sau khi khởi động một môi trường làm việc, Oracle kết hợp môi trường làm việc với cơ sở dữ liệu được xác định.

Điều này được gọi là lắp ghép cơ sở dữ liệu. Cơ sở dữ liệu sau đó được mở, tạo cho những người dùng hợp lệ có thể truy xuất.

Nhiều môi trường làm việc có thể tiến hành đồng thời trên cùng máy tính, mỗi truy xuất cơ sở dữ liệu vật lý của riêng nó. Trong hệ thống song song hàng loạt (MPP) có cụm liên kết, Oracle Parallel Server cho phép nhiều môi trường làm việc lắp ghép một cơ sở dữ liệu đơn.

Thông tin bổ sung :
Xem Oracle8 Parallel Server Concepts và Administrator, những thông tin về Oracle Parallel Server.

Nếu bạn đang dùng Trusted Oracle, xem tài liệu Trusted Oracle của bạn về những thông tin các môi trường làm việc trong môi trường đó.

Chỉ người điều hành cơ sở dữ liệu có thể khởi động một môi trường làm việc và mở cơ sở dữ liệu. Nếu cơ sở dữ liệu được mở, thì chỉ người điều hành cơ sở dữ liệu có thể ngưng cơ sở dữ liệu ấy để nó được đóng. Khi cơ sở dữ liệu được đóng, người dùng không thể truy xuất thông tin mà nó chứa.

Bảo mật khởi động và đóng cơ sở dữ liệu được điều khiển qua các kết nối Oracle với các đặc quyền của người điều hành. Những người dùng bình thường không có quyền điều khiển ở trạng thái hiện hành của một cơ sở dữ liệu Oracle.

Kết nối với các đặc quyền của người điều hành

Cơ sở dữ liệu khởi động và đóng là những tùy chọn quản lý đầy năng lực và hạn chế người dùng kết nối Oracle

bằng các đặc quyền của người dùng điều hành. Tùy theo hệ điều hành, một trong các điều khiển sau thiết lập các đặc quyền người điều hành đối với một người dùng:

Các đặc quyền hệ điều hành của người dùng cho phép người ấy kết nối sử dụng các đặc quyền người dùng điều hành.

Người dùng được cho các đặc quyền SYSDBA hoặc SYSOPER và cơ sở dữ liệu sử dụng các tập tin mật khẩu để thẩm định các người điều hành cơ sở dữ liệu. Cơ sở dữ liệu có mật khẩu để đăng nhập INTERNAL, và người dùng biết được mật khẩu đó.

Để bảo mật tốt hơn, các người dùng kết nối với các đặc quyền người điều hành chỉ có thể kết nối với các server chuyên dùng (không phải các server dùng chung).

Khi bạn kết nối với các đặc quyền người dùng điều hành, bạn được sắp xếp vào giản đồ do SYS làm chủ. Điều này cho bạn truy xuất tất cả những đối tượng trong giản đồ SYS.

Để có thêm thông tin về các tập tin mật khẩu và các giản đồ thẩm định tính hợp lệ đối với các người điều hành cơ sở dữ liệu, xem "Điều khiển Truy Xuất Cơ Sở Dữ Liệu".

Thông tin bổ sung :
Đối với thông tin về các đặc quyền người điều hành hoạt động trên hệ điều hành của bạn như thế nào, xem tài liệu Oracle xác định hệ điều hành của bạn.

Các tập tin tham số

Để khởi động một môi trường làm việc, Oracle đọc tập tin tham số - một tập tin văn bản chứa một danh sách

của những tham số cấu hình (các tham số khởi hoạt) cho môi trường làm việc và cơ sở dữ liệu đó. Bạn xác lập các tham số này thành các giá trị riêng biệt để khởi hoạt nhiều bộ nhớ và các xác lập tiến trình của một môi trường làm việc Oracle Phần lớn các tham số khởi hoạt thuộc một trong những nhóm sau:

Những tham số đặt tên những thư (chẳng hạn như các tập tin)

Những tham số xác lập các giới hạn (chẳng hạn như các số cực đại)

Những tham số ảnh hưởng đến dung lượng (chẳng hạn như kích thước của SGA) được gọi là các tham số khả biến

Trong thứ khác, các tham số khởi hoạt để cho Oracle biết:

Tên của cơ sở dữ liệu để khởi động môi trường làm việc ấy

Sử dụng bao biêu bộ nhớ cho các cấu trúc bộ nhớ trong SGA

Những gì thực hiện với các tập tin nhật ký hoạt động trực tuyến được điền vào

Những tên và các vị trí của những tập tin điều khiển của cơ sở dữ liệu

Các tên của các đoạn hồi phục riêng trong cơ sở dữ liệu

Thí dụ về một tập tin tham số

Sau là một thí dụ về tập tin tham số tiêu biểu:

```
db_block_buffers = 550
db_name = ORA8PROD
db_domain = US.ACME.COM
# license_max_users = 64
# control_files = filename1, filename2
```

```
# log_archive_dest = c:\logarch
log_archive_format = arch%S.ora
log_archive_start = TRUE
log_buffer = 64512
log_checkpoint_interval = 256000
# rollback_segments = rs_one, rs_two
```

Oracle xử lý các trực kiện chuỗi được xác định cho các tham số National Language Support (NLS) trong tập tin ấy như chúng ở trong tập hợp ký tự cơ sở dữ liệu.

Thay đổi các giá trị tham số

Người điều hành cơ sở dữ liệu có thể điều chỉnh các tham số khả biến để cải tiến hiệu suất của hệ thống cơ sở dữ liệu. Chính xác là những tham số nào ảnh hưởng nhất đến một hệ thống là chức năng của nhiều đặc tính cở sở dữ liệu và các biến.

Các giá trị tham số được sửa đổi chỉ có ảnh hưởng khi môi trường làm việc khởi động và đọc tập tin tham số. Một số tham số còn được thay đổi tự động bằng cách sử dụng lệnh ALTER SESSION và ALTER SYSTEM khi môi trường làm việc chạy.

> Thông tin bổ sung :
> Để mô tả tất cả các tham số khởi hoạt, xem Oracle8 Reference. Để có thông tin về các tham số và ảnh hưởng đến SGA, xem "Size của SGA".

Môi trường làm việc và khởi động cơ sở dữ liệu

Có ba bước đối với khởi động một cơ sở dữ liệu Oracle và tạo cho nó có thể sử dụng toàn bộ hệ thống là:

Khởi động một môi trường làm việc.
Lắp ghép cơ sở dữ liệu.
Mở cơ sở dữ liệu.

Một người điều hành cơ sở dữ liệu có thể thực hiện những bước này bằng cách sử dụng Oracle Enterprise Manager.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle Enterprise Manager Administrator.

Khởi động một môi trường làm việc

Khi Oracle khởi động một môi trường làm việc, trước hết nó đọc một tập tin tham số để xác định những giá trị của các tham số khởi hoạt và sau đó nó phân bổ một SGA - vùng bộ nhớ dùng chung được sử dụng cho thông tin cơ sở dữ liệu - và tạo các xử lý nền. Ở điểm này, không có cơ sở dữ liệu nào được kết hợp với các cấu trúc bộ nhớ và các xử lý.

Xem "Các Cấu Trúc Bộ Nhớ", thông tin về SGA và "Cấu Trúc Xử Lý", thông tin về các xử lý nền.

Chế độ khởi động môi trường làm việc giới hạn

Bạn có thể khởi động một môi trường làm việc trong chế độ giới hạn (hoặc thay đổi một môi trường làm việc có sẵn ở trong chế độ giới hạn sau này). Các kết nối giới hạn chỉ những người dùng đã được cho đặc quyền hệ thống RESTRICTED SESSION.

Ép một môi trường làm việc khởi động trong các trường hợp bất thường

Trong các trường hợp bất thường, một môi trường làm việc trước đó không thể đóng "dễ dàng", thí dụ, một trong các xử lý của môi trường làm việc ấy không thể được kết thúc thích hợp, cơ sở dữ liệu có thể cho trở lại một lỗi khi khởi động môi trường làm việc bình thường. Để giải quyết vấn đề này, bạn phải kết thúc tất cả các xử lý Oracle còn lại của môi trường làm việc trước đó trước khi khởi động môi trường làm việc mới.

Lắp ghép một cơ sở dữ liệu

Môi trường làm việc lắp ghép một cơ sở dữ liệu để kết hợp cơ sở dữ liệu với môi trường làm việc đó. Sau khi lắp ghép cơ sở dữ liệu ấy, môi trường làm việc tìm các tập tin điều khiển cơ sở dữ liệu và mở chung. (Các tập tin điều khiển được xác định trong tham số khởi hoạt CONTROL_FILES trong tập tin tham số được sử dụng để khởi động môi trường làm việc ấy). Sau đó Oracle đọc các tập tin điều khiển để nhận các tên của những tập tin dữ liệu của cơ sở dữ liệu và các tập tin nhật ký Redo Log.

Ở điểm này, cơ sở dữ liệu vẫn còn đóng và chỉ người điều hành cơ sở dữ liệu truy xuất được. Người điều hành cơ sở dữ liệu có thể tiếp tục đóng cơ sở dữ liệu khi hoàn tất các điều hoạt bảo trì riêng. Tuy nhiên, cơ sở dữ liệu chưa có thể sử dụng cho các điều hoạt bình thường.

Các chế độ lắp ghép một cơ sở dữ liệu với Parallel Server

Lưu ý:
Các đặc trưng được mô tả trong chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với

Parallel Server Option. Xem Getting to know Oracle8 và Oracle Enterprise Edition.

Nếu Oracle cho phép nhiều môi trường làm việc để lắp ghép cùng cơ sở dữ liệu đồng thời, người điều hành cơ sở dữ liệu có thể chọn để chạy cơ sở dữ liệu ở chế độ độc quyền hoặc dùng chung.

Chế độ độc quyền

Nếu môi trường làm việc thứ nhất lắp ghép một cơ sở dữ liệu thực hiện như vậy trong chế độ độc quyền, chỉ môi trường làm việc đó có thể lắp ghép cơ sở dữ liệu. Các phiên bản của Oracle không hỗ trợ tùy chọn Parallel Server, chúng cho phép một môi trường làm việc để lắp ghép cơ sở dữ liệu chỉ trong chế độ độc quyền.

Chế độ dùng chung

Nếu môi trường làm việc thứ nhất lắp ghép cơ sở dữ liệu được khởi động ở chế độ dùng chung (còn được gọi là chế độ "song song", các môi trường làm việc khác được khởi động trong chế độ dùng chung cũng có thể lắp ghép cơ sở dữ liệu ấy.

Số môi trường làm việc có thể lắp ghép cơ sở dữ liệu tùy theo số cực đại được xác định trước, mà bạn có thể xác định khi tạo cơ sở dữ liệu ấy.

Thông tin bổ sung :
Xem Oracle Parallel Server Concepts và Administration về những thông tin sử dụng nhiều môi trường làm việc với cơ sở dữ liệu đơn.

Mở một cơ sở dữ liệu

Mở một cơ sở dữ liệu được lắp ghép tạo cho nó có thể sử dụng cho các điều hoạt cơ sở dữ liệu bình thường. Bất kỳ người dùng hợp lệ có thể kết nối với một cơ sở dữ liệu mở và truy xuất thông tin của nó. Thường một người điều hành cơ sở dữ liệu mở cơ sở dữ liệu để tạo cho nó có thể sử dụng chung.

Khi bạn mở cơ sở dữ liệu đó, Oracle mở các tập tin dữ liệu trực tuyến và các tập tin ghi nhật ký Redo Log trực tuyến. Nếu một không bảng đã ngoại tuyến khi cơ sở dữ liệu đã đóng trước, thì không gian bảng và các tập tin dữ liệu tương ứng sẽ vẫn ngoại tuyến khi bạn mở lại cơ sở dữ liệu ấy. Xem "Đưa Các Không Gian Bảng Trực Tuyến Và Ngoại Tuyến".

Nếu bất kỳ một tập tin dữ liệu hoặc tập tin nhật ký hoạt động không có khi bạn cố mở cơ sở dữ liệu ấy, thì Oracle cho trở lại một lỗi. Bạn phải thực hiện phục hồi sao lưu của bất kỳ tập tin cơ sở dữ liệu nào bị hư trước khi bạn có thể mở cơ sở dữ liệu ấy.

Phục hồi môi trường làm việc

Nếu cơ sở dữ liệu đã đóng một cách bất thường, bởi vì người điều hành hủy ngang môi trường làm việc của nó hoặc vì sự cúp điện, Oracle tự động thực hiện phục hồi môi trường làm việc khi cơ sở dữ liệu được mở lại. Xem "Sự Cố Môi trường làm việc Của Cơ Sở Dữ Liệu".

Giành được đoạn hồi phục

Khi bạn mở cơ sở dữ liệu, môi trường làm việc cố gắng giành một hoặc nhiều các đọa hồi phục. Xem "SYSTEM

doạn hồi phục" và "Các Môi trường làm việc Oracle và Các Kiểu Của Các Đoạn Hồi Phục".

Giải quyết giao dịch phân phối không chắc chắn

Đôi khi một cơ sở dữ liệu có thể đóng một cách bất thường với một hoặc nhiều giao dịch phân phối không chắc chắn (không được cam kết và không hồi phục). Khi bạn mở lại cơ sở dữ liệu và phục hồi môi trường làm việc hoàn tất, thì xử lý nền RECO tự động giải quyết ngay và liên tiếp bất kỳ giao dịch được phân phối không chắc chắn. Xem thêm "Các Cơ Sở Dữ Liệu Được Phân Phối".

Thông tin bổ sung :
Xem "Các Hệ Thống Cơ Sở Dữ Liệu Được Phân Phối Của Oracle8" về những thông thường phục hồi từ các sai sót của giao dịch.

Cơ sở dữ liệu và đóng môi trường làm việc

Có ba bước để đóng một cơ sở dữ liệu và môi trường làm việc được kết hợp của nó là:

Đóng cơ sở dữ liệu
Tháo ghép cơ sở dữ liệu
Đóng môi trường làm việc

Một nhà điều hành cơ sở dữ liệu có thể thực hiện các bước này bằng cách sử dụng Oracle Enterprise Manager. Oracle tự động thực hiện tất cả ba bước bất kỳ khi nào một môi trường được đóng.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle Enterprise Administrator.

Đóng cơ sở dữ liệu

Khi bạn đóng một cơ sở dữ liệu, Oracle ghi tất cả dữ liệu của cơ sở dữ liệu và phục hồi dữ liệu trong SGA vào các tập tin dữ liệu và các tập tin nhật ký hoạt động tương ứng. Kế tiếp, Oracle đóng tất cả các tập tin dữ liệu và các tập tin nhật ký hoạt động trực tuyến. (Bất kỳ các tập tin dữ liệu ngoại tuyến nào của các không gian bảng ngoại tuyến sẽ được đóng rồi). Nếu bạn mở lại cơ sở dữ liệu liên tiếp, bất kỳ không gian bảng đã ngoại tuyến và các tập tin dữ liệu của nó vẫn còn ngoại tuyến và được đóng tương ứng. Ở điểm này, cơ sở dữ liệu được đóng và không thể truy xuất đối với các điều hoạt bình thường. Các tập tin điều khiển vẫn mở sau khi cơ sở dữ liệu được đóng nhưng vẫn được lắp ghép.

Đóng cơ sở dữ liệu bằng cách hủy ngang môi trường làm việc

Trong một ít trường hợp khẩn cấp, bạn có thể hủy ngang môi trường làm việc của một cơ sở dữ liệu mở trong để đóng và ngưng hoàn toàn cơ sở dữ liệu tức thời. Tiến trình này nhanh vì điều hoạt ghi tất cả dữ liệu trong các vùng đệm của SGA vào các tập tin dữ liệu và các tập tin ghi nhật ký Redo Log được bỏ qua. Mở lại cơ sở dữ liệu liên tiếp đòi hỏi phục hồi môi trường làm việc mà Oracle tự động thường.

Lưu ý:
Nếu một sự cố hệ thống hoặc điện bị cúp xảy ra khi cơ sở dữ liệu mở, môi trường làm việc thực sự bị "hủy ngang" và phục hồi môi trường làm việc được thực hiện khi cơ sở dữ liệu được mở lại.

Tháo lắp cơ sở dữ liệu

Một khi cơ sở dữ liệu được đóng, Oracle tháo lắp cơ sở dữ liệu để tách nó ra khỏi môi trường làm việc. Ở điểm này, môi trường làm việc vẫn còn trong bộ nhớ của máy tính.

Sau khi cơ sở dữ liệu được tháo lắp, Oracle đóng các tập tin điều khiển của cơ sở dữ liệu.

Đóng môi trường làm việc

Bước cuối cùng ngưng cơ sở dữ liệu là đóng môi trường làm việc ấy. Khi bạn đóng một môi trường làm việc, SGA được xóa khỏi bộ nhớ và các xử lý nền được kết thúc.

Đóng môi trường làm việc bất thường

Trong các trường hợp bất thường, đóng một môi trường làm việc không thể xảy ra hoàn toàn; tất cả các cấu trúc bộ nhớ không thể bị xóa khỏi bộ nhớ hoặc một trong các xử lý nền không thể được kết thúc. Khi những gì còn lại của cơ sở dữ liệu trước đó tồn tại, môi trường làm việc kế tiếp khởi động rất có thể không thực hiện được. Trong các trường hợp như thế, người điều hành cơ sở dữ liệu có thể buộc môi trường làm việc mới khởi động bằng cách trước hết xóa những gì còn lại của môi trường làm việc trước và sau đó khởi động một môi trường làm việc mới hoặc bằng cách đưa lệnh SHUTDOWN ABORT trong Oracle Enterprise Manager.

Thông tin bổ sung :
Để có thông tin chi tiết về môi trường làm việc và khởi động và đóng cơ sở dữ liệu, xem Hướng dẫn của Oracle8 Administrator.

Chương 6

Các cấu trúc bộ nhớ

Chương này để cập đến các cấu trúc bộ nhớ và các xử lý trong hệ thống cơ sở dữ liệu của Oracle:

- **Giới thiệu các cấu trúc bộ nhớ Oracle**
- **Vùng toàn cục hệ thống (SGA)**
- **Các vùng toàn cục chương trình (PGA)**
- **Cửa sổ vùng sắp xếp**
- **Bộ nhớ ảo**
- **Các vùng mã. phần mềm**

Giới thiệu các cấu trúc bộ nhớ Oracle

Oracle sử dụng bộ nhớ để lưu trữ những thông tin khác nhau:

Mã chương trình được tiến hành

Thông tin về phiên được kết nối, dù hiện chúng không hoạt động

Thông tin cần thiết khi chương trình thực hiện (thí dụ. trạng thái hiện hành của một truy vấn mà các hàng đang được truy nạp)

Thông tin được dùng chung giữa các xử lý của Oracle (thí dụ, thông tin khóa)
Dữ liệu được trữ cũng được lưu trữ vĩnh viễn trên bộ nhớ ngoại vi (thí dụ, các khối dữ liệu và các mục nhập nhật ký Redo Log)

Các cấu trúc bộ nhớ cơ bản được kết hợp với Oracle gồm:
Các Vùng Mã Phần Mềm
Vùng Toàn Cục Hệ Thống (SGA):
- Vùng Nhớ Cung Cấp Cơ Sở Dữ Liệu
- Vùng Đệm Ghi Nhật ký Redo Log
- Vùng Chia Dùng Chung

Các Vùng Toàn Cục Chương Trình (PGA):
- Các Vùng Ngăn Xếp
- Các Vùng Dữ Liệu

Các Vùng Sắp Xếp

Vùng toàn cục hệ thống (SGA)

Một vùng toàn cục hệ thống (SGA) là một nhóm các cấu trúc bộ nhớ dùng chung chứa dữ liệu và điều khiển thông tin cho một môi trường làm việc cơ sở dữ liệu Oracle. Nếu nhiều người dùng được kết nối đồng thời với cùng một môi trường làm việc, dữ liệu trong SGA của môi trường làm việc ấy được "dùng chung" giữa các người dùng. Do đó, SGA đôi khi được tham chiếu như là "vùng toàn cục dùng chung".

Như được mô tả trong "Tổng quan về Môi trường làm việc Oracle", một SGA và các xử lý Oracle cấu tạo một môi trường làm việc Oracle. Oracle tự động phân bổ bộ nhớ cho một SGA khi bạn khởi động một môi trường làm việc và hệ điều hành giành lại bộ nhớ ấy khi bạn đóng môi trường làm việc ấy. Mỗi môi trường làm việc có SGA riêng.

SGA ghi-đọc; tất cả người dùng được kết nối với môi trường làm việc cơ sở dữ liệu đa xử lý có thể đọc thông tin được chứa trong SGA của môi trường làm việc ấy và nhiều xử lý ghi vào SGA khi Oracle thực hiện.

SGA chứa các cấu trúc dữ liệu sau:
- Cache cùng đệm cơ sở dữ liệu
- Vùng đệm ghi nhật ký Redo Log
- Vùng chia dùng chung
- Cache từ điển dữ liệu
- Các thông tin hỗn hợp khác

Thành phần của SGA chứa thông tin tổng quát về trạng thái của cơ sở dữ liệu và môi trường làm việc mà các xử lý nền cần truy xuất; điều này được gọi là SGA cố định. Không có dữ liệu người dùng được lưu trữ ở đây.

SGA còn có thông tin được truyền thông giữa các xử lý, chẳng hạn như thông tin khóa.

Nếu hệ thống sử dụng kiến trúc server đa xâu và các hàng sắp trả lời và một số nội dung của các vùng toàn cục chương trình trong SGA. (Xem "Các Vùng Toàn Cục Chương Trình (PGA)", và "Yêu Cầu Trình Phân Công và Các Hàng Sắp Trả Lời".

Vùng nhớ cung cấp cơ sở dữ liệu

Vùng nhớ cung cấp cơ sở dữ liệu là một phần của SGA lưu trữ các bản sao của những khối dữ liệu được đọc từ các tập tin dữ liệu. Tất cả những xử lý người dùng được kết nối đồng thời với truy xuất môi trường làm việc dùng chung vùng nhớ cung cấp cơ sở dữ liệu.

Vùng nhớ cung cấp cơ sở dữ liệu và Cache SQL dùng chung được phân đoạn logic vào nhiều tập hợp. Tổ chức này vào nhiều tập hợp làm giảm sự tranh chấp trên các hệ thống đa xử lý.

Tổ chức của vùng nhớ cung cấp cơ sở dữ liệu

Các vùng đệm trong Cache được tổ chức thành hai danh sách: danh sách đánh dấu những thay đổi dữ liệu và Least Recently Used list (LRU). Danh sách đánh dấu những thay đổi dữ liệu lưu trữ các vùng đệm bị sửa đổi, chứa dữ liệu đã bị sửa đổi nhưng chưa được ghi vào đĩa. Least Recently Used list (LRU) lưu giữ các vùng đệm trống, các vùng đệm có chân và các vùng đệm bị sửa đổi chưa được di chuyển vào danh sách đánh dấu những thay đổi dữ liệu. Các vùng đệm trống chưa bị sửa đổi và có thể sử dụng. Các vùng đệm có chân đang được xử lý cùng một lúc.

Khi một Oracle xử lý các truy xuất một vùng đệm, xử lý ấy di chuyển vùng đệm vào cuối danh sách Most Recently Used - (MRU) của danh sách LRU. Khi nhiều vùng đệm tiếp tục được di chuyển vào cuối MRU của danh sách LRU, các vùng đệm bị sửa đổi "tăng" theo phần cuối LRU của danh sách LRU.

Lần đầu tiên một xử lý người dùng Oracle đòi hỏi một phần dữ liệu riêng, nó tìm dữ liệu ấy trong vùng nhớ cung cấp cơ sở dữ liệu. Nếu tiến trình tìm được dữ liệu trong cache ấy rồi (đạt kết quả cạche), thì nó có thể đọc dữ liệu trực tiếp từ bộ nhớ. Nếu xử lý không tìm được dữ liệu trong Cache (mất cache), nó phải sao chép khối dữ liệu từ một tập tin dữ liệu trên đĩa vào vùng đệm trong cache trước khi truy xuất dữ liệu. Dữ liệu truy xuất

cache đạt kết quả nhanh hơn truy xuất dữ liệu qua một cache mất.

Trước khi đọc một khối dữ liệu vào cache, tiến trình ấy trước hết phải tìm được một vùng đệm trống. Nó tìm danh sách LRU, khởi động ít nhất cuối danh sách mới được sử dụng. Tiến trình ấy tìm cho tới khi nó tìm thấy vùng đệm trống hoặc cho tới khi nó đã tìm thấy giới hạn ngưỡng của các vùng đệm.

Nếu tiến trình người dùng tìm thấy một vùng đệm bị sửa đổi như nó tìm danh sách LRU, nó di chuyển vùng đệm đó vào danh sách đánh dấu những thay đổi dữ liệu và tiếp tục tìm. Khi tiến trình ấy tìm thấy vùng đệm trống, nó đọc khối dữ liệu từ đĩa vào vùng đệm ấy và di chuyển vùng đệm đó vào cuối MRU của danh sách LRU.

Nếu một tiến trình người dùng Oracle tìm giới hạn ngưỡng của các vùng đệm mà không thấy một vùng đệm trống, thì tiến trình ấy ngưng tìm danh sách LRU và phát tín hiệu xử lý nền DBW0 để ghi một số vùng đệm bị sửa đổi vào đĩa. Để có thêm thông tin về xử lý DBW0 (hoặc nhiều xử lý DBWn), xem "DBWn Trình Ghi Cơ Sở Dữ Liệu".

Thuật toán LRU và duyệt toàn bộ bảng

Khi tiến trình người dùng thực hiện duyệt toàn bộ bảng, nó đọc các khối dữ liệu của bảng vào các vùng đệm và đưa chúng vào cuối LRU (thay vì cuối MRU) của danh sách LRU. Chính vì bảng tác vụ duyệt toàn bộ thường chỉ cần tóm tắt, vì vậy các khối dữ liệu phải được chuyển đi nhanh để có nhiều khối thường được dùng hơn trong cache.

Bạn có thể điều khiển xử lý mặc định của các khối liên quan trong bảng duyệt trên cơ sở từng bảng. Để xác định các khối dữ liệu của bảng phải được đặt vào cuối MRU của danh sách khi duyệt toàn bộ bảng, sử dụng mệnh đề CACHE khi tạo hoặc thay đổi bảng hoặc cụm liên kết. Bạn có thể cần xác định xử lý này cho các bảng tra cứu nhỏ hoặc các bảng có lịch sử tĩnh lớn để tránh I/O trên các truy xuất liên tiếp của bảng.

Thông tin bổ sung :
Xem Oracle8 SQL Referencevề những thông tin mệnh đề CACHE.

Kích thước của vùng nhớ cung cấp cơ sở dữ liệu

Tham số khởi hoạt DB_BLOCK_BUFFERS xác định số vùng đệm trong cache vùng đệm cở sở dữ liệu. Mỗi vùng đệm, trong cache là kích thước của khối dữ liệu Oracle. (được tham số khởi hoạt DB_BLOCK_SIZE xác định); do đó, mỗi vùng đệm cơ sở dữ liệu trong cache có thể lưu giữ một khối dữ liệu đơn được đọc từ một tập tin dữ liệu.

Cache có kích thước giới hạn, vì vậy không phải tất cả dữ liệu trên đĩa được điền vào cache. Khi cache đầy, các bỏ sót cache liên tiếp gây cho Oracle đọc dữ liệu bị sửa đổi trong cache vào đĩa để tạo không gian trống cho dữ liệu mới. (Nếu một vùng đệm không bị bị sửa đổi, nó không cần được ghi vào đĩa trước khi một khối mới có thể được đọc vào vùng đệm ấy). Truy xuất liên tiếp bất kỳ dữ liệu nào được ghi vào đĩa dẫn đến bỏ sót cache bổ sung.

Kích thước của cache ánh hưởng khả năng xảy ra yêu cầu dữ liệu sẽ dẫn đến kết quả cache. Nếu cache ấy lớn, nó

rất có thể chứa dữ liệu được yêu cầu. Gia tăng kích thước của cache làm tăng số phần trăm các yêu cầu dữ liệu dẫn đến các kết quả cache.

Thông tin bổ sung :
Xem Oracle8 Tuning thông tin về cache vùng đệm.

Nhiều vùng chia của vùng đệm

Bạn có thể cấu hình vùng nhớ cung cấp cơ sở dữ liệu với các vùng chia của vùng đệm tách riêng để lưu giữ dữ liệu trong cache vùng đệm hoặc tạo cho các vùng đệm có thể sử dụng dữ liệu mới ngay sau khi dùng các khối dữ liệu. Các đối tượng giản đồ riêng biệt (các bảng, các cụm liên kết, các chỉ mục và các vùng chia) sau đó có thể được gán vào vùng chia của vùng đệm thích hợp để điều khiển cách định thời hạn các khối dữ liệu của cache ấy. Vùng chia của vùng đệm KEEP giữ lại các khối dữ liệu của đối tượng giản đồ trong bộ nhớ.
Vùng chia của vùng đệm RECYCLE loại thải các khối dữ liệu khỏi bộ nhớ ngay khi chúng không còn được cần đến. Vùng chia của vùng đệm DEFAULT chứa các khối dữ liệu của các đối tượng giản đồ không được gán vào bất kỳ vùng chia của vùng đệm, cũng như các đối tượng giản đồ được gán cụ thể cho vùng chia DEFAULT.

Các tham số khởi hoạt cấu hình các vùng chia của vùng đệm KEEP và RECYCLE là BUFFER_POOL_KEEP và BUFFER_POOL_RECYCLE.

Thông tin bổ sung :
Xem Oracle8 Tuning về thông tin các vùng chia của vùng đệm và xem Oracle8 SQL Reference về cú pháp của tùy chọn BUFFER_POOL mệnh đề STORAGE.

Vùng đệm ghi nhật ký Redo Log

Vùng đệm ghi nhật ký Redo Log là một vùng đệm luân chuyển trong SGA lưu giữ thông tin về những thay đổi đối với cơ sở dữ liệu ấy. Thông tin này được lưu trữ trong các mục nhập redo. Các mục nhập redo chứa những thông tin cần thiết để tái cấu tạo hoặc thực hiện lại, thay đổi đối với cơ sở dữ liệu do các điều hoạt INSERT, UPDATE, DELETE, CREATE, ALTER hoặc DROP. Các mục nhập Redo được sử dụng để phục hồi cơ sở dữ liệu, nếu cần.

Các mục nhập redo được các xử lý Oracle server sao chép từ không gian bộ nhớ của người dùng vào vùng đệm ghi nhật ký Redo Log trong SGA. Các mục nhập thực hiện lại chiếm không gian liên tiếp trong vùng đệm. LGWR xử lý nền ghi vùng đệm ghi nhật ký Redo Log vào tập tin nhật ký Redo Log trực tuyến hoạt động (hoặc các nhóm tập tin) trên đĩa.

> Thông tin bổ sung :
> Xem "Xử Lý Trình Ghi Nhật ký hoạt động (LGWR) về những thông tin vùng đệm ghi nhật ký Redo Log được ghi vào đĩa như thế nào và xem Sao lưu Oracle8 và Hướng Dẫn Phục Hồi về những thông tin các tập tin ghi nhật ký Redo Log và các nhóm.

Tham số khởi hoạt LOG_BUFFER xác định kích thước (bằng byte) của vùng đệm ghi nhật ký Redo Log. Nói chung, các giá trị lớn hơn làm bớt I/O tập tin ghi nhật ký hoạt động, nhất là nếu các giao dịch dài hoặc nhiều. Xác lập mặc định gấp bốn lần kích thước khối dữ liệu tối đa cho hệ điều hành server.

Vùng chia dùng chung

Phần của vùng chia dùng chung của SGA chứa ba lãnh vực chính: cache thư viện, cache từ điển và các cấu trúc điều khiển. Hình 6-1 trình bày nội dung của vùng chia dùng chúng.

Tổng kích thước của vùng chia dùng chung được tham số khởi hoạt SHARED_POOL SIZE quyết định. Giá trị tham số mặc định này là 3.500.000 byte. Tăng giá trị của tham số này làm tăng dung lượng bộ nhớ dự trữ cho vùng chia dùng chung và do đó làm tăng không gian dự trữ cho các vùng SQL dùng chung.

Cache thư viện

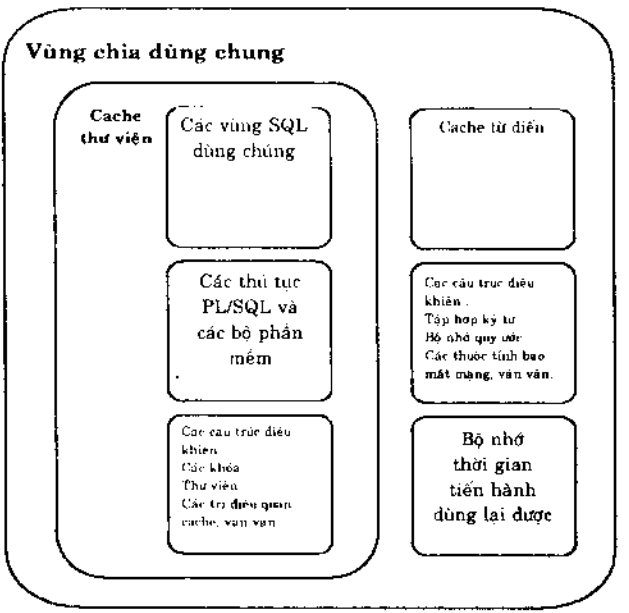
Cache thư viện gồm các vùng SQL, các vùng SQL riêng, các thủ tục PL/SQL và các bộ phần mềm, các cấu trúc điều khiển, chẳng hạn như các khóa và các xử lý cache thư viện.

Các vùng SQL dùng chung phải được nhiều người dùng sử dụng, vì vậy cache thư viện được chứa trong vùng chia dùng chung trong SGA. Kích thước của cache thư viện (cùng với kích thước của cache từ điển dữ liệu) bị kích thước của vùng chia dùng chung giới hạn.

Các vùng SQL dùng chung và các vùng SQL riêng

Oracle biểu thị mỗi câu lệnh SQL mà nó tiến hành với vùng SQL dùng chung và một vùng SQL riêng. Oracle nhận biết khi hai người dùng đang thực hiện cùng câu lệnh SQL và sử dụng lại vùng SQL dùng chung cho những người dùng đó. Tuy nhiên, mỗi người dùng phải có một bản sao vùng SQL riêng của câu lệnh.

Hình 6-1 Nội dung của vùng chia dùng chung



Vùng SQL dùng chung

Một vùng SQL dùng chung chứa cây phân ngữ và kế hoạch tiến hành cho câu lệnh SQL đơn hoặc cho các câu lệnh SQL đồng nhất. Oracle tiết kiệm bộ nhớ bằng cách sử dụng một vùng SQL dùng chung cho nhiều câu lệnh DML giống nhau, nhất là khi nhiều người dùng tiến hành cùng trình ứng dụng.

Thông tin bổ sung :
Xem Oracle Tuning thông tin về tiêu chuẩn xác định các câu lệnh SQL đồng nhất.

Oracle phân bổ bộ nhớ từ vùng đệm dùng chung khi câu lệnh SQL được phân ngữ; kích thước của bộ nhớ này tùy theo tính phức tạp của câu lệnh. Nếu câu lệnh SQL đòi

hỏi một vùng SQL dùng chung mới và toàn bộ vùng chia dùng chung đã được phân bổ rồi, Oracle có thể thôi phân bổ các mục của vùng chia ấy bằng cách sử dụng một thuật toán Least Recently Used được sửa đổi cho tới khi có đủ không gian trống cho vùng SQL dùng chung của câu lệnh mới đó. Nếu Oracle thôi phân bổ một vùng SQL dùng chung, câu lệnh SQL được kết hợp phải được phân ngữ lại và được gán lại thành vùng SQL dùng chung khác khi nó được tiến hành tiếp

Các vùng SQL riêng

Một vùng SQL riêng chứa dữ liệu, chẳng hạn như thông tin kết hợp và các vùng đệm thời gian thực hiện. Mỗi phiên đưa câu lệnh SQL có vùng SQL riêng. Mỗi người dùng đệ trình một câu lệnh SQL có vùng SQL riêng của họ sử dụng một vùng SQL dùng chung đơn; nhiều vùng SQL riêng có thể được kết hợp với vùng SQL dùng chung giống nhau. (Xem "Các Kết Nối Và Các Phiên" các thông tin về các phiên).

Vùng SQL riêng có vùng liên tục và vùng thời gian thực hiện:

Vùng liên tục chứa thông tin kết hợp liên tục khi thực hiện, mã để chuyển đổi kiểu dữ liệu (trong trường hợp kiểu dữ liệu được xác định không giống với kiểu dữ liệu của cột được lựa), và thông tin trạng thái khác (như các số con trỏ từ xa hoặc tái quy hay trạng thái của một truy vấn song song). Kích thước của vùng liên tục tùy theo số của kết hợp và các cột được xác định trong câu lệnh. Thí dụ, vùng liên tục lớn hơn nếu nhiều cột được xác định trong một truy vấn.

Vùng thời gian thực hiện chứa thông tin được sử dụng khi câu lệnh SQL đang được tiến hành. Kích thước của vùng thời gian thực hiện tùy theo kiểu và tính phức tạp của câu lệnh SQL được tiến hành và những kích thước

được câu lệnh xử lý. Nói chung, vùng thời gian chạy phần ít hơn đối với các câu lệnh INSERT, UPDATE và DELETE so với các câu lệnh SELECT.

Oracle tạo vùng thời gian thực hiện là bước đầu tiên của yêu cầu tiến hành. Đối với các câu lệnh INSERT, UPDATE và DELETE, Oracle giải phóng vùng thời gian thực hiện sau khi câu lệnh được tiến hành. Đối với các truy vấn, Oracle giải phóng vùng thời gian thực hiện chỉ sau khi tất cả các hàng được truy nạp hoặc truy vấn được hủy bỏ.

Vị trí của một vùng SQL riêng phụ thuộc vào kiểu kết nối được thiết lập cho phiên. Nếu phiên được kết nối bằng một server chuyên dùng, các vùng SQL riêng được định vị trong PGA của người dùng. Tuy nhiên, nếu một phiên được kết nối bằng server đa xâu, các vùng liên tục và đối với các câu lệnh SELECT, các vùng thời gian thực hiện được lưu giữ trong SGA.

Các con trỏ và các vùng SQL

Developer ứng dụng của chương trình Oracle Precompiler hoặc chương trình OCI có thể mở các con trỏ rõ ràng hoặc xử lý các vùng SQL riêng và sử dụng chung làm tài nguyên có tên suốt thời gian tiến hành chương trình ấy. Các con trỏ tái quy mà Oracle đưa ra một số câu lệnh SQL còn sử dụng các vùng SQL dùng chung. Xem thêm "Các Con Trỏ"

Quản lý các vùng SQL riêng là trách nhiệm của tiến trình người dùng. Phân bổ và thôi phân bổ các vùng SQL riêng phụ thuộc nhiều vào công cụ ứng dụng bạn đang sử dụng, dù số vùng SQL riêng mà xử lý người dùng có thể phân bổ luôn luôn bị giới hạn bởi tham số

OPEN_CURSORS. Giá trị mặc định của tham số này là 50.

Một vùng SQL riêng tiếp tồn tại cho tới khi con trỏ tương ứng được đóng hoặc xử lý câu lệnh được giải phóng. Dù Oracle giải phóng vùng thời gian thực hiện sau khi câu lệnh hoàn tất, nhưng vùng liên tục vẫn chờ. Những Developer ứng dụng phải đóng tất cả các con trỏ mở sẽ không được sử dụng lại để giải phóng vùng liên tục và tối thiểu hóa dùng lượng bộ nhớ cần thiết cho ứng dụng của người dùng.

Đối với các truy vấn mà xử lý số lượng lớn sắp xếp đòi hỏi dữ liệu, các Developer ứng dụng sẽ hủy bỏ truy vấn ấy nếu một phần kết quả của truy nạp được thỏa đáng. Thí dụ, trong một ứng dụng Oracle Office, người dùng có thể lựa từ danh sách trên 60 khuôn mẫu để tạo một thông điệp thư. Khi Oracle Office hiển thị 10 tên khuôn mẫu đầu tiên, nếu người dùng chọn một trong các khuôn mẫu này thì ứng dụng sẽ hủy bỏ tiến trình của phần còn lại của truy vấn, thay vì tiếp tục cố hiển thị nhiều tên khuôn mẫu hơn.

Các đơn vị chương trình PL/SQL và vùng chia dùng chung

Oracle xử lý những đơn vị chương trình PL/SQL (các thủ tục, các hàm, các bộ phần mềm, các khối nặc danh và các bộ kích khởi cơ sở dữ liệu) giống như phương thức nó xử lý các câu lệnh SQL riêng biệt. Oracle phân bổ một vùng dùng chung để lưu giữ biểu mẫu được biên dịch, được phân ngữ của một đơn vị chương trình. Oracle phân bổ một vùng riêng để lưu giữ những giá trị xác định đối với phiên thực hiện đơn vị chương trình, kể cả các biến

cục bộ, toàn cục và phần mềm (cũng được coi là lập môi trường làm việc phần mềm) và các vùng đệm để tiến hành SQL. Nếu nhiều người dùng thực hiện cùng đơn vị chương trình, sau đó một vùng dùng dung đơn được tất cả người dùng sử dụng, trong khi mỗi người dùng duy trì bản sao tách riêng vùng SQL riêng của họ, lưu giữ các giá trị xác định với phiên của họ.

Các câu lệnh SQL riêng được chứa trong một đơn vị chương trình PL/SQL được xử lý trong các phiên trước đó. Dù các ban đầu của chúng trong đơn vị chương trình PL/SQL, các câu lệnh SQL này sử dụng vùng dùng chung để lưu giữ các biểu thị được phân ngữ và vùng riêng cho mỗi phiên tiến hành câu lệnh ấy.

Cache từ điển

Từ điển dữ liệu là một tập hợp các bảng cơ sở dữ liệu và các khung nhìn chứa những thông tin tham chiếu về cơ sở dữ liệu, các cấu trúc và những người dùng của nó. Oracle truy xuất từ điển dữ liệu khi phân ngữ của các câu lệnh SQL. Truy xuất này cần thiết đối với điều hoạt liên tục của Oracle. Xem "Từ Điển Dữ Liệu".

Từ điển dữ liệu được Oracle truy xuất rất thường mà hai vị trí trong bộ nhớ được thiết kế để lưu giữ dữ liệu từ điển. Một vùng được gọi là cache từ điển dữ liệu, cũng được coi là cache hàng vì nó lưu giữ dữ liệu bằng các hàng thay vì bằng các vùng đệm (lưu giữ toàn bộ những khối dữ liệu). Vùng khác trong bộ nhớ lưu giữ dữ liệu từ điển là cache thư viện. Tất cả những xử lý người dùng Oracle dùng chung hai cache này để truy xuất thông tin của từ điển dữ liệu.

Phân bổ và dùng lại bộ nhớ trong vùng chia dùng chung

Nói chung, bất kỳ mục (vùng SQL dùng chung hoặc hàng từ điển) trong vùng chia dùng chung vẫn duy trì cho tới khi nó được xóa tương ứng với thuật toán LRU có sửa đổi. Bộ nhớ đối với các mục không được sử dụng thường được giải phóng nếu cần không gian cho các mục mới phải được phân bổ không gian nào đó trong vùng chia dùng chung. Một thuật toán LRU có sửa đổi cho phép các mục của vùng chia dùng chung được nhiều phiên sử dụng để duy trì trong bộ nhớ bau lâu chúng còn cần thiết, dù tiến trình ấy đầu tiên đã tạo các kết thúc mục. Do đó, phần điều hành chung và xử lý các câu lệnh SQL được kết hợp với hệ thống Oracle nhiều người dùng giảm tối đa.

Khi một câu lệnh SQL được đệ trình để Oracle thực hiện, Oracle tự động thực hiện các bước phân bổ bộ nhớ sau:

Oracle kiểm vùng chia dùng chung để xem một vùng SQL dùng chung có còn tồn tại cho một câu lệnh đồng nhất không. Nếu có, thì vùng SQL dùng chung được sử dụng để tiến hành các môi trường làm việc mới kế tiếp của câu lệnh ấy. Hoặc, nếu không có vùng SQL dùng chung cho một câu lệnh, thì Oracle phân bổ một vùng SQL dùng chung mới trong vùng chia dùng chung. Trọng hai trường hợp này, vùng SQL riêng của người dùng được kết hợp với vùng SQL dùng chung chứa câu lệnh ấy.

Lưu ý:
Vùng SQL dùng chung có thể được xóa từ vùng chia dùng chung, dù vùng SQL dùng chung tương ứng với một con trỏ mở chưa được sử dụng vào lúc nào đó. Nếu con trỏ mở được sử dụng tiếp theo để thực thi

câu lệnh của nó. Oracle phân ngữ lại câu lệnh ấy và một vùng SQL dùng chung mới được phân bổ vào vùng chia dùng chung.

Oracle phân bổ một vùng SQL riêng thay cho phiên. Vị trí chính xác của vùng SQL riêng phụ thuộc vào kết nối được thiết lập cho phiên (xem Các Vùng SQL và Các Vùng SQL Riêng).

Oracle còn xóa một vùng SQL dùng chung từ vùng chia dùng chung trong các trường hợp sau:
Khi lệnh ANALYZE được sử dụng để cập nhật hoặc xóa các thống kê của một bảng, cụm liên kết hoặc chỉ mục, thì tất cả các vùng SQL dùng chung chứa các câu lệnh tham chiếu đối tượng giản đồ phân tích được xóa từ vùng chia dùng chung. Lần kế tiếp một câu lệnh xóa được tiến hành, câu lệnh ấy được phân ngữ trong vùng SQL dùng chung mới để phản ánh các thống kê mới cho đối tượng giản đồ đó.

Nếu một đối tượng giản đồ được tham chiếu trong một câu lệnh SQL và đối tượng đó sau đó được sửa đổi bằng cách nào đó, thì vùng SQL dùng chung không hợp lệ (được đánh dấu không hợp lệ) và câu lệnh ấy phải được phân ngữ lại vào lần kế tiếp khi được thực hiện. Xem "Quản lý sự phụ thuộc Oracle" về thông tin không hợp lệ của các câu lệnh SQL và các phát hành lệnh phụ thuộc.

Nếu bạn thay đổi tên cơ sở dữ liệu toàn cục của cơ sở dữ liệu, tất cả thông tin được xóa từ vùng chia dùng chung.

Người điều hành có thể xóa thủ công tất cả thông tin trong vùng chia dùng chung để truy xuất hiệu suất (liên

quan đến vùng chia dùng chung, không liên quan đến cache vùng đệm dữ liệu) có thể được mong muốn sau khi môi trường làm việc khởi động mà không đóng môi trường làm việc hiện hành.

Kích thước của SGA

Kích thước của SGA được môi trường làm việc khởi động xác định. Đối với hiệu suất trong hầu hết các hệ thống, toàn bộ SGA sẽ so khớp với bộ nhớ thực. Nếu nó không so khớp với bộ nhớ thực và bộ nhớ ảo (xem "Bộ Nhớ Ảo") được dùng để lưu trữ các thành phần của nó, hiệu suất hệ thống cơ sở dữ liệu tổng quát có thể làm giảm khủng khiếp vì các phần của SGA được lập trang (được ghi và đọc từ đĩa) bởi hệ điều hành. Số lượng của bộ nhớ được chuyên dùng cho tất cả các vùng dùng chung trong SGA cũng có ảnh hưởng đến hiệu suất; xem Oracle8 Tuning.

Kích thước của SGA được các tham số khởi hoạt xác định. Các tham số ảnh hưởng đến kích thước của SGA nhất là:

DB_BLOCK_SIZE	Kích thước tính bằng byte của một khối dữ liệu đơn và vùng đệm của cở sở dữ liệu.
DB_BLOCK_BUFFERS	Số vùng đệm của cơ sở dữ liệu, mỗi kích thước của DB_BLOCK_SIZE, được phân bổ cho SGA. (tổng số không gian được phân bổ cho cache vùng đệm của cơ sở dữ liệu trong SGA theo số lần DB_BLOCK_SIZE nhân với DB_BLOCK_BUFFERS.)
LOG_BUFFER	Số byte được phân bổ cho vùng đệm ghi nhật ký Redo Log.

SHARED_POOL_SIZE	Kích thước tính bằng byte của vùng được dành cho các lệnh SQL dùng chung và PL/SQL.

Bộ nhớ được phân bổ cho một SGA của môi trường làm việc được hiển thị vào lúc khởi động môi trường làm việc khi sử dụng Oracle Enterprise Manager (hoặc Server Manger). Bạn cũng có thể hiển thị kích thước SGA của môi trường làm việc hiện hành bằng cách sử dụng lệnh Server Manager SHOW với tùy chọn SGA.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle Enterprise Manager Administrator về kích thước SGA với Oracle Enterprise Manager (hoặc Server Manager).

Xem Oracle8 Tuning về các tham số khởi hoạt ở trên và chúng ảnh hưởng đến SGA như thế nào. Xem thêm cài đặt Oracle hoặc hướng dẫn người dùng để xác định hệ điều hành của bạn.

Điều khiển sử dụng bộ nhớ của SGA

Nhiều tham số khởi hoạt có thể sử dụng để điều khiển SGA sử dụng bộ nhớ như thế nào. Để có thông tin chi tiết về những tham số này, xem Oracle8 Reference.

Bộ nhớ vật lý

Các tham số LOCK_SGA và LOCK SGA_AREAS khóa toàn bộ SGA hoặc các vùng SGA đặc biệt vào bộ nhớ vật lý.

Địa chỉ khởi động SGA

Các tham số SHARED_MEMORY_ADDRESS và HI_SHARED_MEMORY_ADDRESS xác định địa chỉ khởi động của SGA vào lúc thực hiện. Những tham số này chỉ được sử dụng trên các nền không xác định địa chỉ khởi động của SGA vào thời gian liên kết. Đối với các nền 64-bit, HI_SHARED_MEMORY_ADDRESS xác định 32 bit thứ tự cao của địa chỉ 64-bit.

Cơ chế cache vùng đệm mở rộng

Các tham số The USE_INDIRECT_DATA_BUFFERS cho phép cơ chế cache vùng đệm mở rộng đối với các nền 32-bit có thể hỗ trợ nhiều hơn 4 GB bộ nhớ vật lý.

Các vùng toàn cục chương trình (PGA)

Một vùng toàn cục chương trình (PGA) là vùng bộ nhớ chứa dữ liệu và điều khiển thông tin cho một xử lý đơn (server hoặc nền). Do đó, đôi khi một PGA được gọi là "vùng toàn cục xử lý".

Một PGA là vùng bộ nhớ không dùng chung mà một tiến trình có thể ghi; PGA dành riêng cho xử lý của server và được đọc và ghi chỉ bởi mã Oracle tác động thay cho tiến trình đó.

Một PGA được Oracle phân bổ khi người dùng kết nối với cơ sở dữ liệu Oracle và phiên được tạo, tuy nhiên điều này khác nhau tùy theo hệ điều hành và cấu hình. (Xem "Kết Nối và Các Phiên").

Nội dung của một PGA

Nội dung của một PGA khác nhau tùy theo môi trường làm việc được kết hợp đang chạy server đa xâu. (Xem Server Đa Xâu").

Không gian ngăn xếp

Một PGA luôn luôn chứa không gian ngăn xếp mà bộ nhớ phân bổ để lưu giữ các biến, các mảng và các thông tin khác của phiên.

Thông tin phiên

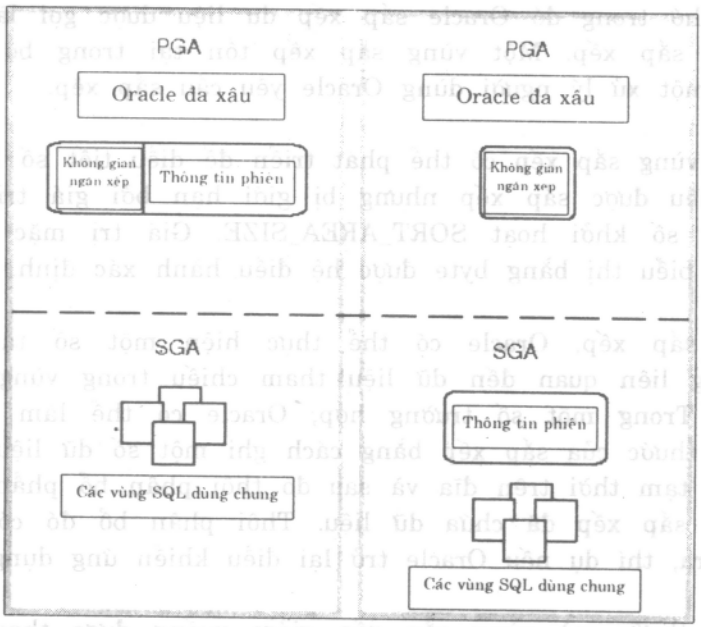
Nếu môi trường làm việc đang chạy không có server đa xâu, thì PGA cũng chứa thông tin về phiên của người dùng, chẳng hạn như các vùng SQL riêng. Nếu môi trường làm việc đang trong cấu hình server đa xâu, thông tin phiên này không ở trong PGA, nhưng được thay vào đó được phân bổ trong SGA.

Hình 6-2 trình bày nơi thông tin phiên được lưu trữ trong các cấu hình khác nhau.

Kích thước của PGA

Kích thước của PGA's cố định và hệ điều hành xác định. Khi client và server trên các máy khác nhau, thì PGA được phân bổ trên server cơ sở dữ liệu vào thời gian kết nối; nếu không đủ bộ nhớ để kết nối, một lỗi Oracle xảy ra với một số hiệu lỗi trong phạm vi đối với hệ điều hành. Một khi được kết nối, người dùng có thể không bao giờ chạy hết không gian PGA; có đủ hoặc không có đủ bộ nhớ để kết nối ở vị trí đầu tiên.

Hình 6-2 Vị trí của thông tin phiên có và không có server đa xâu



Các tham số khởi hoạt sau ảnh hưởng đến kích thước của các PGA:
OPEN_LINKS
DB_FILES
LOG_FILES

Kích thước của không gian ngăn xếp trong mỗi PGA được tạo thay cho các xử lý nền của Oracle (chẳng hạn như DBW0 và LGWR) bị ảnh hưởng bởi một số tham số bổ sung.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle xác định về PGA.

Các vùng sắp xếp

Sắp xếp đòi hỏi không gian trong bộ nhớ. Các phần của bộ nhớ trong đó Oracle sắp xếp dữ liệu được gọi là các vùng sắp xếp. Một vùng sắp xếp tồn tại trong bộ nhớ của một xử lý người dùng Oracle yêu cầu sắp xếp.

Một vùng sắp xếp có thể phát triển để điều tiết số lượng dữ liệu được sắp xếp nhưng bị giới hạn bởi giá trị của tham số khởi hoạt SORT_AREA_SIZE. Giá trị mặc định được biểu thị bằng byte được hệ điều hành xác định.

Khi sắp xếp, Oracle có thể thực hiện một số tác vụ không liên quan đến dữ liệu tham chiếu trong vùng sắp xếp. Trong một số trường hợp, Oracle có thể làm giảm kích thước của sắp xếp bằng cách ghi một số dữ liệu vào đoạn tạm thời trên đĩa và sau đó thôi phân bổ phần của vùng sắp xếp đã chứa dữ liệu. Thôi phân bổ đó có thể xảy ra, thí dụ nếu Oracle trở lại điều khiển ứng dụng ấy.

Kích thước mà vùng sắp xếp giảm xuống được tham số khởi hoạt SORT_AREA_RETAINED_SIZE quyết định. Giá trị của tham số này được biểu thị bằng byte. Giá trị tối thiểu tương đương với một khối cơ sở dữ liệu; giá trị tối đa (và mặc định) là giá trị của tham số khởi hoạt SORT_AREA_SIZE.

Bộ nhớ được giải phóng khi một sắp xếp được giải phóng để cùng tiến trình Oracle sử dụng, nhưng nó không được giải phóng cho hệ điều hành.

Nếu số lượng dữ liệu được sắp xếp không được điền vào vùng sắp xếp, thì dữ liệu được chia thành các phần nhỏ hơn để cho thích hợp. Mỗi phần nhỏ sau đó được sắp xếp riêng. Các phần được sắp xếp riêng được gọi là "các

phần thực hiện". Sau khi sắp xếp tất cả các phần thực hiện, Oracle hợp nhất chúng để cho kết quả cuối cùng.

Ghi trực tiếp sắp xếp

Ghi trực tiếp sắp xếp cung cấp một phương pháp hòa hợp tự động để phái sinh kích thước và số vùng đệm ghi trực tiếp dựa trên kích thước vùng sắp xếp. Bộ nhớ cho các vùng đệm được lấy từ vùng sắp xếp, vì vậy chỉ cần tham số hòa hợp. Hơn nữa, mô hình chi phí của bộ ưu hóa được cung cấp.

Nếu bộ nhớ và không gian tạm thời còn nhiều trên hệ thống của bạn và bạn thực hiện các sắp xếp lớn vào đĩa, sự xác lập tham số khởi hoạt SORT_DIRECT_WRITES có thể làm tăng hiệu suất sắp xếp.

Thông tin bổ sung :
Xem Oracle8 Tuning về các vùng sắp xếp và tham số SORT_DIRECT_WRITES.

Bộ nhớ ảo

Trên nhiều hệ điều hành, Oracle khai thác bộ nhớ ảo - một đặc trưng hệ điều hành cung ứng nhiều bộ nhớ hơn được chỉ bộ nhớ thực cung cấp và linh hoạt hơn sử dụng bộ nhớ chính.

Bộ nhớ ảo kích thích bộ nhớ sử dụng tổ hợp bộ nhớ thực (chính) và bộ nhớ thứ cấp (thường là không gian đĩa). Hệ điều hành truy xuất bộ nhớ ảo bằng ách làm cho bộ nhớ thứ cấp nhìn giống như bộ nhớ chính đối với các trình ứng dụng.

Đề nghị:
Thường, lưu trữ toàn bộ SGA trong bộ nhớ thực là tốt nhất. trên các nhiều nền, bạn có thể khóa SGA hoặc các thành phần của nó vào bộ nhớ thực bằng tham số LOCK_SGA và LOCK_SGA_AREAS.

Các vùng mã phần mềm

Các vùng mã phần mềm là các phần của bộ nhớ được sử dụng để lưu trữ mã đang được thực thi hoặc có thể được tiến hành. Mã Oracle được lưu trữ trong vùng phần mềm thường ở vị trí khác so với các chương trình của người dùng - một vị trí độc quyền hoặc được bảo vệ.

Các vùng phần mềm thường tĩnh về kích thước, chỉ thay đổi khi phần mềm được cập nhật hoặc cài đặt lại. Kích thước cần thiết của các vùng này khác nhau tùy theo hệ điều hành.

Các vùng phần mềm chỉ đọc và có thể được cài đặt dùng chung hoặc không dùng chung. Khi có thể, mã Oracle được dùng chung để tất cả người dùng Oracle có thể truy xuất nó mà không phải sao chép nhiều trong bộ nhớ. Những kết quả này để tiết kiệm bộ nhớ chính và tăng hiệu suất tổng quát.

Các chương trình người dùng có thể được dùng chung hoặc không được dùng chung. Một số công cụ và những tiện ích của Oracle (chẳng hạn như SQL*Forms và SQL*Plus) có thể được cài đặt dùng chung, nhưng một số không thể được. Nhiều môi trường làm việc của Oracle có thể dùng cùng vùng mã Oracle với các cơ sở dữ liệu khác nhau nếu chạy trên cùng máy tính.

Thông tin bổ sung :
Tùy chọn của phần mềm cài đặt dùng chung không thể sử dụng cho tất cả các hệ điều hành (thí dụ, trên các PC chạy MS DOS). Xem thêm tài liệu hệ điều hành Oracle xác định.

Chương 7

Cấu trúc xử lý

Chương này đề cập các xử lý trong hệ thống cơ sở dữ liệu Oracle và các cấu hình khác nhau có thể sử dụng hệ thống Oracle. Nó gồm:

- **Giới thiệu Các xử lý**
- **Oracle đơn xử lý**
- **Oracle đa xử lý**
- **Các khác nhau về cấu hình Oracle**
- **Các thí dụ về cách Oracle hoạt động**
- **Giao diện chương trình**

Giới thiệu các xử lý

Tất cả người dùng Oracle được kết nối phải thực hiện các môđun mã để truy xuất một môi trường làm việc cơ sở dữ liệu của Oracle:

Ứng dụng hoặc công cụ Oracle	Một người dùng cơ sở dữ liệu thực hiện một ứng dụng cơ sở dữ liệu (chẳng hạn như chương trình biên dịch trước) hoặc một công cụ Oracle (chẳng hạn như ứng dụng Oracle

	Forms hoặc SQL*Plus), đưa các câu lệnh SQL vào cơ sở dữ liệu Oracle.
Mã Oracle server	Mỗi người dùng có mã server nào đó tiến hành thay cho họ để diễn dịch và xử lý các câu lệnh SQL.

Những Môđun mã này được tiến hành bằng các xử lý. Một xử lý là một "xâu điều khiển" hoặc cơ chế trong một hệ điều hành có thể tiến hành một loạt các bước. (Một số hệ điều hành dùng tác vụ thời hạn hoặc tác vụ). Một xử lý thường có vùng nhớ tự nhân trong đó nó chạy.

Cấu trúc xử lý khác nhau tùy theo cấu hình khác nhau của Oracle, phụ thuộc vào hệ điều hành và chọn lựa các tùy chọn của Oracle.

Oracle đơn xử lý

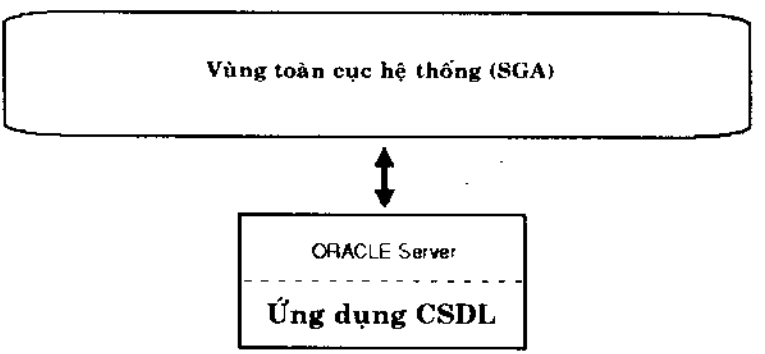
Oracle đơn xử lý (còn được gọi Oracle một người dùng) là một hệ thống cơ sở dữ liệu trong đó một xử lý thực hiện tất cả các phần của mã Oracle server và trình ứng dụng của một người dùng. Các xử lý khác nhau không tách sự thực hiện của môi trường làm việc ra trình ứng dụng client.

Hình 7-1 trình bày một môi trường làm việc Oracle xử lý đơn. Đơn xử lý thực hiện tất cả mã được kết hợp với ứng dụng cơ sở dữ liệu và Oracle.

Chỉ một người dùng có thể truy xuất môi trường làm việc Oracle trong môi trường xử lý đơn; nhiều người dùng không thể truy xuất cơ sở dữ liệu ấy đồng thời. Thí dụ, Oracle chạy dưới hệ điều hành MS-DOS trên PC chỉ có thể được truy xuất bằng một người dùng vì MS-DOS không thể chạy da xử lý.

Cấu hình này thường không phải Oracle đa xử lý, vì nó không khai thác được xử lý phân phối bình thường kết hợp với các điều hoạt cơ sở dữ liệu.

Hình 7-1 Môi trường làm việc Oracle đơn xử lý



Xử lý đơn

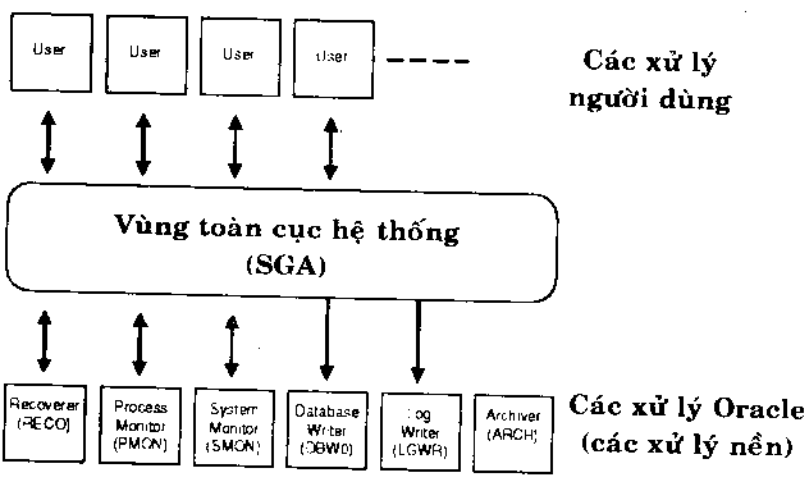
Oracle đa xử lý

Oracle đa xử lý (còn được gọi là Oracle đa người dùng) sử dụng nhiều xử lý để thực hiện các phần khác nhau của mã Oracle và một xử lý tách riêng cho mỗi người dùng được kết nối. Mỗi xử lý trong môi trường làm việc Oracle đa xử lý thực hiện một tác vụ riêng. Bằng cách chia công việc của Oracle và các ứng dụng cơ sở dữ liệu vào nhiều xử lý, nhiều người dùng các ứng dụng có thể kết nối với môi trường làm việc cơ sở dữ liệu đơn đồng thời khi hệ thống duy trì hiệu suất tốt nhất.

Phần lớn những hệ thống cơ sở dữ liệu là đa người dùng, vì một trong các lợi ích chính của cơ sở dữ liệu quản lý dữ liệu nhiều người dùng cần cùng một lúc.

Hình 7-2 minh họa môi trường làm việc Oracle đa xử lý. Mỗi người dùng được kết nối xử lý người tách riêng và nhiều xử lý nền thực hiện Oracle. Hình này có thể biểu thị nhiều người dùng đồng thời chạy một ứng dụng trên cùng máy bằng Oracle; cấu hình đặc biệt này thường chạy trên máy tính lớn hoặc máy vi tính.

Hình 7-2 Môi trường làm việc Oracle đa xử lý



Trong một hệ thống đa xử lý, các xử lý có thể được phân loại thành hai nhóm: các xử lý người dùng và các xử lý Oracle. Các xử lý người dùng thực hiện ứng dụng hoặc mã công cụ Oracle và các xử lý Oracle xử lý mã Oracle server.

Các xử lý người dùng

Khi người dùng chạy một trình ứng dụng (chẳng hạn như chương trình Pro*C) hoặc một công cụ Oracle (chẳng hạn như Oracle Enterprise Manager hoặc SQL*Plus) Oracle tạo

một xử lý người dùng để chạy ứng dụng của người dùng đó.

Các kết nối và các phiên

Thuật ngữ "kết nối" và "phiên" có quan hệ sát với thuật ngữ "xử lý của người dùng", nhưng khác nhau về ý nghĩa. Một kết nối là một đường dẫn truyền thông giữa xử lý người dùng và một môi trường làm việc Oracle. Đường dẫn truyền thông được thiết lập để sử dụng các cơ chế truyền thông liên xử lý (trên một máy tính thực hiện cả hai mà người xử lý một Oracle) hoặc phần mềm mạng (khi các máy tính khác nhau thực hiện ứng dụng cơ sở dữ liệu Oracle và truyền thông qua mạng).

Một phiên là một kết nối cd của người dùng với một môi trường làm việc Oracle bằng một xử lý người dùng. Thí dụ, khi người dùng khởi động SQL*Plus, người dùng đó phải cung cấp tên người dùng và mật khẩu hợp lệ và sau đó một phiên được thiết lập cho người dùng ấy. Một phiên kéo dài từ thời gian gian người dùng kết nối cho tới khi người dùng thôi kết nối hoặc tồn tại ứng dụng cơ sở dữ liệu.

Nhiều phiên có thể được tạo và tồn tại đồng thời cho một người dùng Oracle đơn bằng cách sử dụng cùng tên người dùng. Thí dụ, một người dùng có tên người dùng/mật khẩu là SCOTT/TIGER có thể kết nói với cùng môi trường làm việc Oracle nhiều lần.

Trong các cấu hình không có server đa xâu, Oracle tạo một xử lý server thay cho mỗi phiên người dùng; tuy nhiên, với server đa xâu, nhiều phiên người dùng có thể dùng chung một xử lý server đơn. Xem Server Đa Xâu".

Các xử lý Oracle

Trong các hệ thống đa xử lý, hai kiểu xử lý điều khiển Oracle: các xử lý server và các xử lý nền.

Oracle tạo các xử lý server để xử lý các yêu cầu của các xử lý người dùng được kết nối với môi trường làm việc ấy. Một số trường hợp khi ứng dụng và Oracle điều hoạt trên cùng máy, nó có thể kết hợp xử lý người dùng và xử lý server tương ứng thành một xử lý để làm giảm phần điều hành chung hệ thống. Tuy nhiên, khi ứng dụng và Oracle điều hạot trên các máy khác nhau, một xử lý người dùng truyền thông với Oracle bằng một xử lý server tách. Xem "Các Khác Nhau Trong Cấu Hình Oracle"

Các xử lý server

Các xử lý server (hoặc thành phần server của các xử lý người dùng/server) được tạo thay cho mỗi ứng dụng của người dùng có thể thực hiện một hoặc nhiều câu lệnh sau:
Phân ngữ và tiến hành các câu lệnh SQL được đưa ra bằng ứng dụng ấy.
Đọc các khối dữ liệu cần thiết từ các tập tin dữ liệu trên đĩa vào những vùng đệm cơ sở dữ liệu dùng chung của SGA, nếu các khối ấy chưa được biểu thị trong SGA.
Các kết quả trở lại bằng cách sao cho ứng dụng ấy có thể xử lý thông tin.

Các xử lý nền

Để tối đa hóa hiệu suất và điều tiết nhiều người dùng, một hệ thống Oracle đa xử lý dùng một số xử lý Oracle bổ sung được gọi là các xử lý nền.

Trên nhiều hệ điều hành, các xử lý nền được tạo tự động khi một môi trường làm việc được khởi động. Trên các hệ điều hành khác, các xử lý server được tạo làm thành phần của cài đặt Oracle.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle xác định về thông tin chi tiết các xử lý này được xử lý như thế nào.

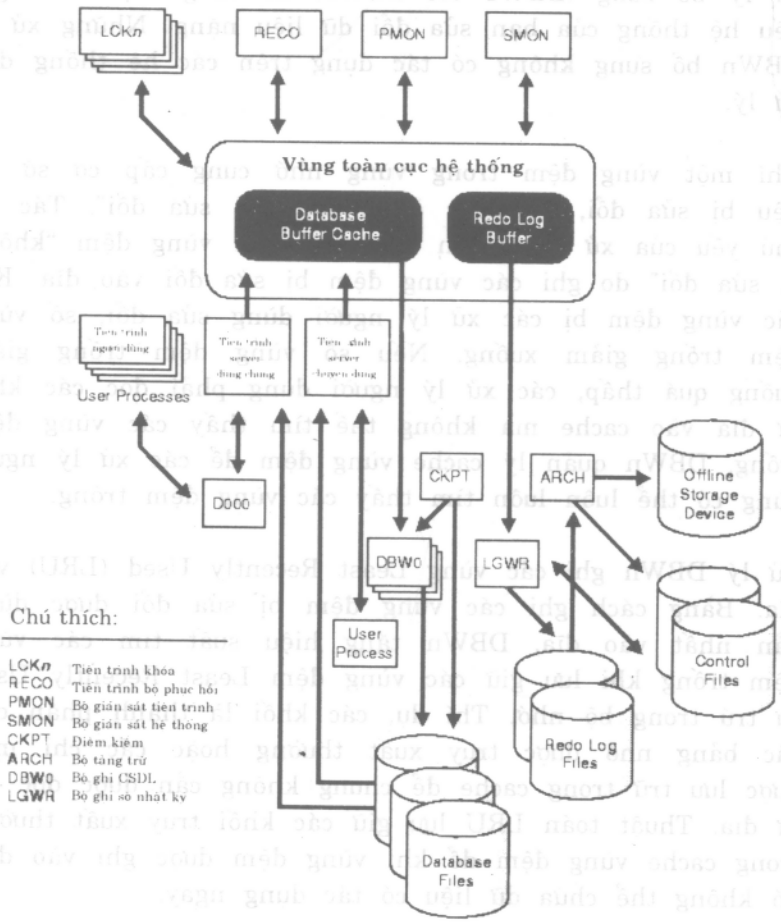
Một môi trường làm việc Oracle có thể có nhiều xử lý nền; không phải tất cả đều được biểu thị. Các xử lý nền trong một môi trường làm việc Oracle gồm:

Database Writer (DBW0 hoặc DBWn) - Trình ghi cơ sơ dữ liệu
Log Writer (LGWR) - Trình ghi nhật ký hoạt động
Checkpoint (CKPT) - Điểm kiểm
System Monitor (SMON) - Trình giám sát hệ thống
Process Monitor (PMON) - Trình giám sát xử lý
Archiver (ARCH) - Trình tàng trữ
Recoverer (RECO) - Trình phục hồi
Lock (LCKn) - Khóa
Job Queue (SNPn) - Hàng đợi tác vụ
Queue Monitor (QMNn) - Trình giám sát hàng đợi
Dispatcher (Dnnn) - Trình phân công
Server (Snnn) - Server

Hình 7-3 minh họa mỗi xử lý nền tương tác với các phần khác của cơ sở dữ liệu của Oracle như thế nào và phần còn lại của phần này mô tả mỗi xử lý.

Thông tin bổ sung :
Oracle Parallel Server không được minh họa trong Hình 7-3; xem Oracle8 Parallel Server Concepts và Adminstrator.

Hình 7-3 Các xử lý nền của môi trường làm việc Oracle đa xử lý



Trình ghi cơ sở dữ liệu - Database Writer (DBWn)

Trình ghi cơ sở dữ liệu (DBWn) ghi nội dụng của những vùng đệm vào các tập tin dữ liệu. Các xử lý DBWn chịu trách nhiệm ghi những vùng đệm đã bị sửa đổi trong cache vùng đệm vào đĩa. (Xem "Vùng nhớ cung cấp cơ sở

dữ liệu". Dù một trình ghi cơ sở dữ liệu (DBW0) thích hợp với hầu hết các hệ thống, bạn có thể cấu hình các xử lý bổ sung (DBW1 tới DBW9) để tăng hiệu suất ghi nếu hệ thống của bạn sửa đổi dữ liệu nặng. Những xử lý DBWn bổ sung không có tác dụng trên các hệ thống đơn xử lý.

Khí một vùng đệm trong vùng nhớ cung cấp cơ sở dữ liệu bị sửa đổi, nó được đánh dấu "bị sửa đổi". Tác vụ chú yếu của xử lý DBWn lưu trữ cache vùng đệm "không bị sửa đổi" do ghi các vùng đệm bị sửa đổi vào đĩa. Khi các vùng đệm bị các xử lý người dùng sửa đổi, số vùng đệm trống giảm xuống. Nếu số vùng đệm trống giảm xuống quá thấp, các xử lý người dùng phải đọc các khối từ đĩa vào cache mà không thể tìm thấy các vùng đệm trống. DBWn quản lý cache vùng đệm để các xử lý người dùng có thể luôn luôn tìm thấy các vùng đệm trống.

Xử lý DBWn ghi các vùng Least Recently Used (LRU) vào đĩa. Bằng cách ghi các vùng đệm bị sửa đổi được dùng gần nhất vào đĩa, DBWn tăng hiệu suất tìm các vùng đệm trống khi lưu giữ các vùng đệm Least Recently Used cư trú trong bộ nhớ. Thí dụ, các khối là thành phần của các bảng nhỏ được truy xuất thường hoặc các chỉ mục được lưu trữ trong cache để chung không cần được đọc lại từ đĩa. Thuật toán LRU lưu giữ các khối truy xuất thường trong cache vùng đệm để khi vùng đệm được ghi vào đĩa, nó không thể chứa dữ liệu có tác dụng ngay.

Tham số khởi hoạt DB_WRITER_PROCESSES xác định số xử lý DBWn. Nếu hệ thống của bạn dùng nhiều xử lý DBWn, bạn phải điều chỉnh giá trị của tham số DB_BLOCK_LRU_LATCHES để xử lý DBWn có cùng số mạch gài (danh sách vùng đệm LRU).

Thông tin bổ sung :
Xem Oracle8 Tuning về thiết bị xác lập
DB_WRITER_PROCESSES và
DB_BLOCK_LRU_LATCHES.

Xử lý DBWn ghi các vùng đệm bị sửa đổi vào đĩa dưới các điều kiện sau:
Khi một xử lý server không thể tìm được một vùng đệm chưa sửa đổi có thể dùng lại sau khi duyệt một số ngưỡng của các vùng đệm, nó phát tín hiệu DBWn để ghi. DBWn ghi các vùng đệm bị sửa đổi vào đĩa bằng một lần ghi dữ liệu đồng thời. (số khối được ghi trong lần ghi dữ liệu đồng thời khác nhau tùy theo hệ điều hành). Khi một điểm kiểm xảy ra, xử lý trình ghi nhật ký hoạt động (LGWR) phát triển tín hiệu DBWn và DBWn ghi tất cả những vùng đệm cần được ghi cho điểm kiểm để hoàn tất. Các điểm kiểm xảy ra vào mỗi lần chuyển đổi ghi sổ, khi LGWR ngưng ghi vào một tập tin nhật ký hoạt động trực tuyến và bắt đầu ghi vào một tập tin khác, vào những thời khoảng đều nhau được các tham số khởi hoạt LOG_CHECKPOINT_INTERVAL và LOG_CHECKPOINT_TIMEOUT.

DBWn định kỳ ghi những vùng đệm để từng vị trí trong xâu (nhật ký hoạt động) thực hiện lại từ sai sót hoặc phục hồi môi trường làm việc nào cần bắt đầu (điểm kiểm gia tăng). Vị trí ghi nhật ký hoạt động này được vùng đệm bị sửa đổi cũ nhất xác định trong cache vùng đệm. Điểm kiểm gia tăng xảy ra khi số vùng đệm bị sửa đổi trong cache lớn hơn ngưỡng được tham số khởi hoạt DB_BLOCK_MAX_DIRTY_TARGET ấn định. Xem "Điểm Kiểm Gia Tăng".

Trong mọi trường hợp, DBWn thực hiện các khối ghi có lô để tăng hiệu quả.

Trên một số nền, xử lý DBW0 có thể có nhiều xử lý server I/O. nếu một trong những xử lý này gặp trở ngại khi ghi vào một đĩa, thì các cách xử lý khác có thể tiếp tục ghi vào các đĩa khác. Những xử lý này không thể được dùng trên những hệ thống có nhiều xử lý DBWn. Tham số khởi hoạt DBWR_IO_SLAVES điều khiển số xử lý server I/O.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle xác định về đa xử lý server I/O trên nền của bạn.
Xem thêm Oracle8 Tuning về cách giám sát và điều hòa hiệu suất của một đơn xử lý DBW0 hoặc đa xử lý DBWn.

Xử lý trình ghi nhật ký hoạt động - Log Writer Process (LGWR)

Xử lý trình ghi nhật ký hoạt động (LGWR) chịu trách nhiệm quản lý vùng đệm ghi nhật ký Redo Log - ghi vùng đệm nhật ký Redo Log vào tập tin ghi nhật ký Redo Log trên đĩa (xem "Vùng Đệm Nhật ký Redo Log"). LGWR ghi tất cả những mục nhập thực hiện lại đã được sao chép vào vùng đệm từ lần nó ghi cuối cùng.

Vùng đệm nhật ký Redo Log là một vùng đệm luân chuyển. Khi LGWR ghi những mục nhập thực hiện lại từ vùng đệm nhật ký Redo Log vào tập tin nhật ký Redo Log, sau đó các xử lý server có thể chép các mục nhập mới trên các mục ấy trong vùng đệm nhật ký Redo Log được ghi vào đĩa. LGWR thường ghi đủ nhanh để bảo đảm không gian đó luôn luôn có thể sử dụng trong vùng đệm cho những mục nhập mới, ngay cả khi truy xuất nhật ký Redo Log nặng.

LGWR ghi một phần liên tục của vùng đệm vào đĩa. Các lần ghi LGWR:
Bản ghi cam kết khi xử lý người dùng cam kết một giao dịch
Các vùng đệm nhật ký Redo Log
Cứ ba giây mỗi lần
Khi vùng đệm nhật ký Redo Log đầy từ một tới một phần ba
Khi xử lý DBWn ghi các vùng đệm bị sửa đổi vào đĩa, nếu cần

Lưu ý:
Trước khi DBWn có thể ghi một vùng đệm bị sửa đổi, tất cả các bản ghi thực hiện lại với các thay đổi đối với vùng đệm phải được ghi vào đĩa (giao thức ghi trước). Nếu DBWn tìm được một số bản ghi thực hiện lại chưa được ghi, nó phát tín hiệu LGWR để ghi các bản ghi thực hiện lại vào đĩa và chờ LGWR hoàn tất vùng đệm nhật ký Redo Log trước khi nó có thể ghi ra những vùng đệm dữ liệu.

LGWR ghi đồng bộ vào nhóm phản chiếu hoạt động của các tập tin nhật ký Redo Log trực tuyến. Nếu một trong các tập tin trong nhóm bị hư hoặc không thể sử dụng, LGWR tiếp tục ghi vào những tập tin khác và ghi sổ một lỗi trong tập tin vệt tìm LGWR và trong tập tin ALERT hệ thống (xem Những Tập Tin Vệt Tìm và Tập Tin ALERT). Nếu tất cả các tập tin trong một nhóm bị hư hoặc nhóm ấy không thể sử dụng vì nó đã bị tăng trữ, thì LGWR không thể tiếp tục thực hiện chức năng.

Khi một người dùng đưa ra câu lệnh COMMIT, LGWR đưa bản ghi cam kết vào vùng đệm nhật ký Redo Log và ghi nó vào đĩa ngay, cùng với những mục nhập thực hiện lại

của giao dịch ấy. Những thay đổi tương ứng với những khối dữ liệu được trì hoãn cho tới khi nó có đủ hiệu lực ghi chung. Điều này được gọi là cơ chế "chuyển giao nhanh". Ghi nguyên tử của mục thực hiện lại chứa bản ghi cam kết của giao dịch là sự kiện đơn xác định giao dịch được cam kết. Oracle cho trở lại mã thành công đối với giao dịch cam kết, dù những vùng đệm dữ liệu chứa được ghi vào đĩa.

Lưu ý:
Đôi khi, nếu cần nhiều không gian vùng đệm, LGWR ghi những mục nhập nhật ký Redo Log trước khi giao dịch được bản ghi cam kết. Những mục nhập này trở thành thường xuyên chỉ nếu giao dịch sau này được bản ghi cam kết.

Khi một người dùng bản ghi cam kết một giao dịch, thì giao dịch ấy được gán một số hiệu thay đổi hệ thống (SCN), mà Oracle ghi song song với những mục nhập thực hiện lại của giao dịch đó. Các SCNs được ghi vào nhật ký Redo Log để những điều hoạt phục hồi có thể đồng bộ hóa trong những cấu hình Oracle Parallel Server và các cơ sở dữ liệu phân phối.

Thông tin bổ sung :
Xem Oracle8 Parallel Server Concepts và Administrator và Hướng dẫn của Oracle8 Administrator về những SCN và chúng được sử dụng như thế nào.

Vào những thời gian hoạt động cao, LGWR có thể ghi vào tập tin nhật ký Redo Log trực tuyến sử dụng các bản ghi cam kết nhóm. Thí dụ, giả sử là một người dùng bản ghi cam kết một giao dịch - LGWR phải ghi những mục nhập của giao dịch vào đĩa và khi điều này xảy ra, những người dùng khác đưa ra các câu lệnh COMMIT.

Tuy nhiên, LGWR không thể ghi vào tập tin nhật ký Redo Log trực tuyến để bản ghi cam kết những giao dịch này cho tới khi nó hoàn tất điều hoạt ghi trước đó. Sau những mục nhập của giao dịch đầu tiên chờ các giao dịch (chưa được bản ghi cam kết) có thể được ghi vào đĩa bằng một điều hoạt đòi hỏi ít I/O hơn nếu những mục nhập giao dịch được xử lý riêng. Do đó, Oracle giảm I/O đĩa tối đa và tăng hiệu suất của LGWR tối đa. Nếu các yêu cầu bản ghi cam kết tiếp tục ở mức độ cao, thì mỗi lần ghi (bởi LGWR) từ vùng đệm nhật ký hoạt động lại có thể chứa nhiều bản ghi cam kết.

Trên một số nền, xử lý LGWR có thể có nhiều xử lý server I/O. Nếu một trong những xử lý này bị trở ngại khi ghi vào một đĩa thì các xử lý khác có thể tiếp tục ghi vào các đĩa khác. Tham số khởi hoạt LGWR_IO_SLAVES điều khiển số xử lý server I/O.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle xác định về nhiều xử lý server I/O trên nền của bạn.

Xem thêm cách giám sát và điều hòa hiệu suất của LGWR, xem Oracle Tuning.

Xử lý điểm kiểm - Checkpoint Process (CKPT)

Khi một điểm kiểm xảy ra, Oracle phải cập nhật các phần đầu có thể tất cả tập tin dữ liệu để ghi chi tiết của điểm kiểm. Điều này được thực hiện bằng xử lý CKPT. Xử lý không ghi những khối dữ liệu vào đĩa; DBWn luôn luôn thực hiện công việc đó.

Những điểm kiểm DBWR thống kê được trình giám sát System Statistics hiển thị trong Oracle Enterprise Manager chỉ số yêu cầu điểm kiểm được hoàn tất.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle8 Administrator về những tác động thay đổi khoảng cách điểm kiểm.

Xem Oracle8 Parellel Server và Administrator về CKPT trong một Oracle Parallel Server.

Trình giám sát hệ thống - System Monitor (SMON)

Xử lý của trình giám sát hệ thống (SMON) thực hiện phục hồi môi trường làm việc vào lúc khởi động môi trường làm việc. SMON còn chịu trách nhiệm xóa các đoạn tạm thời không còn sử dụng để kết hợp những khối lưu riêng trống liên tiếp để tạo ra các khối dữ liệu lớn hơn của không gian trống có thể sử dụng. SMON "đánh thức" thường xuyên kiểm xem nó có cần thiết không. Các xử lý khác có thể gọi SMON nếu chúng phát hiện một nhu cầu để SMON đánh thức.

Trong môi trường Oracle Parallel Server, SMON còn thực hiện phục hồi môi trường làm việc đối với môi trường làm việc hoặc CPU bị sai sót.

Thông tin bổ sung :
Xem Oracle Parallel Server Concepts và Administrator về SMON trong một Oracle Parallel Server.

Trình giám sát xử lý - Process Monitor (PMON)

Trình giám sát xử lý (PMON) thực hiện phục hồi xử lý khi xử lý người dùng không thực hiện được. PMON chịu trách nhiệm xóa cache vùng đệm dữ liệu và giải phóng những tài nguyên mà xử lý người dùng đã sử dụng. Thí dụ, nó xác lập lại trạng thái của bảng giao dịch động,

giải phóng các khóa và xóa ID xử lý khỏi danh sách những xử lý hoạt động.

PMON còn định kỳ kiểm trạng thái của trình phân công và các xử lý server và khởi động lại bất kỳ xử lý nào đã chết (một không phải bất kỳ xử lý nào mà Oracle đã cố ý kết thúc).

Giống như SMON, PMON "đánh thức" đều đặn để kiểm xem nó có cần thiết không và có thể được gọi nếu một xử lý khác phát hiện yêu cầu đến nó.

Xử lý trình phục hồi - Recoverer Process (RECO)

Xử lý của trình phục hồi (RECO) là xử lý nền được dùng với cấu hình cơ sở dữ liệu được phân phối tự động giải quyết những sai sót liên quan đến những giao dịch được phân phối.

Xử lý RECO của một nút tự động kết nối với những cơ sở dữ liệu khác trong một giao dịch được phân phối không chắc chắn. Khi xử lý RECO tái thiết lập một kết nối giữa các server cơ sở dữ liệu liên quan, nó tự động giải quyết tất cả các giao dịch không chắc chắn, xóa khỏi bảng chờ giải quyết của mỗi cơ sở dữ liệu bất kỳ hàng nào tương ứng với những giao dịch không chắc chắn được giải quyết.

Nếu xử lý RECO không kết nối được với server ở xa, RECO tự động cố kết nối lại một thời khoảng đã định. Tuy nhiên, RECO chờ một thời lượng gia tăng (phát triển theo hàm mũ) trước khi nó cố thực hiện một kết nối khác.

Thông tin bổ sung :
Xem thêm thông tin về phục hồi giao dịch được phân phối, xem Oracle8 Distributed Database Systems.

Xử lý RECO hiện hữu chỉ nếu môi trường làm việc ấy cho phép những giao dịch được phân phối và nếu tham số DISTRIBUTED_TRANSACTIONS lớn hơn zêrô. Nếu tham số khởi hoạt này bằng zêrô, thì RECO không được tạo khi môi trường làm việc khởi động.

Xử lý của trình tàng trữ - Archiver Process (ARCH)

Xử lý của trình tàng trữ (ARCH) sao chép những tập tin nhật ký Redo Log trực tuyến vào thiết bị lưu trữ được chỉ định khi chúng đầy. ARCH hiện hữu chỉ khi nhật ký Redo Log được dùng ở chế độ ARCHIVELOG và tàng trữ tự động được cho phép.

Thông tin bổ sung :
Xem thêm thông tin về tàng trữ nhật ký Redo Log trực tuyến, xem "Redo Log" và Oracle8 Backup và Recovery Guide.

Xem tài liệu hệ điều hành Oracle xác định về chi tiết sử dụng xử lý ARCH.

Các xử lý khóa - Lock Processes (LCKn)

Với tùy chọn Parallel Server có tối đa 10 xử lý khóa (LCK0, . . ., LCK9) cung cấp khóa liên môi trường làm việc. Một xử lý LCK đơn (LCK0) đủ hiệu lực cho phần lớn những hệ thống Oracle Parallel Server systems.

Thông tin bổ sung :
Xem Oracle8 Parallel Server Concepts và Administrator về những thông tin xử lý nền này.

Các xử lý hàng đợi tác vụ - Job Queue Processes (SNPn)

Với cấu hình cơ sở dữ liệu được phân phối, tối đa 36 xử lý hàng đợi tác vụ (SNP0, ..., SNP9, SNPA, ..., SNPZ) có thể tự động tô bổ những ảnh chụp nhanh báng. Những xử lý này đánh thức định kỳ và tô bổ bất kỳ ảnh chụp nhanh được lập lịch trình để được tự động tô bổ. Nếu nhiều hàng đợi tác vụ được sử dụng, thì các xử lý dùng dung tác vụ của các ảnh chụp tô bổ.

Không giống như các xử lý nền Oracle khác, sai sót của một xử lý SNPn không gây chó môi trường làm việc thực hiện được. Nếu xử lý SNPn không thực hiện được, Oracle khởi động lại nó.

Những xử lý này cùng thực hiện các yêu cầu tác vụ được phần mềm DBMS_JOB tạo và truyền các thông điệp hàng đợi tới các hàng trên những cơ sở dữ liệu khác (xem "Oracle Advanced Queuing".

Thông tin bổ sung :
Xem Hướng dẫn của Oracle8 Administrator về thông tin xử lý nền và các hàng đợi tác vụ.

Các xử lý trình giám sát hàng đợi - Queue Monitor Processes (QMNn)

Xử lý trình giám sát hàng đợi là một xử lý nền tùy chọn cho Oracle Advanced Queuing (Oracle AQ) giám sát các hàng đợi thông điệp. Bạn có thể cấu hình tới 10 xử lý trình giám sát hàng đợi. Những xử lý này, giống như những xử lý SNPn khác với các xử lý nền Oracle khác trong đó sai sót xử lý không gây cho môi trường làm việc không thực hiện được.

Xem "Oracle Advanced Queuing" thông tin về những hàng đợi và xử lý trình giám sát hàng đợi.

Các xử lý trình phân công - Dispatcher Processes (Dnnn)

Các xử lý trình phân công hỗ trợ cấu hình đa xâu bằng cách cho phép các xử lý người dùng dùng chung một số giới hạn những xử lý server. (Xem Server Đa Xâu). Với server đa xâu, ít cần đến xử lý server dùng chung hơn cho cùng số người dùng; do đó, server đa xâu có thể hỗ trợ một số người dùng nhiều hơn, nhất là trong các môi trường client/server nơi ứng dụng client và server điều hoạt trên các máy khác nhau.

Bạn có thể tạo nhiều xử lý trình phân công; ít nhất một trình phân công phải được tạo cho mỗi giao thức mạng được dùng với Oracle. Người điều hành Oracle sẽ khởi động một số xử lý trình phân công tối ưu phụ thuộc vào sự giới hạn của hệ điều hành về số kết nối trên mỗi xử lý và có thể bổ sung hoặc xóa các xử lý trình phân công khi môi trường làm việc thực hiện.

> Lưu ý:
> Mỗi xử lý người dùng kết nối với một trình phân công phải thực hiện như vậy qua Net8 hoặc SQL*Net Version 2, dù cả hai xử lý đang chạy trên cùng một máy.

Trong cấu hình server đa xâu, xử lý Listener mạng chờ những yêu cầu kết nối từ các ứng dụng client và định tuyến mỗi ứng dụng với xử lý trình phân công. Nếu nó không thể kết nối với một ứng dụng khác với trình phân công, xử lý Listener khởi động một xử lý server chuyên dùng và kết nối ứng dụng client với server chuyên dùng. Xử lý Listener không phải là thành phần của môi trường

làm việc Oracle; đúng hơn, nó là thành phần của các xử lý mạng hoạt động với Oracle.

Thông tin bổ sung :
Xem "Server đa xâu" và Hướng dẫn của Oracle Net8 Administrator về người lắng nghe mạng.

Các xử lý server dùng chung - Shared Server Processes (Snnn)

Mỗi xử lý máy chủ dùng chung đáp ứng nhiều yêu cầu client trong cấu hình server đa xâu. Xem thêm "Các Xử Lý Server Dùng Chung".

Các tập tin vệt tìm và tập tin ALERT

Mỗi server và xử lý nền có thể ghi vào một tập tin vệt tìm được kết hợp. Khi một xử lý phát hiện một lỗi bên trong, nó kết suất thông tin về lỗi ấy vào tập tin vệt tìm. Nếu lỗi bên trong xảy ra và thông tin được ghi vào tập tin vệt tìm, người dùng điều hành sẽ liên lạc hỗ trợ Oracle.

Thông tin bổ sung :
Xem "Server Đa Xâu" và Hướng dẫn của Oracle Net8 Administrator về những thông điệp lỗi.

Tất cả những tên tập tin của các tập tin vệt tìm được kết hợp với xử lý nền chứa tên của xử lý ấy đã phát sinh tập tin vệt tìm. Một ngoại trừ là những tập tin vệt tìm được các xử lý hàng đợi tác vụ phát sinh (SNPn).

Thông tin bổ sung trong các tập tin vệt tìm có thể cung cấp hướng dẫn cho các ứng dụng hòa hợp hoặc môi trường làm việc. Các xử lý nền luôn luôn ghi những tiến trình này vào tập tin vệt tìm khi thích hợp. Tuy nhiên,

các xử lý server ghi thông tin hòa hợp vào tập tin vệt tìm nếu tham số khởi hoạt SQL_TRACE được xác lập là TRUE đối với môi trường làm việc hoặc phiên ấy. (thông tin về các lỗi bên trong luôn luôn được ghi vào các tập tin vệt tìm).

Mỗi phiên có thể cho phép hoặc vô hiệu hóa vệt tìm ghi nhật ký hoạt động thay cho xử lý server được kết hợp bằng cách sử dụng lệnh ALTER SESSION của SQL với tham số SQL_TRACE. Thí dụ, câu lệnh sau cho phép ghi vào tập tin vệt tìm cho phiên ấy:
ALTER SESSION SET SQL_TRACE = TRUE;

Mỗi cơ sở dữ liệu còn có một tập tin ALERT. Tập tin ALERT của cơ sở dữ liệu là một nhật ký hoạt động theo trình tự thời gian những tài nguyên và các lỗi, gồm
Tất cả các lỗi bên trong (ORA-600), các lỗi phân hủy khối dữ liệu (ORA-1578) và các lỗi bế tắc (ORA-60) xảy ra các điều hoạt quản lý, chẳng hạn như các câu lệnh CREATE/ALTER/DROP DATABASE/TABLESPACE/ROLLBACK SEGMENT của SQL và các câu lệnh STARTUP, SHUTDOWN, ARCHIVE LOG và RECOVER của Oracle Enterprise Manager hoặc Server Manager
Nhiều thông điệp và các lỗi liên quan đến những chức năng của server dùng chung và các xử lý trình phân công
Những lỗi khi tự động tô bố một hình chụp nhanh

Oracle sử dụng tập tin ALERT để lưu giữ bản ghi của những sự kiện này là một thay thế hiển thị thông tin trên thiết bị điều khiển của người điều hành. (nhiều hệ thống còn hiển thị thông tin này trên thiết bị điều khiển.) Nếu hoạt động điều hành thành công, thông điệp được ghi trong tập tin ALERT "hoàn tất" song song với tem thời gian.

Các thay đổi trong cấu hình Oracle

Trong môi trường làm việc người dùng Oracle đa xử lý, mã cho những người dùng được kết nối có thể được cấu hình bằng một trong ba cách sau:

Oracle Tác Vụ Đơn	Đối với mỗi người dùng, cả ứng dụng cơ sở dữ liệu và mã Oracle server được kết hợp trong một xử lý người dùng đơn.
Server Chuyên Dùng (Oracle Hai Tác Vụ)	Đối với mỗi người dùng, ứng dụng cơ sở dữ liệu được một xử lý khác tiến hành (một xử lý người dùng) so với xử lý thực hiện mã Oracle server (một xử lý server chuyển dùng).
Server Đa Xâu	Ứng dụng cơ sở dữ liệu được một xử lý khác tiến hành (một xử lý người dùng) so với xử lý thực hiện mã Oracle server; mỗi xử lý server thực hiện mã Oracle server (một xử lý server dùng chung) có thể đáp ứng nhiều xử lý người dùng.

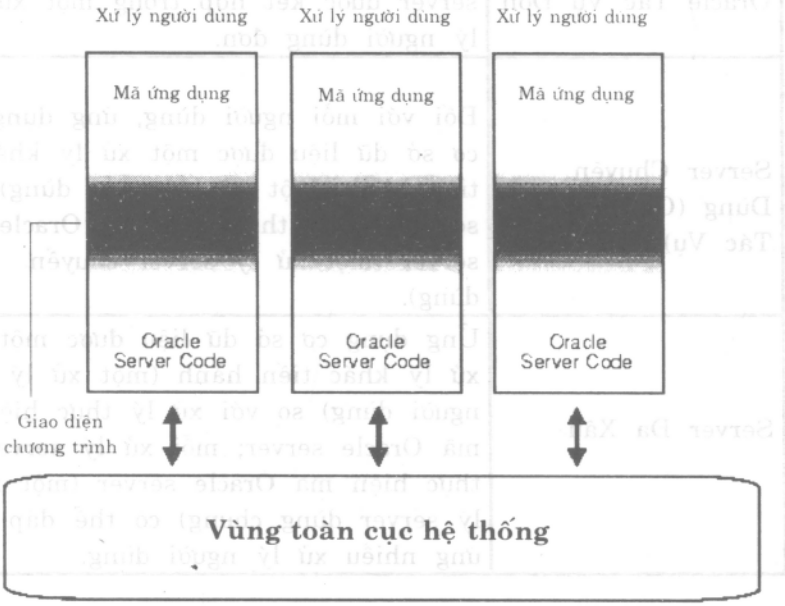
Các phần sâu mô tả mỗi sự khác nhau chi tiết hơn.

Thông tin bổ sung :
Một số hệ điều hành cung ứng sự chọn lựa cấu hình; xem tài liệu hệ điều hành Oracle xác định chi tiết về các tùy chọn.

Cấu hình tác vụ đơn

Hình 7-4 minh họa cấu hình Oracle tác vụ đơn, ứng dụng cơ sở dữ liệu và mã Oracle server tất cả đều chạy trong cùng xử lý, được gọi là xử lý người dùng.

Hình 7-4 Oracle sử dụng các xử lý kết hợp người dùng/server



Cấu hình của Oracle này chỉ khả thi trong các hệ điều hành có thể duy trì sự tách riêng giữa ứng dụng cơ sở dữ liệu và mã Oracle bằng một xử lý đơn (chẳng hạn như VAX VMS). Sự tách riêng này cần thiết cho tính nguyên vẹn và tính riêng tư của dữ liệu. Một số hệ điều hành, chẳng hạn như UNIX, không thể cung cấp sự tách riêng này và do đó phải dùng các xử lý tách riêng chạy mã ứng dụng và mã server.

Lưu ý:
Giao diện chương trình chịu trách nhiệm về sự tách riêng và bảo vệ mã Oracle server và chịu trách nhiệm chuyển dữ liệu giữa ứng dụng cơ sở dữ liệu và chương trình người dùng Oracle. Xem "Giao Diện Chương Trình".

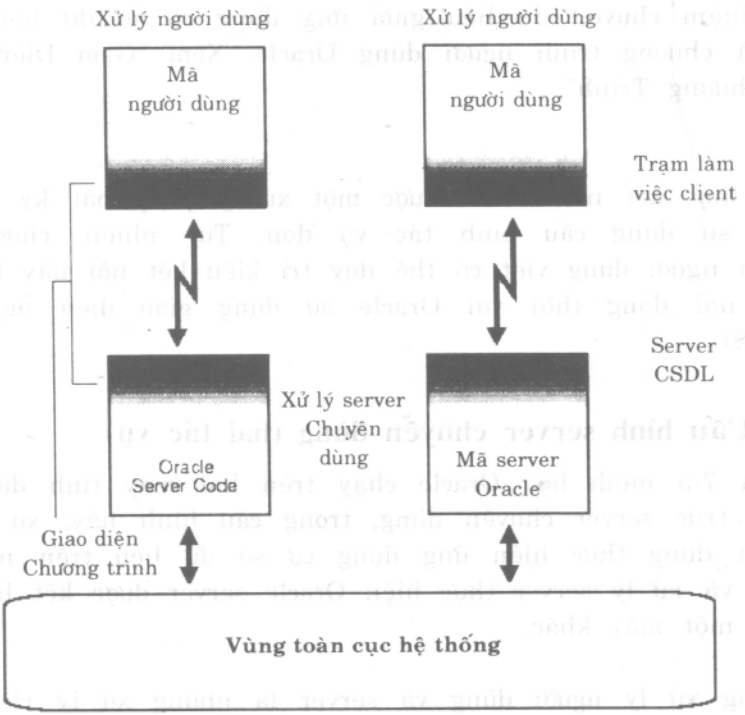
Chỉ một kết nối Oracle được một xử lý phép bất kỳ lúc nào sử dụng cấu hình tác vụ đơn. Tuy nhiên, chương trình người dùng viết có thể duy trì kiểu kết nối này khi kết nối đồng thời với Oracle sử dụng giao diện mạng (Net8).

Cấu hình server chuyển dùng (hai tác vụ)

Hình 7-5 minh họa Oracle chạy trên hai máy tính dùng kiến trúc server chuyển dùng, trong cấu hình này, xử lý người dùng thực hiện ứng dụng cơ sở dữ liệu trên một máy và xử lý server thực hiện Oracle server được kết hợp trên một máy khác.

Những xử lý người dùng và server là những xử lý riêng biệt, tách rời. Xử lý server tách riêng được tạo thay cho mỗi xử lý người dùng được gọi là xử lý server chuyên dùng (hoặc xử lý bóng) vì xử lý server này chỉ đóng vai trò thay thế cho xử lý người dùng được kết hợp.
Cấu hình này duy trì tỷ lệ một - một giữa số xử lý người dùng và server. Ngay cả khi người dùng không tạo một yêu cầu cơ sở dữ liệu hoạt động, xử lý server chuyên dùng vẫn duy trì (dù nó không hoạt động và có thể phân hết trang trên một số hệ điều hành).

Hình 7-5 Oracle sử dụng các xử lý server chuyên dùng



Hình 7-5 trình bày những xử lý người dùng và server chạy trên các máy tính được kết nối trên mạng. Tuy nhiên, kiến trúc server chuyên dùng còn được sử dụng nếu cùng máy tính ấy thực hiện cả ứng dụng client và mã Oracle server nhưng hệ điều hành server không thể duy trì sự tách riêng hai chương trình nếu chúng chạy trong một xử lý đơn. (UNIX là một thí dụ về hệ điều hành như thế.)

Trong cấu hình server chuyên dùng, những xử lý người dùng và server truyền thông bằng cách sử dụng các cơ chế khác nhau:
Nếu hệ thống ấy được cấu hình để xử lý người dùng và xử lý server chuyển dùng chạy trên cùng máy tính, thì giao dịch chương trình sử dụng cơ chế truyền thông của hệ điều hành server để thực hiện tác vụ của nó.
Nếu xử lý người dùng và xử lý server chuyên dùng chạy trên các máy tính khác nhau, thì giao diện chương trình cung cấp các cơ chế truyền thông (chẳng hạn như phần mềm mạng và Net8) giữa các chương trình.

Thông tin bổ sung :

Các liên kết truyền thông này là đòi hỏi và cài đặt phụ thuộc; xem tài liệu hệ điều hành Oracle xác định và tài liệu Net8.

Kiến trúc server chuyên dùng đôi khi có thể dẫn đến không hiệu quả. Xét một hệ thống mục nhập đơn đặt hàng với các xử lý server chuyển dùng. Khách hàng đặt một đơn đặt hàng khi một nhân viên kế toán nhập đơn đặt hàng đó vào cơ sở dữ liệu. Đối với hầu hết giao dịch ấy, nhân viên đó thảo luận với khách hàng khi xử lý server chuyên dùng với xử lý người dùng của nhân viên kế toán ấy vẫn chưa hoạt động. Xử lý server ấy không cần đến trong phần lớn giao dịch đó và hệ thống chậm hơn khi các nhân viên kế toán nhập các đơn đặt hàng. Đối với các ứng dụng như trước, thì kiến trúc server đa xâu có thể thích hợp hơn.

Xem "Thí Dụ Về Oracle Sử Dụng Các Xử Lý Server Chuyên Dùng" về minh họa chi tiết các server chuyển dùng.

Server đa xâu

Cấu hình server đa xâu cho phép nhiều xử lý người dùng dùng chung rất ít xử lý server. Các xử lý người dùng kết nối với xử lý nền của trình phân công để định tuyến các yêu cầu của client đối với xử lý server dùng chung có thể sử dụng kết tiếp.

Lợi điểm của cấu hình server đa xâu là phần điều hành chung hệ thống đó giảm xuống làm tăng số người dùng có thể được hỗ trợ. Một số ít các xử lý server dùng chung có thể thực hiện cùng số lượng xử lý bằng với các xử lý server chuyển dùng và dùng lượng bộ nhớ cần thiết cho mỗi người dùng tương đối nhỏ.

Một số các xử lý khác nhau cần thiết trong hệ thống server đa xâu:
Xử lý Listener mạng kết nối các xử lý người dùng với các trình phân công hoặc các server chuyên dùng (xử lý Listener là thành phần của Net8, không phải của Oracle).
Một hoặc nhiều xử lý trình phân công
Một hoặc nhiều xử lý server dùng chung

Server đa xâu đòi hỏi Net8 hoặc SQL*Net Version 2.

Lưu ý:
Để sử dụng các server dùng chung, xử lý người dùng phải kết nối bằng Net8 hoặc SQL*Net Version 2, dù xử lý ấy chạy trên cùng máy như môi trường làm việc Oracle.

Khi một môi trường làm việc khởi động, xử lý của Listener mạng mở và thiết lập một đường dẫn truyền thông mà các người dùng kết nối với Oracle. Sau đó, mỗi xử lý của trình phân công tạo xử lý Listener và địa chỉ mà trình phân công nghe các yêu cầu kết nối. Ít nhất một xử lý trình phân công phải được cấu hình và khởi

động cho mỗi giao thức mạng mà các client của cơ sở dữ liệu sẽ sử dụng.

Khi một xử lý người dùng đưa ra một yêu cầu kết nối, Listener khảo sát yêu cầu ấy và xác định xem xử lý người dùng có thể dùng một xử lý server dùng chung không. Nếu có, Listener cho trở lại địa chỉ của xử lý trình phân công có mức tải thấp nhất và xử lý người dùng kết nối với trình phân công ấy trực tiếp.

Một số xử lý người dùng không thể truyền thông với trình phân công (chẳng hạn như các xử lý kết nối sử dụng phiên bản trước 2 SQL*Net) vì vậy xử lý Listener mạng không thể kết nối chúng với một trình phân công. Trong trường hợp này, hoặc nếu xử lý người dùng yêu cầu một server chuyển dùng (xem "Các Điều Hoạt Hạn Định Của Server Đa Xâu"), Listener tạo một server chuyển dùng và thiết lập kết nối thích hợp.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle Net8 Administrator's về Listener mạng.

Yêu cầu của trình phân công và các hàng đợi trả lời

Một yêu cầu từ một người dùng là một lệnh gọi giao diện chương trình đơn là thành phần của câu lệnh SQL của người dùng ấy. Khi người dùng đưa ra một lệnh gọi, trình phân công của nó đưa yêu cầu ấy vào hàng đợi yêu cầu, nơi nó được một xử lý server dùng chung có thể sử dụng kế tiếp lựa.

Hàng đợi yêu cầu trong SGA và phổ biến đối với tất cả các xử lý trình phân công của một môi trường làm việc. Các xử lý server dùng chung kiểm hàng đợi yêu cầu

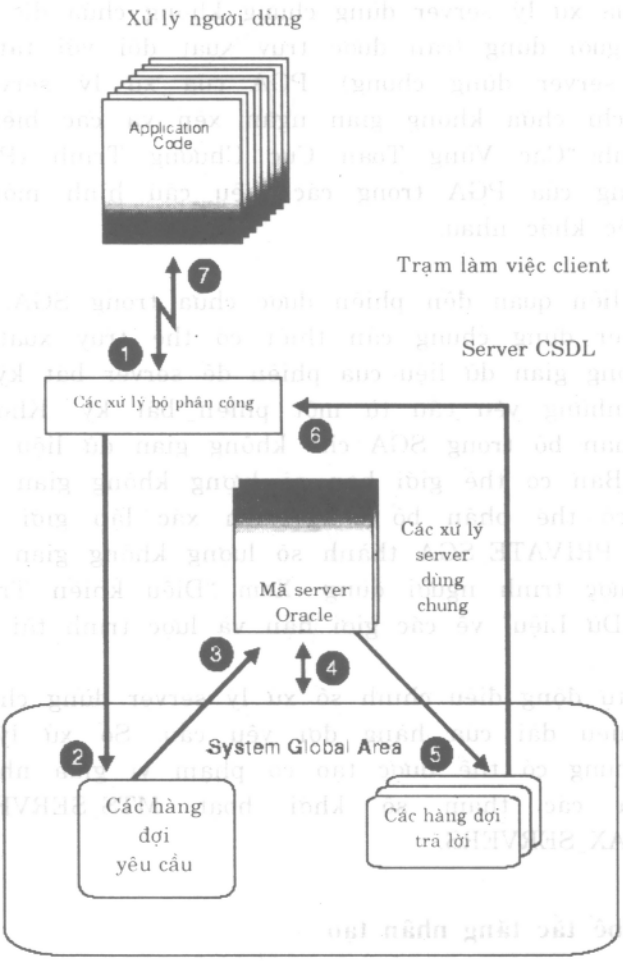
chung về các yêu cầu mới, lựa các yêu cầu mới theo phương thức vào trước-ra trước. Một xử lý server dùng chung lựa một yêu cầu trong hàng đợi và đưa ra tất cả các lệnh gọi cần thiết đối với cơ sở dữ liệu để hoàn tất yêu cầu đó.

Khi server ấy hoàn tất yêu cầu, nó chuyển trả lời cho hàng đợi yêu cầu của trình phân công. Mỗi trình phân công có hàng đợi trả lời riêng trong SGA. Sau đó trình phân công đưa yêu cầu hoàn tất trở lại xử lý người dùng thích hợp.

Thí dụ, trong hệ thống nhập đơn đặt hàng mà mỗi xử lý người dùng của nhân viên kế toán kết nối với trình phân công và mỗi yêu cầu được nhân viên kế toán đưa ra được gửi tới trình phân công đó, để đưa yêu cầu ấy vào hàng đợi yêu cầu. Xử lý server dùng chung có thể sử dụng kế tiếp lựa yêu cầu ấy, phục vụ nó và đưa trả lời vào hàng đợi trả lời. Khi yêu cầu của nhân viên kế toán được hoàn tất, thì nhân viên kế toán vẫn còn kết nối với trình phân công nhưng xử lý server phục vụ đã xử lý yêu cầu ấy được giải phóng và có thể sử dụng cho các yêu cầu khác. Khi một nhân viên kế toán nói chuyện với khách hàng, một nhân viên kế toán khác có thể sử dụng cùng xử lý server dùng chung.

Hình 7-6 minh họa các xử lý người dùng truyền thông với trình phân công qua giao diện chương trình như thế nào và trình phân công ấy truyền thông với các yêu cầu của người dùng đối với các xử lý server dùng chung như thế nào.

Hình 7-6 Cấu hình server đa xâu và các xử lý server dùng chung



Các xử lý server dùng chung

Các xử lý server dùng chung và các xử lý server chuyên dùng cung cấp cùng tính năng, ngoại trừ các xử lý server dùng chung không được kết hợp với xử lý người dùng

riêng. Thay vì, xử lý server dùng chung đáp ứng bất kỳ yêu cầu client nào trong cấu hình server đa xau.

PGA của xử lý server dùng chung không chứa dữ liệu liên quan người dùng (cần được truy xuất đối với tất cả các xử lý server dùng chung). PGA của xử lý server dùng chung chỉ chứa không gian ngăn xếp và các biến xử lý xác định. "Các Vùng Toan Cục Chương Trình (PGA)" về nội dung của PGA trong các kiểu cấu hình môi trường làm việc khác nhau.

Tất cả liên quan đến phiên được chứa trong SGA. Mỗi xử lý server dùng chung cần thiết có thể truy xuất tất cả các không gian dữ liệu của phiên để server bất kỳ có thể xử lý những yêu cầu từ một phiên bất kỳ. Không gian được phân bổ trong SGA cho không gian dữ liệu của mỗi phiên. Bạn có thể giới hạn số lượng không gian mà một phiên có thể phân bổ bằng cách xác lập giới hạn tài nguyên PRIVATE_SGA thành số lượng không gian yêu cầu trong lược trình người dùng. Xem "Điều khiển Truy Xuất Cơ Sở Dữ Liệu" về các giới hạn và lược trình tài nguyên.

Oracle tự động điều chỉnh số xử lý server dùng chung dựa trên chiều dài của hàng đợi yêu cầu. Số xử lý server dùng chung có thể được tạo có phạm vi giữa những giá trị của các tham số khởi hoạt MTS_SERVERS và MTS_MAX_SERVERS.

Các bế tắc tăng nhân tạo

Với số xử lý server dùng chung giới hạn, khả năng của một bế tắc "nhân tạo" có thể tăng. Một bế tắc nhân tạo có thể xảy ra trong trường hợp sau:

Một người dùng đòi hỏi một khóa độc quyền trên tài nguyên bằng cách đưa câu lệnh SELECT với mệnh đề FOR UPDATE hoặc câu lệnh LOCK TABLE.

Xử lý server dùng chung xử lý yêu cầu khóa được giải phóng khi câu lệnh ấy hoàn tất.

Những người dùng khác cố truy xuất tài nguyên được khóa. Mỗi xử lý server dùng chung giới hạn xử lý người dùng mà nó đang phục vụ cho tới khi tài nguyên cần thiết bị khóa trở thành sử dụng được. Cuối cùng tất cả server dùng chung có thể bị giới hạn vào những xử lý người dùng chờ những tài nguyên được khóa.

Người dùng ban đầu cố đệ trình một yêu cầu mới (chẳng hạn như câu lệnh COMMIT hoặc ROLLBACK) để giải phóng khóa giành được trước đó, nhưng không thể vì tất cả các xử lý máy chủ hiện đang được sử dụng cùng một lúc.

Khi Oracle phát hiện một bế tắc nhân tạo, thì các xử lý máy chủ dùng chung mới tự động được tạo theo nhu cầu cho tới khi người dùng ban đầu đệ trình một yêu cầu giải phóng các tài nguyên bị khóa gây ra các bế tắc nhân tạo. Nếu số xử lý dùng chung tối đa (tham số MTS_MAX SERVERS ấn định) đã được khởi động, thì người điều hành cơ sở dữ liệu phải giải quyết thủ công bằng cách tháo kết nối người dùng. Điều này giải phóng xử lý server dùng chung, để giải quyết bế tắc nhân tạo.

Nếu các bế tắc nhân tạo xảy ra quá thường trên hệ thống của bạn, bạn nên tăng giá trị của MTS_MAX SERVERS.

Những điều hoạt hạn định của server đa xâu

Các hoạt động quản lý nào đó không thể được thực hiện khi được kết nối với một xử lý trình phân công, gồm đóng hoặc khởi động một môi trường làm việc và phục hồi môi trường. Một thông báo lỗi được đưa ra nếu bạn cố thực hiện những hoạt động này dù được kết nối với xử lý trình phân công.

Những hoạt động này thường được thực hiện khi được kết nối với các đặc quyền người điều hành. Khi bạn muốn kết nối với các đặc quyền người điều hành trong một cấu hình hệ thống với các server đa xâu, bạn phải khai báo chuỗi kết nối của mình muốn dùng một xử lý server chuyên dùng (SRVR=DEDICATED) thay vì xử lý trình phân công.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle xác định hoặc Hướng dẫn của Oracle Net8 Administrator về cú pháp chuỗi kết nối thích hợp.

Xem "Thí Dụ Về Oracle Sử Dụng Server Đa Xâu" về minh họa chi tiết cách sử dụng cấu hình server đa xâu.

Các thí dụ về Oracle hoạt động như thế nào

Nay các cấu trúc bộ nhớ, các xử lý và các cấu hình khác nhau của hệ thống cơ sở dữ liệu Oracle được đề cập, cần hiểu rõ tất cả các thành phần hoạt động cùng với nhau. Các phần sau giải thích rõ về các cấu hình Oracle server hai tác vụ và đa xâu.

Thí dụ về Oracle sử dụng các xử lý server chuyên dùng

Thí dụ sau là một minh họa đơn giản về kiến trúc server chuyên dùng. Các bước sau chỉ trình bày mức độ căn bản nhất về các điều hoạt mà Oracle thực hiện.

Một server cơ sở dữ liệu hiện đang chạy Oracle sử dụng nhiều xử lý nền.

Một xử lý người dùng trên một trạm làm việc client chạy ứng dụng cơ sở dữ liệu, chẳng hạn như SQL*Plus. Ứng dụng client cố thiết lập một kết nối với server sử dụng trình điều khiển Net8.

Server cơ sở dữ liệu hiện đang chạy trình điều khiển Net8 thích hợp. Xử lý Listener mạng trên server cơ sở dữ liệu phát hiện yêu cầu kết nối từ ứng dụng cơ sở dữ liệu client và tạo một xử lý server chuyên dùng trên server cơ sở dữ liệu ấy thay cho xử lý người dùng.

Người dùng thực hiện câu lệnh SQL đơn. Thí dụ, người dùng chèn một hàng vào bảng.

Xử lý server chuyên dùng nhận câu lệnh. Ở điểm này, hai đường dẫn có thể tiếp theo sau để tiếp tục xử lý câu lệnh SQL:

- Nếu vùng chia dùng chung chứa một vùng SQL dùng chung chỏ một câu lệnh SQL đồng nhất, thì xử lý server sử dụng vùng SQL dùng chung có sẵn để tiến hành câu lệnh SQL của client.
- Nếu vùng chia dùng chung không chứa vùng SQL dùng chung cho câu lệnh đồng nhất, thì một vùng SQL dùng chung mới được phân bổ cho câu lệnh ấy trong vùng chia dùng chung.

Cả hai trường hợp, vùng SQL riêng được tạo trong PGA của phiên và xử lý server chuyên dùng kiểm các

đặc quyền truy xuất của người dùng đối với dữ liệu được yêu cầu.

Xử lý server truy hồi các khối dữ liệu từ tập tin dữ liệu thực, nếu cần hoặc sử dụng những khối dữ liệu đã được lưu trữ trong cache vùng đệm trong SGA của môi trường làm việc ấy.

Xử lý server thực hiện câu lệnh SQL được lưu trữ trong vùng SQL dùng chung. Dữ liệu đầu tiên bị thay đổi trong SGA. Nó được ghi vĩnh viễn vào đĩa khi xử lý DBW0 xác định điều đó được thực hiện hiệu quả nhất. Xử lý LGWR ghi giao dịch trong tập tin nhật ký Redo Log trực tuyến trên yêu cầu cam kết kế tiếp từ người dùng ấy.

Nếu yêu cầu ấy thành công, thì server gửi một thông điệp qua mạng tới người dùng đó. Nếu không thành công, một thông báo lỗi thích hợp được cam kết.

Trong toàn bộ thủ tục, các xử lý nền khác đang chạy và theo dõi bất kỳ điều kiện nào đòi hỏi sự can thiệp. Hơn nữa, Oracle quản lý các giao dịch khác và tránh để tranh chấp giữa các giao dịch khác nhau yêu cầu cùng dữ liệu.

Thí dụ về Oracle sử dụng server đa xâu

Thí dụ sau là một minh họa đơn giản về kiến trúc đa xâu:

Một server cơ sở dữ liệu hiện đang chạy cấu hình server đa xâu.

Một xử lý người dùng trên trạm làm việc client chạy ứng dụng cơ sở dữ liệu, chẳng hạn như SQL*Forms. Ứng dụng khác cô thiết lập một kết nối với server cơ sở dữ liệu sử dụng trình điều khiển Net8 thích hợp.

Server cơ sở dữ liệu hiện đang chạy trình điều khiển Net8 thích hợp. Xử lý Listener mạng trên server cơ sở

dữ liệu phát hiện yêu cầu kết nối của xử lý người và xác định xử lý người dùng sẽ được kết nối như thế nào. Nếu người dùng đang sử dụng Net8 or SQL*Net Version 2, thì Listener thông báo xử lý người dùng để kết nối lại sử dụng địa chỉ của một xử lý trình phân công có thể sử dụng.

Lưu ý:
Nếu xử lý người dùng kết nối với SQL*Net Version 1 hoặc 1.1, thì người dùng nghe SQL*Net tạo một xử lý server chuyên dùng thay cho xử lý người dùng và phần còn lại của thí dụ điều hoạt như được mô tả ở thí dụ trên. (Các xử lý người dùng phải kết nối với Net8 hoặc SQL*Net Version 2 để sử dụng xử lý server dùng chung).

Người dùng đưa ra câu lệnh SQL. Thí dụ, người dùng cập nhật một hàng vào bảng.
Xử lý trình phân công đưa yêu cầu của xử lý người dùng vào hàng đợi yêu cầu trong SGA và được tất cả các xử lý trình phân công dùng chung.
Một xử lý máy chủ dùng chung có thể sử dụng kiểm hàng đợi yêu cầu của trình phân công và lựa câu lệnh SQL kế tiếp trên hàng đợi. Sau đó nó xử lý các xử lý câu lệnh SQL như được mô tả trong bước 5, 6 và 7 của thí dụ trước đó. (trong bước 5, các thành phần của vùng SQL riêng được tạo trong SGA.)
Một khi xử lý server dùng chung kết thúc xử lý câu lệnh SQL, thì xử lý ấy đưa kết quả vào hàng đợi trả lời của xử lý trình phân công đã gửi yêu cầu ấy.
Xử lý trình phân công kiểm tra hàng đợi trả lời của và gửi các yêu cầu hoàn tất trở lại xử lý người dùng đã đưa ra yêu cầu ấy.

Giao diện chương trình

Giao diện chương trình là tầng phần mềm giữa ứng dụng cơ sở dữ liệu và Oracle. Giao diện chương trình:
Cung cấp hàng rào bảo mật để tránh bị truy xuất phát hoại SGA của các xử lý người dùng khác
Đóng vai trò là một cơ chế truyền thông, định dạng các yêu cầu thông tin, chuyển dữ liệu và các lỗi bẫy và cho trở lại
Chuyển đổi và phiên dịch dữ liệu, nhất là giữa các kiểu máy tính khác nhau hoặc với các kiểu dữ liệu chương trình người dùng bên ngoài

Mã Oracle đóng vai trò là một server thực hiện các tác vụ cơ sở dữ liệu thay cho một ứng dụng (một client), chẳng hạn như truy nạp các hàng từ các khối dữ liệu. Nó gồm nhiều thành phần được cung cấp bằng phần mềm Oracle và phần mềm hệ điều hành xác định.

Cấu trúc giao diện chương trình

Giao diện chương trình của các phần sau:
Giao diện gọi Oracle (OCI) hoặc thư viện Oracle thời gian thực hiện (SQLLIB)
Bên client hoặc người dùng của giao diện chương trình (còn được gọi là UPI)
Các trình điều khiển Net8 khác nhau (phần mềm các truyền thông xác định)
Phần mềm các truyền thông hệ điều hành
Server hoặc bên Oracle của giao diện chương trình (còn được gọi là OPI)

Cả bên người dùng và Oracle của giao diện chương trình thực hiện phần mềm Oracle như thực hiện các trình điều khiển.

Net8 là thành phần của giao diện chương trình cho phép trình ứng dụng client và Oracle server cư trú các máy tính tách riêng trong mạng truyền thông của bạn.

Các trình giao diện điều khiển

Các trình điều khiển là các thành phần của phần mềm chuyển tải dữ liệu, thường qua một mạng. Chúng thực hiện các điều hoạt giống kết nối, giải kết nối, phát tín hiệu lỗi và kiểm tra lỗi. Các trình điều khiển được xác định đối với một giao thức truyền thông. Luôn luôn có trình điều khiển mặc định.

Bạn có thể cài đặt nhiều trình điều khiển (chẳng hạn như các trình điều khiểu không đồng bộ hoặc DECnet) và lựa một trình điều khiển làm trình điều khiển mặc định, nhưng cho phép một người dùng riêng biệt sử dụng các trình điều khiển khác bằng cách xác định trình điều khiển được yêu cầu vào thời gian kết nối. Các xử lý khác có thể sử dụng những trình điều khiển khác nhau. Một xử lý đơn có thể có các kết nối đồng thời hoặc nhiều cơ sở dữ liệu (cục bộ hoặc từ xa) sử dụng các trình điều khiển Net8.

Hướng dẫn và cài đặt, tài liệu Net8 cho hệ thống của bạn chi tiết về chọn lựa và cài đặt các trình điều khiển, bổ sung các trình điều khiển mới sau khi cài đặt. Tài liệu Net8 mô tả để lựa một trình điều khiển vào thời gian thực hiện khi truy xuất Oracle.

Thông tin bổ sung :
Xem Hướng dẫn của Oracle Net8 Administrator về Net8.

Phần mềm truyền thông hệ điều hành

Phần mềm mức thấp nhất kết nối bên người dùng với bên Oracle của giao diện người dùng là phần mềm truyền thông được hệ điều hành server cung cấp. Thí dụ như DECnet, TCP/IP, LU6.2 và ASYNC.

Thông tin bổ sung :
Phần mềm truyền thông có thể được công Oracle cung cấp nhưng thường phải mua riêng của nhà kinh doanh phần mềm hoặc nhà cung cấp phần mềm bên thứ ba. Xem tài liệu hệ điều hành Oracle xác định về phần cứng truyền thông hệ thống của bạn.

Phần IV

DBMS QUAN HỆ ĐỐI TƯỢNG

Phần IV mô tả mô hình quản lý Oracle về quản lý cơ sở dữ liệu và các mở rộng đối tượng đối với mô hình đó.

Phần IV gồm các chương sau:

Chương 8 "Các Đối Tượng Giản Đồ"
Chương 9 "Các Bảng Phân Hoạch Và Các Chỉ Mục"
Chương 10 "Các Tập Tin Dữ Liệu Cài Sẵn"
Chương 11 "Các Kiểu Dữ Liệu Người Dùng Xác Định (Tùy Chọn Đối Tượng)"
Chương 12 "Sử Dụng Các Kiểu Dữ Liệu Người Dùng Xác Định"
Chương 13 "Các Khung Nhìn Đối Tượng"

Chương 8

Các đối tượng giản đồ

Chương này đề cập các kiểu đối tượng cơ sở dữ liệu khác nhau trong giản đồ người dùng. Nó gồm:

Tổng quan về các đối tượng giản đồ
Các bảng
Các khung nhìn
Trình phát sinh chuỗi
Các từ đồng nghĩa
Các chỉ mục
Các bảng chỉ tổ chức
Các cụm liên kết
Các cụm liên kết trị số

Thông tin bổ sung :
Nếu bạn đang dùng Trusted Oracle, xem tài liệu Trusted Oracle về các đối tượng giản đồ trong môi trường đó.

Tổng quan về các đối tượng giản đồ

Được kết hợp với mỗi người dùng cơ sở dữ liệu là một giản đồ. Một giản đồ là tập hợp các đối tượng giản đồ.

Các thí dụ về những đối tượng giản đồ gồm các bảng, các khung nhìn, các chuỗi, các từ đồng nghĩa, các chỉ mục, các cụm liên kết, các liên kết cơ sở dữ liệu, các ảnh chụp nhan, các thủ tục, các hàm và các bộ phần mềm.
Các đối tượng giản đồ là các cấu trúc lưu trữ dữ liệu. Các đối tượng giản đồ không có quan hệ tương ứng một-một với các tập tin vật lý trên đĩa lưu trữ thông tin của chung. Tuy nhiên, Oracle lưu trữ một đối tượng giản đồ logic trong không gian bảng của cơ sở dữ liệu. Dữ liệu của mỗi đối tượng được chứa vật lý trong một hoặc nhiều tập tin dữ liệu của không gian bảng. Đối với một số đối tượng, chẳng hạn như các bảng, các chỉ mục và các cụm liên kết, mà bạn có thể ấn định không gian đĩa Oracle phân bổ bao nhiêu cho các tập tin dữ liệu của không gian bảng đó.

Không có mối quan hệ giữa các giản đồ và các không gian bảng: một không gian bảng có thể chứa các đối tượng của các giản đồ khác và các đối tượng cho một giản đồ có thể được chứa trong các không gian bảng khác nhau. Hình 8-1 minh họa mối quan hệ giữa các đối tượng, các không gian bảng và các tập tin dữ liệu.

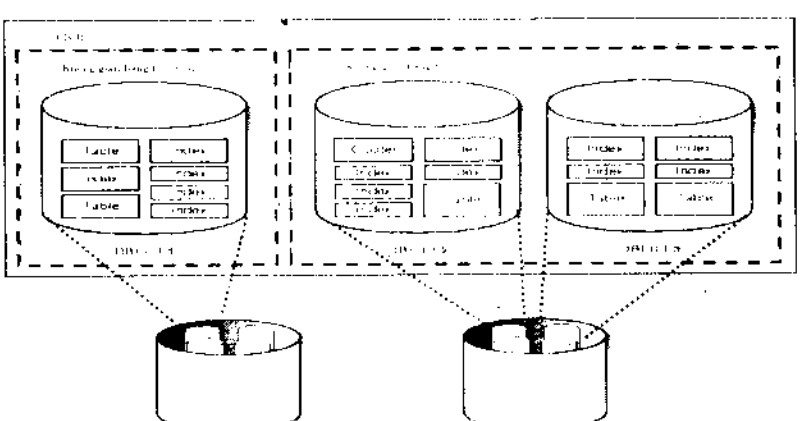
Thông tin bổ sung :
Chương này giải thích các bảng, các khung nhìn, các chuỗi, các từ đồng nghĩa, các chỉ mục và các cụm liên kết. Các loại đối tượng giản đồ khác được giải thích chỗ khác trong tài liệu này hoặc trong các tài liệu khác. Nhất là:
Các liên kết cơ sở dữ liệu được mô tả ở "Các Cơ Sở Dữ Liệu Được Phân Phối".
Các khung nhìn đối tượng được mô tả ở "Các Khung Nhìn Đối Tượng" và ở "Hướng Dẫn Của Developer Ứng Dụng Oracle8".
Các thủ tục, các hàm và các bộ phần mềm được mô tả

ở "Các Thủ Tục Và Bộ Phần Mềm".
Các bộ kích khởi được mô tả ở "Các Bộ kích khởi Cơ Sở Dữ Liệu".
Các ảnh chụp nhanh được mô tả ở "Sao chép của Oracle8".

Các bảng

Các bảng là đơn vị cơ bản của bộ nhớ dữ liệu trong cơ sở dữ liệu của Oracle. Dữ liệu được lưu trữ trong các hàng và các cột. Bạn xác định một bảng với một tên bảng (chẳng hạn như EMP) và tập hợp các cột. Bạn cho mỗi cột một tên cột (chẳng hạn như EMPNO, ENAME và JOB), một kiểu dữ liệu (chẳng hạn như VARCHAR2, DATE hoặc NUMBER) và độ rộng (độ rộng có thể được kiểu dữ liệu xác định, như trong DATE) hoặc độ chính xác và tỷ lệ (đối với các cột của chỉ kiểu dữ liệu NUMBER datatype). Một hàng là một tập hợp gồm thông tin cột tương ứng với một bản ghi đơn.

Hình 8-1 Các đối tượng giản đồ, các không gian bảng và các tập tin dữ liệu



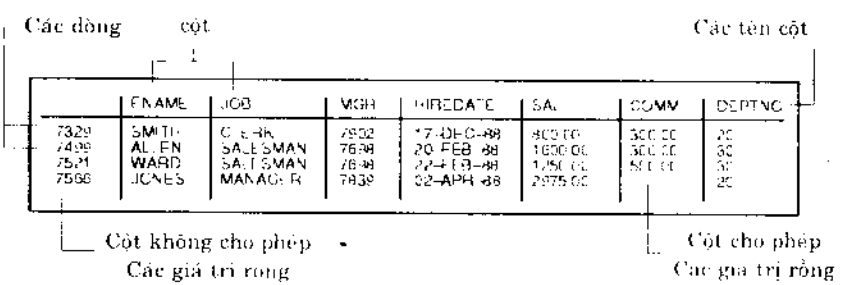
Xem "Các Kiểu Dữ Liệu Lập Sẵn" về các kiểu dữ liệu của Oracle.

Bạn có thể tùy chọn để xác định các nguyên tắc cho mỗi cột của bảng. Những nguyên tắc này được gọi là những hạn định tính toàn vẹn. Thí dụ là hạn định tính toàn vẹn NOT NULL. Hạn định này buộc cột chứa một giá trị trong mỗi hàng. Xem "Tính Toàn Vẹn Dữ Liệu" về các hạn định tính toàn vẹn.

Một khi bạn tạo một bảng, bạn chèn các hàng dữ liệu cách sử dụng các lệnh SQL. Dữ liệu bảng sau đó có thể được truy vấn, xóa hoặc cập nhật bằng cách dùng SQL.

Hình 8-2 trình bày bảng mẫu có tên EMP.

Hình 8-2 Bảng EMP



Dữ liệu bảng được lưu trữ như thế nào

Khi bạn tạo một bảng không có cụm liên kết, Oracle tự động phân bổ một đoạn dữ liệu trong không gian bảng để lưu trữ dữ liệu tương lai của bảng. Bạn có thể điều khiển phân bổ không gian cho đoạn dữ liệu của bảng và sử dụng không gian dự trữ này trong các cách sau:

- Bạn có thể điều khiển số lượng không gian được phân bổ cho đoạn dữ liệu ấy.
- Bạn có thể điều khiển cách sử dụng không gian trống trong các không gian dữ liệu cấu tạo các khối lưu riêng của đoạn dữ liệu bằng cách xác lập các tham số PCTFREE và PCTUSED đối với đoạn dữ liệu ấy.

Oracle lưu trữ dữ liệu cho bảng có cụm liên kết trong đoạn dữ liệu được tạo cho cụm liên kết ấy. Các tham số bộ nhớ không thể được ấn định khi một bảng có cụm liên kết được tạo hoặc thay đổi; các tham số bộ nhớ được xác lập cho cụm liên kết ấy lưu trữ điều khiển tất cả các bảng trong cụm liên kết ấy.

Không gian bảng chứa đoạn dữ liệu của bảng không có cụm liên kết là không gian bảng mặc định của chủ nhân bảng hoặc không gian bảng nhất là có tên trong câu lệnh CREATE TABLE. Xem "Các Xác Lập Không Gian Bảng Và Các Chỉ Tiêu".

Định dạng hàng và kích thước

Oracle lưu trữ mỗi hàng của bảng cơ sở dữ liệu bằng một hoặc nhiều phần hàng. Nếu toàn bộ một hàng có thể được chèn vào khối dữ liệu đơn, Oracle lưu trữ hàng ấy bằng một phần hàng. Tuy nhiên, nếu tất cả dữ liệu của một hàng không thể được chèn vào một khối dữ liệu đơn hoặc một cập nhật một hàng có sẵn gây cho hàng ấy tăng khối dữ liệu nhan, thì Oracle lưu trữ hàng sử dụng nhiều phần hàng. Một khối dữ liệu thường chỉ chứa hàng sử dụng một phân đoạn hàng trên mỗi hàng. Khi Oracle phải lưu trữ một hàng bằng nhiều phân đoạn hàng, nó được "lập chuỗi xích" qua nhiều khối. Các phân đoạn của

hàng được lập chuỗi xích được lập chuỗi xích lại với nhau bằng cách sử dụng ROWIDs của các phân đoạn. Xem "Lập Chuỗi Xích Hàng Và Di Trú".

Mỗi phân đoạn hàng được lập chuỗi xích hoặc giải chuỗi xích có phần đầu hàng và dữ liệu cho tất cả hoặc một số cột của hàng. Các cột riêng biệt có thể còn nối các phân đoạn hàng thành các các khối dữ liệu.

Hình 8-3 trình bày định dạng của một phân đoạn hàng.

Phần đầu hàng đứng trước dữ liệu và có thông tin về
các phân đoạn hàng
lập chuỗi xích (cho các phân đoạn hàng được lập chuỗi xích)
các cột trong phân đoạn hàng
các khóa cụm liên kết (cho dữ liệu có cụm liên kết)

Một hàng không có cụm liên kết chứa đầy đủ trong một khối có ít nhất ba byte của phần đầu hàng. Sau thông tin phần đầu hàng, mỗi hàng chứa chiều dài cột và dữ liệu. Chiều dài cột đòi hỏi một byte cho các cột lưu trữ 250 byte hoặc ít hơn hay ba byte cho các cột lưu trữ hơn 250 byte và đứng trước dữ liệu cột ấy. Không gian đòi hỏi cho dữ liệu cột tùy theo kiểu dữ liệu. Nếu kiểu dữ liệu của cột có chiều dài khả biến, thì không gian cần lưu trữ một giá trị có thể tăng lên và rút xuống với các cập nhật dữ liệu.

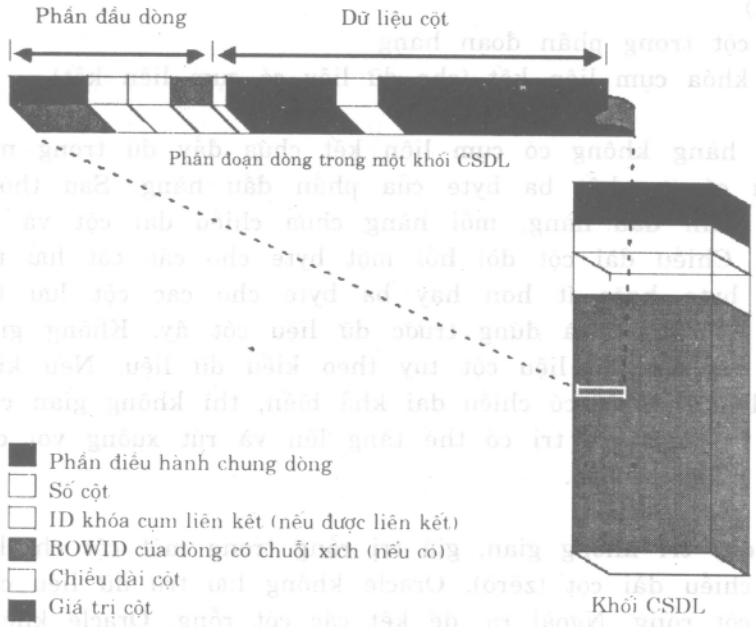
Để duy trì không gian, giá trị rỗng trong một cột chỉ lưu trữ chiều dài cột (zêrô). Oracle không lưu trữ dữ liệu cho các cột rộng. Ngoài ra, để kết các cột rỗng, Oracle không lưu trữ chiều dài cột vì phần đầu hàng phát tín hiệu bắt đầu một hàng mới (thí dụ, ba cột cuối cùng của bảng

rỗng, do đó không có thông tin được lưu trữ cho các cột đó).

Lưu ý:
Mỗi hàng sử dụng hai byte trong thư mục hàng của phần đầu khối dữ liệu.

Các hàng có cụm liên kết chứa cùng thông tin bằng các hàng không có cụm liên kết. Ngoài ra, chúng chứa thông tin tham chiếu khóa cụm liên kết mà chúng có quan hệ. Xem "Các Cụm Liên Kết".

Hình 8-3 Định dạng của một phân đoạn hàng



Thứ tự cột

Thứ tự cột tương tự đối với tất cả các hàng trong một bảng cho trước. Các cụm liên kết thường được lưu trữ theo thứ tự mà chúng đã được liệt kê trong câu lệnh CREATE TABLE, nhưng điều này không phải bảo đảm. Thí dụ, nếu bạn tạo một bảng với một cột có kiểu dữ liệu LONG, Oracle luôn luôn lưu trữ cột này cuối cùng. Hơn nữa, nếu một bảng được thay đổi để một cột mới được bổ sung, cột mới trở thành là cột cuối cùng được lưu trữ.

Nói chung, bạn phải cố đặt các cột thường chứa giá trị rỗng cuối cùng để các hàng chiếm ít không gian hơn. Lưu ý, dù nếu bảng bạn đang tạo cũng có cột LONG, thì các lợi ích đặt các cột rỗng thường xuyên cuối cùng bị mất.

Các ROWID của các phân đoạn hàng

ROWID nhận dạng mỗi phân đoạn hàng bằng vị trí hoặc địa chỉ của nó. Một khi được gán, phân đoạn hàng cho trước tiếp tục duy trì ROWID của nó cho tới khi hàng tương ứng được xóa hoặc được Export và được Import bằng cách sử dụng các tiện ích IMPORT và EXPORT. Nếu các giá trị khóa cụm liên kết của một hàng thay đổi, thì hàng ấy lưu giữ cùng ROWID, nhưng vẫn có ROWID điểm trỏ bổ sung cho các giá trị mới.

Bởi vì các ROWID là không thay đổi trong suốt thời gian tồn tại của một phân đoạn hàng, nó cần để tham chiếu các ROWID trong các câu lệnh SQL, chẳng hạn như SELECT, UPDATE và DELETE. Xem "Kiểu Dữ Liệu ROWID".

Các giá trị rỗng

Một giá trị rỗng không có giá trị trong một cột của hàng. Các giá trị rỗng chỉ bị mất, không biết rõ hoặc dữ liệu không thể ứng dụng. Một giá trị rỗng sẽ không được sử dụng để ám chỉ bất kỳ giá trị khác, chẳng hạn như như zêrô. Một cột cho phép các giá trị rỗng từ khi hạn định tính nguyên vẹn NOT NULL hoặc PRIMARY KEY được xác định cho cột ấy, trong trường hợp hàng không thể được chèn nếu không có giá trị cho cột đó.

Các giá trị rỗng được lưu trữ trong cơ sở dữ liệu, nếu chúng rớt vào giữa các cột có các giá trị dữ liệu. Trong các trường hợp này, chúng đòi hỏi một byte để lưu trữ chiều dài của cột ấy (zêrô). Các giá trị rỗng cuối trong một hàng đòi hỏi lưu trữ vì phần đầu hàng mới phát tín hiệu các cột còn lại trong hàng trước đó rỗng. Trong các bảng có nhiều cột, các cột rất có thể có các giá trị rỗng sẽ được xác định cuối cùng để duy trì không gian đĩa.

Hầu hết các so sánh giữa các giá trị rỗng và các giá trị khác được định nghĩa không đúng hoặc không sai, nhưng biết rõ. Để nhận dạng các giá trị rỗng trong SQL, sử dụng IS NULL khẳng định. Dùng hàm NVL của SQL để chuyển đổi các giá trị rỗng thành các giá trị không rỗng.

> Thông tin bổ sung :
> Xem Oracle8 SQL Reference Về Các So Sánh Sử Dụng IS NULL và hàm NVL.

Các giá trị rỗng không phải là chỉ mục, ngoại trừ khi giá trị cột có khóa cụm liên kết rỗng hoặc chỉ mục ấy là một chỉ mục bitmap, (xem "Các Chỉ Mục Bitmap Và Các Giá Trị Rỗng").

Các giá trị mặc định của các cột

Bạn có thể gán một cột của bảng bằng giá trị mặc định để khi một hàng mới được chèn và một giá trị cho cột ấy bị bỏ sót, thì giá trị mặc định tự động cung cấp. Các giá trị cột mặc định hoạt động như là câu lệnh INSERT thực sự ấn định giá trị mặc định.

Giá trị mặc định hợp lệ có trực kiện hoặc biểu thức bất kỳ không tham chiếu cột LEVEL, ROWNUM hoặc PRIOR. Các giá trị mặc định có thể gồm các hàm SYSDATE, USER, USERENV và UID của SQL. Kiểu dữ liệu của trực kiện hoặc biểu thức mặc định phải so khớp hoặc có thể chuyển đổi thành kiểu dữ liệu cột.

Nếu giá trị mặc định không được xác định rõ ràng cho một cột, thì giá trị mặc định cho cột đó được ngầm định là NULL.

Chèn giá trị mặc định và kiểm hạn định tính nguyên vẹn

Kiểm hạn định tính nguyên vẹn xảy ra sau khi hàng có giá trị mặc định được chèn. Thí dụ, ở Hình 8-4, một hàng được chèn vào bảng EMP liên kết các giá trị cho số phòng của nhân viên. Vì không có giá trị nào được cung cấp, Oracle chèn giá trị mặc định "20" của cột DEPTNO. Sau khi chèn giá trị mặc định, Oracle kiểm hạn định tính nguyên vẹn FOREIGN KEY được xác định trong cột DEPTNO.

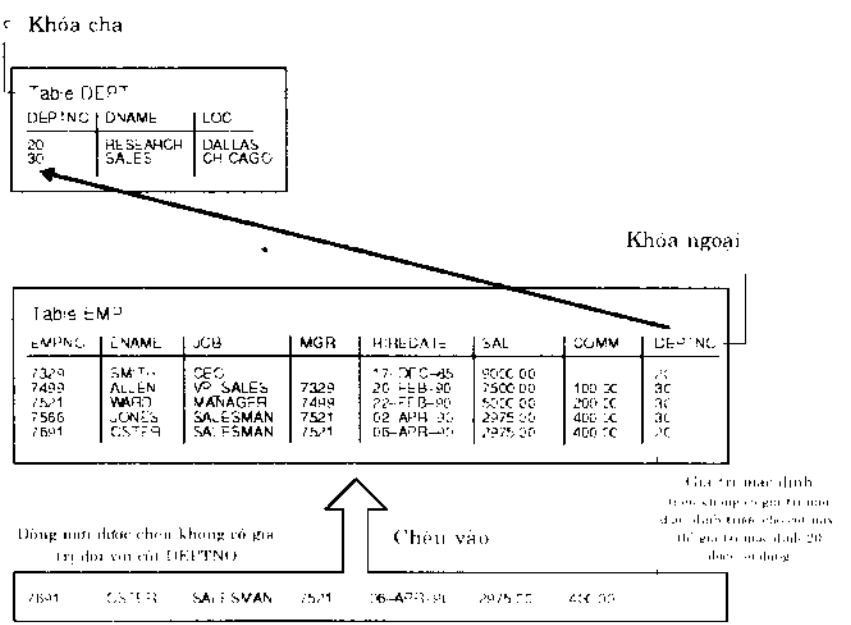
Xem thêm "Tính Nguyên Vẹn Dữ Liệu" về các hạn định tính nguyên vẹn.

Các bảng được lồng

Trong cơ sở dữ liệu quan hệ đối tượng Oracle, bạn có thể tạo một bảng với một cột có kiểu dữ liệu là một bảng khác. Nghĩa là, các bảng có thể được lồng nhau trong các bảng khác bằng những giá trị trong một cột. Oracle server lưu trữ dữ liệu bảng lồng nhau "ngoài tuyến" từ các hàng của bảng cha bằng cách sử dụng bảng lưu trữ được kết hợp với cột bảng được lồng. Hàng cha bây giờ chứa một giá trị định danh duy nhất được kết hợp với môi trường làm việc của bảng được lồng.

Thông tin bổ sung :
Xem Hướng Dẫn Của Developer Ứng Dụng Oracle8.

Hình 8-4 Các giá trị cột DEFAULT

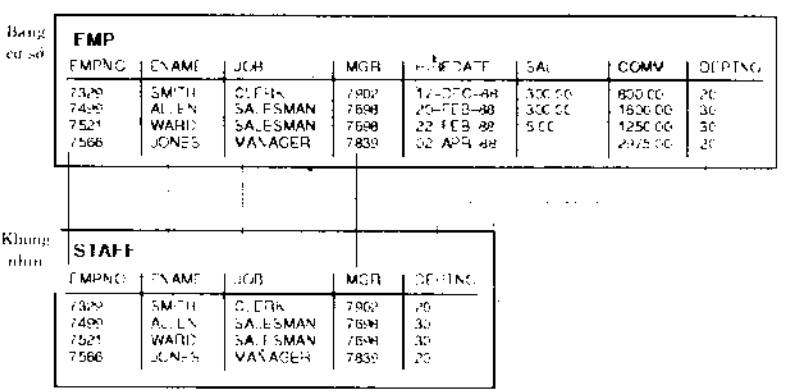


Các khung nhìn

Khung nhìn là một biểu thị đáp ứng nhu cầu của dữ liệu được chứa trong một hoặc nhiều bảng (hoặc các khung nhìn khác). Một khung nhìn dùng kết xuất của truy vấn hoặc coi nó là một bảng; do đó, một khung nhìn có thể được cho là một "truy vấn được lưu trữ" hoặc một "bảng ảo". Bạn có thể dùng các khung nhìn trong hầu hết các vị trí nơi một bảng cơ sở được sử dụng.

Thí dụ, bảng EMP có nhiều cột và nhiều hàng thông tin. Nếu bạn chỉ muốn người dùng thấy năm trong các cột này hoặc chỉ các hàng riêng biệt, bạn có thể tạo một khung nhìn của bảng đó cho các người dùng khác truy xuất. Hình 8-5 trình bày một thí dụ về khung nhìn được gọi là STAFF được phái sinh từ bảng EMP cơ sở. Lưu ý khung nhìn chỉ trình bày năm trong các cột đó của bảng cơ sở.

Hình 8-5 Một thí dụ về khung nhìn



Vì các khung nhìn được phát sinh từ các bảng, chúng có nhiều sự tương đồng. Thí dụ, bạn có thể xác định các khung nhìn tối đa 1.000 cột, giống như một bảng. Bạn có thể truy vấn các khung nhìn và với một số hạn chế bạn

có thể cập nhật, chèn vào và xóa khỏi khung nhìn. Tất cả các điều hoạt được thực hiện trên khung nhìn thực sự ảnh hưởng đến dữ liệu trong bảng cơ sở nào đó của khung nhìn và liên quan đến các hạn định tính toàn vẹn và các bộ kích khởi của các bảng cơ sở.

Thông tin bổ sung :

Xem Oracle8 SQL Reference.

Lưu ý :
Bạn không thể xác định rõ ràng các hạn định tính nguyên vẹn và các bộ kích khởi trên các khung nhìn, nhưng bạn có thể xác định chúng cho các bảng cơ sở nằm dưới được tham chiếu bằng khung nhìn.

Bộ nhớ cho các khung nhìn

Không giống như một bảng, một khung nhìn không được phân bổ bất kỳ không gian bộ nhớ nào, cũng không thực hiện một khung nhìn thực sự chứa dữ liệu; đúng hơn, một khung nhìn được xác định bằng một truy vấn dữ liệu trích ra hoặc phái sinh dữ liệu từ các bảng mà khung nhìn tham chiếu. Các bảng này được gọi là các bảng cơ sở. Các bảng cơ sở đối tượng trở thành các bảng thực hoặc có thể là chính các khung nhìn (kể cả các ảnh chụp nhanh). Vì một khung nhìn dựa trên các đối tượng khác, một khung nhìn không đòi hỏi bộ nhớ nào khác là bộ nhớ để định nghĩa cho của khung nhìn (bộ truy vấn định trước) trong từ điển dữ liệu.

Các khung nhìn được sử dụng như thế nào

Các khung nhìn cung cấp phương tiện để biểu thị một trình bày dữ liệu khác nhau cư trú trong các bảng cơ sở. Các khung nhìn rất hữu hiệu vì chúng cho bạn đáp ứng

thích hợp sự trình bày của dữ liệu với các kiểu người dùng khác nhau.

Các khung nhìn thường được sử dụng để cung cấp mức bảo mật bảng bằng cách hạn chế truy xuất tập hợp các hàng và/hoặc các bảng được xác định trước thí dụ, Hình 8-5 trình bày khung nhìn STAFF không biểu thị các cột SAL hoặc COMM của EMP bảng cơ sở.

Để ẩn tính phức tạp dữ liệu, thí dụ, một khung nhìn đơn có thể được xác định với một liên kết là một tập hợp các cột liên quan hoặc các hàng trong nhiều bảng. Tuy nhiên, khung nhìn ẩn sự kiện mà thông tin này thực sự phát sinh từ nhiều bảng.

Để đơn giản hóa các lệnh cho người dùng, thí dụ các khung nhìn cho phép người dùng lựa thông tin từ nhiều bảng mà không thực sự biết cách thường một liên kết.

Để biểu thị dữ liệu trong một phối cảnh khác từ bảng cơ sở ấy, thí dụ, các cột của một khung nhìn có thể được đặt tên lại mà không ảnh hưởng đến các bảng mà khung nhìn ấy căn cứ vào.

Để cách ly các ứng dụng khỏi bị các thay đổi trong các định nghĩa của các bảng cơ sở, thí dụ, nếu bộ truy vấn xác định của khung nhìn tham chiếu ba cột của một bảng bốn cột và cột thứ năm được bổ sung vào bảng, định nghĩa của khung nhìn không bị ảnh hưởng và tất cả các ứng dụng sử dụng khung nhìn ấy không bị ảnh hưởng.

Để biểu thị bộ truy vấn dữ liệu không thể được biểu thị nếu không sử dụng một khung nhìn, thí dụ, một khung nhìn có thể được xác định liên kết một khung nhìn GROUP BY với một bảng hoặc một khung nhìn có thể

được xác định liên kết một khung nhìn UNION với một bảng.

Thông tin bổ sung :
Xem Oracle8 SQL Reference về GROUP BY hoặc UNION.

Để lưu các bộ truy vấn dữ liệu phức tạp, thí dụ một truy vấn có thể thực hiện các tính toán mở rộng bằng thông tin bảng. Lưu truy vấn này như một khung nhìn, các tính toán có thể được thực hiện mỗi lần khung nhìn được truy vấn.

Cơ chế của các khung nhìn

Oracle lưu trữ một định nghĩa của khung nhìn trong từ điển dữ liệu bằng văn bản của truy vấn xác định khung nhìn ấy. Khi bạn tham chiếu một khung nhìn bằng một câu lệnh SQL, thì Oracle hợp nhất câu lệnh tham chiếu khung nhìn với truy vấn xử lý khung nhìn rồi sau đó phân ngữ câu lệnh được hợp nhất trong vùng SQL dùng chung và tiến hành nó. Oracle phân ngữ một câu lệnh tham chiếu một khung nhìn trong một vùng SQL dùng chung mới chỉ nếu vùng SQL dùng chung chứa câu lệnh đồng nhất không tồn tại. Do đó, bạn đạt được lợi ích là SQL dùng chung khi bạn sử dụng các khung nhìn.

Các tham số NLS

Các khung nhìn phẩm định chứa các trực kiện chuỗi hoặc các hàm SQL có các tham số NLS làm các đối số (chẳng hạn như TO_CHAR, TO_DATE, and TO_NUMBER). Oracle dùng các giá trị mặc định cho các tham số này từ các tham số NLS cho phiên làm việc đó. Bạn có thể chạy đè các giá trị mặc định bằng cách xác định những tham số NLS cụ thể trong định nghĩa của khung nhìn ấy.

Sử dụng các chỉ mục

Oracle xác định xem sử dụng những chỉ mục nào cho một truy vấn dữ liệu dựa trên một khung nhìn bằng cách biến đổi truy vấn dữ liệu ban đầu khi hợp nhất nó với truy vấn của khung nhìn đang xác định truy vấn. Xét khung nhìn

```
CREATE VIEW emp_view AS
SELECT empno, ename, sal, loc
FROM emp, dept
WHERE emp.deptno = dept.deptno AND dept.deptno =
10;
```

Bây giờ xét các truy vấn người dùng đưa ra sau:

```
SELECT ename
FROM emp_view
WHERE empno = 9876;
```

Truy vấn cuối cùng được Oracle cấu tạo:

```
SELECT ename
FROM emp, dept
WHERE emp.deptno = dept.deptno AND
dept.deptno = 10 AND
emp.empno = 9876;
```

Trong tất cả các trường hợp có thể xảy ra, Oracle hợp nhất một truy vấn dữ liệu dựa trên một khung nhìn với truy vấn xác định của khung nhìn ấy (và truy vấn của các khung nhìn cơ sở). Oracle tối ưu hóa truy vấn hợp nhất như bạn đã đưa ra truy vấn ấy mà không tham chiếu các khung nhìn. Do đó, Oracle có thể sử dụng những chỉ mục trên các cột bảng cơ sở được tham chiếu bất kỳ, xem các cột này được tham chiếu trong định

nghĩa khung nhìn hoặc truy vấn người dùng dựa trên khung nhìn ấy.
Trong một số trường hợp, Oracle không thể hợp nhất định nghĩa khung nhìn với truy vấn được người dùng đưa ra. Trong các trường hợp như vậy, Oracle không thể dùng tất cả các chỉ mục trên các cột tham chiếu.

Các sự phụ thuộc và các khung nhìn

Vì một khung nhìn được một truy vấn dữ liệu xác định để tham chiếu các đối tượng khác (các bảng, các ảnh chụp nhanh hoặc các khung nhìn khác), một khung nhìn bị phụ thuộc vào các đối tượng được tham chiếu. Oracle tự động xử lý các phụ thuộc đối với các khung nhìn. Thí dụ, nếu bạn giảm một bảng cơ sở của một khung nhìn rồi sau đó tạo lại nó, Oracle xác định xem bảng cơ sở mới có thể chấp nhận định nghĩa có sẵn của khung nhìn ấy không. Xem "Quản Lý Sự Phụ thuộc Oracle" về những phụ thuộc trong cơ sở dữ liệu.

Các khung nhìn liên kết cập nhật

Một khung nhìn liên kết được xác định bằng một khung nhìn với hơn một bảng hoặc khung nhìn trong mệnh đề FROM của nó và không dùng bất kỳ các mệnh đề sau: DISTINCT, AGGREGATION, GROUP BY, START WITH, CONNECT BY, ROWNUM và xác lập các điều hoạt (UNION ALL, INTERSECT, v.v.).

Một khung nhìn liên kết cập nhật là một khung nhìn liên kết liên quan đến hai hoặc nhiều bảng cơ sở hay các khung nhìn, nơi các điều hoạt UPDATE, INSERT và DELETE được cho phép. Từ điển dữ liệu hiển thị ALL_UPDATABLE_COLUMNS, DBA_UPDATABLE_COLUMNS và USER_UPDATABLE_COLUMNS chứa thông tin chỉ rõ

những cột nào của khung nhìn có thể cập nhật. Bảng 8-1 liệt kê các nguyên tắc cho các khung nhìn liên kết cập nhật.

Bảng 8-1 Các nguyên tắc cho INSERT, UPDATE và DELETE trên các khung nhìn liên kết trên các Join View

Nguyên tắc	Mô tả
Nguyên tắc chung	Điều hoạt INSERT, UPDATE hoặc DELETE bất kỳ trên một khung nhìn liên kết chỉ có thể sửa đổi một lần.
Nguyên tắc UPDATE	Tất cả các cột cập nhật của một khung nhìn liên kết phải ánh xạ vào các cột của một bảng khóa được dự trữ. Nếu khung nhìn được xác định bằng mệnh đề WITH CHECK OPTION, sau đó tất cả các cột liên kết và tất cả các cột của các bảng được lặp lại không cập nhật được.
Nguyên tắc DELETE	Các hàng của một khung nhìn liên kết có thể được xóa bao lâu có một bảng khóa dự trữ chính xác trong liên kết ấy. Nếu khung nhìn được xác định bằng mệnh đề WITH CHECK OPTION và bảng khóa dự trữ được lặp lại, sau đó các hàng không thể xóa khỏi khung nhìn đó.
Nguyên tắc INSERT	Câu lệnh INSERT không rõ ràng hoặc ám chỉ tham chiếu các cột của một bảng không có khóa dự trữ. Nếu khung nhìn liên kết được xác định với mệnh đề WITH CHECK OPTION, các câu lệnh INSERT được cho phép.

Các khung nhìn không cập nhật được có thể bị sửa đổi bằng cách sử dụng các bộ kích khởi INSTEAD OF. Xem "Các Bộ Kích Khởi INSTEAD OF".

Các khung nhìn đối tượng

Trong CSDL quan hệ đối tượng Oracle, các khung nhìn đối tượng cho bạn truy hồi, cập nhật, chèn và xóa dữ liệu quản lý như chung đã được lưu trữ bằng các kiểu đối tượng. Bạn còn có thể xác định các khung nhìn có các cột là các kiểu dữ liệu đối tượng, chẳng hạn như các đối tượng, REFs và các tập hợp (các bảng được lồng nhau và các VARRAY).

Thông tin bổ sung :
Xem "Các Khung Nhìn Đối Tượng" và Hướng dẫn của Oracle Application Developer.

Bộ phát sinh chuỗi

Bộ phát sinh chuỗi cung cấp hàng loạt chuỗi số. Bộ phát sinh chuỗi rất hữu dụng trong các môi trường nhiều người dùng để phát sinh các số chuỗi chuỗi duy nhất mà không có phần điều hành chung đĩa I/O hoặc khóa giao dịch. Do đó, bộ phát sinh chuỗi làm giảm "sự nối tiếp hóa" nơi các câu lệnh của hai giao dịch phải phát sinh các số chuỗi cùng một lúc. Bằng cách tránh sự nối tiếp hóa tạo ra khi nhiều người dùng cho nhau để phát sinh và sử dụng số chuỗi, bộ phát sinh chuỗi làm tăng thông suất giao dịch và người dùng phải chờ ít hơn đáng kể.

Các số chuỗi là các số nguyên Oracle được xác định trong cơ sở dữ liệu tối đa 38 chữ số. Một định nghĩa chuỗi chỉ thông tin chung: tên của chuỗi, xem nó tăng hoặc giảm, các khoảng cách giữa các số và thông tin khác. Một phần

quan trọng của định nghĩa chuỗi là xem Oracle sẽ trữ các tập hợp của các số chuỗi được phát sinh trong bộ nhớ.

Oracle lưu trữ những định nghĩa của tất cả những chuỗi cho một CSDL riêng biệt bằng các hàng trong một bảng từ điển dữ liệu đơn trong không gian bảng SYSTEM. Do đó, tất cả những định nghĩa chuỗi luôn luôn khả dụng, vì không gian bảng SYSTEM luôn luôn trực tuyến.

Các số chuỗi được những câu lệnh SQL sử dụng để tham chiếu chuỗi. Bạn có thể đưa ra một câu lệnh để phát sinh một số chuỗi mới hoặc sử dụng số chuỗi hiện hành. Một khi câu lệnh trong phiên làm việc của người dùng phát sinh một số chuỗi, thì số chuỗi riêng biệt chỉ khả dụng với phiên làm việc đó; mỗi người dùng tham chiếu một chuỗi để truy xuất riêng nó, số chuỗi hiện hành.

Các số chuỗi được phát sinh độc lập với các bảng. Do đó, bộ phát sinh số chuỗi cần sử dụng để phát sinh các khóa chính đối với dữ liệu tự động và điều hợp các khóa qua nhiều hàng hoặc bảng. Các số chuỗi riêng biệt có thể được bỏ qua nếu chung được phát sinh và được sử dụng trong một dịch vụ cuối cùng đã được hồi phục trạng thái cũ. Các ứng dụng có thể tạo các phân chia để trữ và sử dụng lại những số chuỗi này, nếu cần.

Thông tin bổ sung :
Đối với các ám chỉ hiệu suất khi sử dụng các chuỗi, xem Hướng dẫn của "Oracle8 Appliation Developer".

Các từ đồng nghĩa

Từ đồng nghĩa là một bí danh cho bảng, khung nhìn, ảnh chụp nhanh, chuỗi, thủ tục, hàm hoặc bộ phần mềm bất kỳ. Vì từ đồng nghĩa chỉ là một bí danh, nó không đòi

hỏi bộ nhớ thay vì định nghĩa của nó trong từ điển dữ liệu.

Các từ đồng nghĩa thường được dùng để bảo mật và cho tiện dụng. Thí dụ, chúng có thể thực hiện như sau:

Dấu tên và chủ nhân của đối tượng
Cung cấp khả năng thông suốt vị trí cho các đối tượng từ xa của một CSDL được phân phối.
Đơn giản hóa các câu lệnh SQL cho những người dùng CSDL

Bạn có thể tạo cả các từ đồng nghĩa chung và riêng. Một từ đồng nghĩa chung được một nhóm người dùng đặc biệt được gọi là PUBLIC và mọi người dùng trong CSDL có thể truy xuất nó. Một từ đồng nghĩa riêng trong một giản đồ của người dùng riêng biệt có quyền điều khiển tính khả dụng của nó so với những người dùng khác.

Các từ đồng nghĩa rất hữu dụng với cả môi trường được phân phối và không được phân phối vì chúng ấn định danh của đối tượng cơ bản, kể cả vị trí của nó trong hệ thống phân phối. Đây là một ưu điểm vì nếu đối tượng cơ bản phải được đặt tên lại hoặc di chuyển, thì chỉ từ đồng nghĩa cần được xác định lại và các ứng dụng dựa trên từ động nghĩa ấy tiếp tục thực hiện chức năng mà không cần sửa đổi.

Các từ đồng nghĩa còn có thể đơn giản hóa các câu lệnh SQL ch những người dùng trong hệ thống CSDL được phân phối. Thí dụ sau trình bày các từ động nghĩa tại sao thường được người điều hành CSDL tạo và như thế nào để ẩn định danh của bảng cơ sở và làm giản tính phức tạp của các câu lệnh SQL. Giả sử như sau:

Một bảng được gọi là SALES_DATA trong giản đồ được người dùng JWARD làm chủ.
Đặc quyền SELECT đối với bảng SALES_DATA được cấp với PUBLIC.

Ở điểm này, bạn sẽ phải truy vấn bảng SALES DATA bằng câu lệnh SQL tương tự với câu lệnh sau:

SELECT * FROM jward.sales_data;

Lưu ý bạn phải có giản đồ chứa bảng ấy cùng với tên bảng để thực hiện truy vấn đó.

Giả sử người điều hành CSDL tạo một từ đồng nghĩa chung bằng câu lệnh SQL:

CREATE PUBLIC SYNONYM sales FOR jward.sales_data;

Sau khi từ đồng nghĩa chung được tạo, bạn có thể truy vấn bảng SALES_DATA bằng câu lệnh SQL:

SELECT * FROM sales;

Lưu ý từ đồng nghĩa chung SALES ẩn tên của bảng SALES_DATA và tên của giản đồ chứa bảng đó.

Các chỉ mục

Các chỉ mục là các cấu trúc tùy chọn được kết hợp với các bảng và các cụm liên kết. Bạn có thể tạo các chỉ mục rõ ràng để tăng tốc độ thực hiện của câu lệnh SQL trên một bảng. Giống như chỉ mục trong tài liệu này giúp bạn định vị thông tin nhanh hơn nếu không có chỉ mục, chỉ mục Oracle là phương tiện chủ yếu để làm giảm đĩa I/O khi được sử dụng thích hợp.

Oracle cung cấp nhiều giản đồ chỉ mục để cung cấp tính năng hiệu suất bổ sung: các chỉ mục B*-tree (hiện phổ biến nhất), các chỉ mục cụm liên kết B*, các chỉ mục cụm liên kết mã số, cửa sổ chỉ mục khóa đảo ngược và các chỉ mục bipmap.

Có hoặc không có chỉ mục không nhất thiết phải thay đổi trình bày câu lệnh SQL bất kỳ. Một chỉ mục chính là một đường nhanh đối với dữ liệu; nó chỉ ảnh hưởng đến tốc độ thực hiện. Giá trị dữ liệu cho trước đã có chỉ mục, thì chỉ mục ấy trỏ trực tiếp vị trí các hàng chứa giá trị đó.

Các chỉ mục độc lập về mặt logic và vật lý với dữ liệu trong bảng được kết hợp. Bạn có thể tạo hoặc giảm chỉ mục bất kỳ lúc nào ma không ảnh hưởng đến các bảng cơ sở hoặc các chỉ mục khác. Nếu bạn bỏ một chỉ mục, thì ứng dụng tiếp tục hoạt động, tuy nhiên, truy xuất của dữ liệu có chỉ mục trước sẽ chậm hơn. Các chỉ mục là các cấu trúc độc lập đòi hỏi không gian lưu trữ.

Oracle tự động duy trì và sử dụng những chỉ mục khi chung được tạo. Oracle tự động phản ánh nhưng thay đổi đối với dữ liệu, chẳng hạn như bổ sung các hàng mới, cập nhật các hàng hoặc xóa các hàng, bằng các chỉ mục thích ứng không có tác động bổ sung của các người dùng.

Hiệu suất truy hồi của dữ liệu có chỉ mục hầu như không thay đổi, ngay cả khi các hàng mới được chen vào. Tuy nhiên, sự hiện hữu của nhiều chỉ mục trên một bảng làm giảm hiệu suất cập nhật, xoa và chèn vì Oracle còn phải cập nhật các chỉ mục được kết hợp với bảng đó.

Bộ tối ưu hóa có thể sử dụng chỉ mục có sẵn để thiết lập một chỉ mục khác. Điều này làm cho việc thiết lập chỉ mục nhanh hơn nhiều.

Các chỉ mục duy nhất và không duy nhất

Các chỉ mục có thể là duy nhất hoặc không. Các chỉ mục duy nhất bảo đảm không có hai hàng của bảng sao chép các giá trị trong các cột xác định chỉ mục đó. Các chỉ mục không duy nhất không bắt buộc sự hạn chế nào trên các giá trị của cột.

Oracle đề nghị bạn không xác định cụ thể các chỉ mục duy nhất trên các bảng; chỉ mục không duy nhất là một khái niệm logic nghiệm ngặt và sẽ được kết hợp với định nghĩa của bảng. Nói cách khác, xác định tính nguyên vẹn UNIQUE giới hạn các cột được yêu cầu. Oracle buộc tính nguyên vẹn UNIQUE giới hạn bằng cách xác định tự động một chỉ mục duy nhất trên một khóa riêng biệt.

Các chỉ mục tổng hợp

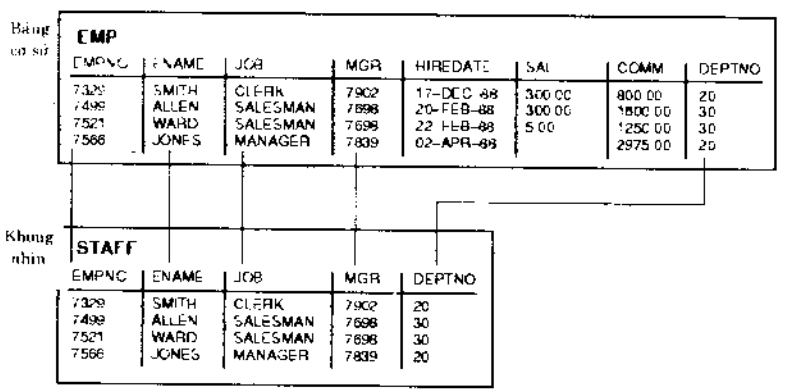
Một chỉ mục tổng hợp (còn được gọi là chỉ mục ghép nối) là một chỉ mục bạn tạo trên nhiều cột trong một bảng. Các cột trong một chỉ mục tổng hợp có thể xuất hiện theo thứ tự bất kỳ và không cần liên tiếp trong bảng.

Các chỉ mục tổng hợp có thể làm tăng tốc độ truy hồi dữ liệu bảng các câu lệnh SELECT trong đó mệnh đề WHERE tham chiếu tất cả hoặc phần đầu của các cột trong chỉ mục tổng hợp đó. Do đó, thứ tự của các cột được sử dụng trong định nghĩa rất cần thiết; nói chung, được truy xuất phổ biến nhất hoặc phần lớn các cột lựa tiến hành trước.

Thông tin bổ sung .
Xem Oracle8 Tuning.

Hình 8-6 minh họa bảng VENDOR_PARTS có một chỉ mục tổng hợp trên các cột VENDOR_ID và PART_NO.

Hình 8-6 Các chỉ mục, các khóa chính, các khóa duy nhất và các khóa ngoại



Không hơn 32 cột có thể tạo thành một chỉ mục tổng hợp thường lệ và đối với chỉ mục bitmap số cột tối đa là 30. Một giá trị khóa không thể vượt một phần nửa (trừ đi phần điều hành) không gian dữ liệu khả dụng trong khối dữ liệu.

Các chỉ mục và các khóa

Dù thuật ngữ thường được sử dụng thay thế lẫn nhau giữa "các chỉ mục" và "các khóa", nhưng bạn nên phân biệt chúng. Các chỉ mục là các cấu trúc thực sự được lưu trữ trong CSDL mà người dùng tạo, thay thế và rút gọn các câu lệnh SQL. Bạn tạo một chỉ mục để cung cấp đường dẫn truy xuất nhanh dữ liệu bảng. Các khóa là

khái niệm logic nghiêm ngặt. Các khóa tương ứng với một đặc trưng khác của Oracle được gọi là các hạn định tính nguyên vẹn.

Các hạn định tính nguyên vẹn tăng cường các nguyên tắc thương mại của CSDL (xem "Tính Nguyên Vẹn Của Dữ Liệu". Vì Oracle sử dụng các chỉ mục để tăng cường một số hạn định tính nguyên vẹn dữ liệu, thuật ngữ khóa và chỉ mục thường được sử dụng thay thế lẫn nhau; tuy nhiên, không nên lầm lẫn chúng với nhau.

Các chỉ mục được lưu trữ như thế nào

Khi bạn tạo một chỉ mục, Oracle tự động phân bổ một đoạn chỉ mục để lưu trữ dữ liệu của chỉ mục ấy trong một không gian bảng. Bạn điều khiển phân bổ của không gian cho phân đoạn chỉ mục và sử dụng không gian dự trữ này bằng các cách sau:

Xác lập các tham số cho phân đoạn chỉ mục để điều khiển sự phân bổ của các khối lưu riêng của phân đoạn chỉ mục.
Xác lập tham số PCTFREE cho phân đoạn chỉ mục để điều khiển không gian trống trong các khối dữ liệu cấu tạo các khối lưu riêng của phân đoạn chỉ mục.

Không gian bảng của một phân đoạn chỉ mục là không gian bản mặc định của chủ nhân hoặc không gian bảng được đặt tên riêng trong câu lệnh CREATE INDEX. Bạn không phải đặt một chỉ mục trong cùng không gian bảng bằng bảng được kết hợp của nó. Hơn nữa, bạn có thể tăng hiệu suất của các truy vấn dữ liệu sử dụng chỉ mục bằng cách lưu trữ một chỉ mục và bảng của nó trong các không gian bảng khác nhau được định vị trên các ổ đĩa khác nhau vì Oracle có thể truy hồi cả chỉ mục và dữ

liệu bảng song song. Xem "Xác Lập Không Gian Bảng Của Người Dùng Và Các Chỉ Tiêu".

Định dạng các khối chỉ mục

Không gian bảng khả dụng đối với dữ liệu chỉ mục là kích thước khối dữ liệu Oracle trừ đi phần điều điều hành chung khối dữ liệu, phần điều hành chung mục nhập, ROWID và một chiều dài byte trên giá trị có chỉ mục. Số byte cần thiết cho phần điều hành chung của một khối chỉ mục phụ thuộc vào hệ điều hành.

Thông tin bổ sung :
Xem tài liệu hệ điều hành Oracle về phần điều hành chung của một khối chỉ mục.

Khi bạn tạo một chỉ mục, Oracle truy nạp và sắp xếp các cột có chỉ mục và lưu trữ ROWID cùng với giá trị chỉ mục cho mỗi hàng. Sau đó Oracle nạp chỉ mục từ bên dưới lên. Thí dụ, xét câu lệnh:

```
CREATE INDEX emp_ename ON emp(ename);
```

Oracle sắp xếp bảng EMP trên cột ENAME. Rồi sau đó nó nạp chỉ mục ấy với ENAME và những giá trị ROWID tương ứng theo thứ tự sắp xếp này. Khi nó sử dụng chỉ mục ấy, Oracle tìm kiếm nhanh qua những giá trị ENAME và sau đó dùng những giá trị ROWID được kết hợp để định vị các hàng có giá trị ENAME được tìm thấy.

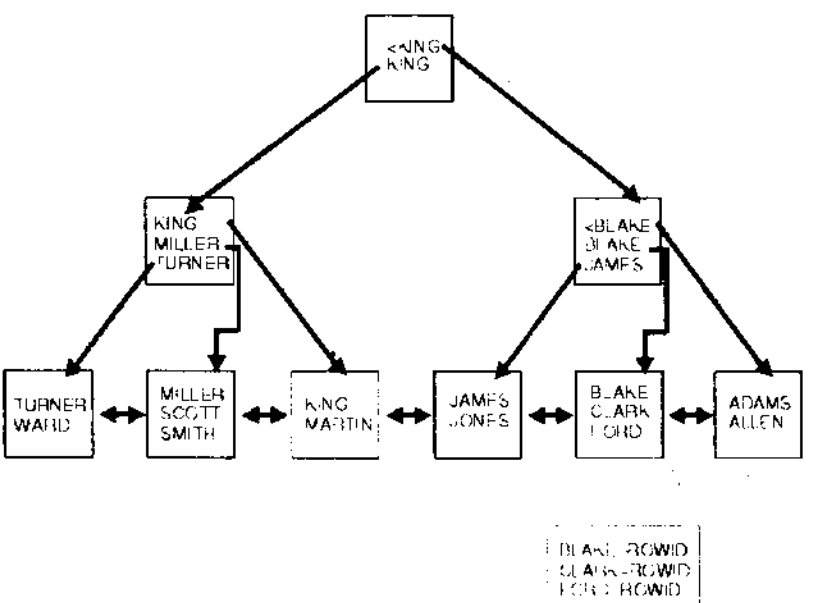
Dù Oracle chấp nhận các từ khóa ASC, DESC, COMPRESS và NOCOMPRESS trong lệnh CREATE INDEX, nhưng chúng không có ảnh hưởng đến dữ liệu chỉ

được lưu trữ sử dụng nén ở cuối trong các nút nhánh nhưng không ở các nút lá.

Cấu trúc bên trong của các chỉ mục

Oracle sử dụng các chỉ mục B*-tree được cân bằng để phân điều thời gian truy xuất một hàng bất kỳ. Lý thuyết của các chỉ mục B*-tree bên ngoài phạm vi của tài liệu này; để có thêm thông tin bạn có thể tham chiếu các văn bản môn tin học liên quan đến các cấu trúc dữ liệu. Hình 8-7 minh họa chỉ mục B*-tree.

Hình 8-7 Cấu trúc bên trong của chỉ mục B*-tree



Các khối dữ liệu bên trên (các khối dữ liệu nhánh) của chỉ mục B*-tree chứa dữ liệu chỉ mục trỏ các khối dữ liệu chỉ mục mức thấp hơn. Các khối dữ liệu chỉ mục mức thấp nhất (các khối lá) chứa mọi giá trị dữ liệu có chỉ

mục và ROWID tương ứng được sử dụng để định vị hàng thực đó; các khối dữ liệu lá được liên kết hai mặt. Các chỉ mục trong các cột chứa dữ liệu ký tự dựa trên những giá trị nhị phân của các ký tự trong tập hợp ký tự CSDL.

Đối với một chỉ mục duy nhất, có một ROWID trên mỗi giá trị dữ liệu. Đối với một chỉ mục không duy nhất, ROWID ở trong khóa theo thứ tự được sắp xếp, vì vậy các chỉ mục không duy nhất được sắp xếp theo khóa chỉ mục và ROWID. Những giá trị khóa chứa tất cả giá trị rỗng không có chỉ mục, ngoại trừ các chỉ mục cụm liên kết. Hai hàng có thể chứa tất cả các giá trị rỗng và không xâm phạm chỉ mục duy nhất.

Các ưu điểm của cấu trúc B*-tree

Cấu trúc B*-tree có các ưu điểm sau:
Tất cả những khối dữ liệu lá của câu ở cùng độ sâu, vì vậy truy hồi bản ghi bất kỳ từ bất kỳ nơi nào trong chỉ mục mất khoảng thời gian gần bằng nhau. Các chỉ mục B*-tree tự động duy trì cân bằng.
Tất cả những khối dữ liệu của câu B* trung bình bằng ba phân tư.
Các B*-tree cung cấp hiệu suất truy hồi tốt nhất cho các truy vấn phạm vi rộng, kể cả so khớp chính xác và các tìm kiếm phạm vi. Chèn, cập nhật và xóa có hiệu quả duy trì thứ tự khóa để truy hồi nhanh.
Hiệu suất B*-tree tốt đối với cả các bảng nhỏ và lớn, và không giảm cấp khi kích thước của bảng tăng.

Đảo ngược các chỉ mục khóa

Tạo một chỉ mục khóa đảo ngược, so sánh với chỉ mục tiêu chuẩn, đảo ngược các số byte của mỗi cột có chỉ mục

(ngoại từ ROWID) khi duy trì thứ tự cột. Một sự sắp xếp như vậy có thể giúp tránh làm giảm hiệu suất trong các chỉ mục trong một môi trường Oracle Parallel Server nơi các sửa đổi đối với chỉ mục ấy được tập trung vào một tập hợp nhỏ các khối dữ liệu lá. Bằng cách đảo ngược các khóa của chỉ mục, các chỉ mục chèn trở thành được phân phối qua tất cả các khóa lá trong chỉ mục ấy.

Sử dụng sắp xếp khóa đảo ngược loại bỏ khả năng thực hiện một phạm vi chỉ mục (scan) đọc truy vấn trên chỉ mục ấy. Vì các khóa liên tiếp về mặt từ vựng không được lưu trữ sát bên nhau trong một chỉ mục khóa đảo ngược, chỉ mục khóa theo truy nạp hoặc duyệt chỉ mục hoàn toàn (bảng) có thể được thực hiện.

Dưới một số trường hợp sử dụng chỉ mục khóa đảo ngược có thể tạo cho ứng dụng OLTP Oracle Parallel Server nhanh hơn. Thí dụ, lưu chỉ mục của các thông điệp thư tín trong Oracle Office: một số người dùng lưu các thông điệp cũ và chỉ mục phải duy trì các điểm trỏ những thông điệp này cũng như hầu hết các thông điệp mới nhất.

Từ khóa REVERSE cung cấp cơ chế đơn giản để tạo một chỉ mục khóa đảo ngược. Bạn có thể xác định từ khóa REVERSE cùng với các định chuẩn chỉ mục tùy chọn bằng câu lệnh CREATE INDEX:

```
CREATE INDEX i ON t (a,b,c) REVERSE;
```

Bạn có thể ấn định từ khóa NOREVERSE thành một chỉ mục đảo ngược REBUILD vào một chỉ mục không có khóa đảo ngược:

```
ALTER INDEX i REBUILD NOREVERSE;
```

Tái thiết lập một chỉ mục khóa đảo ngược mà không có từ khóa NOREVERSE tạo ra một sự tái thiết lập một chỉ mục khóa đảo ngược. Bạn không thể tái thiết lập một chỉ mục bình thường thành chỉ mục khóa đảo ngược; thay vì thế bạn phải sử dụng lệnh CREATE.

Các chỉ mục bitmap

Lưu ý:
Những chỉ mục bitmap chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition. Xem Getting Know Oracle8 Enterprise Edition về các đặc trưng khả dụng trong Oracle8 và Oracle8 Enterprise Edition.

Mục đích của chỉ mục là để cung cấp những điểm trỏ các hàng trong bảng chứa một giá trị khóa cho trước. Trong một chỉ mục bình thường, điều đạt được bằng cách lưu danh sách của các ROWIDs cho mỗi khóa tương ứng với các hàng với giá trị khóa đó. (Oracle lưu trữ mỗi giá trị khóa được lặp lại với mỗi ROWID được lưu trữ.) Trong chỉ mục bitmap, bitmap cho mỗi giá trị khóa được sử dụng thay cho một danh sách các ROWID.

Mỗi bit trong bitmap tương ứng với ROWID có thể xảy ra và nếu bit ấy được xác lập, có nghĩa là hàng ấy có ROWID tương ứng chứa giá trị khóa. Một hàm ánh xạ chuyển đổi vị trí bit thành ROWID thực, vì vậy chỉ mục bitmap cung cấp cùng tính năng như một chỉ mục bình thường dù nó sử dụng biểu thị khác nhau bên trong. Nếu số giá trị khóa khác nhau nhỏ, thì những chỉ mục rất hiệu quả về không gian.

Lập chỉ mục bitmap hợp nhất những chỉ mục có hiệu quả cần tương ứng với nhiều điều kiện trong mệnh đề

WHERE. Các hàng thỏa dáng một số điều kiện, nhưng tất cả những điều kiện được lọc trước khi chính bảng ấy được truy xuất. Điều này cải thiện được thời gian trả lời, thường rất lớn.

Những lợi ích đối với các ứng dụng kho

Lập chỉ mục bitmap khai thác được những ứng dụng kho có số lượng dữ liệu lớn và những truy vấn không định trước, nhưng ở mức các giao dịch đồng thời thấp. Đối với các ứng dụng như thế, lập chỉ mục bitmap cung cấp:

Thời gian trả lời giản đối với các lớp truy vấn không định trước
Giảm đáng kể không gian sử dụng so với những kỹ thuật lập chỉ mục khác
Hiệu suất tăng rất nhiều ngay trên phần cứng đầu cuối rất thấp
DML song song và nạp rất có hiệu quả

Lập chỉ mục đầy đủ một bảng lớn với chỉ mục B*-tree truyền thống có thể rất tốn phí về mặt không gian vì chỉ mục có thể lớn hơn dữ liệu trong bảng gấp nhiều lần. Các chỉ mục bitmap thường chỉ là một phần của kích thước dữ liệu có chỉ mục trong bảng.

Những chỉ mục bitmap không thích hợp đối với những ứng dụng OLTP có số lượng lớn những giao dịch đồng thời sửa đổi dữ liệu. Những chỉ mục chủ yếu để hỗ trợ quyết định (DSS) trong những ứng dụng kho nơi người dùng thường truy vấn dữ liệu hơn là cập nhật nó.

Các chỉ mục bitmap được tích hợp với cách tiếp cận tối ưu hoa dựa trên chi phí Oracle và động cơ thực hiện. Chúng có thể được sử dụng tổ hợp với những phương

pháp tiến hành Oracle một cách liên lạc. Thí dụ, bộ tối ưu hóa có thể quyết định để thực hiện một liên kết trị số giữa hai bảng sử dụng một chỉ mục bitmap trên một bảng và chỉ mục B*-tree trên một bảng khác. Bộ tối ưu hóa xét các chỉ mục bitmap và các phương pháp truy xuất khả dụng khác, chẳng hạn như các chỉ mục B*-tree thường lệ và duyệt toàn bộ bảng, chọn phương pháp hiệu quả nhất để đưa quan hệ song song vào tài khoản nơi thích hợp.

Truy vấn song song và DML song song hoạt động với những chỉ mục bitmap như với những chỉ mục truyền thống. Những chỉ mục bitmap trên các bảng được phân hoạch phải là các chỉ mục cục bộ; xem "Phân Hoạch Chỉ Mục" về Parallel tạo chỉ mục và những chỉ mục ghép nối cũng được hỗ trợ.

Số phần tử

Những ưu điểm sử dụng các chỉ mục bitmap lớn nhất đối với những cột có số phần tử thấp: nghĩa là các cột trong đó số giá trị riêng biệt nhỏ so với số hàng trong bảng. Nếu những giá trị trong cột được lặp lại hơn 100 lần, thì cột ấy là một viên tuyển cho chỉ mục bitmap. Ngay cả các cột với số lần được lặp lại thấp hơn (và do đó số phần tử cao hơn) có thể là các ứng viên nếu chúng có chiều hướng liên quan đến những điều kiện phức tạp trong các mệnh đề của truy vấn.

Thí dụ, trên một bảng với một triệu hàng, thì một cột có 10.000 giá trị riêng biệt là một ứng viên cho một chỉ mục bitmap. Chỉ mục bitmap trên cột này có thể thực hiện ngoài chỉ mục B*-tree, nhất là khi cột này thường được truy vấn chung với các cột khác.

Những chỉ mục B*-tree có hiệu quả nhất đối với dữ liệu có số phần tử cao: nghĩa là dữ liệu có nhiều giá trị có thể xảy ra, chẳng hạn như CUSTOMER NAME hoặc PHONE_NUMBER. Một chỉ mục B*-tree thường lệ có thể lớn hơn dữ liệu có chỉ mục nhiều lần. Được sử dụng thích hợp, những chỉ mục bitmap có thể nhỏ hơn chỉ mục B*-tree tương ứng đáng kể.

Trong những truy vấn dữ liệu không định trước và các trường hợp tương tự, những chỉ mục bitmap có thể làm tăng hiệu suất truy vấn lên rất nhiều. Những điều kiện AND và OR trong mệnh đề WHERE của một truy vấn dữ liệu có thể giải quyết nhanh bằng cách thực hiện những phép toán boolean tương ứng trực tiếp trên các bitmap trước khi chuyển đổi bitmap kết quả thành các ROWID. Nếu số kết quả hàng nhỏ, thì truy vấn ấy có thể được trả lời rất nhanh mà không sắp xếp lại duyệt toàn bộ bảng của bảng ấy.

Thí dụ chỉ mục bitmap

Bảng 8-2 trình bày một phần của dữ liệu khách hàng của một công ty.

Bảng 8-2 Thí dụ chỉ mục Bitmap

CUSTOMER #	MARITAL_STATUS	REGION	GENDER	INCOME_LEVEL
101	Độc thân	Đông	Nam	Thuế thu nhập_1
102	Đã kết hôn	Trung tâm	Nữ	Thuế thu nhập_4
103	Đã kết hôn	Tây	Nữ	Thuế thu nhập _2
	Ly dị	Tây	Nam	Thuế thu

104				nhập_1
105	Độc thân	Trung tâm	Nữ	Thuộc thu nhập_2
106	Đã kết hôn	Trung tâm	Nữ	Thuộc thu nhập_3

Vì MARITAL STATUS, REGION, GENDER và INCOME LEVEL đều là các cột có số phần tử thấp (chỉ có ba giá trị có thể xảy ra đối với tình trạng hôn nhân và vùng, hai giá trị. Xem thêm xảy ra đối với phái tính và bốn đối với mức lợi tức) nó thích hợp để tạo những chỉ mục bitmap trên những cột này. Chỉ mục bitmap sẽ không được tạo trên cột CUSTOMER# vì đây là cột có số phần tử cao. Thay vào đó, một chỉ mục B*-tree duy nhất trên cột này theo thức tự sẽ cung cấp sự trình bày và truy hồi hiệu quả nhất.

Bảng 8-3 minh họa chỉ mục bitmap đối với cột REGION trong thí dụ này. Nó gồm ba bitmap tách riêng, mỗi bitmap cho một vùng.

Bảng 8-3 Bitmap mẫu

REGION='đông'	REGION='trung tâm'	REGION='tây'
1	0	0
0	1	0
0	0	1
0	0	1
0	1	0

0	1	0

Mỗi mục nhất (hoặc "bit") trong bitmap tương ứng với một hàng đơn của bảng CUSTOMER. Giá trị của mỗi bit phụ thuộc vào những giá trị của hàng tương ứng trong bảng ấy. Thí dụ, bitmap REGION='đông' chứa một giá trị làm bit thứ nhất của nó; bởi vì đây là vùng "đông" có một zêrô cho các bit khác của nó vì không có hàng khác của bảng chứa "phía đông" làm giá trị của chung cho REGION.

Một nhà phân tích điều tra các khuynh hướng nhân khẩu của khách hàng có thể hỏi, "Có bao nhiêu trong bốn khách hàng đã lập gia đình sống ở vùng trung tâm hay vùng phía tây?" điều này tương ứng với truy vấn SQL sau:

```
SELECT COUNT(*) FROM CUSTOMER
    WHERE MARITAL_STATUS = 'married' AND REGION
IN ('central','west');
```

Những chỉ mục bitmap có thể xử lý truy vấn này hiệu quả nhất chỉ bằng cách đếm số chỉ mục trong bitmap kết quả, như được minh họa ở Hình 8-8. Để nhận dạng những khách hàng cụ thể đáp ứng tiêu chuẩn ấy, bitmap kết quả sẽ được sử dụng để truy xuất bảng.

Các chỉ mục bitmap và các giá trị rỗng

Các chỉ mục bitmap gồm các hàng có những giá trị NULL, không giống như phần lớn các kiểu chỉ mục khác. Lập chỉ mục rỗng có thể hữu dụng đối với các kiểu câu lệnh SQL, chẳng hạn như những truy vấn với hàm nhóm COUNT.

Hình 8-8 Thực hiện một truy vấn sử dụng các chỉ mục bitmap

Trạng thái = "đã kết hôn"		Vùng = 'trung tâm'		Vùng = 'phía tây'						
0		0		0		0		0		0
1		1		0		1		1		1
1	AND	0	OR	1	=	1	AND	1	=	1
0		0		1		0		1		0
0		1		0		0		1		0
1		1		0		1		1		1

Thí dụ 1:

SELECT COUNT(*) FROM EMP;

Chỉ mục bitmap bất kỳ có thể được sử dụng cho truy vấn này vì tất cả các hàng có chỉ mục, gồm những chỉ mục có dữ liệu NULL. Nếu các giá trị rỗng không có chỉ mục, thì bộ tối ưu hóa sẽ chỉ có thể sử dụng những chỉ mục trên các cột với các hạn định NOT NULL.

Thí dụ 2:

SELECT COUNT(*) FROM EMP WHERE COMM IS NULL;

Truy vấn này có thể được tối ưu hóa với một chỉ mục bitmap trên COMM.

Thí dụ 3:

```
SELECT COUNT(*) FROM CUSTOMER
    WHERE GENDER = 'M' AND STATE != 'CA';
```

Truy vấn này có thể được trả lời bằng cách tìm bitmap đối với GENDER = 'M' và trừ đi bitmap đối với STATE = 'CA'. Nếu STATE có thể có những giá trị rỗng (nghĩa là nếu nó không có hạn định NOT NULL), thì các bitmap đối với STATE = 'NULL' cũng phải được trừ đi kết quả đó.

Các chỉ mục bitmap trên các bảng được phân hoạch

Giống như những chỉ mục khác, bạn có thể tạo những chỉ mục bitmap trên các bảng phân hoạch. Hạn chế duy nhất là những chỉ mục bitmap phải là cục bộ đối với bảng được phân hoạch - chúng không thể là những chỉ mục toàn cục. (Các chỉ mục bitmap chỉ được hỗ trợ trên các bảng không được phân hoạch).

Để có thêm thông tin về các bảng được phân hoạch và các mô tả về những chỉ mục toàn cục và cục bộ, xem "Các Bảng Được Phân Hoạch Và Các Chỉ Mục".

Thông tin bổ sung:
Oracle* Tuning về cách sử dụng những chỉ mục bitmap.

Các bảng có tổ chức chỉ mục

Một bảng có tổ chức chỉ mục khác với một bảng thường lệ trong đó dữ liệu đối với bảng ấy được lưu giữ trong chỉ mục được kết hợp của nó. Các thay đổi đối với dữ liệu bảng, chẳng hạn như bổ sung các hàng mới, cập nhật các hàng hoặc xóa các hàng, chỉ dẫn đến cập nhật chỉ mục ấy.

Bảng có tổ chức chỉ mục giống như một bảng thường lệ với một chỉ mục trên một hoặc nhiều cột của nó, nhưng thay vì duy trì hai bộ nhớ tách rời cho bảng ấy và chỉ mục B*-tree, hệ thống CSDL chỉ duy trì một một chỉ mục B*-tree đơn có cả giá trị khóa được mã hóa và các giá trị cột được kết hợp đối với hàng tương ứng. Thay vì có hàng ROWID làm phần tử thứ hai của mục nhập chỉ mục, hàng dữ liệu thực được lưu trữ trong chỉ mục B*-tree. các hàng dữ liệu được thiết lập trên khóa chính đối với bảng ấy và mỗi mục nhập chỉ mục B*-tree có <primary_key_value, non_primary_key_column_values>.

Các bảng có tổ chức chỉ mục thích hợp với việc truy xuất dữ liệu bằng khóa chính hoặc bằng khóa bất kỳ là một tiền tố hợp lệ của khóa chính. Không có sao chép những giá trị khóa vì chỉ những giá trị cột không có khóa được lưu trữ với khóa ấy.

Những ứng dụng điều tác bảng có tổ chức chỉ mục giống như một bảng thường lệ, sử dụng các câu lệnh SQL. Tuy nhiên, hệ thống CSDL thực hiện tất cả các phép toán bằng cách điều tác chỉ mục B*-tree tương ứng.

Bảng 8-4 tóm tắt những khác biệt giữa các bảng có tổ chức chỉ mục và các bảng thường lệ.

Bảng 8-4 So sánh các bảng có tổ chức chỉ mục với các bảng thường lệ

Bảng thường lệ	Bảng có tổ chức chỉ mục
ROWID nhận dạng duy nhất một hàng; khóa chính có thể được ấn định tùy chọn	Khóa chính nhận dạng duy nhất một hàng; khóa chính phải được xác định
Cột ROWID ngầm định; cho phép thiết lập các chỉ mục thứ cấp vật lý	Cột ROWID không ngầm định; không thể có các chỉ mục thứ cấp vật lý
ROWID dựa trên truy xuất	Khóa chính dựa trên truy xuất
Duyệt liên tiếp cho trở lại tất cả các hàng	Duyệt hoàn toàn chỉ mục cho trở lại tất cả các hàng theo thứ tự khóa chính
Hạn định UNIQUE và bộ kích khởi cho phép	Hạn định UNIQUE không cho phép nhưng các bộ kích khởi được cho phép
Bảng có thể được lưu trữ trong một cụm liên kết có các bảng khác	Bảng có tổ chức chỉ mục không thể lưu trữ trong một cụm liên kết

Phân phối, sao chép dữ liệu và phân hoạch hỗ trợ	Phân phối, sao chép dữ liệu và phân hoạch không được hỗ trợ

Thông tin bổ sung :
Xem Hướng dẫn của Oracle8 Administrator về cách tạo và duy trì các bảng có tổ chức chỉ mục.

Các lợi ích của các bảng có tổ chức chỉ mục

Vì các hàng được lưu trữ bằng chỉ mục, các bảng có tổ chức chỉ mục cung cấp truy xuất dữ liệu bảng dựa trên khóa nhanh hơn đối với các truy vấn dữ liệu liên quan đến tìm kiếm so khớp và/hoặc phạm vi chính xác. Các đòi hỏi của bộ nhớ ít hơn vì các cột khóa được phải sao chép trong bảng và chỉ mục. Hàng dữ liệu được lưu trữ bằng khóa chỉ chứa những giá trị cột không khóa. Ngoài ra, đặt hàng dữ liệu bằng khóa loại trừ bộ nhớ bổ sung đòi hỏi các ROWID để liên kết những giá trị khóa với các hàng tương ứng trong trường hợp một chỉ mục cho một bảng thường lệ.

Các bảng có tổ chức chỉ mục với Row Overflow Area

Chỉ mục B*-tree thường rất nhỏ vì chúng chỉ gồm cặp <key, ROWID>. Tuy nhiên, trong các bảng có tổ chức chỉ mục, những mục nhập của chỉ mục B*-tree có thể rất lớn vì chúng gồm cặp <key, non key column values>. Nếu mục nhập chỉ mục rất lớn thì các nút lá có thể kết thúc lưu trữ bằng một hàng hoặc phần đoạn hàng do đó hủy tính chất liên kết server mật độ cao của chỉ mục B*-tree.

Oracle cung cấp một mệnh đề Row Overflow Area để khắc phục vấn đề này. Bạn có thể xác định không gian bảng

tràn trên cùng như một giá trị ngưỡng. Giá trị ngưỡng được xác định bằng phần trăm kích thước của khối dữ liệu.

Nếu kích thước hàng lớn hơn giá trị ngước xác định, thì những giá trị cột không có khóa đối với hàng ấy vượt ngưỡng được lưu trữ trong không gian bảng tràn trên được xác định. Trong trường hợp như vậy, mục nhập chỉ mục chứa một cặp <key, rowhead>, nơi đầu hàng chứa phần bắt đầu của phần còn lại của các cột. Nó giống như một phân đoạn hàng thường lệ, ngoại trừ nó trỏ một phân đoạn hàng tràn trên chứa những giá trị cột còn lại.

Các ứng dụng liên quan các bảng có tổ chức chỉ mục

Các bảng có tổ chức chỉ mục rất hữu dụng đối với các kiểu ứng dụng sau:

Các ứng dụng truy hồi thông tin (IR)
Các ứng dụng không gian
Các ứng dụng OLAP

Các ứng dụng truy hồi thông tin

Ứng dụng truy hồi thông tin hỗ trợ các tìm kiếm dựa vào nội dụng trên các tập hợp tài liệu. Để cung cấp một khả năng như vậy, các ứng dụng IR duy trì một chỉ mục đảo ngược đối với tập hợp tài liệu ấy. Chỉ mục đảo ngược có các mục nhập tiêu biểu với dạng <token, document_id, occurrence_data> đối với mỗi từ riêng biệt trong một tài liệu. Ứng dụng ấy thực hiện một tìm kiếm dựa vào nội dung bằng cách duyệt chỉ mục đảo ngược đó để tìm các thẻ bài quan tâm.

Bạn có thể xác định một bảng thường lệ với mô hình có chỉ mục đảo ngược. Để tăng tốc độ truy hồi dữ liệu từ bảng đó, bạn còn có thể xác định một chỉ mục trên cột tương ứng với thẻ bài ấy. Tuy nhiên, giản đồ này có các sai sót sau:

Truy hồi dữ liệu diễn biến từ chỉ mục đảo ngược bằng cách sử dụng chỉ mục nhận một truy nạp dựa trên ROWID phụ trên mỗi hàng. Truy vấn IR dựa vào nội dung tiêu biểu để truy nạp tất cả những mục nhập chỉ mục đảo ngược đối với các thuật ngữ truy vấn xác định. Vì các bản sao chép dữ liệu tiêu chuẩn hơn là sự ngoại lệ trong các ứng dụng IR, một thuật ngữ truy vấn đơn có thể chứa hàng ngàn bản sao dữ liệu. Do đó, một truy nạp dựa vào ROWID trên mỗi phần điều hành chung hàng có thể rất quan trọng, nó ảnh hưởng rất nhiều đến hiệu suất tìm kiếm của IR.

Sao chép dữ liệu của cột (thẻ bài) có khóa trong bảng và trong chỉ mục dẫn đến lãng phí bộ nhớ. Vì chỉ mục đảo ngược có thể rất lớn, nhu cầu bộ nhớ sẽ không chấp nhận được.

Trong một số trường hợp, hiệu suất truy hồi có thể được tăng lên nhờ vào xác định chỉ mục ghép nối trên nhiều cột của bảng chỉ mục đảo ngược. Chỉ mục ghép nối cho phép truy hồi có tổ chức chỉ mục khi dữ liệu diễn biến không cần thiết (nghĩa là, đối với các truy vấn Boolean). Trong các trường hợp như vậy, phải tránh những truy nạp ROWID của các bản ghi bảng đảo ngược. Khi truy vấn liên quan đến một thuộc tính gần (thí dụ, từ ngữ "Oracle Corporation"), tiếp cận chỉ mục ghép nối vẫn đòi hỏi bảng chỉ mục đảo ngược để được truy xuất. Hơn nữa, thiết lập và duy trì một chỉ mục ghép nối mất nhiều thời gian hơn sử dụng một chỉ mục cột đơn trên thẻ bài ấy. Ngoài ra, phần điều hành chung cao hơn như nhiều cột của khóa (thẻ bài) được sao chép trong bảng và chỉ mục.

Sử dụng một bảng có tổ chức chỉ mục để tạo mô hình một chỉ mục đảo ngược khắc phục các vấn đề được mô tả ở trên:

Vì hàng dữ liệu được lưu trữ cùng với khóa, truy hồi dữ liệu diễn biến đối với một chỉ mục đảo ngược liên quan qua chỉ mục và nhận các hàng dữ liệu từ các nút là thích hợp.

Chỉ những giá trị cột không có khóa được lưu trữ bằng khóa trong chỉ mục ấy. Do đó, không có sao chép dữ liệu. Ngoài ra, điều này tránh phải có phần điều hành chung ROWID bổ sung nếu một chỉ mục được duy trì trên bảng thường lệ.

Hơn nữa, vì các bảng có tổ chức chỉ mục có thể hiển thị đối với các ứng dụng, chúng thích hợp để hỗ trợ lập chỉ mục liên quan nơi ứng dụng và CSDL liên kết quản lý những chỉ mục ứng dụng xác định.

Các ứng dụng không gian

Các ứng dụng không gian có thể khai thác được các bảng có tổ chức chỉ mục khi chúng sử dụng biểu mẫu chỉ mục đảo ngược nào đó để duy trì những chỉ mục ứng dụng xác định.

Các ứng dụng không gian duy trì các chỉ mục đảo ngược để xử lý những truy vấn không gian. Thí dụ, một chỉ mục không gian cho các đối tượng cư trú trong tập hợp của các hình có thể được tạo mô hình như một chỉ mục đảo ngược nơi mỗi mục nhập thuộc biểu mẫu ấy:
<grid id, spatial object id, spatial object data>

Các bảng có tổ chức chỉ mục thích hợp để lập mô hình những chỉ mục đảo ngược như vậy vì chúng cung cấp hiệu suất truy hồi cần thiết trong khi tiết kiệm được bộ nhớ tối đa.

Các ứng dụng OLAP

Các ứng dụng xử lý phân tích trực tuyến (OLAP) thường điều tác các khối đa kích thước. Để cho phép truy hồi nhanh các thành phần của các khối đa kích thước, chung duy trì một chỉ mục đảo ngược để anh xạ một tập hợp những giá trị kích thước vào tạp hợp các trang. Một mục nhập trong chỉ mục đảo ngược thuộc biểu mẫu:
<dimension_value, list_of_pages>

Chỉ mục đảo ngược được những ứng dụng OLAP duy trì có thể dễ dàng được tạo mô hình như một bảng có tổ chức chỉ mục.

Các cụm liên kết

Các cụm liên kết là một phương thức lưu trữ dữ liệu bảng tùy chọn. Một cụm liên kết là một nhóm các bảng dùng chung cùng những khối dữ liệu vì chúng dùng chung các cột thông thường và thường được sử dụng cùng với nhau.
Thí dụ, bảng EMP và DEPT dùng chung cột DEPTNO. Khi bạn tạo cụm liên kết các bảng EMP và DEPT (Hình 8-9, "Dữ liệu bảng có cụm liên kết), Oracle lưu trữ vật lý tất cả các hàng cho mỗi bộ phận từ các bảng EMP và DEPT trong cùng các khối dữ liệu.

Vì các cụm liên kết được lưu trữ các hàng liên quan của các bảng khác nhau cùng trong các khối dữ liệu giống nhau, sử dụng thích hợp các cụm liên kết cung ứng hai lợi ích chính:

Đĩa I/O giảm và thời gian truy xuất tăng lên cho các liên kết của các bảng có cụm liên kết.

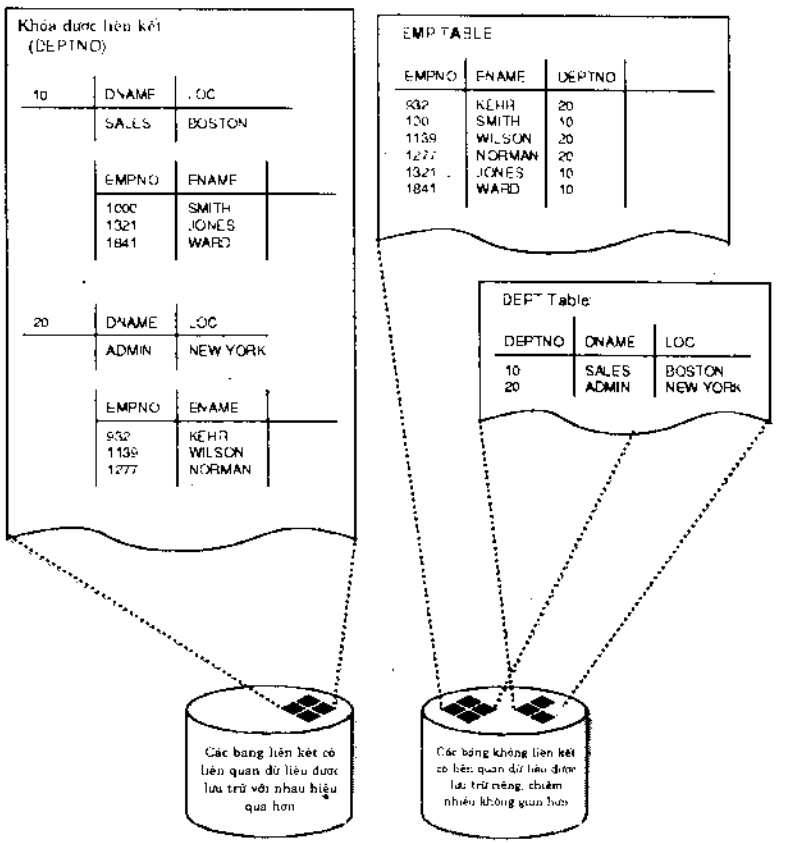
Trong một cụm liên kết, giá trị khóa của cụm liên kết là giá trị của các cột có khóa cụm liên kết đối với một hàng riêng biệt. Mỗi giá trị khóa cụm liên kết chỉ được lưu trữ một lần mỗi giá trị trong cụm liên kết và chỉ mục cụm liên kết, bất kể bao nhiêu hàng của các bảng khác nhau chứa giá trị ấy.

Do đó, có thể ít đòi hỏi bộ nhớ để lưu trữ bảng liên quan và dữ liệu chỉ mục trong một cụm liên kết hơn trong định dạng bảng không có cụm liên kết. Thí dụ, ở Hình 8-9 lưu ý mỗi khóa cụm liên kết (mỗi DEPTNO) được lưu trữ chỉ một lần đối với nhiều hàng như thế để chứa cùng giá trị trong cả bảng EMP và DEPT.

Các nhận xét hiệu suất

Các cụm liên kết có thể làm giảm hiệu suất của các câu lệnh INSERT khi so sánh với lưu trữ một bảng tách riêng với chỉ mục của riêng nó. Bất lợi này liên quan tới việc sử dụng không gian và số khối dữ liệu phải đến để duyệt một bảng; vì nhiều bảng có dữ liệu trong mỗi khối dữ liệu, nhiều khối dữ liệu phải được sử dụng để lưu trữ một bảng có cụm liên kết hơn nếu bảng đó đã được lưu trữ không có cụm liên kết.

Hình 8-9 Dữ liệu bảng có cụm liên kết



Để nhận dạng dữ liệu sẽ lưu trữ tốt hơn được lưu trữ bằng biểu mẫu có cụm liên kết so với không có cụm liên kết, tìm các bảng có liên quan bằng các hạn định tính nguyên vẹn tham chiếu và các bảng thực hiện được truy xuất cùng với nhau bằng liên kết. Nếu bạn tạo cụm liên kết các bảng trên các cột được sử dụng để liên kết dữ liệu bảng, bạn làm giảm số khối dữ liệu phải được truy xuất để xử lý truy vấn ấy; tất cả còn cần thiết cho một liên kết trên một khóa cụm liên kết trong cùng khối dữ

liệu. Do đó, hiệu suất đối với các liên kết tăng lên. Tương tự, tạo cụm liên kết một bảng riêng biệt có thể hữu dụng. Thí dụ, bảng EMP có thể được tạo cụm liên kết trên cột DEPTNO để tạo cụm liên kết các hàng cho những nhân viên trong cùng bộ phận. Điều này sẽ khai thác được nếu các ứng dụng thường xử lý các hàng theo từng bộ phận.

Giống như các chỉ mục, các cụm liên kết không ảnh hưởng đến thiết kế ứng dụng. Sự tồn tại của một cụm liên kết thông suốt với những người dùng và các ứng dụng. Bạn truy xuất dữ liệu được lưu trữ trong một bảng có cụm liên kết bằng SQL giống như dữ liệu được lưu trữ trong một bảng không có cụm liên kết..

Thông tin bổ sung :
Để có thêm thông tin về các quan hệ hiệu suất bằng cách sử dụng các cụm liên kết, xem Oracle8 Tuning.

Định dạng của các khối dữ liệu có cụm liên kết

Nói chung, các khối dữ liệu có cụm liên kết có định dạng giống với các khối dữ liệu không có cụm liên kết với bổ sung dữ liệu trong thư mục bảng. Tuy nhiên, Oracle lưu trữ tất cả các hàng dùng chung cùng giá trị khóa cụm trong khối dữ liệu tương tự.

Khi bạn tạo một cụm, xác định số lượng không gian trung bình cần thiết để lưu trữ tất cả các hàng cho một giá trị khóa cụm liên kết bằng cách sử dụng tham số SIZE của lệnh CREATE CLUSTER. SIZE xác định số khóa cụm liên kết tối đa có thể được lưu trữ trên mỗi khối dữ liệu.

Thí dụ, nếu mỗi khối dữ liệu có 1700 byte của không gian khả dụng và kích thước khóa cụm liên kết được ấn định là 500 byte, mỗi khối dữ liệu có thể lưu trữ các hàng cho ba khóa cụm liên kết tiềm năng. Nếu SIZE lớn hơn số lượng không gian khả dụng trên mỗi khối dữ liệu, thì mỗi khối được lưu giữ các hàng cho chỉ một giá trị khóa cụm liên kết.

Dù số giá trị khóa cụm liên kết tối đa trên mỗi khối dữ liệu được SIZE ấn định, nhưng Oracle thực sự không dự trữ không gian cho mỗi giá trị khóa cụm liên kết và cũng không bảo đảm nó số khóa cụm liên kết được gán vào khối dữ liệu. Thí dụ, nếu SIZE xác định ba giá trị khóa cụm liên kết được phép trên mỗi khối dữ liệu, điều này không ngăn chặn số hàng cho một giá trị khóa cụm liên kết không sử dụng tất cả không gian khả dụng trong khối ấy. Nếu nhiều hàng tồn tại cho một khóa định trước hơn có thể thích hợp với một khối đơn, thì khối ấy được lập chuỗi xích theo nhu cầu.

Một giá trị khóa cụm liên kết chỉ được lưu trữ một lần trọng khối dữ liệu.

Khóa cụm liên kết

Khóa cụm liên kết là cột hoặc nhóm các cột mà các bảng có cụm liên kết thường dùng. Bạn ấn định số cột của khóa cụm liên kết khi tạo cụm liên kết ấy. Bạn tiếp tục ấn định cùng số cột khi tạo mọi bảng được bổ sung vào cụm liên kết.

Đối với mỗi cụm được ấn định làm thành phần của khóa cụm liên kết (khi tạo cụm liên kết ấy), mọi bảng được tạo trong cụm liên kết ấy phải có một cột so khớp với kích thước và kiểu cột trong khóa cụm liên kết. Không

hơn 16 cột có thể tạo thành khóa cụm liên kết và giá trị khóa cụm liên kết không thể vượt một phân nửa (trừ đi phần điều hành chung nào đó) không gian dữ liệu khả dụng trong một khối dữ liệu. Khóa cụm liên kết không thể có cột LONG hoặc LONG RAW.

Bạn có thể cập nhật những giá trị dữ liệu trong các cột có cụm liên kết của một bảng. Tuy nhiên, vì sự bố trí dữ liệu phụ thuộc vào khóa cụm liên kết, thay đổi khóa cụm liên kết đối với một hàng có thể làm cho Oracle định vị lại hàng ấy về vật lý. Do đó, các cụm thường được cập nhật không là các ứng viên tốt cho khóa cụm liên kết.

Chỉ mục cụm liên kết

Bạn phải tạo một chỉ mục trên các cột khóa cụm liên kết sau khi bạn đã tạo một cụm liên kết. Một chỉ mục cụm liên kết là một chỉ mục được xác định riêng cho một cụm liên kết. Một chỉ mục như thế có các mục nhập cho mỗi giá trị khóa cụm liên kết.

Để định vị một hàng trong cụm liên kết, chỉ mục cụm liên kết được sử dụng để tìm giá trị khóa cụm liên kết trỏ khối dữ liệu được kết hợp với giá trị khóa cụm liên kết đó. Do đó, Oracle truy xuất một hàng cho trước với ít nhất hai I/O (có thể hơn tùy theo số mức phải được đi qua trong chỉ mục).

Bạn phải tạo một chỉ mục cụm liên kết trước khi bạn có thể thực hiện bất kỳ các câu lệnh DML (kể cả các câu lệnh INSERT và SELECT) dựa trên các bảng có cụm liên kết. Do đó, bạn không thể nạp dữ liệu vào bảng có cụm liên kết cho tới khi bạn tạo chỉ mục cụm liên kết ấy.

Giống như một chỉ mục bảng, Oracle lưu trữ chỉ mục cụm liên kết trong một đoạn chỉ mục. Do đó, bạn có thể đặt một cụm liên kết trong một không gian bảng và chỉ mục cụm liên kết trong một không gian bảng khác.

Một chỉ mục cụm liên kết không giống như một chỉ mục bảng trong các phương thức sau:
Các khóa hoàn toàn rỗng có một mục nhập trong chỉ mục cụm liên kết.
Các mục nhập chỉ mục trỏ khối dữ liệu đầu tiên trong chuỗi xích cho một giá trị khóa cụm liên kết cho trước.
Một chỉ mục cụm liên kết chứa một mục nhập trên mỗi giá trị khóa cụm liên kết, hơn là một mục nhập trên mỗi hàng cụm liên kết.
Không có chỉ mục bảng không ảnh hưởng đến người dùng, nhưng dữ liệu có cụm liên kết không thể được truy xuất trừ khi có một chỉ mục cụm liên kết.

Nếu bạn giảm chỉ mục cụm liên kết, dữ liệu trong cụm liên kết ấy vẫn duy trì nhưng thành không sử dụng được cho tới khi bạn tạo một chỉ mục cụm liên kết. Bạn có thể muốn giảm một chỉ mục cụm liên kết để di chuyển chỉ mục cụm liên kết tới một không gian bảng khác hoặc để thay đổi những đặc điểm bộ nhớ của nó; tuy nhiên, bạn phải tạo lại chỉ mục của cụm liên kết để cho phép truy xuất dữ liệu trong cụm liên kết ấy.

Các cụm liên kết trị số

Gán trị số là một phương thức tùy chọn lưu trữ dữ liệu để tăng hiệu suất truy hồi dữ liệu. Để sử dụng gán trị số, bạn tạo một cụm liên kết trị số và nạp các bảng vào cụm liên kết ấy. Oracle lưu trữ các hàng của một bảng trong một cụm liên kết trị số về vật lý và truy hồi chung theo các kết quả của hàm trị số.

Oracle sử dụng một hàm trị số để phát sinh sự phân phối những giá trị số, được gọi là các trị số được dựa vào các giá trị khóa cụm liên kết. Khóa của một cụm liên kết trị số (giống như khóa của một cụm liên kết chỉ mục) có thể là một cột đơn hoặc khóa tổng hợp (nhiều khóa cột). Để tìm hoặc lưu trữ một hàng trong cụm liên kết trị số, Oracle áp dụng hàm trị số dịch vụ giá trị khóa cụm liên kết của hàng ấy; trị số kết quả tương ứng với khối dữ liệu trong cụm liên kết mà Oracle sau đó đọc hoặc ghi thay cho câu lệnh được đưa ra.

Một cụm liên kết trị số là một thay thế đối với bảng không có liên kết cụm bằng một chỉ mục hoặc một cụm liên kết chỉ mục. Với một bảng có chỉ mục hoặc cụm liên kết chỉ mục, Oracle định vị các hàng trong bảng sử dụng những giá trị khóa mà Oracle lưu trữ trong một chỉ mục tách riêng.

Để tìm hoặc lưu trữ một hàng trong một bảng có chỉ mục hoặc cụm liên kết, ít nhất hai I/O phải được thực hiện (nhưng thường nhiều hơn): một hoặc nhiều I/O để tìm hoặc lưu trữ giá trị khóa trong chỉ mục và một I/O đọc hoặc giữa hàng ấy vào bảng trỏ cụm liên kết. Trái lại, Oracle sử dụng một hàm trị số để định vị hàng trong cụm liên kết trị số (không cần có I/O). Vì vậy, ít nhất một điều hoạt I/O cần phải đọc hoặc ghi hàng vào cụm liên kết trị số.

Dữ liệu được lưu trữ trong một cụm liên kết trị số như thế nào

Cụm liên kết trị số lưu trữ các hàng liên quan cùng với nhau trong cùng các khối dữ liệu. Các hàng trong một cụm liên kết trị số được lưu trữ cùng với nhau dựa trên trị số của chung.

Lưu ý:
Trái lại, một cụm liên kết chỉ mục lưu trữ các hàng liên quan của các bảng có cụm liên kết cùng với nhau dựa trên mỗi giá trị khóa cụm liên kết của hàng ấy.

Oracle phân bổ một số lượng bộ nhớ ban đầu cho đoạn dữ liệu của cụm liên kết. Oracle căn cứ số lượng của bộ nhớ ban đầu được phân bổ cho một cụm liên kết trị số trên số và kích thước trung bình được đoán trước về các hàng của khóa trị số trong cụm liên kết.

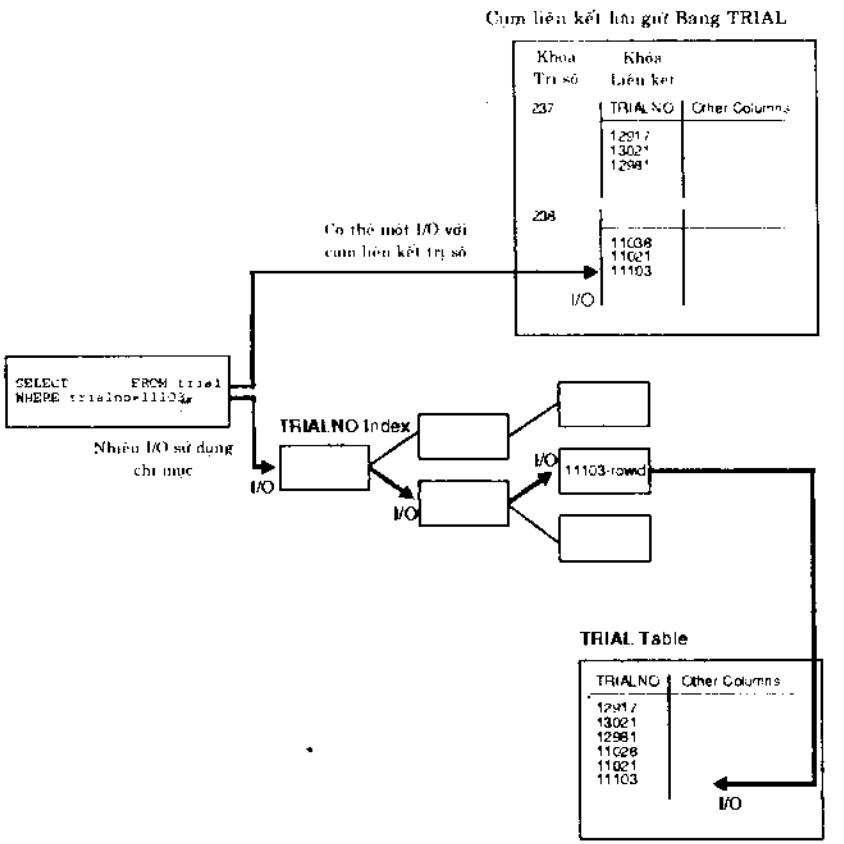
Hình 8-10 minh họa truy hồi dữ liệu cho một bảng trong một cụm liên kết trị số cũng như một bảng có chỉ mục. Các phần sau giải thích thêm về những điều hoạt bên trong của bộ nhớ cụm liên kết trị số.

Các giá trị khóa trị số

Để tìm hoặc lưu trữ một hàng trong cụm liên kết trị số, Oracle áp dụng hàm trị số vào giá trị cụm liên kết của hàng. Trị số kết quả tương ứng với khối dữ liệu trong cụm liên kết mà Oracle sau đó đọc hoặc ghi thay cho một câu lệnh được đưa ra. Số giá trị gán trị số cho một cụm trị số được tạo cố định và được tham số HASHKEYS của lệnh CREATE CLUSTER.

Giá trị của HASHKEYS giới hạn số những trị số cửa sổ được hàm trị số phát sinh được sử dụng cho cụm liên kết ấy. Oracle làm tròn số bạn ấn định cho HASHKEYS thành số nguyên tố gần nhất. thí dụ, xác lập HASHKEYS thành 100 nghĩa là đối với giá trị khóa cụm liên kết bất kỳ mà hàm trị số phát sinh những giá trị từ 0 tới 100 (sẽ có 101 trị số).

Hình 8-10 Gán trị số so với lập chỉ mục: bộ nhớ dữ liệu và thông tin



Do đó, sự phân phối các hàng trong cụm liên kết trị số được tập hợp giá trị trực tiếp điều khiển cho tham số HASHKEYS. Với số khóa trị số lớn hơn đối với số hàng cho trước, thì có khả năng va chạm (hai giá trị khóa cụm có cùng trị số) giảm xuống. Giảm tối đa các va chạm rất quan trọng vì các khối tràn trên (vì I/O phụ) có thể cần lưu trữ các hàng với các trị số va chạm.

Số khóa trị số tối đa được gán trên mỗi khối dữ liệu được tham số SIZE xác định của lệnh CREATE CLUSTER. SIZE là một dự đoán của tổng số không gian bằng byte cần thiết để dữ liệu số hàng trung bình được kết hợp với mỗi trị số. Thí dụ, nếu không gian trống khả dụng trên mỗi khối dữ liệu bằng 1700 byte và SIZE được xác lập 500 byte, ba khóa trị số được gán trên mỗi khối dữ liệu.

> Lưu ý:
> Tầm quan trọng về tham số SIZE của các cụm trị số tương tự với tham số SIZE đối với các cụm liên kết chỉ mục. Tuy nhiên, với các cụm chỉ mục, SIZE áp dụng vào các hàng có cùng giá trị khóa cụm liên kết thay vì cùng trị số.

Dù số tối đa của những giá trị khóa trị số trên mỗi khối dữ liệu được SIZE xác định, Oracle không thực sự dự trữ không gian cho mỗi giá trị khóa trị số trong khối dữ liệu ấy. Thí dụ, nếu SIZE xác định ba giá trị khóa trị số được phép trên mỗi khối, thì điều này không ngăn cản các hàng đối với một giá trị khóa trị số không dùng tất cả không gian khả dụng trong khối ấy. Nếu có nhiều hàng cho một giá trị khóa trị số hơn có thể thích hợp với một khối đơn, thì khối dữ liệu ấy được lập chuỗi xích theo nhu cầu.

Lưu ý mỗi trị số không được lưu trữ làm thành phần của hàng; tuy nhiên, giá trị khóa cụm liên kết đối với mỗi hàng được lưu trữ. Do đó, khi xác định giá trị thích hợp cho SIZE, thì giá trị khóa cụm liên kết phải có đối với mọi hàng được lưu trữ.

Các hàm trị số

Hàm trị số là hàm được áp dụng vào giá trị khóa cụm liên kết để cho trở lại một giá trị bằng trị số. Sau đó Oracle sử dụng giá trị bằng trị số để định vị hàng trong khối dữ liệu thích hợp của cụm liên kết trị số. Tác vụ của một hàm trị số để cung cấp phân phối tối đa những hàng giữa những trị số khả dụng của cụm liên kết. Để đạt được mục đích này, hàm trị số phải giảm tối đa số lần va chạm.

Sử dụng hàm trị số bên trong của Oracle

Khi bạn tạo một cụm liên kết, bạn có thể sử dụng hàm trị số bên trong của Oracle hoặc bỏ qua sử dụng của hàm này. Hàm trị số bên trong cho phép khóa cụm liên kết để là một cột đơn hoặc khóa tổng hợp.

Hơn nữa, khóa cụm liên kết có thể gồm các cột của kiểu dữ liệu bất kỳ (ngoại trừ LONG và LONG RAW). Hàm trị số bên trong cung ứng phân phối đầy đủ những giá trị khóa cụm liên kết trong những khóa trị số khả dụng tạo ra tối đa va chạm đối với kiểu khóa cụm liên kết bất kỳ.

Ấn định khóa cụm liên kết bằng hàm trị số

Trong các trường hợp nơi khóa cụm liên kết là một định danh duy nhất được phân phối cùng kiểu trên phạm vi của nó, bạn có thể muốn bỏ qua hàm trị số bên trong và đơn giản hóa cột gán trị số.

Thay vì sử dụng hàm trị số bên trong để phát sinh một giá trị bằng trị số, Oracle kiểm giá trị khóa cụm liên kết. Nếu giá trị khóa cụm liên kết nhỏ hơn HASHKEYS, thì giá trị bằng trị số là giá trị khóa cụm liên kết ấy;

tuy nhiên, nếu giá trị khóa cụm liên kết bằng hoặc lớn hơn HASHKEYS, Oracle chia giá trị khóa cụm liên kết bằng số được ấn định cho HASHKEYS và số dư bằng giá trị số; nghĩa là giá trị số là giá trị khóa cụm, số các khóa trị số.

Sử dụng tham số HASH IS của lệnh CREATE CLUSTER để ấn định cột khóa cụm liên kết nếu những giá trị khóa cụm liên kết được phân phối đều toàn bộ cụm liên kết. Khóa cụm liên kết phải gồm một cột đơn chỉ chứa những số nguyên. Nếu hàm trị số bên trong được bỏ qua và không cung cấp giá trị khóa cụm liên kết không có số nguyên, thì điều hoạt (câu lệnh INSERT hoặc UPDATE) được hồi phục trạng thái cũ và cho trở lại một lỗi.

Ấn định một hàm trị số người dùng xác định

Bạn còn có thể ấn định biểu thức SQL bất kỳ bằng hàm trị số cho một cụm liên kết trị số. Nếu những giá trị khóa cụm liên kết không được phân phối đều trong cụm liên kết, bạn phải xét để tạo hàm trị số của riêng mình để phân phối các hàng cụm liên kết đầy đủ trong những giá trị số ấy.

Thí dụ, nếu bạn có một cụm liên kết trị số chứa thông tin nhân viên và khóa cụm liên kết là mã vùng địa chỉ của nhân viên, có thể nhiều nhân viên sẽ gán trị số vào giá trị số giống nhau. Để giảm bớt vấn đề này, bạn có thể đặt biểu thức sau trong mệnh đề HASH IS của lệnh CREATE CLUSTER command:

```
MOD((emp.home_area_code + emp.home_prefix +
emp.home_suffix), 101)
```

Biểu thức dùng cột mã vùng và bổ sung tiền tố điện thoại và các cột hậu tố được chia cho số giá trị số (trong

trường hợp này là 101) và sau đó sử dụng số dư làm giá trị số. Kết quả là các hàng cụm liên kết được phân phối đều hơn trong những giá trị số khác nhau.

Sự phân bổ không gian cho một cụm trị số

Giống như những kiểu đoạn khác, sự phân bổ những khối lưu riêng khi tạo một cụm liên kết trị số được những tham số INITIAL, NEXT và MINEXTENTS của mệnh đề STORAGE kiểm soát. Tuy nhiên, với những cụm liên kết trị số, một phần bắt đầu của không gian được gọi là bảng trị số được phân bổ vào lúc tạo để tất cả các khóa trị số của cụm liên kết có thể được ánh xạ, với tổng không gian bằng SIZE * HASHKEYS. Do đó, phân bổ không bắt đầu cho một cụm liên kết trị số cũng phụ thuộc vào những giá trị của SIZE và HASHKEYS. Giá trị lớn hơn của (SIZE*HASHKEYS) và giá trị được ấn định bằng mệnh đề STORAGE (INITIAL, NEXT, v.v...) được sử dụng.

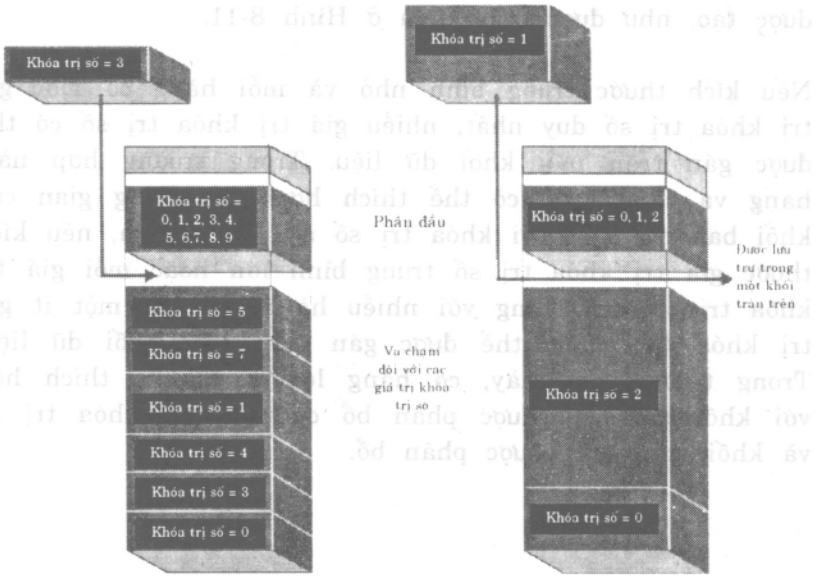
Không gian được phân bổ liên tiếp vào cụm liên kết trị số được sử dụng để lưu trữ các tràn trên của các hàng từ các khối dữ liệu đã đầy. Thí dụ, giả sử khối dữ liệu ban đầu cho một khóa trị số cho trước đầy. Người dùng chèn một hàng vào bảng có cụm liên kết sao cho nó được lưu trữ trong một khối dữ liệu đầy; do đó, hàng ấy không thể được chèn vào khối ban đầu (khối ban đầu) được phân bổ cho khóa trị số. Thay vào đó, hàng ấy được chèn vào một khối tràn trên được lập chuỗi xích với khối ban đầu của khóa trị số.

Các va chạm thông thường có thể hoặc không thể dẫn đến nhiều khối tràn trên hơn trong một cụm liên kết trị số (do đó làm giảm hiệu suất truy hồi). Nếu va chạm xảy ra và không có không gian trong khối ban đầu được phân bổ cho khóa trị số, một khối tràn trên phải được phân

bổ để lưu giữ hàng mới. Khả năng này xảy ra phụ thuộc nhiều vào kích thước trung bình của mỗi giá trị khóa trị số và dữ liệu tương ứng được ấn định khi cụm liên kết được tạo, như được minh họa ở Hình 8-11.

Nếu kích thước trung bình nhỏ và mỗi hàng có một giá trị khóa trị số duy nhất, nhiều giá trị khóa trị số có thể được gán trên mỗi khối dữ liệu. Trong trường hợp này, hàng va chạm nhỏ có thể thích hợp với không gian của khối ban đầu đối với khóa trị số ấy. Tuy nhiên, nếu kích thước giá trị khóa trị số trung bình lớn hoặc mỗi giá trị khóa trị số tương ứng với nhiều hàng, thì chỉ một ít giá trị khóa trị số có thể được gán trên mỗi khối dữ liệu. Trong trường hợp này, có hàng lớn sẽ không thích hợp với khối ban đầu được phân bổ cho giá trị khóa trị số và khối tràn trên được phân bổ.

Hình 8-11 Các va chạm và các khối tràn trên trong một cụm liên kết trị số



Chương 9

Các bảng phân hoạch và các chỉ mục

Chương này mô tả các bảng phân hoạch và các chỉ mục và giải thích một số nhận xét quản lý để phân hoạch. Gồm các chủ đề sau:

- **Phân hoạch để làm gì?**
- **Các ưu điểm của phân hoạch**
- **Mô hình phân hoạch cơ bản**
- **Các nguyên tắc phân hoạch các bảng và các chỉ mục**
- **Các khóa phân hoạch DML**
- **Các điều hoạt bảo trì**
- **Các đặc quyền cho các bảng phân hoạch và các chỉ mục**
- **Phần mở rộng SQL: tên bảng phân hoạch mở rộng**

Lưu ý:
Các đặc trưng được mô tả trong chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với Partitioning Option. Xem Getting To Know Oracle8 và

Oracle8 Enterprise Edition về những đặc trưng và tùy chọn với Oracle8 Enterprise Edition.

Giới thiệu phân hoạch

Phần này giải thích phân hoạch có thể trợ giúp bạn quản lý các bảng lớn và các chỉ mục trong cơ sở dữ liệu Oracle như thế nào.

Lưu ý:

Oracle chỉ hỗ trợ phân hoạch cho các bảng và các chỉ mục; nó không hỗ trợ phân hoạch các bảng có cụm liên kết và các chỉ mục của chúng hoặc các ảnh chụp nhanh.

Phân hoạch để làm gì?

Phân hoạch lập địa chỉ vấn đề then chốt hỗ trợ rất nhiều bảng và các chỉ mục bằng cách cho bạn phân tích chúng thành các phân đoạn nhỏ hơn và có thể quản lý được nhiều phần nhỏ hơn được gọi là các phân hoạch.

Một khi các phân hoạch được xác định, các câu lệnh SQL có thể truy xuất và điều tác các phân hoạch thay vì toàn bộ các bảng hoặc các chỉ mục. Các phân hoạch rất hữu dụng trong các ứng dụng kho dữ liệu, thường lưu trữ và phân tích số lượng lớn dữ liệu lịch sử.

Tất cả những phân hoạch của một bảng hoặc chỉ mục có những thuộc tính logic giống nhau, dù những thuộc tính vật lý có thể khác. Thí dụ, tất cả các phân hoạch trong một bảng dùng chung cùng cột và các định nghĩa hạn định; và tất cả các phân hoạch trong một chỉ mục dùng chung cùng các cột chỉ mục. Tuy nhiên, các định chuẩn và các thuộc tính vật lý khác, chẳng hạn như PCTFREE, PCTUSED, INITRANS và MAXTRANS có thể thay đổi đối

với các phân hoạch khác nhau của cùng bảng hoặc chỉ mục.

Mỗi phần cứng được lưu trữ trong một đoạn tách riêng. Bạn có thể tùy chọn lưu trữ phân hoạch trong một không gian bảng tách riêng để có các ưu điểm sau:

- Bạn có thể chứa dữ liệu bị hư gây ảnh hưởng.
- Bạn có thể sao lưu và phục hồi mỗi phân hoạch độc lập.
- Bạn có thể cân bằng nạp I/O bằng cách ánh xạ các phân hoạch vào các ổ đĩa.

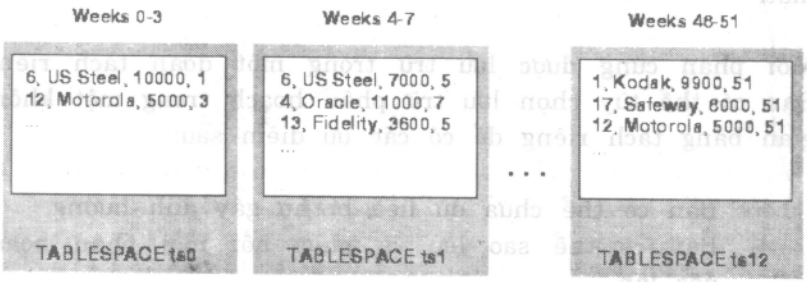
Phần "Mô Hình Phân Hoạch Căn Bản" cung cấp thêm thông tin về các khái niệm phân hoạch.

Thí dụ về bảng được phân hoạch

Trong Hình 9-1, bảng kinh doanh chứa dữ liệu lịch sử được chia cho số tuần thành 13 phân hoạch bốn tuần. Câu lệnh SQL này tạo bảng được phân hoạch:.

```
CREATE TABLE sales ( acct_no NUMBER(5), acct_name
CHAR(30), amount_of_sale NUMBER(6), week_no INTEGER )
PARTITION BY RANGE ( week_no ) ...
 (PARTITION sales1 VALUES LESS THAN ( 4 )
TABLESPACE ts0, PARTITION sales2 VALUES LESS
THAN ( 8 ) TABLESPACE ts1, ...
PARTITION sales13 VALUES LESS THAN ( 52 )
TABLESPACE ts12 );
```

Hình 9-1 Bảng kinh doanh được phân hoạch theo tuần



Thông tin bổ sung :

Để có thêm thí dụ về các bảng được phân hoạch, xem hướng dẫn của Oracle8 Administrator.

Xóa bỏ phân hoạch

Oracle server phối hợp trí tuệ thông minh để nhận biết các phân hoạch rõ ràng. Kiến thức này được khai thác trong các câu lệnh SQL tối ưu hóa để đánh dấu các phân hoạch cần được truy xuất, loại trừ ("xóa bỏ") các phân hoạch không cần thiết khỏi truy xuất bằng các câu lệnh SQL đó.

Đối với mỗi câu lệnh SQL, tùy theo tiêu chuẩn lựa chọn được ấn định, các phân hoạch không cần thiết có thể được loại trừ. Thí dụ, nếu một truy vấn chỉ liên quan đến dữ liệu kinh doanh Q1, thì không cần truy hồi dữ liệu cho tổng thể ba phần tư còn lại. Sự xóa bỏ thông minh như thế có thể làm giảm khối lượng dữ liệu rất nhiều, dẫn đến tăng hiệu suất truy vấn lên đáng kể.

Nếu bộ tối ưu hóa xác định tiêu chuẩn lựa chọn được sử dụng để xóa bỏ đáp ứng cho tất cả những hàng trong

phân hoạch được truy xuất, nó xóa những tiêu chuẩn đó khỏi danh sách thuộc tính (mệnh đề WHERE) khi thẩm định để tăng hiệu suất.

Xóa bỏ phân hoạch có thể loại trừ các phân hoạch chỉ mục ngay cả khi các phân hoạch của bảng cơ sở không thể bị loại trừ, nếu chỉ mục và bảng được phân hoạch trên các cột khác nhau. Bạn thường có thể tăng hiệu suất của các điều hoạt trên các bảng lớn bằng cách tạo các chỉ mục được phân hoạch để giảm số lượng dữ liệu mà các câu lệnh SQL của bạn cần truy xuất hoặc sửa đổi.

Khả năng xóa bỏ các phân hoạch không cần thiết khỏi các câu lệnh SQL làm tăng hiệu suất và tính khả dụng cho nhiều mục đích, kể cả nạp mức phân hoạch, xả, sao lưu, phục hồi, tái tổ chức và thiết lập chỉ mục.

Các ưu điểm của phân hoạch

Phần này nhận dạng các lớp của những cơ sở dữ liệu có thể khai thác nhờ vào sử dụng phân hoạch và biểu thị đặc điểm của chúng liên quan đến các vấn đề:

- Các Cơ Sở Dữ Liệu Rất Lớn (VLDB)
- Giảm Thời Gian Tái Xuống Đối Với Bảo Trì Theo Lịch Trình
- Hiệu Suất I/O
- Hiệu Suất Lập Dải Đĩa So Với Tính Khả Dụng
- Sự Thông Suốt Phân Hoạch

Các cơ sở dữ liệu rất lớn (VLDBs)

Cơ sở dữ liệu rất lớn - Very Large Database (VLDB) chứa hàng trăm gigabyte hoặc thậm chí một vài terabyte dữ liệu. Phân hoạch cung cấp hỗ trợ cho các VLDB chứa phần lớn dữ liệu có cấu trúc thay vì dữ liệu không có cấu trúc. Những VLDB này thường có được kích thước của

chúng nhờ vào sự hiện hữu của một vài đối tượng dữ liệu rất lớn (các bảng và các chỉ mục) thay vì sự hiện hữu của số rất lớn những đối tượng dữ liệu.

Có hai biệt loại chính của VLDB:
Các CSDL Xử Lý Giao Dịch Trực Tuyến - On-Line Transaction Processing (OLTP) được gán đối với những số lượng lớn các giao dịch đồng thời, nơi giao dịch tương đối là một điều hoạt đơn xử lý một số lượng nhỏ dữ liệu.
Các Hệ Thống Hỗ Trợ Quyết Định - Decision Support Systems (DSS) được thiết kế cho những truy vấn rất phức tạp cần truy xuất và xử lý những số lượng dữ liệu lớn.

VLDB có thể được biểu thị đặc tính bằng một cơ sở dữ liệu OLTP nếu phần lớn tác vụ tải của nó là OLTP. Tương tự một VLDB có thể được biểu thị đặc điểm bằng một cơ sở dữ liệu DSS nếu phần lớn tác vụ tải của nó gồm những truy vấn DSS.
Phân hoạch hỗ trợ cả OLTP VLDBs và DSS VLDBs hiệu quả.

Các cơ sở dữ liệu lịch sử

Những cơ sở dữ liệu lịch sử là kiểu DSS VLDB phổ biến nhất. Cơ sở dữ liệu lịch sử chứa hai mệnh đề của các bảng, các bảng lịch sử và các bảng của công ty.

> Các bảng lịch sử mô tả những giao dịch kinh doanh của một công ty trên một khoảng thời gian gần nhất, chẳng hạn như 24 giờ cuối cùng. Có hai kiểu bảng lịch sử:
>
> > Các bảng cơ sở chứa thông tin hàng cơ sở (thí dụ, doanh số, các ngân phiếu và các đơn đặt hàng).

Các bảng cuộn chứa thông tin tóm tắt được phái sinh từ thông tin cơ sở sử dụng các điều hoạt, chẳng hạn như GROUP BY, AVERAGE và COUNT.

Khoảng thời gian được phản ánh trong bảng lịch sử là một cửa sổ cuộn, vì vậy định kỳ nhà điều hành CSDL (DBA) xóa tập hợp những hàng được mô tả những giao dịch cũ nhất và phân bổ không gian cho tập hợp những hàng mô tả những giao dịch mới. Thí dụ, vào ngày khóa sổ kinh doanh 30 tháng 4 năm 1997, DBA xóa những hàng (và tất cả những mục nhập hỗ trợ) mô tả những giao dịch tháng 5 năm 1995 và phân bổ không gian cho những giao dịch tháng 5 năm 1997.

Đại đa số dữ liệu trong VLDB lịch sử được lưu trữ trong một vài bảng rất lớn trình bày những vấn đề đặc biệt do kích thước và đòi hỏi cuộn ra dữ liệu cũ và cuộn vào dữ liệu mới.

Các bảng của công ty mô tả những thực thể kinh doanh của công ty (thí dụ, các bộ phận, các vị trí và các sản phẩm). Thông tin này thay đổi chậm theo thời gian và không được sửa đổi theo lịch trình định kỳ. Dù các bảng của công ty không lớn, nhưng chúng ảnh hưởng hiệu suất của nhiều truy vấn DSS tiến hành thời gian lâu gồm các liên kết của một bảng lịch sử với các bảng của công ty.

Phân hoạch lập địa chỉ vấn đề hỗ trợ các bảng lịch sử lớn và những chỉ mục của chúng bằng cách chia dữ liệu lịch sử thành các phân hoạch liên quan thời gian có thể được quản lý độc lập và bổ sung hoặc xóa thuận tiện.

Các cơ sở nhiệm vụ tối quan trọng

Các CSDL OLTP có nhiệm vụ tối quan trọng trình bày tính khả dụng đặc biệt và các vấn đề hiệu suất dù chúng không lớn lắm. Thí dụ, có thể cần phải thực hiện những hoạt động bảo trì theo lịch trình hoặc phục hồi bảng 10-gigabyte trong một khoảng thời gian rất ngắn, có lẽ một giờ hoặc ít hơn. Ngoài ra, DBA có thể cần một mức độ điều khiển bố trí dữ liệu khó đạt được khi một bảng hoặc chỉ mục trải rộng trên nhiều ổ đĩa.

Phân hoạch có thể tăng tính khả dụng của những CSDL tối quan trọng nếu các bảng tối quan trọng và những chỉ mục được chia thành các phân hoạch để giảm các cửa sổ bảo trì xuống, thời gian phục hồi và tác động của các lỗi. Bạn còn có thể tăng hiệu suất truy xuất một bảng tối quan trọng hoặc chỉ mục bằng những tham số điều khiển hiệu suất trên phương thức phân hoạch.

Giảm thời gian tải xuống để bảo trì theo lịch trình

Các phân hoạch cho phép những hoạt động quản lý dữ liệu như tải, tạo chỉ mục và xóa bỏ dữ liệu ở mức phân hoạch, thay vì trên toàn bộ bảng dẫn đến làm giảm thời gian đáng kể cho những hoạt động này.

Phân hoạch có thể giảm tác động của thời gian tải xuống cho các hoạt động bảo trì đáng kể:
Bằng cách đưa ra các hoạt động bảo trì phân hoạch điều tác trên một phân hoạch riêng thay vì trên toàn bộ bảng hoặc chỉ mục.
Bằng cách cung cấp phân hoạch độc lập để các hoạt động bảo trì có thể được thực hiện đồng thời trên các phân hoạch khác nhau.

Các hoạt động bảo trì phân hoạch

Các hoạt động bảo trì phân hoạch nhanh hơn toàn bộ một bảng hoặc các hoạt động bảo trì chỉ mục. Tốc độ tăng có thể đạt được bằng với tỷ số:

(# bản ghi trong toàn bộ bảng hoặc chỉ mục) / (# bản ghi trong phân hoạch)

với điều kiện không có các cấu trúc được lưu trữ liên phân hoạch (các chỉ mục toàn cục và các hạn định tính nguyên vẹn liên quan).

Để giảm thêm thời gian tải xuống, một hoạt động bảo trì phân hoạch có thể khai thác các đặc trưng hiệu suất có thể sử dụng cho bảng và các hoạt động mức chỉ mục, chẳng hạn như các tùy chọn PARALLEL, NOLOGGING và DIRECT (hoặc APPEND) nơi có thể áp dụng.

Tính độc lập phân hoạch

Tính độc lập phân hoạch đối với các hoạt động bảo trì phân hoạch tạo cho nó có thể thực hiện các hoạt động bảo trì đồng thời trên các phân hoạch khác nhau của cùng bảng hoặc chỉ mục cũng như các hoạt động SELECT và DML đồng thời trên các phân hoạch không bị ảnh hưởng bởi các hoạt động bảo trì.

Thí dụ, bạn có thể đưa Direct Path Load vào các phân hoạch PA và PB cùng một lúc, trong khi các ứng dụng đang tiến hành các hoạt động SQL SELECT và DML chuẩn trên các phân hoạch khác.

Tính độc lập phân hoạch rất quan trọng đối với các điều hoạt liên quan đến di chuyển dữ liệu. Những điều hoạt như thế có thể mất thời gian (hàng phút, hàng giờ hoặc

ngay cả hàng ngày). Phân hoạch có thể giảm cửa sổ không thể sử dụng trên các phân hoạch khác còn thời gian ngắn (vài giây) khi các điều hoạt liên quan đến di chuyển dữ liệu, với điều kiện không có các cấu trúc được lưu trữ liên phân hoạch (các chỉ mục toàn cục và các hạn định tính nguyên vẹn liên quan).

Tính độc lập phân hoạch không cần thiết đối với các điều kiện ngắn (không di chuyển dữ liệu) vì những điều hoạt này hoàn tất trong một thời gian ngắn.

Giảm thời gian tải xuống vì các sai sót dữ liệu

Một số điều hoạt bảo trì là các sự kiện không được dự tính đòi hỏi phải phục hồi từ các lỗi phần cứng hoặc phần mềm làm mất dữ liệu hoặc bị sai sót. Phục hồi các sai sót phần cứng và nhiều lỗi phần mềm hệ thống được thực hiện bằng cách chạy lệnh RECOVER trên một CSDL, không gian bảng hoặc tập tin dữ liệu. Bảng hoặc chỉ mục bất kỳ có các bản ghi trong một không gian bảng hoặc tập tin dữ liệu được phục hồi vẫn không khả dụng khi phục hồi. Tính khả dụng tăng rất quan trọng đối với những cơ sở dữ liệu OLTP nhiệm vụ tối quan trọng..
Bởi vì các phân hoạch độc lập với nhau, tính bất khả dụng của một phân đoạn (hoặc một tập hợp con của các phân đoạn) không ảnh hưởng đến truy xuất đối với phần còn lại của dữ liệu.

Các phân hoạch lưu trữ trong các không gian bảng tách riêng cung cấp các ưu điểm sau:

Thời gian tải xuống do sự tiến hành lệnh bằng lệnh RECOVER giảm xuống bởi vì đơn vị phục hồi (một không gian bảng) nhỏ hơn.

Những tài nguyên đĩa cần được phục hồi một không gian ngoại tuyến (các đoạn hồi phục trạng thái cũ

được trì hoãn). Những ứng dụng người dùng vẫn có thể truy xuất các phân hoạch khác. Điều này là một thí dụ khác của tính độc lập phân hoạch.

Hiệu suất DSS

Những truy vấn trên các bảng rất lớn trình bày các vấn đề hiệu suất đặc biệt. Một truy vấn không định trước đòi hỏi duyệt bảng có thể mất thời gian, vì nó phải khảo sát mọi hàng trong bảng ấy; không có cách nào để nhận dạng và bỏ qua các tập hợp con của những hàng không thích hợp. Vấn đề đặc biệt quan trọng đối với các bảng lịch sử, vì nhiều truy vấn tập trung truy xuất trên những hàng mới được phát sinh.

Các phân hoạch giúp giải quyết vấn đề hiệu suất DSS này. Một truy vấn không định trước chỉ đòi hỏi các hàng tương ứng với một phân hoạch đơn (hoặc phạm vi của phân hoạch) có thể được thực hiện bằng cách duyệt phân hoạch thay vì duyệt bảng.

Thí dụ, một truy vấn yêu cầu dữ liệu được phát sinh trong tháng 10 năm 1997 có thể chỉ duyệt những hàng được lưu trữ trong phân hoạch tháng 10 năm 1997 thay vì các hàng được phát sinh trong nhiều năm hoạt động. Điều này làm tăng thời gian trả lời và nó cũng có thể giảm không gian đĩa tạm thời đáng kể đối với những truy vấn cần sắp xếp.

Hiệu suất I/O

Phân hoạch có thể điều khiển dữ liệu trải rộng trên các thiết bị vật lý như thế nào. Để cân bằng sử dụng I/O, bạn có thể ấn định nơi lưu trữ các phân hoạch của một bảng hoặc chỉ mục.

Với mức điều khiển vị trí này, bạn có thể điều tiết các nhu cầu đặc biệt của những ứng dụng đòi hỏi thời gian trả lời nhanh bằng cách giảm tranh chấp đĩa và sử dụng các thiết bị nhanh hơn. Nói cách khác, dữ liệu được truy xuất thường xuyên, chẳng hạn như dữ liệu đã cũ, có thể được di chuyển vào những đĩa chậm hoặc được lưu trữ trong các hệ thống con hỗ trợ một phân cấp bộ nhớ.

Lập dải đĩa: hiệu suất so với tính khả dụng

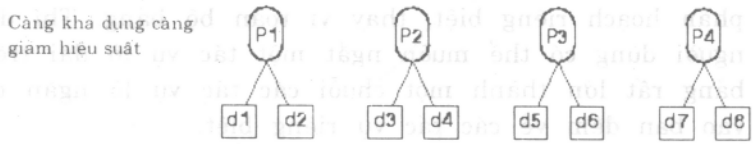
Lập dải đĩa và phân hoạch là hai công cụ có thể tăng hiệu suất bằng cách làm giảm tranh chấp các cần của ổ đĩa. Công cụ sử dụng hoặc trong đó tỷ lệ sử dụng chung với nhau như thế nào là vấn đề quan trọng cần xét khi ấn định CSDL vật lý. Những vấn đề này phải được xét không những liên quan đến hiệu suất mà còn liên quan đến tính khả dụng và tính độc lập phân hoạch.

Hình 9-2 trình bày hai cực độ của phân hoạch kết hợp và lập dải. Cả hai (a) và (b) trình bày bốn phân hoạch trải rộng trên tám đĩa, nhưng (a) lập dải mỗi phân hoạch thành hai đĩa riêng của nó.

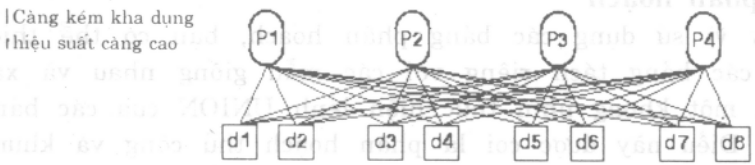
Các đặc điểm hiệu suất tốt hơn ở (b), nhưng nếu một lỗi đĩa đơn bất kỳ xảy ra, thì tất cả các phân hoạch bị ảnh hưởng bất lợi.
Những đặc tính khả dụng tốt hơn ở (a) vì lỗi của một đĩa đơn chỉ ảnh hưởng đến một phân hoạch.

Các cấu hình trung gian cũng có thể xảy ra, nơi các tập hợp con của các phân hoạch được lập dải trên các tập hợp con của các đĩa.

Hình 9-2 Các phân hoạch và lập dải đĩa



(a) mỗi phân hoạch cư trú trên một dải tập hợp con của tất cả các đĩa



(b) mọi phân hoạch được lập dải qua tất cả các đĩa

Trao đổi giữa hiệu suất và tính khả dụng phải được quyết định khi xác định cách phân hoạch các bản và các chỉ mục và cách lập dải các đĩa mà chúng được lưu trữ.

Những CSDL nhiệm vụ tối quan trọng mà nó được đề nghị tính độc lập phân hoạch và tính khả dụng ưu tiên, do đó mỗi phân hoạch bạn muốn lập dải trên các đĩa sẽ được lập dải thành tập hợp riêng các ổ đĩa của nó để sẽ có đủ các ổ đĩa đặt được tính song song I/O cần thiết cho các truy xuất phân hoạch đó.

Tính thông suốt phân hoạch

Đại đa số những trình ứng dụng đòi hỏi tính thông suốt phân hoạch là các chương trình sẽ được mở rộng đối với dữ liệu, xem dữ liệu chúng truy xuất được phân hoạch chưa và nó được phân hoạch như thế nào.

Tuy nhiên, một vài trình ứng dụng có thể khai thác các phân hoạch bằng cách yêu cầu truy xuất cụ thể đối với một phân hoạch riêng biệt, thay vì toàn bộ bảng. Thí dụ, một người dùng có thể muốn ngắt một tác vụ lô dài trên một bảng rất lớn thành một chuỗi các tác vụ lô ngắn để làm vào ban đêm về các tác vụ riêng biệt.

Phân hoạch thủ công với các khung nhìn phân hoạch

Thay vì sử dụng các bảng phân hoạch, bạn có thể thiết lập các bảng tách riêng với các mẫu giống nhau và xác định một khung nhìn thực hiện lệnh UNION của các bảng này. Điều này được coi là phân hoạch thủ công và khung nhìn được coi là khung nhìn phân hoạch.

Các khung nhìn phân hoạch đã là biểu mẫu duy nhất của phân hoạch có thể sử dụng trong Oracle7 Release 7.3. Chúng không được đề nghị đối với những ứng dụng trong Oracle8. Những khung nhìn phân hoạch được tạo cho những CSDL Oracle7 có thể được chuyển đổi thành các bảng được phân hoạch bằng cách sử dụng tùy chọn EXCHANGE PARTITION của lệnh ALTER TABLE.

Lưu ý:

- Oracle8 chỉ hỗ trợ các khung nhìn phân hoạch cho tính tương thích về trước với Oracle7 Release 7.3.

Thông tin bổ sung:

Xem Oracle Migration và Hướng dẫn của Oracle8 Administrator về các chỉ lệnh chuyển đổi các khung nhìn phân hoạch thành các bảng được phân hoạch.

Ý tưởng cơ bản hậu thuẫn các khung nhìn phân hoạch để chia bảng lớn thành nhiều bảng vật lý sử dụng tiêu chuẩn phân hoạch (mệnh đề WHERE hoặc hạn định

CHECK), sau đó gắn ghép các bảng nhỏ lại với nhau thành một với khung nhìn UNION ALL. Sau đó bạn có thể xác định các tập hợp "những chỉ mục cơ sở" với các định chuẩn khóa đồng nhất trên các bảng cơ sở để cung cấp những khả năng chỉ mục khi khung nhìn UNION ALL được sử dụng. Các khung nhìn phân hoạch phải được lập chỉ mục để hoạt động thích hợp.

Những truy vấn sử dụng phạm vi khóa để lựa từ một khung nhìn phân hoạch, chỉ truy xuất các bảng cơ sở nằm trong phạm vi khóa đó. Bộ tối ưu hóa có thể sử dụng các dự kiến thực hiện đơn giản các bảng cơ sở của khung nhìn phân hoạch. (Trái lại, bộ tối ưu hóa sử dụng một kế hoạch thực hiện đơn giản cho tất cả các phân hoạch trong một bảng được phân hoạch.)

Phân hoạch thủ công với các khung nhìn có một số bất lợi:

- Tính phức tạp cấu hình
  Người điều hành CSDL chịu trách nhiệm xác định chính xác các bảng cơ sở và các chỉ mục tương ứng với các phân hoạch và duy trì những định nghĩa này. Sự tương đương của các điều hoạt DDL di chuyển dữ liệu trên các phân hoạch (tách, di chuyển, v.v...) phải được thực hiện bằng các ký mã Export/Import hoặc SQL.
- Thiếu tính thông suốt phân hoạch
  Một số điều hoạt SQL phải được thực hiện bằng cách sử dụng các bảng cơ sở thay vì khung nhìn UNION ALL. Thí dụ, INSERT tham chiếu một bảng cơ sở và cần có mã người dùng để có được tên bảng xuất hiện trong câu lệnh INSERT.
- Thiếu hiệu suất
  Một số điều hoạt SQL trên khung nhìn UNION ALL có thể thực hiện không chính xác vì bộ tối

ưu hóa không khai thác được tất cả những chỉ mục cơ sở có sẵn.

- Sử dụng bộ nhớ kém
  Một truy vấn biên dịch SQL điều hoạt trên một khung nhìn UNION ALL sao chép thông tin mô tả bên trong đối với tất cả các bảng hỗ trợ khung nhìn ấy.
- Các hạn chế DDL
  Các chỉ mục toàn cục và các hạn định tính toàn vẹn tham chiếu không thể được xác định trên khung nhìn UNION ALL.
- Các hạn chế tải
  Không thể tải trực tiếp trên một khung nhìn UNION ALL.

Mô hình phân hoạch cơ bản

Phân hoạch được ấn định với các tùy chọn thành các lệnh CREATE TABLE và CREATE INDEX. Sau khi tạo một bảng phân hoạch hoặc chỉ mục, bạn có thể sử dụng các câu lệnh ALTER TABLE hoặc ALTER INDEX để sửa đổi các thuộc tính phân hoạch của nó. Cú pháp phân hoạch đối với các câu lệnh CREATE TABLE vàCREATE INDEX rất giống nhau.

Câu lệnh CREATE TABLE ấn định:

- Các thuộc tính logic của bảng, chẳng hạn như các định nghĩa cột và các hạn định.
- Các thuộc tính vật lý của bảng.
- Nếu bảng không được phân hoạch, thì đây là các thuộc tính vật lý thực của đoạn được kết hợp với bảng.
- Nếu bảng được phân hoạch, thì các thuộc tính mức bảng này ấn định các mặc định đối với các phân hoạch riêng biệt của bảng.

- Đối với bảng được phân hoạch, còn có một định chuẩn phân hoạch gồm:
  - Thuật toán mức bảng được sử dụng để ánh xạ các hàng vào các phần hoạch.
  - Một danh sách các mô tả phân hoạch, mỗi mô tả cho một phân hoạch trong bảng.

  Mỗi mô tả phân hoạch gồm một mệnh đề xác định thông tin bổ sung, thông tin mức phân hoạch về thuật toán được sử dụng để ánh xạ vào các phân hoạch. Mệnh đề này còn có thể ấn định tên phân hoạch và các thuộc tính vật lý cho phân hoạch ấy.

Các hạn chế kiểu dữ liệu

Đối với các bảng được phân hoạch, các thuộc tính logic có các hạn chế bổ sung. Các bảng được phân hoạch không thể có bất kỳ cột nào với các kiểu dữ liệu LONG hoặc LONG RAW, các kiểu dữ liệu LOB (BLOB, CLOB, NCLOB hoặc BFILE), hay kiểu dữ liệu đối tượng.

Nếu một bảng (hoặc chỉ mục) được phân hoạch trên một cột có kiểu dữ liệu DATE, thì các mô tả phân hoạch sẽ sử dụng mặt che định dạng TO_DATE; trái lại không thể xóa bỏ phân hoạch. Xem "Mặt Che Định Dạng TO DATE".

Các hạn chế bitmap

Bạn có thể tạo các chỉ mục bitmap trên các bảng phân hoạch với hạn chế các chỉ mục bitmap phải cục bộ đối với bảng phân hoạch - chúng không thể là các chỉ mục toàn cục. (Xem Phân Hoạch Chỉ Mục"

Sự tối ưu hóa dựa trên chi phí

Bộ tối ưu hóa dựa trên chi phí được sử dụng khi câu lệnh SQL truy xuất các bảng được phân hoạch hoặc các chỉ mục; nguyên tắc dựa vào sự tối ưu hóa không thể sử

dụng đối với các phân hoạch. Một kế hoạch thực hiện đơn được sử dụng cho tất cả phân hoạch của một bảng được phân hoạch.

Môn thống kê có thể được tập hợp bằng phân hoạch sử dụng lệnh ANALYZE. Điều quan trọng để tập hợp thống kê bất kỳ khi nào tính chất dữ liệu trong bảng được phần cứng thay đổi nhiều. Môn thống kê có thể được tìm thấy trong các khung nhìn từ điển dữ liệu này:
ALL_TAB_PARTITIONS, DBA_TAB_PARTITIONS, USER_TAB_PARTITIONS
ALL_IND_PARTITIONS, DBA_IND_PARTITIONS, USER_IND_PARTITIONS
ALL_PART_COL_STATISTICS, DBA_PART_COL_STATISTICS, USER_PART_COL_STATISTICS

Phân hoạch phạm vi

Phân hoạch phạm vi ánh xạ vào các phân hoạch dựa trên các phạm vi của những giá trị cột. Phân hoạch phạm vi được xác định bằng định chuẩn phân hoạch cho một bảng hoặc chỉ mục:

PARTITION BY RANGE (column_list)

Và bằng các định chuẩn cho mỗi phân hoạch riêng biệt:

VALUES LESS THAN (value_list)

Trong đó:
column_list là một danh sách có thứ tự của các cột xác định phân hoạch mà hàng hoặc một mục nhập thuộc về. Những cột này được gọi là các cột phân hoạch.
Những giá trị trong các cột phân hoạch của một hàng riêng biệt cấu tạo khóa phân hoạch của hàng đó.

value_list là một danh sách có thứ tự của những giá trị đối với các cột trong column_list.

Mỗi giá trị trong value_list phải là một trực kiện hoặc một hàm TO_DATE() hoặc hàm RPAD() với các đối số không thay đổi. (Xem "Các Biên Phân Hoạch Và Các Khóa Phân Hoạch" về các khóa phân hoạch được so sánh với các biên phân hoạch như thế nào và các khóa phân hoạch nhiều cột được xử lý như thế nào.)

value_list được chứa trong định chuẩn phân hoạch cho phân hoạch ith xác định một biên trên mở (không bao gồm) cho phân hoạch được tham chiếu như bằng biên phân hoạch.

Biên phân hoạch cho phân hoạch ith còn xác định một biên trên mở (không bao gồm) cho phân hoạch ith phải nhỏ hơn đối với phân hoạch (i+1)th.

Trong phân hoạch ith, tất cả các hàng (hoặc các hàng được trỏ bằng các mục nhập chỉ mục) có các khóa phân hoạch nhỏ hơn biên phân hoạch đối với phân hoạch đó. Trừ khi phân hoạch ith là phân hoạch thứ nhất trong bảng hoặc chỉ mục, tất cả các khóa phân hoạch trong phân hoạch ith còn so sánh lớn hơn hoặc bằng biên phân hoạch đối với phân hoạch (i-1)th. (Xem "Các Biên Phân Hoạch Và Các Khóa Phân Hoạch" về các khóa phân hoạch so với các biên phân hoạch như thế nào và các khóa phân hoạch nhiều cột được xử lý như thế nào.)

Thí dụ, trong bảng sau có bốn phân hoạch (mỗi phân hoạch cho mỗi quý kinh doanh), một hàng với sale_year=1997, sale_month=7, và sale_day=18 có khóa phân hoạch (1997, 7, 18) thuộc phân hoạch thứ ba và sẽ được lưu trữ trong tsc không gian bảng. Một hàng với

sale_year=1997, sale_month=7 và sale_day=1 có khóa phân hoạch (1997, 7, 1) và cũng có trong phân hoạch thứ ba, được lưu trữ trong tsc không gian bảng.

```
CREATE TABLE sales
    ( invoice_no NUMBER,
      sale_year   INT NOT NULL,
      sale_month  INT NOT NULL,
      sale_day    INT NOT NULL )
  PARTITION BY RANGE (sale_year, sale_month, sale_day)
    ( PARTITION sales_q1 VALUES LESS THAN (1994, 04, 01)
        TABLESPACE tsa,
      PARTITION sales_q2 VALUES LESS THAN (1994, 07, 01)
        TABLESPACE tsb,
      PARTITION sales_q3 VALUES LESS THAN (1994, 10, 01)
        TABLESPACE tsc,
      PARTITION sales_q4 VALUES LESS THAN (1995, 01, 01)
        TABLESPACE tsd );
```

Các tên phân hoạch

Mọi phân hoạch có tên phải thích hợp với các nguyên tắc thông thường để đặt tên các đối tượng giản đồ và các thành phần của nó. Nhất là:

Tên của một phân hoạch phải duy nhất trong tất cả những phân hoạch thuộc cùng bảng cha.

Tên của một phân hoạch chỉ mục phải duy nhất trong tất cả phân hoạch thuộc cùng chỉ mục cha.

Bạn có thể đặt tên một phân hoạch lại; tuy nhiên, bạn không thể tạo bất kỳ từ đồng nghĩa nào trên một tên phân hoạch.

Thông tin bổ sung :

Xem Oracle8 SQL Reference về những nguyên tắc đặt tên các đối tượng giản đồ.

Tham chiếu một phân hoạch

Các tên phân hoạch có thể được tham chiếu tùy chọn trong các câu lệnh DDL và DML, trong các câu lệnh tiện ích như Import/Export và SQL*Loader. Chúng luôn luôn xuất hiện trong ngữ cảnh với tên của bảng cha hoặc chỉ mục và chúng không bao giờ được hợp chuẩn bằng một tên giản đồ. (Tên giản đồ có thể được sử dụng để hợp chuẩn bảng cha hoặc chí mục.)
Thí dụ:

```
ALTER TABLE admin.patient_visits DROP PARTITION
pv_dec92
```

Xem "Mở Rộng SQL: Tên Bảng Mở Rộng Phân Hoạch" về những phân hoạch tham chiếu trong các câu lệnh SQL.

Các biên phân hoạch và các khóa phân hoạch

Phần này mô tả một khóa phân hoạch của hàng được so sánh với một tập hợp các biên trên và dưới như thế nào để xác định hàng thuộc về phân hoạch nào.

Các biên phân hoạch

Mỗi bảng và phân hoạch chỉ mục không bao gồm *upper bound* (biên trên) được mệnh đề VALUES LESS THAN xác định. Mỗi phân hoạch trừ phân hoạch đầu tiên cũng bao gồm cả *lower bound* (biên dưới) được VALUES LESS THAN xác định trên phân hoạch dưới kế tiếp.

Các biên phân hoạch xác định chung một sắp xếp thứ tự của các phân hoạch trong một bảng hoặc chỉ mục. Phân hoạch "thứ nhất" là phân hoạch cùng với mệnh đề VALUES LESS THAN thấp nhất và phân hoạch "cuối" hoặc "cao nhất" là phân hoạch với mệnh đề VALUES LESS THAN cao nhất.

Nếu bạn cố chèn một hàng vào một bảng và khóa phân hoạch của hàng ấy lớn hơn hoặc bằng với biên phân hoạch cho phân hoạch cao nhất trong bảng, thì lệnh chèn ấy sẽ không thực hiện được.

Các khóa phân hoạch

Một khóa phân hoạch gồm một danh sách có thứ tự tối đa 16 cột. Khóa phân hoạch của hàng là một danh sách có thứ tự những giá trị của nó cho các cột phân hoạch ấy.

Khóa phân hoạch có thể không chứa cột giả LEVEL, ROWID hoặc MLSLABEL hay một cột của kiểu ROWID.

Khi những giá trị ký tự so sánh trong các khóa phân hoạch và các biên phân hoạch, thì những ký tự được so sánh theo những giá trị nhị phân của chúng. Tuy nhiên, nếu một ký tự hơn một byte, Oracle so sánh giá trị nhị phân của mỗi byte, không phải của ký tự.

So sánh còn sử dụng những nguyên tắc so sánh được kết hợp với kiểu dữ liệu cột (thí dụ, so sánh có chêm khoảng trắng chêm (blank-padded) được thực hiện cho kiểu dữ liệu ANSI CHAR). Những tham số NLS, nhất là các tham số khởi hoạt NLS_SORT, NLS_LANGUAGE và biến môi trường NLS_LANG không có tác động đến so sánh đó.

MAXVALUE

Bạn có thể ấn định từ khóa MAXVALUE cho giá trị bất kỳ trong value_list của biên phân hoạch. Từ khóa này

biểu thị một giá trị "vô hạn" ảo sắp xếp cao hơn giá trị bất kỳ khác đối với kiểu dữ liệu ấy, kể cả giá trị rỗng.

Thí dụ, bạn có thể phân hoạch bảng văn phòng ở trạng thái CHAR(10) column) vào ba phần cứng với các biên phân hoạch sau:
VALUES LESS THAN ('I'): Chứa các trạng thái có tên bắt đầu với A tới H.
VALUES LESS THAN ('S'): Chứa các trạng thái có tên với I tới R.
VALUES LESS THAN (MAXVALUE): Chứa các trạng thái có tên bắt đầu với S tới Z, cùng với mã đặc biệt cho các vùng không thuộc nước Mỹ.

Các giá trị rỗng - Null

NULL không thể được ấn định bằng một giá trị trong một value_list của biên phân hoạch. Một chuỗi rỗng cũng không thể được ấn định bằng một giá trị trong một value_list biên phân hoạch, vì nó được coi như một NULL trong server CSDL.

Đối với mục đích gán các hàng vào các phân hoạch, Oracle sắp xếp các giá trị rỗng lớn hơn tất cả những giá trị khác trừ MAXVALUE. Các giá trị rỗng sắp xếp nhỏ hơn MAXVALUE.

Điều này có nghĩa là nếu được phân hoạch trên một cột có thể có giá trị rỗng và cột ấy phải chứa các giá trị rỗng, thì phân hoạch cao nhất sẽ có một biên phân hoạch của MAXVALUE cho cột đó. Trái lại, các hàng chứa những giá trị rỗng sẽ ánh xạ trên phân hoạch cao nhất trong bảng và lệnh chèn sẽ không thực hiện được.

Mặt che định dạng TO_DATE

Nếu khóa phân hoạch có một cột có kiểu dữ liệu DATE, bạn phải ấn định các biến phân hoạch sử dụng mặt che định dạng TO_DATE(); trái lại loại trừ ("xóa bỏ") phân hoạch sẽ không hoạt động.

Thí dụ, bạn có thể tạo một bảng kinh doanh sử dụng cột DATE:

```
CREATE TABLE sales
    ( invoice_no NUMBER,
      sale_date DATE NOT NULL )
  PARTITION BY RANGE (sale_date)
    ( PARTITION sales_q1
        VALUES LESS THAN (TO_DATE('94-04-01','YY-MM-DD'))
        TABLESPACE tsa,
      PARTITION sales_q2
        VALUES LESS THAN (TO_DATE('94-07-01','YY-MM-DD'))
        TABLESPACE tsb,
      PARTITION sales_q3
        VALUES LESS THAN (TO_DATE('94-10-01','YY-MM-DD'))
        TABLESPACE tsc,
      PARTITION sales_q4
        VALUES LESS THAN (TO_DATE('95-01-01','YY-MM-DD'))
        TABLESPACE tsd );
```

Bạn cũng cần phải dùng mặt che định dạng TO_DATE() khi bạn truy vấn hoặc sửa đổi dữ liệu trong bảng kinh doanh, thí dụ:

```
SELECT * FROM sales
  WHERE sale_date < TO_DATE('94-06-15','YY-MM-DD');
```

Các khóa phân hoạch nhiều cột

Khi một bảng hoặc chỉ mục được phân hoạch trên nhiều cột, thì mỗi biên phân hoạch và khóa phân hoạch là một danh sách (hoặc vectơ) những giá trị. Trong trường hợp

này, các khóa được sắp xếp thứ tự theo các nguyên tắc so sánh vectơ ANSI SQL2 (điều này cũng là cách các khóa chỉ mục nhiều cột được sắp xếp thứ tự trong Oracle)

Đối với các vectơ V1 và V2 chứa cùng số giá trị, Vx[i] là giá trị ith trong Vx. Giả sử là V1[i] và V2[i] có các kiểu dữ liệu tương thích:
V1 = V2 nếu và chỉ nếu V1[i] = V2[i] đối với tất cả i.
V1 < V2 nếu và chỉ nếu V1[i] = V2[i] đối với tất cả i < n và V1[n] < V2[n] đối với n nào đó.
V1 > V2 nếu và chỉ nếu V1[i] = V2[i] đối với tất cả i < n và V1[n] > V2[n] đối với n nào đó.

Nghĩa là, nếu bạn muốn biết nếu một khóa phân hoạch PK nhỏ hơn hoặc bằng biên phân hoạch PB, bạn so sánh các giá trị tương ứng trong PK và PB cho tới khi tìm thấy một cặp không bằng và cặp xác định.

Thí dụ, nếu biên phân hoạch đối với phân hoạch P là (7, 5, 10) và biên phân hoạch cho phân hoạch thấp nhất kế tiếp là (6, 7, 3):
Các khóa (6, 9, 11) và (7, 3, 15) thuộc phân hoạch P, bởi vì:
Khóa (6, x, x) nhỏ hơn (7, x, x) và (6, 9, x) lớn hơn (6, 7, x)
Khóa (7, 3, x) nhỏ hơn (7, 5, x) và lớn hơn (6, 7, x)
Các khóa (6, 5, 0) và (7, 5, 11) thuộc các phân hoạch.

Nếu MAXVALUE xảy ra là một phần tử của một value list biên phân hoạch, thì các giá trị của tất cả các phần tử sau thích hợp. Thí dụ, một biên phân hoạch (10, MAXVALUE, 5) tương đương với biên phân hoạch (10, MAXVALUE, 6) hoặc với một biên phân hoạch (10, MAXVALUE, MAXVALUE).

Các khóa phân hoạch nhiều cột hữu dụng khi khóa chính đối với bảng chứa nhiều cột, nhưng không được phân phối đều trên cột quan trọng nhất trong khóa ấy. Thí dụ, giả

sử bảng supplier_parts chứa thông tin về nhà cung cấp nào cung cấp các bộ phận nào và khóa chính đối với bảng ấy là (suppnum, partnum). Không đủ phân hoạch trên suppnum vì một số nhà cung cấp cung cấp hàng trăm ngàn bộ phận, trong khi các nhà cung cấp khác chỉ cung cấp một ít bộ phận đặc biệt. Thay vào đó, bạn có thể phân hoạch bảng trên (suppnum, partnum).
Các khóa phân hoạch nhiều cột còn hữu dụng khi bạn biểu thị ngày tháng bằng ba cột CHAR thay vì cột DATE.

Các hạn định ngầm định do các biên phân hoạch bắt buộc

Nếu bạn ấn định một biên phân hoạch khác hơn MAXVALUE đối với phân hoạch cao nhất trong một bảng, điều này buộc một CHECK ẩn trên bảng. Hạn định này không được ghi trong từ điển dữ liệu (nhưng chính biên phân hoạch được ghi)

Phân hoạch tương đương

Hai bảng hoặc chỉ mục được *equipartitioned* (phân hoạch tương đương) nếu chúng có các thuộc tính phân hoạch logic đồng nhất. Chúng không buộc phải có cùng kiểu đối tượng giản đồ; thí dụ, một bảng và một chỉ mục có thể được phân hoạch tương đương.

Nếu A và B là các bảng được phân hoạch hoặc các chỉ mục, nơi A[i] là phân hoạch trong A và B[i] là phân hoạch ith trong B, thì A và B được phân hoạch tương đương nếu tất cả yếu tố sau đúng:
Chúng có cùng số phân hoạch N.
Chung có cùng số cột phân hoạch M.

Đối với mọi 1 <= i <= N, A[i] và B[i] có cùng biên phân hoạch.
Nếu Apcol[i] là cột phân hoạch ith trong A và Bpcol[i] là cột phân hoạch ith trong B, thì các yếu tố sau cũng phải đúng:
Đối với 1 <= i <= M, Apcol[i] và Bpcol[i] có cùng kiểu dữ liệu, kể cả chiều dài, độ chính xác và tỷ lệ.

A[i] và B[i] có thể khác nhau về những thuộc tính vật lý; đặc biệt chúng không phải cư trú trong cùng không gian bảng.

Phân hoạch tương đương quan trọng cần xét khi thiết kế CSDL.

Nó làm giảm thời gian tải xuống và lượng dữ liệu không thể sử dụng trong các hoạt động bảo trì phân hoạch và các hoạt động phục hồi không gian bảng. Thí dụ, nếu một bảng và các chỉ mục của nó sau đó được phân hoạch tương đương thì tác động tách một phân hoạch bị giới hạn vào một phân hoạch bảng và các phân hoạch chỉ mục tương ứng, nhưng nếu một bảng có một chỉ mục không được phân hoạch tương đương thì tách một phân hoạch của bảng làm cho nó cần phải tổ chức lại toàn bộ chỉ mục ấy.

Phân hoạch tương đương giữa các bảng làm tăng kế hoạch thực hiện bằng cách giảm số các sắp xếp lớn và các liên kết phải được thực hiện. (Cải tiến này chỉ áp dụng đối với sự thực hiện nối tiếp của các câu lệnh SQL.)

Nó tạo cho không gian báng phục hồi không hoàn toàn (giờ và ngày tháng) trên các tập hợp con dữ liệu dễ dàng hơn. Thí dụ, bạn có thể phân hoạch tương đương

một bảng liên quan và chỉ mục khóa chính hoặc một bảng cha và một bảng con. Sau đó bạn có thể phục hồi các phân hoạch tương ứng giờ và ngày tháng.

Thí dụ về phân hoạch tương đương

Hình 9-3 trình bày bốn đối tượng giản đồ liên quan logic được phân hoạch tương đương:
ACCOUNTS là một bảng với hai phân hoạch được phân hoạch phạm vi trên cột ACCOUNT_NO. Phân hoạch thứ nhất chứa các số tài khoản tối đa 1000. Phân hoạch thứ hai chứa các số tài khoản tối đa 2000.
ACCOUNTS_IX là một chỉ mục trên cột ACCOUNT_NO trong bảng ACCOUNTS. Giống như bảng, chỉ mục được phân hoạch phạm vi trên ACCOUNT_NO vào hai phân hoạch có cùng các biên phân hoạch như các phân hoạch của ACCOUNTS.
CHECKS là một bảng có hai phân hoạch được phân hoạch phạm vi trên cột ACCT_NO. Các phân hoạch của nó có cùng các biên phân hoạch như các phân hoạch của bảng ACCOUNTS. ACCT_NO là khóa ngoại tham chiếu ACCOUNT_NO trong ACCOUNTS.
CHECKS_IX là một chỉ mục trên các cột (ACCT_NO, CHECK_NO) trong CHECKS. Nó được phân hoạch phạm vi trên ACCT_NO thành hai phân hoạch có cùng các biên phân hoạch như các phân hoạch của ACCOUNTS.

Quan hệ logic giữa bốn đối tượng giản đồ được trình bày ở bên trái của Hình 9-3; phân hoạch vật lý được trình bày ở bên phải. (Các tam giác biểu thị những chỉ mục và các hình chữ nhật biểu thị các bảng.)

Các nguyên tắc phân hoạch các bảng và các chỉ mục

Phần này mô tả những nguyên tắc để tạo các bảng được phân hoạch, các chỉ mục và các thuộc tính vật lý của các phân hoạch.

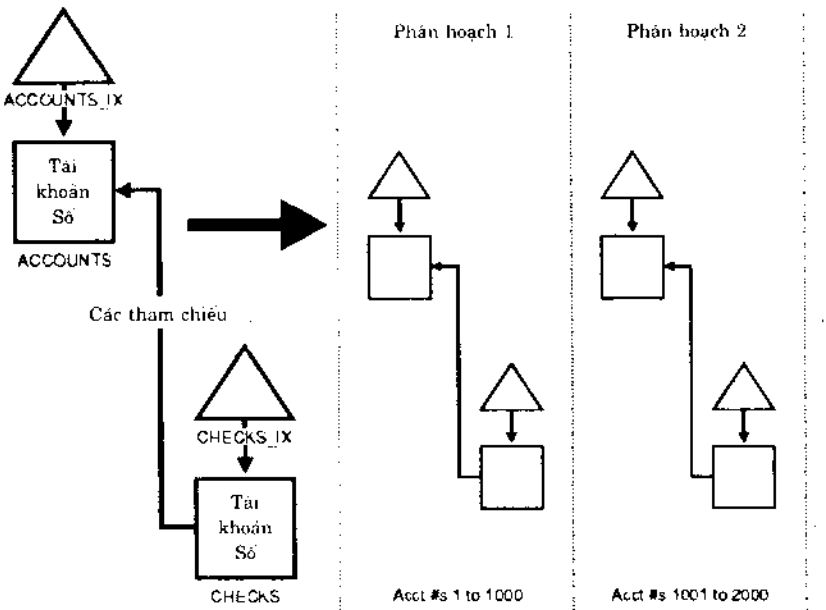
Phân hoạch bảng

Các nguyên tắc các bảng phân hoạch đơn giản:
Một bảng có thể được phân hoạch phạm vi, với điều kiện:
Nó không là thành phần của một cụm liên kết.
Nó không chứa các kiểu dữ liệu LOBs, LONG hoặc LONG RAW hay các kiểu đối tượng.
Nó không phải là một bảng có tổ chức chỉ mục.
Bạn có thể trộn các chỉ mục có phân hoạch và không có phân hoạch với các bảng có phân hoạch và không có phân hoạch:
Một bảng có phân hoạch có thể có các chỉ mục phân hoạch và/hoặc không phân hoạch.
Một bảng không có phân hoạch có thể có các chỉ mục có phân hoạch và/hoặc không có phân hoạch. (Chỉ các chỉ mục toàn cục có thể được tạo trên các bảng không có phân hoạch - Xem "Các Chỉ Mục Toàn Cục").

Các thuộc tính vật lý của các phân hoạch bảng

Các thuộc tính vật lý mặc định được ấn định đầu tiên khi câu lệnh CREATE TABLE tạo một bảng được phân hoạch. Vì không có đoạn nào tương ứng với chính bảng được phân hoạch ấy, những thuộc tính này sẽ chỉ được sử dụng để phái sinh các thuộc tính vật lý của các phân hoạch thành viên. Những thuộc tính vật lý mặc định có thể được sửa đổi sau bằng cách sử dụng ALTER TABLE MODIFY DEFAULT ATTRIBUTES.

Hình 9-3 Các bảng được phân hoạch tương đương và các chỉ mục



Những thuộc tính vật lý của các phân hoạch bảng được tạo bằng CREATE TABLE và ALTER TABLE ADD PARTITION được xác định như sau:
Những giá trị của các thuộc tính vật lý được ấn định (rõ ràng hoặc mặc định) đối với bảng cơ sở được sử dụng bất kỳ khi nào giá trị của một thuộc tính phân hoạch tương ứng đã được ấn định.

Các thuộc tính vật lý của một phân hoạch bảng có sẵn có thể được sửa đổi bằng ALTER TABLE MOVE PARTITION và ALTER TABLE MODIFY PARTITION. Các thuộc tính kết quả được xác định như sau:

Các giá trị của những thuộc tính vật lý của một phân hoạch trước khi câu lệnh ấy đã phát hành được sử dụng bất kỳ khi nào một giá trị mới không được ấn định.

Lưu ý:
ALTER TABLE MOVE PARTITION có thể được sử dụng để thay đổi không gian bảng trong đó một phân hoạch cư trú.

Các thuộc tính vật lý của các phân hoạch bảng được ALTER TABLE SPLIT PARTITION tạo ra được xác định như sau:
Những giá trị của các thuộc tính vật lý của phân hoạch được tách được sử dụng bất kỳ khi nào một giá trị mới không được ấn định. (Điều này cũng áp dụng để tách chỉ mục toàn cục - các thuộc tính mất được thừa kế từ phân hoạch chỉ mục đang được tách ra.)

Những thuộc tính vật lý của tất cả phân hoạch của tính toán những phân hoạch của một bảng có thể được sửa đổi bằng ALTER TABLE, tham số tên bảng NOLOGGING thay đổi chế độ ghi sổ của tên bảng thành NOLOGGING.

Phân hoạch chỉ mục

Các nguyên tắc phân hoạch các chỉ mục giống với những nguyên tắc đối với các bảng:
Một chỉ mục có thể được phân hoạch phạm vi với các ngoại trừ sau:

- Chỉ mục không phải là một chỉ mục cụm liên kết.
- Chỉ mục không được xác định trên một bảng có cụm liên kết.
- Một chỉ mục bitmap trên một bảng được phân hoạch phải là một chỉ mục cục bộ.
-

Bạn có thể trộn các chỉ mục có các bảng có phân hoạch và không có phân hoạch:

- Một bảng được phân hoạch có thể có các chỉ mục có phân hoạch và/hoặc không có phân hoạch.
- Một bảng không có phân hoạch có thể có các chỉ mục B*-tree có phân hoạch và/hoặc không có phân hoạch.
- Các chỉ mục bitmap trên các bảng không có phân hoạch không thể được phân hoạch phạm vi.

Tuy nhiên, các chỉ mục có phân hoạch phức tạp hơn các bảng có phân hoạch vì có bốn kiểu chỉ mục có phân hoạch phạm vi: có tiền tố cục bộ, không có tiền tố cục bộ, có tiền tố toàn cục và không có tiền tố toàn cục. Những kiểu này được mô tả bên dưới. Oracle hỗ trợ ba trong bốn kiểu (các chỉ mục không có tiền tố toàn cục không hữu dụng trong các ứng dụng thực).

Các chỉ mục cục bộ

Trong một chỉ mục cục bộ, tất cả các khóa trong một phân hoạch chỉ mục đặc biệt chỉ tham chiếu các hàng được lưu trữ trong một phân hoạch bảng cơ sở. Một chỉ mục cục bộ được tạo bằng cách ấn định thuộc tính LOCAL.

Oracle cấu tạo chỉ mục cục bộ để nó được phân hoạch tương ứng với bảng cơ sở. Oracle phân hoạch phạm vi chỉ mục trên cùng các bảng như bảng cơ sở để tạo cùng số phân hoạch và tạo cho chúng cùng các biên phân hoạch như các phân hoạch tương ứng của bảng cơ sở. Oracle còn duy trì phân hoạch chỉ mục tự động bằng các phân hoạch trong bảng cơ sở được bổ sung, bỏ hoặc tách. Điều này bảo đảm là chỉ mục vẫn được phân hoạch tương đương với bảng ấy.

Phân hoạch tương đương một bảng và chỉ mục của nó có các ưu điểm sau:

- Chỉ một phân hoạch chỉ mục có tác dụng khi một hoạt động bảo trì (thay vì SPLIT PARTITION) được thực hiện trên một phân hoạch bảng cơ sở.
- Thời gian của một hoạt động bảo trì phân hoạch duy trì tỷ lệ với kích thước phân hoạch nếu bảng ấy được phân hoạch chỉ có những chỉ mục cục bộ.
- Các chỉ mục cục bộ hỗ trợ tính độc lập phân hoạch.
- Các chỉ mục cục bộ hỗ trợ loại thải dữ liệu cũ và nhập dữ liệu mới vào các bảng lịch sử.

Oracle có thể khai thác sự kiện một chỉ mục cục bộ được phân hoạch tương đương với bảng cơ sở để phát sinh các kế hoạch truy xuất truy vấn tốt hơn.

Các chỉ mục cục bộ đơn giản hóa tác vụ phục hồi không hoàn toàn không gian bảng. Để phục hồi một phân hoạch của bảng thành giờ và ngày tháng, bạn cũng phải phục hồi các mục nhập chỉ mục tương ứng vào cùng giờ và ngày tháng. Cách duy nhất để thực hiện điều này với một chỉ mục cục bộ; sau đó bạn có thể phục hồi bảng tương ứng và các phân hoạch chỉ mục cùng với nhau.

Các chỉ mục có tiền tố cục bộ

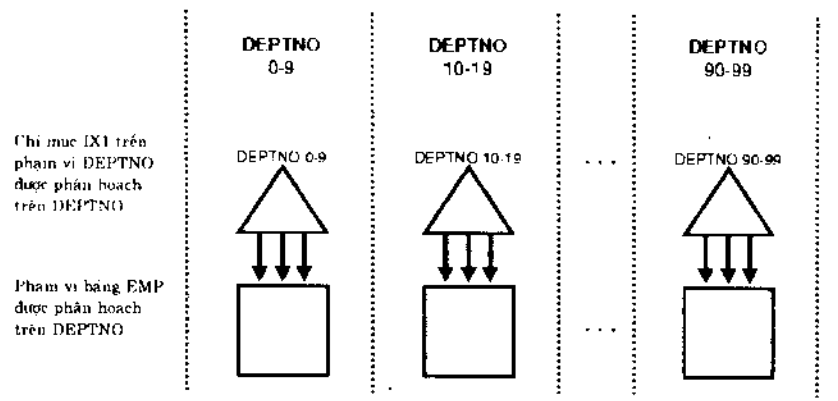
Một chỉ mục cục bộ có tiền tố nếu nó được phân hoạch trên tiền tố bên trái của các cột chỉ mục.

Thí dụ, nếu các bảng kinh doanh và chỉ mục sales_ix cục bộ của nó được phân hoạch trên cột week_num, thì chỉ mục sales_ix có tiền tố cục bộ nếu nó được xác định trên các cột (week_num, xaction_num). Nói cách khác, nếu chỉ mục sales_ix được xác định trên cột product_num thì nó không có tiền tố.

Hình 9-4 trình bày một thí dụ về chỉ mục có tiền tố cục bộ.

Các chỉ mục có tiền tố cục bộ có thể duy nhất hoặc không duy nhất.

Hình 9-4 Chỉ mục có tiền tố cục bộ



Các chỉ mục không có tiền tố cục bộ
Một chỉ mục cục bộ là chỉ mục không có tiền tố cục bộ nếu nó không được phân hoạch trên tiền tố bên trái của các cột chỉ mục.

Bạn không thể có một chỉ mục không có tiền tố cục bộ duy nhất trừ khi khóa chỉ mục là một tập hợp con của khóa phân hoạch.

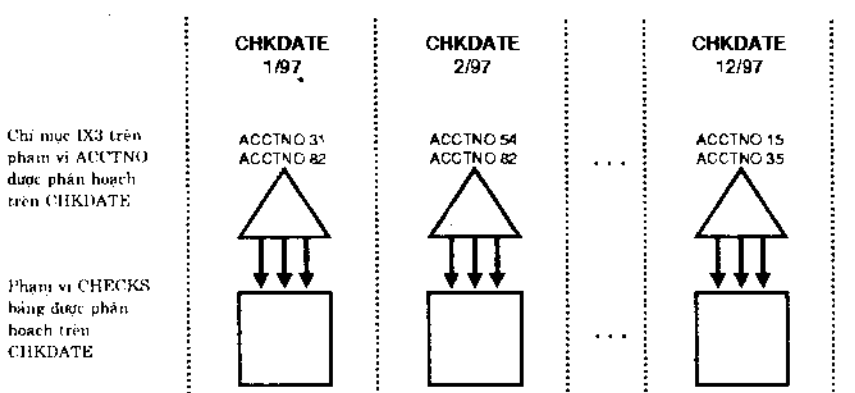
Hình 9-5 trình bày một thí dụ về một chỉ mục không có tiền tố cục bộ.

Các chỉ mục toàn cục

Trong một *global index* (chỉ mục toàn cục), các khóa trong phân hoạch chỉ mục đặc biệt có thể tham chiếu các hàng được lưu trữ trong hơn một phân hoạch bảng cơ sở. Một chỉ mục toàn cục được tạo bằng cách ấn định thuộc tính GLOBAL (điều này mặc định). Người dùng chịu trách nhiệm xác định phân hoạch ban đầu của chỉ mục toàn cục vào lúc tạo và duy trì phân hoạch. Các phân hoạch chỉ mục có thể được bỏ hoặc tách ra nếu cần.

Bình thường, một chỉ mục toàn cục không được phân hoạch tương đương với bảng cơ sở. Không có cách nào ngăn chặn một chỉ mục không được phân hoạch tương đương với bảng cơ sở, nhưng Oracle không khai thác phân hoạch tương đương khi phát sinh các kế hoạch truy vấn hoặc thực hiện các hoạt động bảo trì phân hoạch. Vì vậy một chỉ mục được phân hoạch tương đương với bảng cơ sở sẽ được tạo là LOCAL.

Hình 9-5 Chỉ mục không có tiền tố cục bộ



Một chỉ mục toàn cục chứa (về mặt khái niệm) một B*-tree đơn với các mục cho tất cả những hàng trong tất cả các phân hoạch. Mỗi phân hoạch chỉ mục có thể chứa các khóa tham chiếu nhiều phân hoạch khác nhau trong bảng.

Phân hoạch cao nhất của chỉ mục toàn cục phải có biên phân hoạch mà tất cả giá trị là MAXVALUE. Điều này bảo đảm tất cả các hàng trong bảng cơ sở có thể được biểu thị trong chỉ mục ấy.

Chỉ mục toàn cục có tiền tố nếu nó được phân hoạch trên tiền tố bên trái của các cột chỉ mục. (Xem Hình 9-6 là một thí dụ). Một chỉ mục toàn cục không có tiền tố nếu nó không được phân hoạch trên tiền tố bên trái của các cột chỉ mục. Oracle không hỗ trợ các chỉ mục không có tiền tố.

Các chỉ mục có tiền tố toàn cục có thể là duy nhất hoặc không duy nhất.
Các chỉ mục toàn cục khó quản lý hơn các chỉ mục cục bộ:

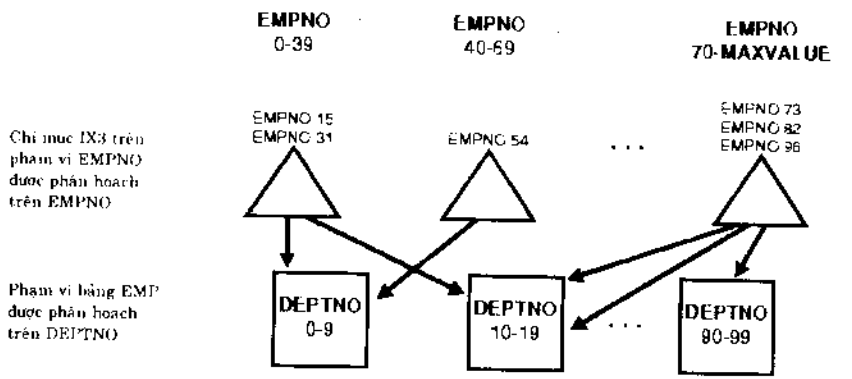
Khi dữ liệu trong phân hoạch bảng cơ sở được di chuyển hoặc xóa (SPLIT, .MOVE, DROP hoặc TRUNCATE), thì tất cả các phân hoạch của một chỉ mục toàn cục đều bị ảnh hưởng. Do đó các chỉ mục toàn cục làm bảo trì phân hoạch (kể cả các thiết kế lại những chỉ mục toàn cục hoặc các phân hoạch chỉ mục) có được thời gian tỷ lệ với kích thước bảng thay vì không phân hoạch và chúng không hỗ trợ tính độc lập phân hoạch.

Khi một phân hoạch bảng cơ sở được phục hồi giờ và ngày tháng, thì tất cả các mục tương ứng trong một chỉ mục toàn cục phải được phục hồi cùng giờ và ngày tháng. Bởi vì các mục nhập này phải được rải qua tất cả các

phẫn hoạch của chỉ mục (được trộn với các mục nhập cho các phân hoạch khác hiện chưa được phục hồi), không có cách nào thực hiện điều này ngoại trừ bằng cách tạo lại toàn bộ chỉ mục toàn cục.

Các chỉ mục không có phân hoạch được coi là các chỉ mục có tiền tố toàn cục.

Hình 9-6 Chỉ mục có tiền tố toàn cục



Tóm tắt các kiểu chỉ mục có phân hoạch

Bảng 9-1 tóm tắt ba kiểu chỉ mục có phân hoạch mà Oracle hỗ trợ.

Nếu một chỉ mục là cục bộ, thì nó được phân hoạch tương đương với bảng cơ sở; trái lại nó là toàn cục.

Một chỉ mục có tiền tố được phân hoạch trên tiền tố bên trái của các cột chỉ mục; trái lại nó không có tiền tố.

Bảng 9-1 Các kiểu của các chỉ mục được phân hoạch

Kiểu chỉ mục	Chỉ mục được phân hoạch tương đương với bảng	Chỉ mục được phân hoạch trên tiền tố bên trái của các cột chỉ mục	Thuộc tính UNIQUE cho phép	Thí dụ		
				Bảng được phân hoạch trên cột	Các cột có chỉ mục	Chỉ mục được phân hoạch trên cột
Có tiền tố cục bộ	Có	Có	Có	A	A B	A
Không có tiền tố cục bộ	Có	Không	Có[1]	A	B	A
Có tiền tố toàn cục	Không[2]	Có	Có	A	B	B
Không có tiền tố toàn cục[3]						

[1] Đối với một chỉ mục không có tiền tố cục bộ duy nhất, chỉ mục ấy là một tập hợp của khóa phân hoạch.
[2] Dù một chỉ mục có phân hoạch toàn cục có thể được phân hoạch tương đương với bảng cơ sở, Oracle không DROP hoặc SPLIT PARTITION.
[3] Kiểu chỉ mục này không được hỗ trợ.

Tầm quan trọng của các chỉ mục không có tiền tố

Các chỉ mục không có tiền tố đặc biệt hữu dụng trong các cơ sở dữ liệu lịch sử. Trong một bảng chứa dữ liệu lịch sử nó phổ biến đối với chỉ mục được xác định trên một cột để hỗ trợ các đòi hỏi của truy xuất nhanh bằng cột đó, nhưng được phân hoạch trên một cột khác (cột tương tự như bảng cơ sở) để hỗ trợ khoảng thời gian để loại thải dữ liệu cũ và nhập vào dữ liệu mới.

Xét bảng kinh doanh được biểu thị trong Hình 9-1 ("Bảng SALES được phân hoạch bằng tuần". Nó chứa dữ liệu có giá trị một năm được chia thành 13 phân hoạch. Nó là phạm vi được phân hoạch trên week_no, bốn tuần thành một phân hoạch. Bạn có thể tạo một sales_ix chỉ mục cục bộ không có tiền tố trên các hoạt động kinh doanh. Chỉ

mục sales_ix được xác định trên acct no vì có các truy vấn cần truy xuất nhanh dữ liệu theo số hiệu tài khoản. Tuy nhiên, nó được phân hoạch trên week_no để so khớp với các báng kinh doanh. Cứ bốn tuần các phân hoạch của các hoạt động kinh doanh và sales_ix được bỏ và các hoạt động kinh doanh mới được bổ sung.

Các ngầm định hiệu suất của các chỉ mục có tiền tố và không có tiền tố

Duyệt một chỉ mục không có tiền tố tốn phí hơn duyệt một chỉ mục có tiền tố.

Nếu một chỉ mục có tiền tố (cục bộ hoặc toàn cục) và Oracle được biểu thị với một thuộc tính liên quan các cột chỉ mục, thì xóa bỏ phân hoạch có thể hạn chế ứng dụng của thuộc tính với tập hợp con của các phân hoạch chỉ mục.

Thí dụ, trong Hình 9-4 "Chỉ Mục Có Tiền Tố Cục Bộ" nếu thuộc tính là DEPTNO=15, bộ tối ưu hóa biết áp dụng chỉ thuộc tính ấy vào phân hoạch thứ hai của chỉ mục ấy. (Nếu thuộc tính ấy liên quan đến một biến kết gán, thì bộ tối ưu hóa sẽ không biết chính xác phân hoạch nào nhưng nó có thể vẫn biết chỉ có một phân hoạch liên quan, trong trường hợp vào thời gian thực hiện chỉ một phân hoạch chỉ mục sẽ được truy xuất).

Khi một chỉ mục không có tiền tố, Oracle thường phải áp dụng một thuộc tính liên quan đến các cột chỉ mục đối với tất cả các phân hoạch chỉ mục N. Điều này đòi hỏi truy tìm một khóa đơn hoặc để thực hiện duyệt phạm vi chỉ mục. Đối với một phạm vi duyệt, Oracle cũng phải kết hợp thông tin từ các phân hoạch chỉ mục. Thí dụ, trong Hình 9-5 ("Chỉ Mục Không Có Tiền Tố Cục Bộ") một chỉ mục cục bộ được phân hoạch trên CHKDATE với

một khóa chỉ mục trên ACCTNO. Nếu thuộc tính là ACCTNO=31, Oracle thăm dò tất cả 12 phân hoạch chỉ mục.

Đương nhiên, nếu còn có thuộc tính trên các cột phân hoạch thì nhiều thăm dò chỉ mục có thể không cần thiết. Oracle khai thác ưu điểm của sự kiện chỉ mục cục bộ được phân hoạch tương đương với bảng cơ sở để thăm dò các phân hoạch dựa trên khóa phân hoạch. Thí dụ, nếu thuộc tính trong Hình 9-5 là CHKDATE<3/97, Oracle chỉ phải thăm dò hai phân hoạch.

Vì vậy đối với một chỉ mục không có tiền tố, nếu khóa phân hoạch là thành phần của mệnh đề WHERE (nhưng không phải khóa chỉ mục) bộ tối ưu hóa xác định các phân hoạch chỉ mục nào để thăm dò dựa trên phân hoạch bảng cơ sở.

Khi nhiều truy vấn và các câu lệnh DML sử dụng các khóa của các chỉ mục không tiền tố cục bộ phải thăm dò tất cả các phân hoạch chỉ mục, điều này làm giảm mức độ của tính độc lập phân hoạch được các chỉ mục như thế cung cấp.

Các nguyên tắc chủ đạo đối với những chỉ mục phân hoạch

Khi quyết định cách phân hoạch các chỉ mục trên một bảng, bạn phải xét sự pha trộn các ứng dụng cần thiết để truy xuất bảng ấy. Có một trao đổi giữa hiệu suất trên một tay (one hand), tính khả dụng và tính năng quản lý trên tay kia.

Sau là một số nguyên tắc chủ đạo bạn nên xét:
Đối với các ứng dụng OLTP:

Các chỉ mục toàn cục và các chỉ mục có tiền tố cục bộ cung cấp hệ điều hành tốt hơn các chỉ mục không có tiền tố cục bộ vì chúng giảm tối đa số lần thăm dò phần cứng chỉ mục.

Các chỉ mục cục bộ hỗ trợ khả dụng hơn khi có các hoạt động bảo trì phân hoạch trên bảng ấy. Các chỉ mục không có tiền tố cục bộ rất hữu dụng đối với các CSDL lịch sử.

Đối với các ứng dụng DSS, các chỉ mục không có tiền tố có thể tăng hiệu suất vì nhiều phân hoạch chỉ mục có thể được duyệt song song bằng những truy vấn phạm vi trên khóa chỉ mục ấy.

> Thí dụ, một truy vấn sử dụng thuộc tính "ACCTNO từ 40 tới 45" trên bảng CHECKS của Hình 9-5 ("Chỉ Mục Không Có Tiền Tố Cục Bộ" tạo cho duyệt song song tất cả những phân hoạch của chỉ mục IX3 không có tiền tố. Nói cách khác, một truy vấn sử dụng thuộc tính "DEPTNO từ 40 tới 45" trên bảng DEPTNO của Hình 9-4 ("Chỉ Mục Có Tiền Tố Cục Bộ" không thể được duyệt song song vì nó truy xuất một phân hoạch đơn của chỉ mục IX1 có tiền tố.

Đối với các bảng lịch sử, các chỉ mục phải là cục bộ nếu có thể. Điều này giới hạn tác động của những điều hoạt phân hoạch bỏ lịch trình thường xuyên.

Các chỉ mục duy nhất trên các cột thay vì các cột phân hoạch phải là toàn cục vì các chỉ mục không có tiền tố cục bộ duy nhất có khóa không chứa khóa phân hoạch mà chúng không hỗ trợ.

Các thuộc tính vật lý của các phân hoạch chỉ mục

Các thuộc tính vật lý mặc định được xác định ban đầu khi một câu lệnh CREATE INDEX tạo chỉ mục có phân hoạch. Vì không có đoạn tương ứng với chính chỉ mục

được phân hoạch, các thuộc tính này chỉ được sử dụng trong phái sinh các thuộc tính vật lý của các phân hoạch thành viên. Những thuộc tính vật lý mặc định có thể được sửa đổi sau bằng cách sử dụng ALTER INDEX.

Các thuộc tính vật lý của các phân hoạch tạo bằng CREATE INDEX được xác định như sau:
Các giá trị của những thuộc tính vật lý được xác định (rõ ràng hoặc theo mặc định) cho chỉ mục được sử dụng bất kỳ khi nào giá trị của thuộc tính phân hoạch tương ứng không được xác định. Xử lý của thuộc tính TABLESPACE của các phân hoạch chỉ mục LOCAL cấu tạo một ngoại lệ quan trọng đối với nguyên tắc này trong đó không có giá trị TABLESPACE do người dùng ấn định, điều đó thuộc phân hoạch tương ứng của bảng cơ bản sẽ được sử dụng.

Các thuộc tính vật lý (thay vì TABLESPACE được giải thích như trên) của các phân hoạch những chỉ mục cục bộ được tạo trong tiến trình xử lý ALTER TABLE ADD PARTITION được xác lập thành các thuộc tính vật lý mặc định của mỗi chỉ mục.

Các thuộc tính vật lý (thay vì TABLESPACE, như được giải thích ở trên) của các phân hoạch chỉ mục được ALTER TABLE SPLIT PARTITION tạo được xác định như sau:
Những giá trị của các thuộc tính vật lý (thay vì TABLESPACE, như được giải thích ở trên) của phân hoạch chỉ mục tách riêng được sử dụng.

Những thuộc tính vật lý của một phân hoạch chỉ mục có thể được sửa đổi bởi ALTER INDEX MODIFY PARTITION và ALTER INDEX REBUILD PARTITION. Những thuộc tính kết quả được xác định như sau:

Các giá trị của những thuộc tính vật lý của phân hoạch trước khi câu lệnh đã được đưa ra được sử dụng bất kỳ khi nào một giá trị mới đã được ấn định. Lưu ý là ALTER INDEX REBUILD PARTITION có thể được dùng để thay đổi không gian bảng trong đó phân hoạch cư trú.

Các thuộc tính vật lý của các phân hoạch toàn cục được tạo bằng ALTER INDEX SPLIT PARTITION được xác định như sau:
Các giá trị của những thuộc tính vật lý của phân hoạch tách riêng được sử dụng bất kỳ khi nào một giá trị mới không được ấn định.

Các thuộc tính vật lý của tất cả những phân hoạch của một chỉ mục có thể được sửa đổi bằng ALTER INDEX, thí dụ, ALTER INDEX *indexname* NOLOGGING thay đổi chế độ ghi sổ nhật ký của tất cả những phân hoạch thuộc *indexname* thành NOLOGGING.

Các khóa phân hoạch DML

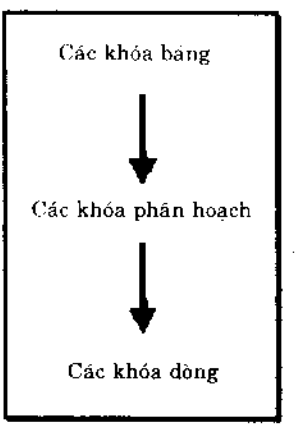
Các khóa bảng DML đồng bộ hóa các câu lệnh DML (INSERT, UPDATE và DELETE) với các câu lệnh DDL và các câu lệnh LOCK TABLE. Các khóa bảng DML còn đồng bộ hóa các lệnh DDL và LOCK TABLE với nhau.

Để cung cấp tính độc lập phân hoạch cho DDL và các điều hoạt tiện ích, Oracle hỗ trợ các khóa phân hoạch DML. Tính độc lập phân hoạch cho bạn thực hiện các điều hoạt DDL và các điều hoạt tiện ích mà không trở thành trạng thái thụ động trên các phân hoạch khác.

Mục đích của một khóa phân hoạch để bảo vệ dữ liệu trong một phân hoạch riêng khi nhiều người dùng đang truy xuất đồng thời phân hoạch đó hoặc các phân hoạch khác trong bảng ấy.

Các khóa phân hoạch rơi vào giữa các khóa bảng và các khóa hàng trong phân cấp khóa DML, như được trình bày ở Hình 9-7.

Hình 9-7 Phân cấp khóa DML



Các khóa phân hoạch có thể cần thiết trong các chế độ tương tự như các khóa bảng: Share (S), Exclusive (X), Row Share (SS), Row Exclusive (SX) và Share Row Exclusive (SSX).

Các nhận xét hiệu suất đối với Oracle Parallel Server

Đề nghị một mức phụ của khóa DML có thể ảnh hưởng đến hiệu suất của những giao dịch ngắn trong môi trường Oracle Parallel Server vì các thông điệp phụ được gửi tới Distributed Lock Manager.

Để tăng hiệu suất trong môi trường Oracle Parallel Server, bạn có thể tắt khóa DML trên các bảng được lựa với câu lệnh ALTER TABLE DISABLE TABLE LOCK vô hiệu hóa

cả bảng và các khóa DML phân hoạch. Các câu lệnh DDL không được cho phép khi khóa DML được vô hiệu hóa.

Thông tin bổ sung :
Xem Oracle8 Parallel Server Concepts và Administrator.

Các hoạt động bảo trì

Phần này gồm các chủ đề sau:

- Các Hoạt Động Bảo Trì Phân Hoạch
- Các Chỉ Mục Quản Lý
- Các Đặc Quyền Đối Với Các Bảng Được Phân Hoạch Và Các Chỉ Mục
- Kiểm Soát Các Bảng Được Phân Hoạch Và Các Chỉ Mục

Đối với các mục đích của chương này, một hoạt động bảo trì là câu lệnh DDL hoặc tiện tích (như Export, Import, SQL*Loader) để thay đổi định nghĩa của bảng hoặc chỉ mục và/hoặc tải hàng loạt dữ liệu hoặc không tải.

Phần lớn các hoạt động bảo trì trên các bảng không phân hoạch và các chỉ mục còn hoạt động trên các bảng có phân hoạch và các chỉ mục. Thí dụ, DROP TABLE có thể giảm một bảng có phân hoạch và Export có thể Export một bảng có phân hoạch. Tuy nhiên, một số hoạt động bảo trì phải được thực hiện trên các phân hoạch riêng thay vì toàn bộ bảng có phân hoạch hoặc chỉ mục. Thí dụ, ALTER TABLE ALLOCATE EXTENT không thể được sử dụng cho một bảng được phân hoạch; thay vào đó, bạn sử dụng ALTER TABLE MODIFY PARTITION ALLOCATE EXTENT cho phân hoạch hoặc các phân hoạch cần các khối lưu riêng mới.

Các hoạt động bảo trì được xét nhanh nếu thời gian yêu cầu của chúng không bị ảnh hưởng bởi kích thước (số những bản ghi) của các đối tượng giản đồ mà chúng hoạt động. Các hoạt động bảo trì nhanh chỉ có kết quả trong từ điển và các thay đổi phần đầu đoạn và không làm cho duyệt dữ liệu và các cập nhật dữ liệu. Chúng cần được thực hiện trong một thời gian ngắn (số giây). Thí dụ, RENAME là một điều hoạt nhanh khi CREATE INDEX không phải là một điều hoạt nhanh.

Các hoạt động bảo trì phân hoạch

Một hoạt động bảo trì phân hoạch sửa đổi một phân hoạch của bảng được phân hoạch hoặc chỉ mục. Thí dụ, bạn có thể bổ sung một phân hoạch mới vào bảng có sẵn hoặc bạn có thể di chuyển phân hoạch tới một không gian bảng khác để cân bằng tải I/O tốt hơn hay bạn có thể tải một phân hoạch.

Một số hoạt động bảo trì phân hoạch là các sự kiện có kế hoạch. Thí dụ, trong một CSDL lịch sử, người điều hành CSDL (DBA) định kỳ bỏ các phân hoạch cũ nhất khỏi CSDL và bổ sung một tập hợp các phân hoạch mới. Điều này làm bỏ và bổ sung điều hoạt xảy ra theo phương thức có lịch trình đều đặn. Một thí dụ khác về hoạt động bảo trì có kế hoạch là Export/Import theo chu kỳ đối với dữ liệu được tái liên kết và bỏ phân đoạn.

Các hoạt động bảo trì khác là các sự kiện không có kế hoạch, phải phục hồi từ các vấn đề ứng dụng hoặc hệ thống. Thí dụ, hoạt động giao dịch không ngoài ý muốn có thể buộc DBA tách phân hoạch để cân bằng lại tải I/O hoặc DBA có thể phải thiết lập lại một hoặc nhiều phân hoạch chỉ mục.

Các hoạt động bảo trì phân hoạch:

- Bổ sung một phân hoạch bảng vào bảng có sẵn
- Sửa đổi một phân hoạch - thay đổi những thuộc tính vật lý của một phân hoạch
- Di chuyển một phân hoạch bảng - di chuyển nó vào không gian bảng khác hoặc liên kết nó lại hay thay đổi bất kỳ tham số của nó (kể cả bất kỳ tham số thời gian tạo của nó)
- Đặt tên lại một phân hoạch
- Bỏ một phân hoạch
- Xóa bỏ một phân hoạch bảng (có hoặc không có không gian hồi phục
- Tách một phân hoạch có sẵn thành hai phân hoạch
- Nạp dữ liệu vào một phân hoạch bảng
- Export dữ liệu từ một phân hoạch bảng
- Import một phân hoạch bảng
- Thiết lập lại một phân hoạch chỉ mục

Mô hình đồng thời đối với các hoạt động bảo trì

Mô hình bảo trì được mô tả ở phần này xác định khi nó có thể tiến hành hơn một DDL và tiện ích trên cùng đối tượng giản đồ cùng một lúc. Nó cũng truy vấn xác định và các hoạt động DDL nào có thể được tiến hành đồng thời với các hoạt động DDL và tiện ích.

Mô hình áp dụng vào tất cả các câu lệnh DDL. Nó cũng áp dụng vào các tiện ích như SQL*Loader.

Các hoạt động một step và ba step

Có hai kiểu hoạt động bảo trì, một Step và ba Step.

Các điều hoạt một Step:

Những hoạt động DML này khóa bảng bị ảnh hưởng ở chế độ Exclusive (X). Các hoạt động chỉ mục khóa bảng cơ sở. Chúng cũng lưu giữ các khóa từ điển Exclusive đối với thời gian điều hoạt.
Những hoạt động này nhanh (thí dụ, ALTER TABLE ADD PARTITION) hoặc chúng không cung ứng khả năng tiến hành các hoạt động đồng thời (thí dụ, cột ALTER TABLE ADD).
Tất cả hoạt động chỉ mục là một Step ngoại trừ CREATE INDEX và ALTER INDEX REBUILD (và ALTER INDEX DROP/SPLIT PARTITION, nếu phân hoạch toàn cục được bỏ hoặc tách là Usable).
Tất cả các câu lệnh Oracle DDL là một Step ngoại trừ CREATE INDEX và MOVE, SPLIT, REBUILD. hoặc EXCHANGE PARTITION.

Các điều hoạt ba Step:

Những hoạt động này giành được các khóa DML ít đòi hỏi trên các bảng có tác động. Chúng chỉ khóa một phân hoạch ở chế độ Exclusive (X) hoặc nếu chúng khóa toàn bộ bảng, chúng khóa nó trong chế độ S hoặc SS hay SX.
Các điều hoạt gồm ba Step:

- Step 1: đọc từ điển khi lưu giữ các khóa từ điển Share. Step 1 mất ít thời gian (số giây). Vào cuối Step này, có được các khóa DML thích hợp, sau đó các khóa từ điển được giải phóng.
- Step 2: duyệt hoặc cập nhật bảng hoặc các bản ghi chỉ mục. Step 2 có thể mất thời gian hơn (hàng phút hoặc hàng giờ).
- Step 3: cập nhật từ điển khi lưu giữ các khóa từ điển Exclusive. Step 3 mất ít thời gian (giây).

Những hoạt động này tiến hành lâu, nhưng chúng cho phép các hoạt động khác tiến hành đồng thời. Chính xác những hoạt động nào có thể tiến hành tùy thuộc vào các khóa DML cụ thể đạt được bằng câu lệnh, được giải thích như sau:

Các hoạt động sau là Step 3:

ALTER TABLE MOVE PARTITION, ALTER TABLE SPLIT PARTITION, ALTER TABLE EXCHANGE PARTITION, Direct Path Load Table Partition: những câu lệnh này khóa bảng cơ sở ở chế độ Row Exclusive (SX) và phân hoạch ở chế độ Exclusive (X).

CREATE INDEX và ALTER INDEX REBUILD PARTITION (đối với chỉ mục toàn cục): những câu lệnh này khóa bảng cơ sở ở chế độ Shared (S).

ALTER INDEX REBUILD PARTITION (đối với chỉ mục cục bộ): câu lệnh này khóa bảng cơ sở ở chế độ Row Share (SS) và phân hoạch bảng cơ sở ở chế độ Shared (S).

ALTER TABLE MODIFY PARTITION REBUILD UNUSABLE LOCAL INDEXES: những câu lệnh này khóa bảng cơ sở ở chế độ Row Share (SS) và phân hoạch bảng cơ sở ở chế độ Shared (S).

Cuối cùng, một số hoạt động có thể thực hiện theo giao thức một Step hoặc ba Step:

Nhóm này gồm ALTER TABLE DROP PARTITION và ALTER TABLE TRUNCATE PARTITION.

> Nếu bảng bị thay đổi không có các chỉ mục toàn cục xác định nó hoặc nếu nó được tham chiếu bằng các hạn định tham chiếu được cho phép, thì các câu lệnh trong nhóm này thực hiện bằng cách sử dụng giao thức ba Step. Trong trường hợp thứ hai, bảng cơ sở được khóa ở chế độ Row Exclusive (SX) và phân hoạch được khóa ở chế độ Exclusive (X).

ALTER INDEX SPLIT PARTITION (chỉ được cho phép các chỉ mục toàn cục).

Nếu phân hoạch được tách là USABLE, câu lệnh thực hiện theo giao thức ba Step và các phân hoạch do SPLIT là USABLE. Mặt khác, nếu phân hoạch được tách là UNUSABLE, thì điều hoạt thực hiện theo giao thức 1 Step và các phân noạch kết quả cũng được đánh dấu UNUSABLE.

ALTER INDEX DROP PARTITION (chỉ được cho phép các chỉ mục toàn cục).

Nếu phân hoạch được bỏ là USABLE, thì câu lệnh thực hiện theo giao thức ba Step; trái lại nó theo giao thức một Step.

Conventional Path SQL*Loader và Import sử dụng SQL INSERT vì vậy chúng được xếp loại là các điều hoạt DML cho các mục đích của mô hình. Export sử dụng SQL SELECT vì vậy nó được xếp loại là điều hoạt truy vấn.

Các hoạt động có thể tiến hành đồng thời

Các nguyên tắc trong phần này có thể được phái sinh từ các định nghĩa về các điều hoạt một Step và ba Step.

Khi một điều hoạt một Step tiến hành:
Bạn có thể tiến hành các truy vấn trên bảng.
Bạn không thể tiến hành bất kỳ điều hoạt khác (DDL, utility hoặc DML).

Vì các truy vấn (các điều hoạt READ) không dùng các khóa DML, các truy vấn được cho phép trên phân hoạch hiện là SPLIT hoặc MOVE khi SPLIT hoặc MOVE đang được xử lý. Tuy nhiên, các đoạn hiện hành được bỏ ở cuối điều hoạt ấy và không gian có thể được sử dụng lại. Một lỗi được phát tín hiệu nếu không gian ấy được sử dụng lại.

Khi ALTER TABLE MOVE PARTITION, ALTER TABLE SPLIT PARTITION, ALTER TABLE EXCHANGE PARTITION hoặc Direct Path Load Table Partition đang tiến hành trên một phân hoạch:
Bạn có thể di chuyển, tách, chuyển đổi hoặc định hướng nạp đường dẫn các phân hoạch khác vào cùng bảng.
Bạn có thể tiến hành các truy vấn trên bảng.
Bạn có thể thực hiện các điều hoạt DML trên bảng ấy với điều kiện chúng không ghi vào phân hoạch đó.
Bạn có thể thiết lập lại bất kỳ phân hoạch chỉ mục nào khác hơn các phân hoạch chỉ mục tương ứng với phân hoạch đó.
Bạn không thể tiến hành bất kỳ hoạt động nào trên bảng hoặc các chỉ mục của nó thay vì các hoạt động được liệt kê ở trên.

Khi CREATE INDEX hoặc ALTER INDEX REBUILD PARTITION hay ALTER INDEX DROP/SPLIT PARTIITON được áp dụng vào phân hoạch Usable (đối với chỉ mục toàn cục) được tiến hành:
Bạn có thể tiến hành các truy vấn trên bảng cơ sở.
Bạn có thể tạo các chỉ mục khác trên bảng, thiết lập lại các phân hoạch trong các chỉ mục có sẵn hoặc bỏ hay tách các phân hoạch có thể sử dụng ở các chỉ mục có sẵn.
Bạn không thể thực hiện bất kỳ hoạt động DML trên bảng hoặc tiến hành bất kỳ hoạt động bảo trì nào trên bảng hoặc các chỉ mục của nó thay vì các hoạt động được liệt kê ở trên.

Khi ALTER INDEX REBUILD PARTITION (đối với chỉ mục cục bộ) được tiến hành trên phân hoạch tương ứng với phân hoạch bảng cơ sở:
Bạn có thể di chuyển, tách hoặc định hướng nạp đường dẫn bất kỳ phân hoạch nào trừ phân hoạch bảng cơ sở.

Bạn có thể tiến hành các truy vấn trên bảng.
Bạn có thể thực hiện các hoạt động DML trên bảng với điều kiện chúng không ghi vào phân hoạch bảng cơ sở.
Bạn có thể thiết lập lại các phân hoạch khác trong chỉ mục ấy. Bạn cũng có thể tạo các chỉ mục khác trên bảng hoặc thiết lập lại các phân hoạch ở các chỉ mục khác.
Bạn có thể tiến hành hoạt động bảo trì bất kỳ trên bảng hoặc các chỉ mục của nó thay vì cửa sổ hoạt động được liệt kê trên.

Một số hoạt động bảo trì trên phân hoạch của một bảng tạo cho các chỉ mục toàn cục của bảng hoặc các phân hoạch chỉ mục trở thành Unusable. Thí dụ như ALTER TABLE MOVE PARTITION. DBA phải tiến hành một kỳ mã gồm chỉ mục toàn cục thiết lập lại bổ sung vào hoạt động bảo trì phân hoạch. Do đó theo quan điểm những hoạt động này truy xuất liên tiếp hoàn toàn bảng. Các hoạt động, chẳng hạn như ALTER TABLE MOVE/SPLIT PARTITION tạo cho Unusable bất kỳ chỉ mục toàn cục không có phân hoạch cũng như tất cả các phân hoạch của các chỉ mục toàn cục có phân hoạch.

Lưu ý các hoạt động phân hoạch bảng đánh dấu tất cả các phân hoạch của những chỉ mục toàn cục cũng đánh dấu một phân hoạch của chỉ mục cục bộ (phân hoạch tương ứng với phân hoạch bảng được điều tác) Unusable.

Tương tự một số hoạt động bảo trì phân hoạch đòi hỏi vô hiệu hóa Referential Integrity Constraints trước hoạt động ấy và sau đó cho phép chúng có tác dụng lại. Thí dụ như ALTER TABLE DROP PARTITION của phân hoạch không trống. DBA phải tiến hành một kỳ mã gồm hạn định cho phép lại để bổ sung vào hoạt động bảo trì phân hoạch. Do đó theo quan điểm của người dùng những hoạt động này truy xuất liên tiếp toàn bộ bảng.

Các truy vấn và các hoạt động bảo trì phân hoạch

Các truy vấn mà thực hiện của nó bắt đầu trước khi gọi một hoạt động bảo trì phân hoạch hoặc trước khi các cập nhật từ điển được thực hiện khi hoạt động bảo trì phân hoạch, truy xuất chính xác bằng Consistent Read dữ liệu của các phân hoạch có tác động như tồn tại vào thời gian ảnh chụp nhanh. Xử lý của các truy vấn như thế sau các cập nhật từ điển được thực hiện không thể đoán trước, nghĩa là một số dữ liệu tồn tại vào thời gian ảnh chụp nhanh có thể được truy hồi hoặc có thể cho trở lại các lỗi.

Các truy vấn sử dụng chỉ mục được phân hoạch và bắt đầu bằng một số trong các phân hoạch chỉ mục được đánh dấu là *Index Unusable*, cho trở lại một lỗi khi chúng thực sự truy xuất một trong các phân hoạch này lần đầu. Điều này xảy ra dù phân hoạch ấy đã thực hiện để có thể sử dụng sau khi truy vấn bắt đầu.

Sự mất hiệu lực con trỏ

Dù nhiều các câu lệnh DDL mới được dựa vào phân hoạch, nhưng sự mất hiệu lực con trỏ vẫn dựa vào bảng. Điều này có nghĩa là câu lệnh DDL sửa đổi bảng T cũng làm mất hiệu lực tất cả các con trỏ phụ thuộc vào T, dù câu lệnh ấy chỉ có tác động trên phân hoạch P của T và các con trỏ không truy xuất phân hoạch P.

Các hoạt động phục hồi được và không phục hồi được

Tất cả hoạt động bảo trì phân hoạch có thể được tiến hành ở chế độ (LOGGING) phục hồi được. Tuy nhiên, một số hoạt động hỗ trợ tùy chọn NOLOGGING:

Parallel CREATE TABLE ... AS SELECT
CREATE INDEX
Direct Path SQL*Loader

Direct-load INSERT

LOGGING mặc định, trừ khi CSDL đang hoạt động ở chế độ NOARCHIVELOG. Trong trường hợp đó, NOLOGGING mặc định. DDL và các câu lệnh không hỗ trợ tùy chọn LOGGING/NOLOGGING luôn luôn tiến hành ở chế độ phục hồi được (LOGGING).

Lưu ý :
[NO]LOGGING không phải là thuộc tính của một hoạt động nhưng của một đối tượng vật lý. Vì thế, bạn không thể ấn định [NO]LOGGING trong INSERT, nhưng đúng hơn nếu bạn muốn thay đổi chế độ ghi sổ của một bảng hoặc (các) chỉ mục liên quan INSERT, bạn cần đưa ra câu lệnh ALTER TABLE/INDEX [NO]LOGGING trước khi đưa ra câu lệnh INSERT. Xem thêm "Chế Độ Ghi Sổ".

Quản lý chỉ mục

Bạn luôn luôn có thể đặt tên lại, thay đổi những thuộc tính bộ nhớ vật lý hoặc thiết lập lại phân hoạch của một chỉ mục toàn cục hoặc cục bộ. Thay đổi một chỉ mục được phân hoạch như thế nào phải được xử lý khác nhau tùy theo chỉ mục là cục bộ hoặc toàn cục.

Các chỉ mục cục bộ

Oracle bảo đảm là phân hoạch của một chỉ mục cục bộ so khớp với phân hoạch của bảng cơ sở. Nó thực hiện điều này bằng cách tạo hoặc bỏ tự động các phân hoạch chỉ mục theo yêu cầu khi bạn thay đổi bảng cơ sở. Bạn không thể bổ sung, bỏ hoặc tách riêng một phân hoạch ở chỉ mục cục bộ.
Đối với mỗi chỉ mục cục bộ:

Khi bạn bổ sung phân hoạch vào bảng cơ sở, Oracle tự động tạo một phân hoạch chỉ mục mới với cùng biên phân hoạch như phân hoạch bảng mới.
Khi bạn bỏ một phân hoạch ở bảng cơ sở, Oracle tự động bỏ phân hoạch chỉ mục tương ứng.
Khi bạn tách một phân hoạch ở bảng cơ sở, Oracle tự động tách phân hoạch chỉ mục tương ứng. Hai phân hoạch chỉ mục mới có cùng các biến phân hoạch như các phân hoạch bảng mới.

> Lưu ý là các phân hoạch chỉ mục cục bộ được tạo do tách một phân hoạch bảng cha được đánh dấu Unusable nếu một phân hoạch bảng tương ứng không rỗng.

Khi Oracle tạo một phân hoạch cục bộ mới (bằng ADD hoặc SPLIT):
Nó cố gán phân hoạch cùng tên như phân hoạch bảng tương ứng. Nếu không thành công, nó phát sinh một tên với dạng SYS_Pnnn (xem "Các Tên Phân Hoạch"). Bạn có thể đặt tên lại phân hoạch ấy sau.
Đối với ADD PARTITION, Oracle tạo một đoạn với các thuộc tính bộ nhớ vật lý của chỉ mục cơ sở. Nếu không gian bảng (thay vì DEFAULT, tạo các phân hoạch chỉ mục cục bộ để được cùng định vị với các phân hoạch bảng cơ sở tương ứng) đã được ấn định cho chỉ mục cha, phân hoạch chỉ mục được đặt vào không gian bảng. Trái lại, không gian bảng trong đó phân hoạch chỉ mục mới cư trú thuộc phân hoạch tương ứng của bảng cơ sở. Bạn có thể sửa đổi những thuộc tính này sau.
Đối với SPLIT PARTITION, các thuộc tính của phân hoạch chỉ mục tách được sử dụng cho các phân hoạch chỉ mục do tách chỉ mục ấy. Các tên phân hoạch là ngoại lệ, dù Oracle sử dụng lại các tên phân hoạch bảng khi có thể. Thí dụ, đối với một phân hoạch bảng với tên TP và chỉ mục cục bộ với tên IP, nếu TP được tách thành TP

và TP1, sau đó các tên của các phân hoạch chỉ mục cục bộ là IP và TP1 (hoặc tên hệ thống phát sinh nếu TP1 đã được sử dụng cho chỉ mục đó). Nếu TP được tách thành TP1 và TP2, sau đó các phân hoạch chỉ mục cục bộ là TP1 và TP2. Nghĩa là nếu tên phân hoạch bảng được sử dụng lại, thì Oracle cũng cố sử dụng lại tên phân hoạch chỉ mục cục bộ. Tuy nhiên, tất cả các thuộc tính khác được thừa kế từ phân hoạch chỉ mục được tách.

Các chỉ mục toàn cục

DBA chịu trách nhiệm duy trì phân hoạch của chỉ mục toàn cục. Bạn có thể bỏ hoặc tách một phân hoạch bằng chỉ mục toàn cục. Tuy nhiên, bạn không thể bổ sung phân hoạch vào chỉ mục toàn cục vì phân hoạch cao của chỉ mục toàn cục luôn luôn có biên phân hoạch MAXVALUE.

Thiết lập lại phân hoạch chỉ mục

Câu lệnh ALTER INDEX REBUILD PARTITION có thể được sử dụng để tái phát sinh một phân hoạch đơn bằng chỉ mục cục bộ hoặc toàn cục. Điều này tiết kiệm cho bạn không phải thực hiện DROP INDEX rồi sau đó CREATE INDEX, sẽ ảnh hưởng đến tất cả những phân hoạch bằng chỉ mục ấy.

ALTER INDEX REBUILD PARTITION có bốn ứng dụng quan trọng:
Để tạo cụm liên kết một phân hoạch chỉ mục để phục hồi không gian và tăng hiệu suất.
Để sửa chữa một phân hoạch chỉ mục trong trường hợp một sai sót môi trường trên bộ đĩa nơi phân hoạch chỉ mục cư trú hoặc sai sót phần mềm của đoạn chứa phân hoạch chỉ mục ấy.

Tái phát sinh một phân hoạch chỉ mục cục bộ sau khi nạp phân hoạch bằng sơ sở bằng Import hoặc SQL*Loader. Các tiện ích này cung ứng tùy chọn hiệu suất để bỏ qua bảo trì chỉ mục, đánh dấu các phân hoạch chỉ mục Index Unusable bị ảnh hưởng và để DBA thiết lập lại chúng sau. (Index Unusable được giải thích ở phần kế tiếp.) Nói khác đi, chiến lược "bỏ chỉ mục rồi tạo lại chỉ mục" có thể được thay thế bằng chiến lược "đánh dấu phân hoạch chỉ mục không dùng được rồi thiết lập lại phân hoạch chỉ mục". Để thiết lập lại các phân hoạch chỉ mục biểu hiện không dùng được bằng các hoạt động bảo trì trên bảng cơ sở.

Thuộc tính không dùng được chỉ mục

Một số hoạt động bảo trì đánh dấu các chỉ mục Index Unusable (IU). Index Unusable là một thuộc tính của chỉ mục không được phân hoạch của một phân hoạch trong chỉ mục được phân hoạch. Khi một chỉ mục hoặc phân hoạch chỉ mục được đánh dấu IU, bạn nhận một lỗi nếu bạn cố thực hiện câu lệnh SELECT hoặc DML đòi hỏi chỉ mục (hoặc phân hoạch).

Khi một phân hoạch chỉ mục đơn được đánh dấu IU, bạn phải thiết lập lại phân hoạch ấy để làm cho nó có hiệu lực trở lại trước khi sử dụng nó. Tuy nhiên, khi một phân hoạch được đánh dấu IU thì các phân hoạch khác của chỉ mục ấy hợp lệ và bạn có thể thực hiện các câu lệnh SELECT hoặc DML đòi hỏi chỉ mục bao lâu các câu lệnh ấy không truy xuất phân hoạch IU.

Bạn cũng có thể tách hoặc đặt tên lại phân hoạch IU trước khi thiết lập lại nó và bạn có thể bỏ một phân hoạch IU của chỉ mục GLOBAL.

Khi một chỉ mục không có phân hoạch được đánh dấu IU, bạn có thể bỏ chỉ mục mà bạn còn có thể bỏ phân hoạch IU của chỉ mục GLOBAL index.x và tạo lại nó. Bạn cũng có thể sử dụng ALTER INDEX REBUILD để thiết lập lại một chỉ mục không được phân hoạch.

Sau kiểu hoạt động bảo trì có thể đánh dấu các phân hoạch chỉ mục Index Unusable. Trong tất cả các trường hợp, bạn phải thiết lập lại các phân hoạch chỉ mục khi hoạt động hoàn tất.

Các hoạt động như Import Partition hoặc đường dẫn quy ước SQL*Loader cung ứng tùy chọn để bỏ qua bảo trì chỉ mục cục bộ. Khi Import hoàn tất, thì các phân hoạch chỉ mục cục bộ được tác động được đánh dấu IU.

Direct path SQL*Loader để các phân hoạch chỉ mục cục bộ được tác động và các chỉ mục toàn cục ở trạng thái IU nếu chỉ mục ấy hết thời hạn liên quan đến dữ liệu mà nó chỉ mục. (Index Unusable trước đây được coi là Direct Load State.) Chỉ mục có thể hết thời hạn vì các nguyên nhân sau:

Chỉ mục không thể được duy trì bằng cách nạp do lỗi quản lý không gian (thí dụ, ngoài các khối lưu riêng).

Người dùng đã yêu cầu tùy chọn SKIP_INDEX_MAINTENANCE.

Các giao dịch bảo trì phân hoạch như ALTER TABLE MOVE PARTITION thay đổi các ROWID. Các hoạt động này đánh dấu phân hoạch chỉ mục cục bộ được tác động và tất cả các phân hoạch chỉ mục toàn cục IU.

Các hoạt động bảo trì phân hoạch như ALTER TABLE TRUNCATE PARTITION hoặc DROP PARTITION xóa các hàng khỏi bảng. Các hoạt động này đánh dấu phân hoạch chỉ mục cục bộ được tác động và tất cả các phân hoạch chỉ mục toàn cục IU.

Các hoạt động bảo trì phân hoạch như ALTER TABLE SPLIT PARTITION sửa đổi định nghĩa của các chỉ mục cục bộ nhưng không tự động thiết lập lại dữ liệu chỉ mục

để so khớp các định nghĩa mới. Những hoạt động này đánh dấu (các) phân hoạch chỉ mục IU cục bộ được tác động. (ALTER TABLE SPLIT PARTITION cũng đánh dấu tất cả các phân hoạch IU toàn cục vì nó dẫn đến các thay đổi thành các ROWID.)

Các hoạt động bảo trì chỉ mục như ALTER INDEX SPLIT PARTITION sửa đổi định nghĩa phân hoạch của chỉ mục nhưng không tự động thiết lập lại các phân hoạch được tác động. Những hoạt động này đánh dấu (các) phân hoạch chỉ mục IU được tác động. Tuy nhiên, nếu bạn tách một phân hoạch Usable của chỉ mục toàn cục, thì các phân hoạch kết quả được tạo có thể sử dụng. Nếu phân hoạch được tách đã được đánh dấu IU, như vậy là các phân hoạch do tách. (Lưu ý bỏ phân hoạch của một chỉ mục toàn cục là IU hoặc không rỗng tạo cho phân hoạch kế tiếp của chỉ mục trở thành IU.)

Các đặc quyền đối với các bảng được phân hoạch và các chỉ mục

Các đặc quyền đối với các phân hoạch được cho trên bảng cha hoặc chỉ mục, không phải trên các phân hoạch riêng.

Nếu người dùng hoặc một vai trò có các đặc quyền cần thiết để thực hiện một hoạt động Oracle trên các bảng không được phân hoạch và các chỉ mục (kể cả các đặc quyền tài nguyên cần thiết), sau đó các hoạt động Oracle được cho phép trên các bảng được phân hoạch và các chỉ mục. Thí dụ:

Nếu bạn có thể tạo các bảng chưa được phân hoạch, thì bạn có thể tạo các bảng được phân hoạch.

Nếu bạn bỏ các chỉ mục không được phân hoạch, thì bạn có thể bỏ các chỉ mục được phân hoạch.

Nếu bạn có thể bổ sung một cột bằng ALTER vào các bảng không được phân hoạch, thì bạn có thể bổ sung một cột bằng ALTER vào các bảng được phân hoạch.

Nếu người dùng hoặc một vai trò có các đặc quyền cần thiết để thực hiện một điều hoạt ALTER trên bảng hoặc chỉ mục, thì các điều hoạt ALTER mới trên các phân hoạch của bảng hoặc chỉ mục có thể được viện dẫn với các ngoại lệ sau:
Đặc quyền DROP ANY TABLE (bổ sung vào đặc quyền ALTER) được một người dùng không phải là chủ nhân để:
ALTER TABLE DROP PARTITION
ALTER TABLE TRUNCATE PARTITION
Các hoạt động sau đòi hỏi chỉ tiêu không gian trong không gian bảng trong đó không gian phải đạt được để bổ sung vào đặc quyền ALTER:

ALTER INDEX MODIFY PARTITION
ALTER INDEX REBUILD PARTITION
ALTER INDEX SPLIT PARTITION
ALTER TABLE ADD PARTITION
ALTER TABLE MODIFY PARTITION
ALTER TABLE MOVE PARTITION
ALTER TABLE SPLIT PARTITION

Kiểm soát các bảng được phân hoạch và chỉ mục

Tất cả các điều hoạt ALTER TABLE PARTITION được kiểm soát giống như các điều ALTER TABLE. Không có các thuộc tính kiểm soát mới được sử dụng các hoạt động.

Phần mở rộng SQL: tên bảng phân hoạch mở rộng

Các điều tác hàng loạt mức phân hoạch bị hạn chế vào chính các hàng của một phân hoạch riêng biệt; thí dụ, người dùng muốn bỏ

một phân hoạch mà không tạo tất cả các chỉ mục toàn cục UNUSABLE mà muốn xóa tất cả các hàng khỏi chính phân hoạch đó.
Những hoạt động được biểu thị rất tự nhiên bằng cách sử dụng cú pháp tên bảng phân hoạch mở rộng. Cố biểu thị cùng hoạt động với thuộc tính mệnh đề WHERE trở thành khá phức tạp nhất là khi khóa phân hoạch phạm vi sử dụng nhiều cột.

Cú pháp định chuẩn bảng cho các câu lệnh DML sau có thể chứa một định chuẩn phân hoạch tùy chọn đối với các bảng không được phân hoạch từ xa:

INSERT
UPDATE
DELETE
LOCK TABLE
SELECT

Thí dụ:
SELECT * FROM schema.table PARTITION part_name;

Cú pháp này cung cấp một cách đơn giản để xem các phân hoạch riêng biệt bằng các bảng: một khung nhìn có thể được tạo chỉ lựa từ một phân hoạch bằng cách sử dụng tên bảng mở rộng phân hoạch và khung nhìn này có thể được sử dụng thay cho một bảng.

Với các khung nhìn như thế bạn còn có thể thiết lập cơ chế truy xuất mức phân hoạch bằng cách cho (viện dẫn) các đặc quyền trên các khung nhìn này tới (từ) những người dùng khác hoặc các vai trò. Để có thể chuyển đổi ứng dụng và tuân thủ cú pháp ANSI bạn có thể sử dụng các khung nhìn để cô lập các ứng dụng của mình với phần mở rộng tư hữu của Oracle.

Sử dụng các tên bảng mở rộng phân hoạch có những hạn chế sau:

Một tên bảng mở rộng phân hoạch không thể tham chiếu một đối tượng giản đồ từ xa.

Một tên bảng mở rộng không thể chứa một dblink hoặc từ đồng nghĩa phiên dịch thành một bảng với một dblink.

Nếu bạn cần sử dụng các phần mềm từ xa, bạn có thể tạo một khung nhìn ở vùng từ xa sử dụng cú pháp tên bảng mở rộng phân hoạch và tham số khung nhìn từ xa.

Cú pháp tên bảng mở rộng phân hoạch không được PL/SQL hỗ trợ.

Câu lệnh SQL sử dụng cú pháp tên bảng mở rộng phân hoạch không thể được sử dụng trong khối PL/SQL, dù nó có thể được sử dụng bằng SQL năng động tới bộ phần mềm DBMS_SQL. Hơn nữa, nếu bạn cần tham chiếu một phân hoạch trong khối PL/SQL bạn có thể sử dụng thay thế bằng các khung nhìn để sau đó sử dụng cú pháp tên bảng mở rộng phân hoạch.

Chỉ các bảng cơ sở được cho phép.

Phần mở rộng phân hoạch phải được xác định với một bảng cơ sở. Không có từ đồng nghĩa, khung nhìn hoặc bất kỳ đối tượng giản đồ nào được cho phép.

Các thí dụ về các tên bảng mở rộng phân hoạch

Các câu lệnh sau chứa các tên bảng mở rộng phân hoạch hợp lệ:

```
SELECT * FROM sales PARTITION (nov95) s
   WHERE s.amount_of_sale > 1000;

UPDATE sales PARTITION (feb96) s
   SET s.account_name = UPPER(s.account_name);

DELETE FROM sales PARTITION (nov95)
   WHERE amount_of_sale != 0;

INSERT INTO sales PARTITION (oct95)
   SELECT * FROM lastest_data;
```

7
INSERT INTO sales PARTITION (oct95) VALUES (...);

INSERT INTO sales PARTITION (oct95) (acct_no, ..., week_no) VALUES (...);

LOCK TABLE sales PARTITION (jun95) IN EXCLUSIVE MODE;

CREATE VIEW sales_feb96 AS
  SELECT * FROM sales PARTITION (feb96);

Chương 10

Các kiểu dữ liệu lập sẵn

Chương này đề cập đến những kiểu dữ liệu lập sẵn của Oracle và chúng ánh xạ vào các kiểu dữ liệu khác Oracle như thế nào. Gồm các chủ đề:

- **Các kiểu dữ liệu Oracle**
- **Các kiểu dữ liệu ký tự**
- **Kiểu dữ liệu NUMBER**
- **Kiểu dữ liệu DATE**
- **Các kiểu dữ liệu RAW và LONG RAW**
- **Kiểu dữ liệu ROWID**
- **Kiểu dữ liệu MISLABEL**

  Các Kiểu Dữ Liệu ANSI, DB2 và SQL/DS

  Chuyển Đổi Dữ Liệu

Các kiểu dữ liệu Oracle

Bạn có thể sử dụng các kiểu dữ liệu lập sẵn trong các định nghĩa sau:

Các Kiểu Dữ Liệu Ký Tự

Kiểu Dữ Liệu CHAR
Kiểu Dữ Liệu VARCHAR2
Kiểu Dữ Liệu VARCHAR
Các Kiểu Dữ Liệu NCHAR và NVARCHAR2
Kiểu Dữ Liệu LONG
Kiểu Dữ Liệu NUBMER
Kiểu Dữ Liệu DATE
Kiểu Dữ Liệu LOB
Kiểu Dữ Liệu BLOB
Các Kiểu Dữ Liệu CLOB và NCLOB
Kiểu Dữ Liệu BFILE
Các Kiểu Dữ Liệu RAW và LONG RAW
Kiểu Dữ Liệu ROWID
Kiểu Dữ Liệu MISLABEL

Thông tin bổ sung :
PL/SQL có các kiểu dữ liệu bổ sung, chẳng hạn như BOOLEAN, các kiểu tham chiếu, các kiểu tổng hợp (các tập hợp và các bản ghi) và các kiểu phụ được người dùng xác định. Xem Hướng dẫn của Người dùng PL/SQL và Tham chiếu về các kiểu dữ liệu PL/SQL.

Các kiểu dữ liệu ký tự

Các kiểu dữ liệu ký tự lưu trữ dữ liệu ký tự (chữ-số) trong các chuỗi với các giá trị byte tương ứng với giản đồ mã hóa ký tự (thường được gọi là tập hợp ký tự hoặc trang mã).

Tập hợp ký tự của cơ sở dữ liệu được thiết lập khi bạn tạo CSDL và không bao giờ thay đổi. Các thí dụ về các bộ ký tự 7-bit ASCII (American Standard Code for Information Interchange - Mã Tiêu Chuẩn Mỹ Để Hoán Đổi Thông Tin), EBCDIC (Extended Binary Coded Decimal Interchange Code - Mã Hoán Đổi Thập Phân Mã Hóa Nhị Phân Mở Rộng), Code Page 500, và

Japan Extended UNIX. Oracle hỗ trợ cả giản đồ mã hóa một byte và nhiều byte.

Thông tin bổ sung:
Xem Hướng dẫn của Oracle8 Application Developer về cách lựa kiểu dữ liệu ký tự.

Kiểu dữ liệu CHAR

Kiểu dữ liệu CHAR lưu trữ các chuỗi ký tự có chiều dài cố định. Khi bạn tạo một bảng với cột CHAR, bạn phải ấn định chiều dài chuỗi (bằng byte, không phải bằng các ký tự) từ 1 tới 2000 đối với chiều rộng CHAR. (Mặc định bằng 1.) Sau đó Oracle bảo đảm là:

Khi bạn chèn hoặc cập nhật một hàng vào bảng, giá trị đối với cột CHAR có chiều dài cố định.

Nếu bạn cho một giá trị ngắn, thì giá trị ấy là đệm khoảng trắng chêm (blank-padded) đối với chiều dài cố định.

Nếu bạn cho giá trị dài hơn với các khoảng trắng đuôi, các khoảng trắng được xóa bỏ khỏi giá trị đối với chiều dài cố định.

Nếu giá trị quá lớn, Oracle cho trở lại một lỗi.

Oracle so sánh các giá trị CHAR sử dụng ngữ nghĩa so sánh khoảng trắng chêm.

Thông tin bổ sung:
Xem Tham chiếu SQL Oracle8 về ngữ nghĩa học so sánh.

Kiểu dữ liệu VARCHAR2

Kiểu dữ liệu VARCHAR2 lưu trữ các chuỗi ký tự có chiều dài khá biến. Khi bạn tạo một bảng với cột VARCHAR2, bạn ấn định chiều dài chuỗi tối đa (bằng byte, không phải bằng các ký tự) từ 1 tới 4000 đối với cột VARCHAR2. Đối với mỗi hàng, Oracle lưu

trữ mỗi giá trị ở cột bằng một trường có chiều dài khả biến (trừ khi giá trị vượt chiều dài tối đa của cột ấy, trong trường hợp đó Oracle cho trở lại một lỗi).

Thí dụ, giả sử bạn khai báo cột VARCHAR2 với kích thước tối đa 50 ký tự. Trong tập hợp ký tự một byte, nếu chỉ cho 10 ký tự cho giá trị cột VARCHAR2 ở một hàng riêng biệt, thì cột trong phân đoạn hàng của hàng chỉ lưu trữ 10 ký tự (10 byte), không phải 50.

Oracle so sánh các giá trị VARCHAR2 sử dụng **nonpadded comparison semantics** (ngữ nghĩa so sánh không có chêm).

Thông tin bổ sung:
Xem Oracle8 SQL Reference.

Kiểu dữ liệu VARCHAR

Kiểu dữ liệu VARCHAR hiện hành đồng nghĩa với kiểu dữ liệu VARCHAR2. Tuy nhiên, trong phiên bản tương lai của Oracle, VARCHAR có thể lưu trữ các chuỗi ký tự có chiều dài khả biến được so sánh với các ngữ nghĩa học so sánh khác nhau. Do đó, bạn phải sử dụng kiểu dữ liệu VARCHAR2 để lưu trữ các chuỗi ký tự có chiều dài khả biến.

Các chiều dài cột đối với các kiểu dữ liệu ký tự và các tập hợp ký tự NLS

Đặc trưng Oracle National Language Support (NLS) cho phép sử dụng các bộ ký tự khác nhau đối với các kiểu dữ liệu ký tự. National Language Support cho phép bạn xử lý dữ liệu ký tự một byte và nhiều byte để chuyển đổi giữa các bộ ký tự. Các phiên client có thể sử dụng các bộ ký tự quốc gia của bộ ký tự CSDL.

Bạn phải xét kích thước của các ký tự khi ấn định chiều dài cột cho các kiểu dữ liệu ký tự. Bạn phải xét vấn đề này khi dự đoán không gian cho các bảng có các cột chứa dữ liệu ký tự.

Thông tin bổ sung:
Xem Oracle8 Reference và Oracle8 Utilities về các đặc trưng NLS của Oracle.

Các kiểu dữ liệu NCHAR và NVARCHAR2

Các kiểu dữ liệu NCHAR và NVARCHAR2 lưu trữ dữ liệu ký tự NLS. Kiểu dữ liệu NCHAR lưu trữ các chuỗi ký tự có chiều dài cố định tương ứng với bộ ký tự quốc gia có chiều dài khả biến hoặc cố định. Kiểu dữ liệu NVARCHAR2 lưu trữ các chuỗi ký tự có chiều dài khả biến.

Khi bạn tạo một bảng với cột NCHAR hoặc NVARCHAR2, bạn ấn định kích thước tối đa là số ký tự (đối với bộ ký tự quốc gia có chiều dài cố định) hoặc số byte (đối với bộ ký tự quốc gia có chiều dài khả biến).

Chiều dài tối đa đối với cột NCHAR là 2000 byte hoặc số ký tự có thể được lưu trữ bằng 2000 byte.

Chiều dài tối đa đối với cột NVARCHAR2 là 4000 byte hoặc số ký tự có thể được lưu trữ bằng 4000 byte.

Thông tin bổ sung:
Xem Oracle8 Reference và Hướng dẫn của người dùng PL/SQL về các kiểu dữ liệu NCHAR và NVARCHAR2.

Các kiểu dữ liệu ký tự LOB Character

Các kiểu dữ liệu LOB đối với dữ liệu ký tự là CLOB và NCLOB. Chúng có thể lưu trữ tối đa 4 gigabyte dữ liệu ký tự (CLOB) hoặc dữ liệu bộ ký tự quốc gia (NCLOB). Những kiểu dữ liệu này được mô tả ở "Các Kiểu Dữ Liệu LOB".

Kiểu dữ liệu LONG

Các cột được xác định là LONG có thể lưu trữ dữ liệu ký tự có chiều dài khả biến chứa tối đa 2 gigabyte thông tin. Dữ liệu LONG là dữ liệu văn bản để được chuyển đổi thích hợp khi di chuyển giữa các hệ thống khác nhau.

Các cột kiểu dữ liệu LONG được sử dụng ở từ điển dữ liệu để lưu trữ các định nghĩa khung nhìn. Bạn có thể sử dụng các cột LONG trong các danh sách SELECT, các mệnh đề SET của các câu lệnh UPDATE và các mệnh đề VALUES của câu lệnh INSERT.

Lưu ý:
Kiểu dữ liệu LONG được cung cấp cho tính tương thích về trước với các ứng dụng có sẵn. Trong các ứng dụng mới, bạn phải sử dụng các kiểu dữ liệu CLOB và NCLOB cho số lượng dữ liệu ký tự lớn.

Thông tin bổ sung:
Kiểu dữ liệu LONG có nhiều hạn chế - xem Hướng dẫn của Oracle8 Application Developer.

Xem thêm "Các kiểu dữ liệu RAW và LONG RAW" về kiểu dữ liệu LONG RAW.

Kiểu dữ liệu NUMBER

Kiểu dữ liệu NUMBER lưu trữ các số dấu chấm động và cố định. Các số của bất kỳ độ lớn ảo có thể được lưu trữ và được bảo đảm khả chuyển giữa các hệ thống điều hoạt Oracle khác nhau, tối đa 38 chữ số chính xác.

Các số sau có thể được lưu trữ ở cột NUMBER:

Các số dương ở phạm vi 1 x 10^-130 tới 9.99..9 x 10^125 (có tối đa 38 chữ số quan trọng)
Các số âm từ -1 x 10^-130 tới 9.99..99 x 10^125 (có tối đa 38 chữ số quan trọng)
zêrô
số vô hạn dương và âm (chỉ được phát sinh bằng cách Import từ CSDL Oracle Version 5 database)

Đối với các cột số bạn có thể ấn định cột ấy là:
column_name NUMBER

Bạn còn có thể tùy chọn ấn định chính xác (tổng các chữ số) và tỷ lệ (số chữ số ở bên phải của dấu thập phân):
column_name NUMBER (precision, scale)

Nếu độ chính xác không được xác định, cột lưu trữ những giá trị được định trước. Nếu không có tỷ lệ được sử dụng, thì tỷ lệ ấy là zêrô.

Oracle bảo đảm tính khả chuyển các số của độ chính xác bằng hoặc nhỏ hơn 38 chữ số. Bạn có thể ấn định tỷ lệ và không có độ chính xác:
column_name NUMBER (*, scale)

Trong trường hợp này, độ chính xác là 38 và tỷ lệ xác định được duy trì.

Khi bạn ấn định các trường bằng số, tốt nhất nên ấn định độ chính xác và tỷ lệ; điều này cung cấp thêm tính nguyên vẹn kiểm dữ liệu nhập.

Bảng 10-1 trình bày các thí dụ về dữ liệu sẽ được lưu trữ sử dụng các yếu tố tỷ lệ khác nhau như thế nào.

Bảng 10-1 Các yếu tố tỷ lệ tác động đến bộ nhớ dữ liệu bằng số như thế nào

Dữ liệu nhập	Được ấn định là	Được lưu trữ là
7,456,123.89	NUMBER	7456123.89
7,456,123.89	NUMBER(*,1)	7456123.9
7,456,123.89	NUMBER(9)	7456124
7,456,123.89	NUMBER(9,2)	7456123.89
7,456,123.89	NUMBER(9,1)	7456123.9
7,456,123.89	NUMBER(6)	(không được chấp nhận, vượt độ chính xác)
7,456,123.89	NUMBER(7,-2)	7456100

Nếu bạn ấn định một tỷ lệ âm, Oracle làm tròn dữ liệu thực thành số được ấn định của các vị trí bên trái của dấu chấm thập phân. Thí dụ, ấn định (7,-2) nghĩa là Oracle sẽ làm tròn các hàng trăm gần nhất, như được trình bày ở Bảng 10-1.

Đối với các số nhập và xuất, ký tự thập phân mặc định Oracle chuẩn là dấu chấm như ở số "1234.56". (Số thập phân là ký tự tách các phần số nguyên và thập phân của một số). Bạn có thể thay đổi ký tự thập phân mặc định với tham số khởi hoạt NLS_NUMERIC_CHARACTERS. Bạn cũng có thể thay đổi nó đối với thời gian của một phiên với câu lệnh ALTER SESSION. Để nhập các số không sử dụng ký tự thập phân mặc định, sử dụng hàm TO_NUMBER.

Định dạng số bên trong

Oracle lưu trữ dữ liệu số ở định dạng chiều dài khả biến. Mỗi giá trị được lưu trữ bằng hệ ký hiệu khoa học, với một byte được sử dụng để lưu trữ số mũ và tối đa 20 byte lưu trữ phần định trị ấy. (Giá trị kết quả bị giới hạn vào 38 chữ số của độ chính xác). Oracle không lưu trữ các số zêrô đầu và cuối. Thí dụ, số 412 được lưu trữ ở định dạng giống như 4.12 x 10^2, với một byte được sử dụng để lưu trữ số mũ (2) và hai byte được sử dụng để lưu trữ ba chữ số quan trọng của phần định trị (4, 1, 2).

Lưu ý điều này, kích thước dữ liệu cột đối với giá trị dữ liệu số NUMBER (p), trong đó p là độ chính xác của giá trị cho trước (không ảnh hưởng đến tỷ lệ) có thể được tính toán bằng cách sử dụng công thức sau:

1 byte	(số mũ)
+ FLOOR(p/2)+1 bytes	(phần định trị)
+ 1 byte	(chỉ đối với số âm nơi số có các chữ số quan trọng nhỏ hơn 38)
số byte dữ liệu	

Zêrô và số vô hạn dương và âm (chỉ được phát sinh khi Import từ các CSDL Version 5 Oracle databases) được lưu trữ bằng cách sử dụng các biểu thị duy nhất: zêrô và mỗi số vô hạn âm đòi hỏi một byte; số vô hạn dương đòi hỏi hai byte.

Kiểu dữ liệu DATE

Kiểu dữ liệu DATE lưu trữ những giá trị giờ và ngày tháng (point-in-time) ở một bảng. Kiểu dữ liệu DATE lưu trữ năm (kể cả thế kỷ), tháng, ngày, giờ, phút và giây (sau không giờ).

Oracle có thể lưu trữ ngày tháng theo kỷ nguyên Julian có phạm vi từ January 1, 4712 BCE tới December 31, 4712 CE (Common Era). Trừ BCE ('BC' bằng mặt che định dạng) được đặc biệt sử dụng, các mục nhập CD là mặc định.

Oracle sử dụng định dạng bên trong của riêng nó để lưu trữ ngày tháng. Dữ liệu ngày tháng được lưu trữ bằng các trường có chiều dài cố định với mỗi trường bảy byte, tương ứng thế kỷ, năm, tháng, ngày, giờ, phút và giây.
Đối với các ngày tháng nhập liệu và xuất dữ liệu, định dạng mặc định Oracle chuẩn là DD-MON-YY, như sau:
'13-NOV-92'

Bạn có thể thay đổi định dạng ngày tháng này đối với một môi trường làm việc có tham số NLS_DATE_FORMAT. Bạn cũng có thể thay đổi nó khi một phiên người dùng bằng câu lệnh ALTER SESSION. Để nhập ngày tháng không có trong định dạng ngày tháng Oracle chuẩn, sử dụng hàm TO_DATE với mặt che định dạng:
TO_DATE ('November 13, 1992', 'MONTH DD, YYYY')

Lưu ý:
Nếu bạn sử dụng định dạng ngày tháng chuẩn DD-MON-YY, YY cho năm ấy vào thế kỷ 20 (thí dụ, 31-DEC-92 là 31 tháng 12 năm 1992). Nếu bạn muốn biểu thị các năm ở bất kỳ thế kỷ nào khác hơn thế kỷ 20, sử dụng mặt che định dạng khác nhau, như được trình bày ở trên.

Oracle lưu trữ thời gian bằng định dạng 24 giờ - HH:MI:SS. Theo mặc định, giờ trong trường ngày tháng là 00:00:00 A.M. (không giờ) nếu không có phần giờ được nhập. Trong một mục nhập chỉ có giờ, thì phần ngày tháng mặc định thành ngày thứ nhất của tháng hiện hành. Để nhập phần giờ của một ngày tháng, sử

dụng hàm TO_DATE với mặt che định dạng để biểu thị phần giờ, như ở

```
INSERT INTO birthdays (bname, bday) VALUES
   ('ANDY',TO_DATE('13-AUG-66     12:56     A.M.','DD-MON-YY
HH:MI A.M.'));
```

Sử dụng ngày tháng Julian

Ngày tháng Julian cho phép ghi ngày tháng liên tiếp bằng số ngày từ một tham chiếu thông thường. (tham chiếu là 01-01-4712 years BCE, vì vậy ngày tháng hiện hành ở trong phạm vi 2,4 triệu). Ngày tháng Julian trên danh nghĩa là một số khác số nguyên, phần phân số là phần của ngày. Oracle sử dụng phép tính gần đúng đơn giản hóa cho các giá trị số nguyên. Ngày tháng Julian có thể được tính toán và được diễn dịch khác nhau; phương pháp tính được Oracle sử dụng bằng bảy chữ số (thường được sử dụng nhất đối với ngày tháng), chẳng hạn như 2449086 cho 08-APR-93.

Lưu ý:
Ngày tháng Julian của Oracle không thể tương thích với ngày tháng Julian được các thuật toán ngày tháng phát sinh.

Mặt che định dạng "J" có thể được sử dụng với các hàm ngày tháng (TO_DATE hoặc TO_CHAR) để chuyển đổi ngày tháng thành ngày tháng Julian. Thí dụ, các truy vấn sau cho trở lại tất cả ngày tháng bằng định dạng ngày tháng Julian:

```
SELECT TO_CHAR (hiredate, 'J') FROM emp;
```

Bạn phải sử dụng hàm TO_NUMBER nếu bạn muốn sử dụng ngày tháng Julian trong các tính toán. Bạn có thể sử dụng hàm TO_DATE để nhập ngày tháng Julian:

```
INSERT INTO emp (hiredate) VALUES (TO_DATE(2448921,
'J'));
```

Ngày tháng số học

Ngày tháng số học Oracle dựa vào khoảng cách gần nhất của các lịch được sử dụng suốt lịch sử. Thí dụ, chuyển đổi từ lịch Julian sang lịch Gregorian, 15-10-1582, được trừ đi 10 trước (05-10-1582 tới 14-10-1582). Không có năm.

Bạn có thể nhập ngày tháng bị mất vào CSDL, nhưng chúng không được biểu thị ở ngày tháng bằng số học và được coi là ngày tháng "thực" kế tiếp. Thí dụ, ngày kế tiếp sau ngày 04-10-1582 là 15-10-1582 và ngày theo sau ngày 05-10-1582 cũng là 15-10-1582.

Lưu ý:
Ngày tháng số học không thể áp dụng cho tất cả các chuẩn ngày tháng của mọi nước (chẳng hạn như ở Châu Á).

Các thế kỷ và năm 2000

Oracle lưu trữ dữ liệu năm với thông tin thế kỷ. Thí dụ, CSDL của Oracle lưu trữ 1996 hoặc 2001, và không chỉ là 96 hoặc 01. Kiểu dữ liệu DATE luôn luôn lưu trữ năm bằng bốn chữ số bên trong và tất cả ngày tháng khác được lưu trữ bên trong CSDL có các năm bốn chữ số. Các tiện ích Oracle, chẳng hạn như Import, Export và phục hồi cũng liên quan chính xác các năm bốn chữ số.

Tuy nhiên, một số ứng dụng có thể được viết với sự thừa nhận về năm (chẳng hạn như thừa nhận tất cả đều là19xx). Do đó các lập trình viên ứng dụng phải xét lại và kiểm tra mã của họ liên quan đến năm 2000.

Thông tin bổ sung:
Để có thêm thông tin về các thế kỷ và các mặt che định dạng

ngày tháng, xem hướng dẫn của Oracle8 Application Develop.

Các kiểu dữ liệu LOB

Các kiểu dữ liệu LOB, BLOB, CLOB, NCLOB và BFILE cho phép bạn lưu trữ các khối dữ liệu không có cấu trúc lớn (chẳng hạn như, văn bản, ảnh đồ họa, các video và các dạng sóng âm thanh) có kích thước tối đa bốn gigabyte. Chúng cung cấp truy xuất đầy đủ, ngẫu nhiên, nhạy phân đoạn dữ liệu.

Các kiểu dữ liệu LOB khác với các kiểu dữ liệu LONG và LONG RAW ở nhiều phương diện. Thí dụ, các kiểu dữ liệu LOB (trừ NCLOB) có thể là các thuộc tính của kiểu đối tượng người dùng xác định nhưng các kiểu dữ liệu LONG không thể. Kích thước tối đa của LOB là bốn gigabyte, nhưng kích thước tối đa của LONG là hai gigabyte. Các LOB hỗ trợ truy xuất dữ liệu ngẫu nhiên, nhưng các LONG chỉ hỗ trợ truy xuất liên tiếp.

Các kiểu LOB có thể được lưu trữ inline - trong một bảng (nội tuyến), ngoại tuyến (trong không gian bảng sử dụng bộ định vị LOB) hoặc trong tập tin bên ngoài (các kiểu dữ liệu BFILE).

Thông tin bổ sung:
Xem hướng dẫn của Oracle Application Developer về bộ nhớ LOB và các bộ định vị LOB.

Bạn có thể sử dụng các câu lệnh SQL để xác định các cột LOB trong một bảng và các thuộc tính LOB trong kiểu đối tượng người dùng xác định. Khi xác định các LOB trong một bảng, bạn có thể xác định cụ thể không gian bảng và các đặc tính bộ nhớ cho mỗi LOB.

Xem "Chế Độ Ghi Sổ Mặc Định" về thuộc tính LOGGING hoặc NOLOGGING của LOB.

Kiểu dữ liệu BLOB

Kiểu dữ liệu BLOB lưu trữ dữ liệu nhị phân không có cấu trúc trong CSDL. Các BLOB có thể lưu trữ tối đa bốn gigabyte dữ liệu nhị phân.

Các BLOB tham gia đầy đủ các giao dịch. Các thay đổi xảy ra đối với giá trị BLOB bằng bộ vỏ bọc DBMS_LOB, PL/SQL hoặc OCI có thể được cam kết hoặc hồi phục về trạng thái cũ. Tuy nhiên, các bộ định vị BLOB không thể nối các giao dịch hoặc các phiên.

Các kiểu dữ liệu CLOB và NCLOB

Các kiểu dữ liệu CLOB và NCLOB lưu trữ tối đa bốn gigabyte dữ liệu ký tự trong CSDL. Các CLOB lưu trữ dữ liệu của bộ ký tự một byte và các NCLOB lưu trữ dữ liệu bộ ký tự quốc gia có chiều dài cố định data (dữ liệu NCHAR).

Các CLOB và các NCLOB tham gia toàn bộ các giao dịch. Các thay đổi xảy ra đối với giá trị CLOB hoặc NCLOB bằng bộ phần mềm DBMS_LOB, PL/SQL hoặc OCI có thể được cam kết hoặc hồi phục về trạng thái cũ. Tuy nhiên, các bộ định vị CLOB và NCLOB không thể nối các giao dịch hoặc các phiên.

Bạn không thể tạo một kiểu đối tượng với các thuộc tính NCLOB, nhưng bạn có thể ấn định các tham số NCLOB theo phương pháp đối với một kiểu đối tượng.

Kiểu dữ liệu BFILE

Kiểu dữ liệu BFILE lưu trữ dữ liệu nhị phân không có cấu trúc trong các tập tin của hệ điều hành ngoài CSDL. Cột BFILE hoặc thuộc tính lưu trữ một bộ định vị tập tin chỉ tập tin bên ngoài chứa dữ liệu ấy. Các BFILE có thể lưu trữ tối đa bốn gigabyte dữ liệu.

Các BFILE là chỉ đọc; bạn không thể sửa đổi chúng. Chúng chỉ hỗ trợ đọc ngẫu nhiên (không liên tiếp) và chúng không tham gia vào các giao dịch. Hệ điều hành cơ sở phải duy trì tính toàn vẹn tập tin và độ bền cho các BFILE. Người điều hành CSDL phải bảo đảm tập tin ấy tồn tại và các xử lý của Oracle được phép đọc của hệ điều hành đọc tập tin ấy.

Các kiểu dữ liệu RAW và LONG

Lưu ý:
Các kiểu dữ liệu RAW và LONG RAW được cung cấp cho tính tương thích về trước với các ứng dụng có sẵn. Đối với các ứng dụng mới, bạn phải sử dụng các kiểu dữ liệu BLOB và BFILE cho các khối lượng dữ liệu nhị phân lớn.

Các kiểu dữ liệu RAW và LONG RAW được sử dụng cho dữ liệu được diễn dịch (không được chuyển đổi khi di chuyển dữ liệu giữa các hệ thống khác nhau) bởi Oracle. Những kiểu dữ liệu này được dành cho dữ liệu nhị phân và các chuỗi byte. Thí dụ, LONG RAW có thể được sử dụng để lưu trữ đồ họa, âm thanh, tài liệu hoặc các mảng dữ liệu nhị phân; sự diễn dịch ấy phụ thuộc vào cách sử dụng.

RAW là kiểu dữ liệu có chiều dài khả biến giống như kiểu dữ liệu ký tự VARCHAR2, ngoài trừ Net8 (kết nối các phiên người dùng với môi trường làm việc) và các tiện ích Import và Export không thực hiện chuyển đổi ký tự khi cam kết dữ liệu RAW hoặc LONG RAW data. Trái lại, Net8 và Import/Export tự động chuyển đổi dữ liệu CHAR, VARCHAR2 và LONG giữa bộ ký tự CSDL và bộ ký tự phiên người dùng (được xác lập bằng tham số NLS_LANGUAGE của lệnh ALTER SESSION), nếu hai bộ ký tự khác nhau.

Khi Oracle tự động chuyển đổi dữ liệu RAW hoặc LONG RAW tới và từ dữ liệu CHAR, dữ liệu nhị phân được biểu thị ở dạng thập lục phân cứ bốn bit dữ liệu RAW với một ký tự thập lục phân. Thí dụ, một byte dữ liệu RAW với các bit 11001011 được biểu thị và được nhập là'CB'.

Dữ liệu LONG RAW không thể được lập chỉ mục, nhưng RAW có thể được lập chỉ mục.

Thông tin bổ sung:
Xem Hướng dẫn của Oracle8 Application Developer về các hạn chế khác về kiểu dữ liệu LONG RAW.

Kiểu dữ liệu ROWID

Oracle sử dụng kiểu dữ liệu ROWID mở rộng để lưu trữ địa chỉ của mỗi hàng trong CSDL. ROWID mở rộng nhận dạng các hàng trong các bảng được phân hoạch và các chỉ mục hiệu quả cũng như các bảng không được phân hoạch và những chỉ mục. Nó hỗ trợ các địa chỉ khối dữ liệu liên quan không gian bảng có thể sử dụng đối với tương thích về trước với các ứng dụng có sẵn.

Các ROWID và kiểu dữ liệu ROWID

Mọi hàng trong bảng không được liên kết được gán một ROWID duy nhất tương ứng với địa chỉ vật lý của một đoạn hàng (đoạn hàng đầu nếu hàng ấy được lập chuỗi xích trong nhiều đoạn hàng). Trong trường hợp các bảng được liên kết, các hàng trong các bảng khác nhau trong cùng khối dữ liệu có thể có cùng ROWID.

Mỗi bảng trong CSDL Oracle có một cột giả có tên ROWID bên trong. Cột giả này không rõ ràng khi liệt kê cấu trúc của một bảng bằng cách tiến hành câu lệnh SELECT * FROM . . . hoặc câu lệnh DESCRIBE . . . sử dụng SQL*Plus. Tuy nhiên, mỗi địa

chỉ hàng có thể được truy hồi với truy vấn SQL sử dụng từ dành riêng ROWID là tên cột, thí dụ:
SELECT ROWID, ename FROM emp;

Hàng được gán ROWID vẫn không bị thay đổi trừ khi hàng ấy được Export và Import (bằng cách sử dụng tiện ích IMPORT và EXPORT). Khi bạn xóa một hàng khỏi một bảng (và sau đó cam kết giao dịch hoàn thiện), ROWID được kết hợp của hàng có thể được gán vào một hàng được chèn trong giao dịch kế tiếp.

Bạn không thể xác lập đối tượng của cột giả ROWID bằng các câu lệnh INSERT hoặc UPDATE và bạn không thể xóa giá trị ROWID. Oracle sử dụng các ROWID trong cột giả ROWID bên trong cho các hoạt động khác nhau như được mô tả ở "Các ROWID Được Sử Dụng Như Thế Nào".

Bạn có thể tham chiếu các ROWID trong cột giả ROWID như các cột của bảng khác (được sử dụng trong các danh sách SELECT và các mệnh đề WHERE), nhưng các ROWID không được lưu trữ trong CSDL, chúng cũng không phải là dữ liệu của CSDL. Tuy nhiên, bạn có thể tạo các bảng chứa các cột có kiểu dữ liệu ROWID, dù Oracle không bảo đảm các đối tượng ấy của mỗi cột như thế là các ROWID hợp lệ.

Các ROWID mở rộng

Các ROWID mở rộng sử dụng một cơ số 64 mã hóa địa chỉ vật lý cho mỗi hàng được lựa. Thí dụ, truy vấn sau
SELECT ROWID, ename FROM emp
  WHERE deptno = 20;

có thể cho trở lại thông tin hàng như sau:
ROWID ENAME
------------------ ----------
AAAAaoAATAAABrXAAA BORTINS

AAAAaoAATAAABrXAAE RUGGLES
AAAAaoAATAAABrXAAG CHEN
AAAAaoAATAAABrXAAN BLUMBERG

ROWID mở rộng có bốn định dạng phân đoạn OOOOOOFFFBBBBBBRRR:

OOOOOO: số hiệu đối tượng dữ liệu nhận dạng đoạn CSDL (AAAAao ở trong thí dụ). Các đối tượng giản đồ trong cùng đoạn, chẳng hạn như một cụm liên kết của các bảng có cùng số hiệu đối tượng dữ liệu.

FFF: tập tin dữ liệu chứa hàng (tập tin AAT ở trong thí dụ). Các số hiệu tính toán là duy nhất trong một CSDL.

BBBBBB: khối dữ liệu chứa hàng (khối AAABrX ở trong thí dụ). Các số hiệu khối liên quan đến tập tin dữ liệu, không liên quan đến không gian bảng. Do đó, hai hàng với các số hiệu khối đồng nhất có thể cư trú ở hai tập tin dữ liệu khác nhau của cùng không gian bảng.

RRR: hàng trong khối dữ liệu.

Bạn có thể truy hồi số hiệu đối tượng dữ liệu từ các khung nhìn từ điển dữ liệu USER_OBJECTS, DBA_OBJECTS và ALL_OBJECTS. Thí dụ, truy vấn sau cho trở lại số hiệu dữ liệu đối với bảng EMP trong giản đồ SCOTT:

```
SELECT DATA_OBJECT_ID FROM DBA_OBJECTS
  WHERE OWNER = 'SCOTT' AND OBJECT_NAME = 'EMP';
```

Bạn cũng có thể sử dụng bộ phần mềm DBMS_ROWID để trích thông tin từ một ROWID mở rộng hoặc chuyển đổi một ROWID từ định dạng mở rộng thành định dạng giới hạn (hoặc ngược lại).

Thông tin bổ sung:
Xem Oracle8 Application Developer về bộ phần mềm DBMS_ROWID.

Các ROWID hạn chế

Các ROWID hạn chế sử dụng biểu thị nhị phân của địa chỉ vật lý cho mỗi hàng được lựa. Khi được truy vấn sử dụng SQL*Plus, biểu thị nhị phân được chuyển đổi thành biểu thị VARCHAR2/hexadecimal. Truy vấn sau

```
SELECT ROWID, ename FROM emp
   WHERE deptno = 30;
```

có thể cho lại thông tin hàng sau:

```
ROWID              ENAME
------------------ ----------
00000DD5.0000.0001 KRISHNAN
00000DD5.0001.0001 ARBUCKLE
00000DD5.0002.0001 NGUYEN
```

Như được trình bày ở trên, một biểu thị VARCHAR2/hexadecimal của ROWID hạn chế ở trong định dạng ba phân đoạn, block.row.file:

- Khối dữ liệu chứa hàng (khối DD5 ở trong thí dụ). Các số hiệu khối có liên quan đến tập tin dữ liệu của chúng, không liên kết đến không gian bảng. Do đó, hai hàng có các số hiệu khối đồng nhất có thể cư trú trong hai tập tin dữ liệu khác nhau của cùng không gian bảng.
- Hàng trong khối dữ liệu chứa hàng (các hàng 0, 1, 2 ở trong thí dụ). Các số hiệu hàng của một khối cho trước luôn luôn bắt đầu bằng 0.
- Tập tin dữ liệu chứa hàng (tập tin 1 ở trong thí dụ). Tập tin dữ liệu thứ nhất của mỗi CSDL luôn luôn là 1 và số hiệu tập tin là duy nhất trong CSDL.

Các thí dụ về sử dụng những ROWID

Bạn có thể sử dụng hàm SUBSTR để ngắt dữ liệu trong ROWID thành các thành phần của nó. Thí dụ, bạn có thể sử dụng SUBSTR để tách ROWID mở rộng thành bốn thành phần (đối tượng CSDL, tập tin, khối và hàng):

```
SELECT ROWID,
     SUBSTR(ROWID,1,6) "OBJECT",
     SUBSTR(ROWID,7,3) "FIL",
     SUBSTR(ROWID,10,6) "BLOCK",
     SUBSTR(ROWID,16,3) "ROW"
     FROM products;

ROWID              OBJECT FIL BLOCK  ROW
------------------ ------ --- ------ ---
AAAA8mAALAAAAQkAAA AAAA8m AAL AAAAQk AAA
AAAA8mAALAAAAQkAAF AAAA8m AAL AAAAQk AAF
AAAA8mAALAAAAQkAAI AAAA8m AAL AAAAQk AAI
```

Hoặc bạn có thể sử dụng SUBSTR để tách ROWID hạn chế thành ba thành phần (khối, hàng và tập tin):

```
SELECT ROWID, SUBSTR(ROWID,15,4) "FILE",
     SUBSTR(ROWID,1,8) "BLOCK",
     SUBSTR(ROWID,10,4) "ROW"
     FROM products;

ROWID              FILE BLOCK    ROW
------------------ ---- -------- ----
00000DD5.0000.0001 0001 00000DD5 0000
00000DD5.0001.0001 0001 00000DD5 0001
00000DD5.0002.0001 0001 00000DD5 0002
```

Các ROWID có thể hữu dụng để phát hiện thông tin về bộ nhớ vật lý của một dữ liệu bảng. Thí dụ, nếu bạn quan tâm đến vị trí vật lý của các hàng của một bảng (chẳng hạn như lập dải bảng), truy vấn sau của ROWID mở rộng cho biết bao nhiêu tập tin dữ liệu chứa các hàng của bảng cho trước:

```
SELECT COUNT(DISTINCT(SUBSTR(ROWID,7,3))) "FILES"
FROM tablename;
```

```
FILES
--------
  2
```

Thông tin bổ sung:
Để có thêm thông tin về cách sử dụng các ROWID, tham chiếu với:
-Oracle8 SQL Reference, Hướng dẫn của Người Dùng PL/SQL và Reference, Oracle8 Tuning và các tài liệu khác về các công cụ và tiện ích Oracle.

Các ROWID và các CSDL khác Oracle

Các ứng dụng CSDL Oracle có thể được thực hiện dựa trên các server CSDL khác Oracle bằng cách sử dụng SQL*Connect hoặc Oracle Open Gateway. Trong các trường hợp như thế, định dạng của các ROWID khác nhau tùy theo các đặc tính của hệ thống khác Oracle. Hơn nữa, phiên dịch không theo tiêu chuẩn theo định dạng VARCHAR2/hexadecimal có thể sử dụng. Các chương trình vẫn có thể sử dụng kiểu dữ liệu ROWID; tuy nhiên, chúng phải sử dụng phiên dịch không tiêu chuẩn thành định dạng thập lục phân có chiều dài tối đa 256 byte.

Thông tin bổ sung:
Tham chiếu với tài liệu hướng dẫn thích hợp về các OCI hoặc các trình biên dịch trước về thông tin chi tiết xử lý các ROWID với các hệ thống khác Oracle.

Các ROWID được sử dụng như thế nào

Oracle sử dụng các ROWID bên trong định vị cấu tạo các chỉ mục. Mỗi khóa trong một chỉ mục được kết hợp với một ROWID chỉ địa chỉ của hàng được kết hợp để truy xuất nhanh.

Các người dùng cuối và các Developer ứng dụng cũng có thể sử dụng các ROWID cho nhiều chức năng quan trọng:

Các ROWID là phương tiện nhanh nhất để truy xuất các hàng riêng biệt.

Các ROWID có thể được sử dụng để xem bảng được tổ chức như thế nào.

Các ROWID là các dấu định danh duy nhất đối với các hàng trong một bảng cho trước.

Trước khi bạn sử dụng các ROWID bằng các câu lệnh DML, chúng sẽ được xác minh và được bảo đảm không thay đổi; các hàng có dụng ý sẽ được khóa vì vậy chúng không thể bị xóa. Trong một số trường hợp, yêu cầu dữ liệu bằng một ROWID không hợp lệ có thể làm cho câu lệnh không thực hiện được.

Bạn cũng có thể tạo các bảng có các cột được xác định bằng cách sử dụng kiểu dữ liệu ROWID. Thí dụ, bạn có thể xác định một bảng ngoại lệ có cột của kiểu dữ liệu ROWID lưu trữ các ROWID của các hàng trong CSDL xâm phạm các hạn định tính nguyên vẹn. Các cột được xác định bằng cách sử dụng kiểu dữ liệu ROWID xử lý giống như các cột của các bảng khác; những giá trị có thể được cập nhật, v.v... Mỗi giá trị trong một cột được xác định bằng kiểu dữ liệu ROWID đòi hỏi sáu byte để lưu trữ dữ liệu cột thích hợp.

Kiểu dữ liệu MLSLABEL

Trusted Oracle cung cấp kiểu dữ liệu MLSLABEL, lưu trữ biểu thị của các nhãn của Trusted Oracle được phát sinh bằng các hệ điều hành bảo mật nhiều mức. Trusted Oracle sử dụng các nhãn để điều khiển các truy xuất CSDL.

Bạn có thể xác định một cột sử dụng kiểu dữ liệu MLSLABEL trong Oracle8 về tính tương thích với các ứng dụng Trusted

Oracle, nhưng chỉ giá trị hợp lệ đối với cột ở trong Oracle8 là NULL.

Khi bạn tạo một bảng trong Trusted Oracle, một cột được gọi là ROWLABEL tự động nối ghép vào bảng ấy. Cột này chứa một nhãn của kiểu dữ liệu MLSLABEL cho mỗi hàng trong bảng.

Thông tin bổ sung:
Xem tài liệu Trusted Oracle documentation về kiểu dữ liệu MLSLABEL, cột ROWLABEL và Trusted Oracle.

Tóm tắt về thông tin kiểu dữ liệu Oracle

Bảng 10-2 tóm tắt những thông tin quan trọng về mỗi kiểu dữ liệu Oracle.

Bảng 10-2 Tóm tắt các kiểu dữ liệu Oracle cài sẵn

Kiểu Dữ Liệu	Mô Tả	Chiều Dài Cột Và Mặc Định
CHAR (size)	Dữ liệu ký tự có chiều dài cố định của số byte kích thước chiều dài	Cố định đối với mọi hàng trong bảng (với các khoảng trắng cuối); kích thước tối đa bằng 2000 byte trên mỗi hàng, kích thước mặc định bằng 1 byte trên mỗi hàng. Xét bộ ký tự (một byte hoặc nhiều byte) trước khi xử lý kích thước.
VARCHAR2 (size)	Dữ liệu ký tự có chiều dài khả biến. Kích thước tối đa phải được ấn định.	Khả biến đối với mỗi hàng, tối đa 4000 byte trên mỗi hàng. Xét bộ ký tự (một byte hoặc nhiều byte) trước khi xử lý kích thước.

NCHAR (kích thước)	Dữ liệu ký tự có chiều dài cố định hoặc số byte tùy theo bộ ký tự quốc gia.	Cố định đối với mỗi hàng trong bảng (với các khoảng trắng cuối). Kích thước cột là số ký tự đối với bộ ký tự quốc gia có chiều rộng khác nhau. Kích thước tối đa được xác định bằng số byte cần thiết để lưu trữ một ký tự với một giới hạn trên 2000 byte trên mỗi hàng. Mặc định là 1 ký tự hoặc 1 byte tùy theo bộ ký tự.
NVARCHAR2 (size)	Dữ liệu ký tự có chiều dài khả biến của những ký tự có kích thước chiều dài hoặc số byte tùy theo bộ ký tự quốc gia. Kích thước tối đa phải được ấn định.	Khả biến đối với mỗi hàng. Kích thước cột là số ký tự đối với bộ ký tự quốc gia có chiều rộng cố định hoặc số byte đối với bộ ký tự quốc gia có chiều rộng khác nhau. Kích thước tối đa được xác định bằng số byte cần thiết để lưu trữ một ký tự với đòi hỏi trên 4000 byte trên mỗi hàng. Mặc định là 1 ký tự hoặc 1 byte tùy theo bộ ký tự.
LONG	Dữ liệu ký tự có chiều dài có thể khác nhau.	Khả biến đối với mỗi hàng trong bảng, tối đa 2^31 - 1 byte hoặc 2 gigabyte trên mỗi hàng.
NUMBER (p, s)	Dữ liệu số có chiều dài khả biến. Độ chính xác tối đa p và/hoặc tỷ lệ s	Khả biến đối với mỗi hàng. Không gian tối đa đòi hỏi cho mỗi cột cho trước là 21 byte trên mỗi hàng.

	là 38.	
DATE	Dữ liệu giờ và ngày tháng có chiều dài cố định có phạm vi từ January 1, 4712 BCE tới December 31, 4712 CE ("A.D.")	Cố định 7 byte đối với mỗi hàng trong bảng. Định dạng mặc định là một chuỗi (chẳng hạn như DD-MON-YY) được ấn định bằng tham số NLS_DATE_FORMAT.
RAW (size)	Dữ liệu nhị phân thô có chiều dài khả biến. Kích thước tối đa phải được xác định.	Khả biến đối với mỗi hàng trong bảng, tối đa 2000 byte trên mỗi hàng.
LONG RAW	Dữ liệu nhị phân thô có chiều dài khả biến.	Khả biến đối với mỗi hàng trong bảng, tối đa 2^31 - 1 byte hoặc 2 gigabyte trên mỗi hàng.
BLOB	Dữ liệu nhị phân.	Tối đa 2^32 - 1 byte hoặc 4 gigabyte.
CLOB	Dữ liệu ký tự một byte.	Tối đa 2^32 - 1 byte hoặc 4 gigabyte.
NCLOB	Dữ liệu bộ ký tự quốc gia một byte hoặc nhiều byte (NCHAR).	Tối đa 2^32 - 1 byte hoặc 4 gigabyte.
BFILE	Dữ liệu nhị phân được lưu trữ trong một tập tin bên ngoài.	Tối đa 2^32 - 1 byte hoặc 4 gigabyte.

ROWID	Dữ liệu nhị phân biểu thị các địa chỉ hàng.	Cố định 10 byte (ROWID mở rộng) hoặc 6 byte (ROWID hạn chế) đối với mỗi hàng trong bảng.
MLSLABEL	Kiểu dữ liệu Trusted Oracle.	Xem tài liệu Trusted Oracle.

Các Kiểu Dữ Liệu ANSI, DB2 và SQL/DS

Các kiểu dữ liệu ANSI chuyển đổi thành kiểu dữ liệu Oracle được trình bày ở Bảng 10-3. Các kiểu dữ liệu ANSI/ISO, NUMERIC, DECIMAL và DEC có thể chỉ ấn định các số có dấu chấm cố định. Đối với các kiểu dữ liệu này, s (scale) mặc định thành 0.

Bảng 10-3 Kiểu dữ liệu ANSI chuyển đổi thành các kiểu dữ liệu Oracle

Kiểu Dữ Liệu ANSI SQL	Kiểu Dữ Liệu Oracle
CHARACTER (n), CHAR (n)	CHAR (n)
NUMERIC (p,s), DECIMAL (p,s), DEC (p,s)	NUMBER (p,s)
INTEGER, INT, SMALLINT	NUMBER (38)
FLOAT (p)	FLOAT (p)
REAL	FLOAT (63)
DOUBLE PRECISION	FLOAT (126)
CHARACTER VARYING(n), CHAR VARYING(n)	VARCHAR2 (n)

Các sản phẩm của IBM, các kiểu dữ liệu SQL/DS và DB2, TIME, TIMESTAMP, GRAPHIC, VARGRAPHIC và LONG VARGRAPHIC không có kiểu dữ liệu Oracle tương ứng và không thể được sử dụng. Các kiểu dữ liệu TIME và TIMESTAMP là các thành phần con của kiểu dữ liệu Oracle DATE.

Bảng 10-4 trình bày các chuyển đổi DB2 DB2 và SQL/DS.

Bảng 10-4 Các chuyển đổi kiểu dữ liệu SQL/DS, DB2 thành các kiểu dữ liệu Oracle

Các Kiểu Dữ Liệu DB2 hoặc SQL/DS	Kiểu Dữ Liệu Oracle
CHARACTER (n)	CHAR (n)
VARCHAR (n)	VARCHAR2 (n)
LONG VARCHAR	LONG
DECIMAL (p,s)	NUMBER (p,s)
INTEGER, SMALLINT	NUMBER (38)
FLOAT (p)	FLOAT (p)
DATE	DATE

Chuyển đổi dữ liệu

Trong một số trường hợp, Oracle cung cấp dữ liệu của một kiểu dữ liệu nơi nó cần có dữ liệu của một kiểu dữ liệu khác. Điều này cho phép khi Oracle có thể tự động chuyển đổi dữ liệu thành kiểu dữ liệu được yêu cầu bằng cách sử dụng một trong các hàm sau:

TO_NUMBER()
TO_CHAR()
TO_DATE()
CHARTOROWID()
ROWIDTOCHAR()
HEXTORAW()
RAWTOHEX()

Thông tin bổ sung:

Các nguyên tắc đối với các chuyển đổi kiểu dữ liệu ngầm định được giải thích ở hướng dẫn của Oracle8 Application Developer.

Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle về các chuyển đổi dữ liệu và kiểu dữ liệu MLSLABEL.

Chương 11

Các kiểu dữ liệu người dùng xác định (tùy chọn các đối tượng)

Các tùy chọn đối tượng cho phép các người dùng xác định những kiểu dữ liệu lập mô hình cấu trúc và xử lý dữ liệu trong các ứng dụng của họ.

Chương này gồm những phần sau:

Giới thiệu
Các kiểu dữ liệu người dùng xác định
Các giao diện ứng dụng

Chú ý:
Các đặc trưng được mô tả ở chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với Objects Option. Bất kỳ khi nào thuật ngữ Oracle server xuất hiện ở trong chương này, nó tham chiếu Oracle8 Enterprise Edition với Objects Option. Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về các đặc trưng và các tùy chọn có thể sử dụng với Oracle8 Enterprise Edition.

Giới thiệu

Các hệ thống quản lý CSDL quan hệ (RDBMSs) là công cụ chuẩn để quản lý dữ liệu doanh nghiệp. Chúng cung cấp truy xuất nhanh, hiệu quả và hoàn toàn đáng tin cậy với những số lượng khổng lồ cho hàng triệu doanh nghiệp khắp nơi trên thế giới hàng ngày.

Tùy chọn Objects tạo cho Oracle một hệ thống quản lý CSDL quan hệ đối tượng (ORDBMS), nghĩa là người dùng có thể xác định các loại dữ liệu bổ sung - ấn định cả cấu trúc dữ liệu và các phương pháp điều hoạt nó - và sử dụng những kiểu này trong mô hình quan hệ. Cách giải quyết này bổ sung giá trị vào dữ liệu được lưu trữ trong một CSDL.

Oracle có tùy chọn Objects lưu trữ dữ liệu doanh nghiệp có cấu trúc ở dạng tự nhiên và cho phép những ứng dụng truy hồi nó bằng cách đó. Vì lý do đó, nó hoạt động hiệu quả với các ứng dụng được phát triển bằng cách sử dụng các kỹ thuật lập trình hướng đối tượng.

Hỗ trợ của Oracle cho các kiểu dữ liệu người dùng xác định làm nó dễ dàng hơn cho các Developer ứng dụng làm việc với dữ liệu phức tạp như hình ảnh, audio và video.

Các mô hình dữ liệu phức tạp

Oracle server cho phép bạn xác định các mô hình phức tạp bằng SQL và tạo cho chúng là thành phần của giản đồ CSDL. Các ứng dụng quản lý và dùng chung dữ liệu của bạn chỉ cần có logic ứng dụng, không cần logic dữ liệu.

Một thí dụ

Thí dụ, công ty của bạn có thể sử dụng các đơn đặt mua hàng để tổ chức việc mua hàng, thanh toán các tài khoản, vận chuyển và các chức năng nhận các tài khoản.

Đơn đặt hàng có nhà cung cấp được kết hợp hoặc khách hàng và số mặt hàng không hạn định. Hơn nữa, các ứng dụng thường cần thông tin trạng thái động về các đơn đặt mua hàng. Tham số bạn có thể cần giá trị hiện hành của số hàng được vận chuyển hoặc chưa vận chuyển.

Các phần sau của chương này trình bày cách bạn có thể xác định đối tượng giản đồ được gọi là kiểu đối tượng, đáp ứng khuôn mẫu cho tất cả dữ liệu của đơn đặt mua hàng trong các ứng dụng của bạn. Một kiểu đối tượng xác định các phần tử được gọi là các thuộc tính để thiết lập đơn vị dữ liệu có cấu trúc như đơn đặt mua hàng. Kiểu đối tượng còn xác định các hoạt động được gọi là các phương pháp mà bạn có thể thực hiện trên đơn vị dữ liệu, chẳng hạn như xác định tổng giá trị của đơn đặt mua hàng.

Bạn có thể tạo các đơn đặt mua hàng so khớp với khuôn mẫu và lưu trữ chúng trong các cột bảng, giống như các số hoặc ngày tháng.

Bạn còn có thể lưu trữ các đơn đặt mua hàng trong các bảng đối tượng, trong đó mỗi hàng của bảng tương ứng với đơn đặt mua hàng và các cột bảng là các thuộc tính của đơn đặt mua hàng.

Vì logic của cấu trúc đơn đặt mua hàng và xử lý trong giản đồ của bạn, các ứng dụng không cần biết chi tiết và không phải theo kịp hầu hết các thay đổi.

Oracle sử dụng thông tin giản đồ về các kiểu đối tượng để đạt được hiệu suất chuyển giao quan trọng. Ứng dụng bên client có thể yêu cầu đơn đặt mua hàng của server và nhận tất cả dữ liệu

thích hợp trong một cam kết đơn. Ứng dụng sau đó có thể tiến hành mà không biết các vị trí bộ nhớ hoặc bổ sung chi tiết, định vị giữa các mục dữ liệu liên quan không có các cam kết thêm từ server.

Các kiểu dữ liệu đa phương tiện

Nhiều hiệu suất của các hệ thống CSDL tăng lên từ quản lý được tối ưu hóa của những kiểu dữ liệu cơ bản như số, ngày tháng và ký tự. Các phương tiện tồn tại để so sánh những giá trị, xác định các phân phối của chúng, thiết lập các chỉ mục đầy đủ và thực hiện các tối ưu hóa khác.

Văn bản, video, âm thanh, đồ họa và dữ liệu không gian là các thí dụ về các thực thể doanh nghiệp quan trọng hầu như không thích hợp với các kiểu dữ liệu cơ bản đó. Oracle có tùy chọn Objects hỗ trợ thiết lập mô hình và bổ sung những kiểu dữ liệu phức tạp này.

C ác kiểu dữ liệu người dùng xác định

Chương 10 "Các Kiểu Dữ Liệu Lập Sẵn" mô tả các kiểu dữ liệu lập sẵn của Oracle. Tùy chọn Objects bổ sung hai biệt loại của các kiểu dữ liệu người dùng xác định:
Các kiểu đối tượng
Các kiểu tập hợp

Các kiểu dữ liệu người dùng xác định sử dụng các kiểu dữ liệu lập sẵn như thiết lập các khối cho các kiểu dữ liệu đối với các kiểu dữ liệu để lập mô hình cấu trúc và xử lý dữ liệu trong các ứng dụng.
Các kiểu người dùng xác định là các đối tượng giản đồ. Sử dụng của chúng liên quan cùng các loại điều khiển quản trị bằng các đối tượng giản đồ khác (xem "Sử Dụng Các Kiểu Dữ Liệu Người Dùng Xác Định".

Kiểu đối tượng

Các kiểu đối tượng là các khái niệm trừu tượng hóa của các thực thể thế giới thực - thí dụ, các đơn đặt mua hàng - các trình ứng dụng liên quan. Một kiểu đối tượng là một đối tượng giản đồ có ba loại thành phần:

Tên đáp ứng để nhận dạng kiểu đối tượng duy nhất trong giản đồ đó.

Các thuộc tính lập mô hình cấu trúc và trạng thái của thực thể thế giới thực. Các thuộc tính là các kiểu được lập sẵn hoặc các kiểu người dùng xác định khác.

Các phương pháp là các hàm và các thủ tục được viết bằng PL/SQL và được lưu trữ trong CSDL hoặc được viết bằng ngôn ngữ như C và được lưu trữ bên ngoài. Các phương pháp bổ sung các hoạt động mà ứng dụng có thể thực hiện trên thực thể thế giới thực.

Một kiểu đối tượng là khuôn mẫu. Đơn vị dữ liệu có cấu trúc so khớp khuôn mẫu được gọi là *object* (đối tượng).

Thí dụ đơn đặt mua hàng

Đây là một thí dụ về cách bạn có thể xác định các kiểu dữ liệu được gọi là external_person, lineitem và purchase_order như thế nào.

Các kiểu đối tượng `external_person` và `lineitem` có các thuộc tính của các kiểu lập sẵn. Kiểu đối tượng `purchase_order` có cấu trúc phức tạp hơn để thích hợp sát với cấu trúc của các đơn đặt hàng thực.

Các thuộc tính của `purchase_order` là `id`, `contact` và `lineitems`. Thuộc tính contact là một đối tượng và thuộc tính `lineitems` là một bảng được lồng (xem "Các Bảng Lồng nhau".

```
CREATE TYPE external_person AS OBJECT (
 name     VARCHAR2(30),
 phone    VARCHAR2(20) );
```

```
CREATE TYPE lineitem AS OBJECT (
 item_name   VARCHAR2(30),
 quantity    NUMBER,
 unit_price  NUMBER(12,2) );

CREATE TYPE lineitem_table AS TABLE OF lineitem;

CREATE TYPE purchase_order AS OBJECT (
 id          NUMBER,
 contact     external_person,
 lineitems   lineitem_table,

 MEMBER FUNCTION
 get_value   RETURN NUMBER );
```

đây là một thí dụ đơn giản hóa. Nó không trình bày cách xác định thân của phương pháp `get_value`. Nó cũng không trình bày đầy đủ tính phức tạp của một đơn đặt mua hàng thực.

> Thông tin bổ sung:
> Xem Hướng dẫn của Oracle8 Application Developer về thí dụ của một đơn đặt mua hàng hoàn chỉnh.

Một kiểu đối tượng là một khuôn mẫu. Xác định nó không dẫn đến phân bổ bộ nhớ. Bạn có thể sử dụng `lineitem`, `external_person` hoặc `purchase_order` bằng các câu lệnh SQL trong phần lớn các vị trí tương tự mà bạn có thể sử dụng các kiểu như NUMBER hoặc VARCHAR2.
Thí dụ, bạn có thể xác định một bảng quản lý để truy tìm liên hệ:

```
CREATE TABLE contacts (
 contact     external_person
 date        DATE );
```

Bảng contact là một bảng quan hệ có kiểu đối tượng xác định một trong các cột của nó. Các đối tượng chiếm các cột của các

bảng quan hệ được gọi là các đối tượng cột (xem "Các Đối Tượng Hàng Và Các Đối Tượng Cột".

Các phương pháp

Trong thí dụ `purchase_order` có một phương pháp được đặt tên là get_value. Mỗi đối tượng đơn đặt mua hàng có phương pháp get_value riêng. Thí dụ, nếu x và y là các biến PL/SQL lưu giữ các đối tượng đơn đặt hàng và w và z là các biến lưu giữ các số, hai câu lệnh sau có thể để w và z có các giá trị khác nhau:

```
w = x.get_value();
z = y.get_value();
```

sau các câu lệnh đó, w có giá trị của đơn đặt mua hàng được tham chiếu bằng biến x; z có giá trị của đơn đặt mua hàng bằng biến y.

Thuật ngữ `x.get_value ()` là một viện dẫn của phương pháp `get_value`. Các định nghĩa phương pháp có thể có các tham số, nhưng `get_value` không cần chúng, vì nó tìm tất cả các đối số của nó giữa các thuộc tính của đối tượng mà viện dẫn của nó được liên kết. Nghĩa là ở câu lệnh đầu tiên của các câu lệnh mẫu, nó tính toán giá trị bằng cách sử dụng các thuộc tính của đơn đặt mua hàng x. Trong câu lệnh thứ hai nó tính toán giá trị bằng cách sử dụng các thuộc tính của đơn đặt mua hàng y. Điều này được gọi là kiểu "selfish style" của viện dẫn phương pháp.

Mọi kiểu đối tượng còn có một phương pháp được xác định ngầm định không được liên kết với các đối tượng riêng biệt, phương pháp bộ cấu tạo của kiểu đối tương.

Các phương pháp bộ cấu tạo kiểu đối tượng

Mọi kiểu đối tượng có một phương pháp bộ cấu tạo hệ thống xác định, nghĩa là một phương pháp tạo cho một đối tượng mới theo

định chuẩn của đối tượng ấy. Tên của phương pháp bộ cấu tạo là tên của kiểu đối tượng ấy. Các tham số của nó có các tên và các kiểu thuộc tính của kiểu đối tượng. Phương pháp bộ cấu tạo là một hàm. Nó cho trở lại đối tượng mới bằng giá trị của nó.
Thí dụ, biểu thức
purchase_order(1000376, external_person ("John Smith","1-800-555-1212"), NULL)

biểu thị một đơn đặt mua hàng theo các thuộc tính sau:
id 1000376
contact external_person("John Smith","1-800-555-1212")
lineitems NULL

Biểu thức `external_person ("John Smith", "1-800-555-1212")` là một viện dẫn của hàm bộ cấu tạo cho kiểu đối tượng `external_person`. Đối tượng mà nó cho trở lại trở thành thuộc tính contact của đơn đặt mua hàng đó.

Xem "Nulls" về các đối tượng rỗng và các thuộc tính rỗng.

Các phương pháp so sánh

Các phương pháp đóng một vai trò so sánh các đối tượng. Oracle có các phương tiện để so sánh hai mục dữ liệu của kiểu được lập sẵn trước (thí dụ, hai số) và xác định xem một số có lớn hơn, bằng hoặc nhỏ hơn số kia. Tuy nhiên, Oracle không thể so sánh hai mục của một kiểu người dùng xác định tùy ý không có hướng dẫn thêm từ bộ xác định. Oracle cung cấp hai cách để xác định một quan hệ thứ tự giữa các đối tượng của một kiểu đối tượng cho trước: các phương pháp ánh xạ và các phương pháp thứ tự.

Các phương pháp ánh xạ sử dụng khả năng của Oracle để so sánh các kiểu lập sẵn. Thí dụ, giả sử bạn đã xác định một kiểu đối tượng được gọi là rectangular có các thuộc tính height và width. Bạn có thể xác định một phương pháp sắp xếp các thuộc

tính height và width của hình chữ nhật. Sau đó Oracle có thể so sánh hai hình chữ nhật bằng cách so sách các vùng của chúng.

Các phương pháp thứ tự thông thường hơn. Phương pháp thứ tự sử dụng logic bên trong của riêng nó để so sánh hai đối tượng của một kiểu đối tượng cho trước. Nó cho một giá trị mã hóa mối quan hệ thứ tự. Thí dụ, nó có thể cho trở lại -1 nếu đối tượng thứ nhất nhỏ hơn, 0 nếu chúng bằng nhau và 1 nếu đối tượng thứ nhất lớn hơn.

Thí dụ, giả sử bạn đã xác định một kiểu đối tượng được gọi là address, với các thuộc tính attributes `street`, `city`, `state` và `zip`. Các thuật ngữ "lớn hơn" và "nhỏ hơn" có thể không có ý nghĩa đối với các địa chỉ trong ứng dụng của bạn, nhưng bạn có thể cần thực hiện các tính toán phức tạp để xác định khi hai địa chỉ bằng nhau.

Khi xác định một kiểu đối tượng, bạn có thể ấn định một phương pháp ánh xạ hoặc một phương pháp thứ tự cho nó, nhưng không ấn định cả hai. Nếu kiểu đối tượng không có phương pháp so sánh, Oracle không thể xác định quan hệ lớn hơn hoặc nhỏ hơn giữa hai đối tượng của kiểu đó. Tuy nhiên, nó có thể cố xác định xem hai đối tượng có kiểu ngang nhau không.

Oracle so sánh hai đối tượng của một kiểu thiếu phương pháp so sánh bằng cách so sánh các thuộc tính tương ứng:
Nếu tất cả thuộc tính không rỗng và bằng nhau, Oracle báo cáo các đối tượng bằng nhau.
Nếu có một thuộc tính mà hai đối tượng có các giá trị không rỗng không bằng nhau, Oracle báo cáo chúng không bằng nhau.
Trái lại, Oracle báo cáo so sánh ấy không thể sử dụng (rỗng).

Thông tin bổ sung:
Đối với các thí dụ về cách ấn định và sử dụng các phương pháp so sánh, xem hướng dẫn của Oracle8 Application

Developer.

Các bảng đối tượng

Một bảng đối tượng là một kiểu bảng đặc biệt lưu giữ các đối tượng và cung cấp khung nhìn quan hệ của các thuộc tính của các đối tượng đó.

Thí dụ, câu lệnh sau xác định bảng đối tượng của kiểu external_person được xác định ban đầu:

```
CREATE TABLE external_person_t OF external_person;
```

Oracle cho phép bạn xem bảng này bằng hai cách:

Bảng một cột trong đó mỗi mục nhập là đối tượng external_person.

Bảng nhiều cột trong đó mỗi thuộc tính của kiểu đối tượng external_person, namely name và phone, chiếm một cột.

Thí dụ, bạn có thể thực hiện các chỉ lệnh sau:

```
INSERT INTO external_person_t VALUES (
    "John Smith",
    "1-800-555-1212" );

SELECT VALUE(p) FROM external_person_t p
    WHERE p.name = "John Smith";
```

Chỉ lệnh thứ nhất chèn một đối tượng đơn đặt mua hàng vào external_person_t bằng một bảng nhiều cột. Lệnh thứ hai lựa từ external_person_t bằng một bảng một cột.

Các đối tượng hàng và các đối tượng cột

Các đối tượng xuất hiện trong các bảng đối tượng được gọi là các bảng hàng. Các đối tượng chỉ xuất hiện trong các cột bảng hoặc bằng các thuộc tính của các đối tượng khác được gọi là các đối tượng cột.

Các REF

Trong mô hình quan hệ, các khóa ngoại biểu thị các mối quan hệ nhiều so với một. Oracle có tùy chọn Objects cung cấp phương tiện hiệu quả hơn để biểu thị các mối quan hệ nhiều so với một khi "một" bên của mối quan hệ là một đối tượng hàng. Oracle cho mọi đối tượng hàng một dấu định danh duy nhất bất biến được gọi là dấu định danh đối tượng. Oracle không cung cấp tài liệu và truy xuất về các dấu định danh đối tượng của cấu trúc bên trong. Cấu trúc này có thể thay đổi bất kỳ lúc nào.

Một dấu định danh đối tượng cho phép đối tượng hàng tương ứng được tham chiếu từ các đối tượng khác hoặc từ các bảng quan hệ. Kiểu dữ liệu lập sẵn được gọi là REF biểu thị các tham chiếu như thế. REF gói riêng một tham chiếu với một đối tượng hàng của một kiểu đối tượng được ấn định.

Khung nhìn đối tượng (xem "Các Khung Nhìn Đối Tượng") là một bảng đối tượng ảo. Các hàng của nó là các đối tượng hàng. Oracle cụ thể hóa các dấu định danh đối tượng mà nó không lưu trữ liên tục từ các khóa chính trong bảng cơ sở hoặc khung nhìn. Oracle sử dụng những dấu định danh này để cấu tạo các REF thành các đối tượng hàng trong khung nhìn đối tượng.

Bạn có thể sử dụng REF để khảo sát hoặc cập nhật đối tượng nó tham chiếu. Bạn cũng có thể sử dụng REF để đạt được bản sao của đối tượng nó tham chiếu. Chỉ những thay đổi bạn có thể tạo ra đối với REF để thay thế nội dung của nó với một tham chiếu với đối tượng khác của cùng kiểu đối tượng hoặc gán nó bằng một giá trị rỗng.

Các REF có phạm vi

Trong khai báo kiểu cột, phần tử của tập hợp hoặc thuộc tính kiểu đối tượng phải là REF, bạn có thể hạn định nó chỉ chứa các tham chiếu đối với bảng đối tượng ấn định. Một REF như thế được gọi là REF có phạm vi. Các REF có phạm vi đòi hỏi ít

không gian bộ nhớ hơn và cho phép truy xuất hiệu quả hơn so với các REF không có phạm vi.

Các REF không theo sát

Có thể nhận dạng một đối tượng bằng REF trở thành không sử dụng được - do xóa đối tượng ấy hoặc thay đổi các đặc quyền. Một REF như thế được gọi là không theo sát. SQL của Oracle cung cấp một thuộc tính (được gọi là IS DANGLING) để cho phép kiểm tra các REF về tình trạng này.

Các REF giải tham chiếu

Truy xuất đối tượng được tham chiếu bằng một REF được gọi là REF giải tham chiếu. Oracle cung cấp toán tử DEREF để thực hiện điều này. Giải tham chiếu các kết quả của REF trong một đối tượng rỗng.

Oracle cung cấp các REF ngầm định, thí dụ, xét câu lệnh sau:

```
CREATE TYPE person AS OBJECT (name    VARCHAR2(30),
  manager REF person );
```

nếu x biểu thị một đối tượng của kiểu person, thì biểu thức là
x.manager.name

biểu thị một chuỗi chứa thuộc tính name của đối tượng person được tham chiếu bằng thuộc tính của trình quản lý x. Biểu thị trên là dạng rút gọn của:
y.name, where y = DEREF(x.manager)

Có được các REF

Bạn có thể có được REF đối với một đối tượng hàng bằng cách lựa đối tượng từ bảng đối tượng của nó và áp dụng toán tử. Thí dụ, bạn có thể đạt được REF đơn đặt mua hàng với số định danh 1000376 như sau:

```
DECLARE OrderRef REF to purchase_order;
```

```
SELECT REF(po) INTO OrderRef
        FROM purchase_order_table po
        WHERE po.id = 1000376;
```

Để có thêm thông tin về các đối tượng bộ nhớ và các REF, xem "Bộ Nhớ Của Kiểu Người Dùng Xác Định".

Thông tin bổ sung :
Các thí dụ về cách sử dụng các REF, xem hướng dẫn của Oracle8 Appliation Developer.

Các kiểu tập hợp

Mỗi kiểu tập hợp mô tả một đơn vị dữ liệu được thiết lập bằng một số các phần tử vô hạn, tất cả kiểu dữ liệu tương tự. Các kiểu tập hợp là các kiểu mảng và các kiểu bảng.

Các kiểu mảng và các kiểu bảng là các đối tượng giản đồ. Các đơn vị dữ liệu tương ứng được gọi là các VARRAY và các bảng lồng nhau. Khi không có sự lầm lẫn, chúng ta thường tham chiếu các kiểu tập hợp bằng VARRAYs và các bảng lồng nhau.

Các kiểu tập hợp có các phương pháp bộ cấu tạo. Tên của phương pháp bộ cấu tạo là tên của kiểu ấy và đối số của nó là danh sách riêng của các phần tử của tập hợp mới. Phương pháp bộ cấu tạo là một hàm. Nó cho trở lại tập hợp mới là giá trị của nó.

Một biểu thức gồm tên kiểu được theo sau bằng dấu ngoặc rỗng biểu thị một lệnh gọi phương pháp bộ cấu tạo để tạo một tập hợp rỗng của kiểu đó. Một tập hợp trống khác với một giá trị rỗng.

Các VARRAY

Mảng là một tập hợp có thứ tự của những phần tử dữ liệu. Tất cả những phần tử của một mảng cho trước thuộc cùng kiểu dữ liệu.

Mỗi phần tử có một chỉ mục là số tương ứng với vị trí của phần tử ấy trong mảng.

Số những phần tử trong một mảng là kích thước của mảng. Oracle cho phép các mảng có kích thước khả biến, đó là lý do tại sao chúng được gọi là các VARRAY. Bạn phải ấn định kích thước tối đa khi khai báo kiểu mảng.

Thí dụ, câu lệnh sau khai báo một kiểu mảng:
CREATE TYPE prices AS VARRAY(10) OF NUMBER(12,2);

Các VARRAY của kiểu giá biểu không nhiều hơn 10 phần tử, mỗi kiểu dữ liệu NUMBER(12,2).

Tạo một kiểu mảng không phân bổ không gian. Nó xác định kiểu dữ liệu mà bạn có thể sử dụng làm
Kiểu dữ liệu cột của một bảng quan hệ.
Một thuộc tính kiểu đối tượng
Biến, tham số A PL/SQL hoặc hàm cho trở lại kiểu.

Một VARRAY thường được lưu trữ nội tuyến, nghĩa là trong cùng không gian bảng như dữ liệu khác trong hàng của nó. Tuy nhiên, nếu nó đủ lớn, Oracle lưu trữ nó bằng BLOB (xem Import/Export Của Các Kiểu Người Dùng Xác Định".

> Thông tin bổ sung:
> Để có thêm thông tin về sử dụng các VARRAY, xem hướng dẫn của Oracle8 Application Developer.

Các bảng lồng nhau

Bảng lồng nhau là một tập hợp những phần tử dữ liệu không có thứ tự, tất cả cùng kiểu dữ liệu. Nó có một cột và kiểu của cột đó là kiểu được lập sẵn hoặc một kiểu đối tượng. Nếu kiểu đối

tượng, bảng còn có thể được xem bằng một bảng nhiều cột với mỗi cột cho một thuộc tính của kiểu đối tượng.

Thí dụ, trong thí dụ đơn đặt mua hàng, câu lệnh sau khai báo kiểu bảng được sử dụng cho các bảng lồng nhau của các danh mục hàng:

```
CREATE TYPE lineitem_table AS TABLE OF lineitem;
```

Định nghĩa kiểu bảng không phân bổ không gian bảng. Nó xác định kiểu mà bạn có thể sử dụng bằng:

Kiểu dữ liệu của cột của bảng quan hệ.
Thuộc tính của kiểu dữ liệu
Biến, tham số PL/SQL hoặc hàm cho trở lại kiểu.

Khi kiểu bảng xuất hiện bằng kiểu của cột trong bảng quan hệ hoặc bằng một thuộc tính của kiểu đối tượng cơ sở của bảng đối tượng, thì Oracle lưu trữ tất cả dữ liệu bảng lồng nhau trong bảng đơn mà nó kết hợp với quan hệ chứa bên trong hoặc bảng đối tượng (xem "Các Bảng Lồng Nhau"). Thí dụ, câu lệnh sau xác định một bảng đối tượng cho kiểu đối tượng `purchase_order`:

```
CREATE TABLE purchase_order_table OF purchase_order
  NESTED TABLE lineitems STORE AS lineitems_table;
```

Hàng thứ hai xác định `lineitems_table` là bảng lưu trữ cho các thuộc tính `lineitems` của tất cả những đối tượng `purchase_order` trong `purchase_order_table`.

Một cách truy xuất thuận tiện những phần tử của bảng lồng nhau riêng biệt phải sử dụng con trỏ lồng nhau.

Thông tin bổ sung:
Xem Oracle8 Reference về những con trỏ lồng nhau và xem hướng dẫn của Oracle8 Application Developer về sử dụng các bảng lồng nhau.

Các giao diện ứng dụng

Oracle cung cấp một số phương tiện để sử dụng các kiểu người dùng xác định trong các trình ứng dụng:

- **SQL**
- **PL/SQL**
- **Pro*C/C++**
- **OCI**
- **OTT**

SQL

Oracle SQL DDL cung cấp hỗ trợ cho những kiểu dữ liệu người dùng xác định:

Xác định kiểu đối tượng, các bảng lồng nhau và các mảng
Xác định các đặc quyền
Xác định các cột bảng của các kiểu người dùng xác định
Xác định các bảng đối tượng

Oracle SQL DML cung cấp hỗ trợ sau cho những kiểu dữ liệu người dùng xác định:

Truy vấn và cập nhật các đối tượng và các tập hợp
Điều tác các REF

Thông tin bổ sung:
Để có mô tả đầy đủ về cú pháp Oracle SQL, xem Oracle8 SQL Reference.

PL/SQL

PL/SQL là một ngôn ngữ thủ tục mở rộng SQL. Nó cung ứng những đặc trưng kỹ thuật phần mềm hiện đại như các bộ phần mềm, gói riêng dữ liệu, ẩn thông tin, quá tải và xử lý ngoại lệ. Nó là ngôn ngữ được sử dụng các thủ tục định sẵn.

PL/SQL cho phép sử dụng trong các hàm và các thủ tục của những đặc trưng SQL hỗ trợ các kiểu người dùng xác định.

Những tham số và các biến của các hàm PL/SQL và các thủ tục có thể là các kiểu người dùng xác định.

PL/SQL cung cấp tất cả những khả năng cần thiết để thực hiện các phương pháp được kết hợp với các kiểu đối tượng. Những phương pháp này (các hàm và các thủ tục) cư trú trên server bằng thành phần của giản đồ của người dùng.

Thông tin bổ sung:
Để mô tả đầy đủ về PL/SQL, xem Hướng dẫn và tham chiếu của người dùng PL/SQL.

Pro*C/C++

Trình biên dịch trước của Oracle Pro*C/C++ cho phép các tham số sử dụng những kiểu dữ liệu người dùng xác định bằng các chương trình C và C++.

Những Developer Pro*C có thể sử dụng Object Type Translator để ánh xạ các kiểu đối tượng Oracle và các tập hợp thành những kiểu dữ liệu C được sử dụng trong ứng dụng Pro*C.

Pro*C cung cấp kiểu thời gian biên dịch kiểm các kiểu đối tượng và các tập hợp, chuyển đổi tự động từ những kiểu cơ sở dữ liệu sang các kiểu dữ liệu C.

Pro*C có cú pháp EXEC SQL để tạo và hủy các đối tượng và cung ứng hai cách để truy cập các đối tượng trong server:
Các câu lệnh SQL và các hàm hay các thủ tục PL/SQL được nhúng trong các chương trình Pro*C.

Một giao diện đơn đối với cache đối tượng (được mô tả ở OCI), trong đó các đối tượng có thể được truy xuất bằng các điểm trỏ nằm ngang hoặc các REF, sau đó được sửa đổi và cập nhật trên server ấy.

Thông tin bổ sung:
Để có thông tin mô tả đầy đủ về trình biên dịch trước Pro*C precompiler, xem Hướng dẫn của Oracle Call Interface Programmer.

OCI

Oracle call interface - Giao Diện Gọi Oracle (OCI) là tập hợp của những giao diện ngôn ngữ C với Oracle server. Nó cung cấp cho những lập trình viên rất nhiều khả năng sử dụng linh hoạt các khả năng của server.

Một thành phần quan trọng của OCI là tập hợp các lệnh gọi để cho phép các trình ứng dụng sử dụng không gian làm việc được gọi cache đối tượng ấy. *Object cache* (cache đối tượng) là một khối bộ nhớ ở bên client để cho phép các chương trình có thể lưu trữ toàn bộ các đối tượng và định vị giữa chúng mà không có các dải chung quanh server.

Cache đối tượng hoàn toàn dưới sự điều khiển và quản lý của các trình ứng dụng sử dụng nó. Oracle server không truy xuất nó. Các trình ứng dụng sử dụng nó phải duy trì khả năng liên kết dữ liệu với server và bảo vệ không gian làm việc tránh truy xuất xung đột đồng thời.

OCI cung cấp các chức năng
Truy xuất các đối tượng trên server sử dụng SQL.
Truy xuất, điều tác và quản lý những đối tượng trong cache đối tượng bằng các con trỏ nằm ngang hoặc các REF.

Chuyển đổi ngày tháng, các chuỗi, các số của Oracle thành các kiểu dữ liệu C.
Quản lý kích thước của bộ nhớ của cache đối tượng.
OCI tăng khả năng đồng thời bằng cách cho những đối tượng riêng biệt được khóa. Nó tăng hiệu suất bằng cách hỗ trợ truy hồi đối tượng phức tạp.

Các Developer OCI có thể sử dụng trình phiên dịch kiểu đối tượng để phát sinh các kiểu dữ liệu C tương ứng với các kiểu đối tượng Oracle.

Thông tin bổ sung:
Để có thông tin mô tả đầy đủ về OCI, xem Hướng dẫn của Oracle Call Interface Programmer.

OTT

Oracle type translator - Trình Phiên Dịch Kiểu Oracle (OTT) là một chương trình tự động phát sinh các khai báo cấu trúc ngôn ngữ C tương ứng với các kiểu đối tượng. Các phương tiện OTT sử dụng trình biên dịch trước Pro*C và bộ phần mềm truy xuất server OCI.

Thông tin bổ sung:
Để có thông tin đầy đủ về OTT, xem hướng dẫn của "Oracle Call interface Programmer và Pro*C/C++ Precompiler Programmer.

Chương 12

Sử dụng các kiểu dữ liệu người dùng xác định

Chương này đề cập các khái niệm chính bạn cần hiểu để sử dụng các kiểu dữ liệu người dùng xác định. Nó gồm các phần chính sau:

- **Tham chiếu và giải pháp tên**
- **Bộ nhớ của các kiểu người dùng xác định**
- **Các đặc quyền về các kiểu người dùng xác định và các phương pháp của chúng**
- **Các phụ thuộc và các kiểu không hoàn chỉnh**
- **Import/Export của các kiểu người dùng xác định**

Lưu ý:
Các đặc trưng được mô tả ở chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với Objects Option.

Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về các đặc trưng và các tùy chọn với Oracle8 Enterprise Edition.

Các tham chiếu và giải pháp tên

Oracle SQL được thiết kế để dễ dàng sử dụng. Thí dụ, nếu *projects* là một bảng có cột được gọi là `assignment` và depts là bảng không chứa cột được gọi là `assignment`, bạn có thể viết

```
SELECT *
FROM projects
WHERE EXISTS
 (SELECT * FROM  depts
        WHERE assignment = task);
```

Oracle xác định mỗi cột thuộc bảng nào. Bạn có thể thực hiện nhưng không phải hội chuẩn các tên cột bằng các tên bảng:

```
SELECT *
FROM projects
WHERE EXISTS
 (SELECT * FROM  depts
        WHERE projects.assignment = depts.task);
```

Bạn có thể thực hiện nhưng phải hội chuẩn các tên cột bằng các bí danh bảng:

```
SELECT *
FROM projects pj
WHERE EXISTS
 (SELECT * FROM  depts dp
        WHERE pj.assignment = dp.task);
```

Các bí danh bảng

Biểu mẫu đầu tiên của câu lệnh SELECT trên viết và hiểu dễ nhất, nhưng nó có thể dẫn đến các kết quả ngoài ý muốn nếu sau đó bạn bổ sung cột `assignment` vào bảng `depts` và quên đổi truy vấn ấy. Oracle tự động biên dịch lại truy vấn đó và phiên bản mới sử dụng cột `assignment` của bảng depts. Trường hợp này được gọi là *inner capture* (đón bắt bên trong).

Để tránh đón bắt bên trong và các diễn dịch sai lầm tương tự về những ý nghĩa có dụng ý của các câu lệnh SQL, Oracle đòi hỏi bạn phải sử dụng các bí danh để hội chuẩn các tham chiếu những phương pháp hoặc các thuộc tính của những đối tượng. Điều này còn áp dụng cho các tham chiếu thuộc tính bằng các REF. Hệ điều hành này được gọi là *capture avoidance rule* (nguyên tắc tránh đón bắt).

Thí dụ, xét các câu lệnh sau:
```
CREATE TYPE person AS OBJECT (ssno VARCHAR(20));
CREATE TABLE ptab1 OF person;
CREATE TABLE ptab2 (c1 person);
```

Những câu lệnh này xác định một kiểu đối tượng person và hai bảng. Bảng thứ nhất là một bảng đối tượng cho các đối tượng của kiểu person. Bảng thứ hai có cột đơn của kiểu person.

Bây giờ xét các truy vấn sau:
```
SELECT    ssno FROM ptab1   ; --Correct
SELECT  c1.ssno FROM ptab2   ; --Wrong
SELECT p.c1.ssno FROM ptab2 p ; --Correct
```

Trong câu lệnh SELECT thứ nhất, `ssno` là tên của cột `ptab1`. Không đòi hỏi thêm hội chuẩn nào khác.
Trong câu lệnh SELECT thứ hai, `ssno` là tên của thuộc tính của đối tượng object trong cột có tên `c1`. Tham chiếu này đòi hỏi một bí danh bảng.
Câu lệnh SELECT thứ ba tương tự như câu lệnh thứ hai, nhưng đòi hỏi phải có bí danh bảng, p.

Hội chuẩn các tham chiếu những thuộc tính đối tượng bằng các tên bảng thay vì các bí danh bảng, dù các tên bảng được các tên giản đồ hội chuẩn thêm, nhưng không thỏa mãn đòi hỏi này.

Thí dụ, trong một truy vấn bạn không thể sử dụng biểu thức

scott.projects.assignment.duedate

để tham chiếu thuộc tính duedate của cột assignment về bảng projects thuộc giản đồ scott.

Các bí danh bảng phải duy nhất trong toàn bộ truy vấn và không được trùng các tên giản đồ có thể xuất hiện hợp lệ trong truy vấn.

Lưu ý:
Oracle đề nghị bạn xác định các bí danh bảng trong tất cả các câu lệnh UPDATE, DELETE và SELECT, và các truy vấn con và sử dụng chúng để hội chuẩn các tham chiếu cột dù có hoặc không có các kiểu đối tượng.

Các lệnh gọi phương pháp không có các đối số

Các phương pháp là các hàm và các trình con. Cú pháp chính xác để viện dẫn chúng sử dụng các dấu ngoặc theo sau tên phương pháp để chứa các đối số gọi bên trong. Để tránh không được rõ ràng, Oracle đòi hỏi các dấu ngoặc rỗng cho các lệnh gọi phương pháp không có các đối số.

Thí dụ, nếu tb là một bảng có cột c của kiểu đối tượng t, và t có phương pháp m không lấy các đối số, truy vấn sau minh họa cú pháp chính xác:
SELECT p.c.m() FROM tb p;

Điều này khác với các nguyên tắc đối với các hàm và các thủ tục PL/SQL, trong đó các dấu ngoặc tùy chọn cho các lệnh gọi không có các đối số.

Bộ nhớ của các kiểu người dùng xác định

Oracle lưu trữ và quản lý dữ liệu của các kiểu người dùng xác định trong các bảng. Nó tự động và ánh xạ ẩn cấu trúc phức tạp

của những kiểu người dùng xác định thành các cấu trúc hình chữ nhật đơn giản của các bảng.

Các thuộc tính mức lá

Cấu trúc của một kiểu đối tượng giống như một cái cây. Các nhánh phát triển từ thân tới các thuộc tính/ nếu một thuộc tính của một kiểu đối tượng, nhánh phát sinh các nhánh phụ cho các thuộc tính ấy của kiểu đối tượng mới.

Cuối cùng mỗi nhánh xuất hiện một đầu ở một thuộc tính của kiểu được lập sẵn hoặc một kiểu tập hợp. Những nhánh này được gọi là các thuộc tính mức lá của kiểu đối tượng ban đầu. Oracle cung cấp cột bảng cho mỗi thuộc tính mức lá.

Những thuộc tính mức lá không tập hợp các kiểu được gọi là các thuộc tính vô hướng mức lá của kiểu đối tượng.

Các đối tượng hàng

Trong một bảng đối tượng, mọi thuộc tính REF hoặc vô hướng mức lá trong đó Oracle lưu trữ dữ liệu thực của nó. Điều này còn đúng với các VARRAY, trừ khi chúng quá lớn (xem "VARRAYs"). Oracle lưu trữ các thuộc tính mức lá của các kiểu bảng trong các bảng tách riêng được kết hợp với bảng đối tượng. Bạn phải khai báo những bảng này là thành phần của khai báo bảng đối tượng (xem, "Các Bảng Lồng Nhau").

Truy xuất các thuộc tính riêng biệt của các đối tượng trong một bảng đối tượng là truy xuất đơn giản các cột của bảng ấy. Truy xuất giá trị của chính đối tượng làm cho Oracle viện dẫn bộ cấu tạo mặc định cho kiểu ấy, bằng cách sử dụng các cột của bảng đối tượng làm các đối số. Nghĩa là, Oracle cung cấp bản sao về đối tượng ấy.

Oracle lưu trữ dấu định danh đối tượng hệ thống phát sinh trong cột ẩn. Oracle sử dụng dấu định danh ấy để cấu trúc các REF cho đối tượng đó.

Các đối tượng cột

Khi một bảng được xác định bằng một kiểu đối tượng, Oracle bổ sung các cột ẩn vào bảng cho những thuộc tính mức lá của kiểu đối tượng. Một cột bổ sung lưu trữ thông tin NULL của đối tượng ấy (nghĩa là, các giá trị rỗng nguyên tử của các đối tượng mức trên cùng và các đối tượng lồng nhau).

Các REF

Oracle cấu tạo một REF cho một đối tượng hàng bằng cách gọi hàm lập sẵn REF trên đối tượng hàng. REF có cấu tạo được thiết lập bằng dấu định danh đối tượng, một số meta dữ liệu của bảng đối tượng và tùy chọn ROWID. Một REF không có phạm vi với ROWID đối với một đối tượng trong bảng đối tượng có kích thước 46 byte. Các REF đối với các khung nhìn đối tượng, các REF không có ROWID và các REF có phạm vi nhỏ hơn.

ROWID trong REF được sử dụng làm một ẩn dụ cho truy xuất có hiệu quả. Khi Oracle giải tham chiếu một mục REF, nó sử dụng ROWID chọn một hàng; nếu dấu định danh đối tượng của hàng có định danh so khớp với một hàng trong REF, thì truy xuất ấy thành công. Trái lại, Oracle sử dụng chỉ mục trên dấu định danh đối tượng để nhận dạng hàng chính xác.

Kích thước của REF trong cột kiểu REF phụ thuộc vào tính chất bộ nhớ được kết hợp với cột ấy. Thí dụ, nếu cột được khai báo là REF WITH ROWID, Oracle lưu trữ ROWID trong cột REF; trái lại, nó hủy bỏ ROWID ấy.

Nếu cột được khai báo là REF với mệnh đề SCOPE thì Oracle không lưu trữ meta dữ liệu của bảng đối tượng và ROWID trong cột đó. Một REF có phạm vi dài 16 byte.

Các bảng lồng nhau

Các hàng của một bảng lồng nhau được lưu trữ trong bảng bộ nhớ tách riêng. Bạn phải cung cấp tên bảng bộ nhớ khi xác định bảng chứa bảng lồng nhau. Nếu định nghĩa bảng có nhiều kiểu bảng trong các cột hoặc trong các kiểu đối tượng xuất hiện trong các định nghĩa cột - bạn phải cung cấp bảng bộ nhớ tách riêng cho mỗi kiểu bảng.

Đối với mỗi bảng lồng nhau trong định nghĩa bảng, bảng bộ nhớ được kết hợp với bảng bộ nhớ chứa các hàng của tất cả môi trường làm việc của bảng lồng nhau cho trước trong các hàng của bảng cha.

Các VARRAY

Tất cả các phần tử của VARRAY được lưu trữ trong một cột đơn. Nếu kích thước của mảng nhỏ hơn 4000 byte, Oracle lưu trữ nó nội tuyến; nếu nó lớn hơn 4000 byte, Oracle lưu trữ nó trong BLOB.

Các đặc tính của các thuộc tính đối tượng

Oracle cho phép bạn ấn định một số đặc tính của các thuộc tính đối tượng:

Các Giá Trị Rỗng
Các Mặc Định
Các Hạn Định
Các Chỉ Mục
Các Bộ Kích Khởi

Các giá trị rỗng

Một đặc tính có thể xảy ra của cột bảng, đối tượng, thuộc tính đối tượng, tập hợp hoặc phần tử của tập hợp là nó có thể rỗng. Điều này có nghĩa là mục đã được khởi hoạt thành NULL hoặc đã được để không khởi hoạt. Thường điều này có nghĩa là giá trị của mục chưa được biết rõ nhưng có thể trở thành khả dụng về sau.

Một đối tượng có giá trị là NULL được gọi là rỗng về mặt nguyên tử. Hơn nữa, các thuộc tính của một đối tượng có thể rỗng. Hai cách sử dụng các giá trị rỗng này khác nhau.

Thí dụ, xét bảng contacts được xác định như sau:

```
CREATE TYPE external_person AS OBJECT (
  name       VARCHAR2(30),
  phone      VARCHAR2(20) );

CREATE TABLE contacts (
  contact    external_person
  date       DATE );
```

Câu lệnh

```
INSERT INTO contacts VALUES (
  external_person (NULL, NULL),
 '24 Jun 1997' );
```

cho kết quả khác với

```
INSERT INTO contacts VALUES (
  NULL,
 '24 Jun 1997' );
```

trong cả hai trường hợp, Oracle phân bổ không gian trong contacts cho một hàng mới và xác lập cột date của nó thành giá trị cho trước. Trong trường hợp thứ nhất, Oracle phân bổ không gian cho một đối tượng trong cột external_person và xác lập

mỗi thuộc tính của nó thành NULL. Trong trường hợp thứ hai, nó xác lập cột external_person thành NULL và không phân bổ không gian cho đối tượng.

Hàng của bảng không thể rỗng. Do đó, Oracle không cho phép bạn xác lập một đối tượng hàng thành NULL. Tương tự, một bảng lồng nhau của các đối tượng không thể chứa một phần tử có giá trị NULL.

Một bảng lồng nhau hoặc mảng có thể rỗng. Một tập hợp rỗng khác với một tập hợp rỗng, nghĩa là tập hợp không chứa các phần tử.

Các mặc định

Khi bạn khai báo một cột bảng thuộc một kiểu đối tượng hoặc kiểu tập hợp, bạn có thể có mệnh đề DEFAULT. Điều này cung cấp đối tượng sử dụng trong các trường hợp trong đó bạn không ấn định rõ giá trị cho cột. Mệnh đề mặc định phải chứa lệnh gọi trực kiện của phương pháp bộ cấu tạo cho đối tượng đó hoặc tập hợp.

Lệnh gọi trực kiện của phương pháp bộ cấu tạo được xác định tái quy để là một lệnh gọi của phương pháp bộ cấu tạo trong đó đối số bất kỳ là các trực kiện hoặc các lệnh gọi trực kiện của các phương pháp bộ cấu tạo.

Thí dụ, xét các câu lệnh sau:

```
CREATE TYPE person AS OBJECT (
  id       NUMBER
  name     VARCHAR2(30),
  address  VARCHAR2(30) );

CREATE TYPE people AS TABLE OF person;
```

Sau là lệnh gọi của phương pháp bộ cấu tạo cho kiểu bảng lồng nhau people:

```
people ( person(1, 'John Smith', '5 Cherry Lane'),
         person(2, 'Diane Smith', NULL) )
```

Thí dụ sau trình bày cách sử dụng các lệnh gọi trực kiện của các phương pháp bộ cấu tạo để ấn định các mặc định:

```
CREATE TABLE department (
 d_no    CHAR(5) PRIMARY KEY,
 d_name  CHAR(20),
 d_mgr   person DEFAULT person(1,'John Doe',NULL),
 d_emps  people DEFAULT people() )
 NESTED TABLE d_emps STORE AS d_emps_tab;
```

Lưu ý là thuật ngữ people( ) là lệnh gọi trực kiện của phương pháp bộ cấu tạo cho một bảng people rỗng.

Các hạn định

Bạn có thể xác định các hạn định trên bảng đối tượng giống như bạn có thể thực hiện trên các bảng khác.

Bạn có thể xác định các hạn định trên các thuộc tính vô hướng mức lá của đối tượng cột với ngoại trừ các REF không có phạm vi (xem Các REF Có Phạm Vi).

Các thí dụ sau minh họa các khả năng.

Thí dụ thứ nhất đặt hạn định khóa chính trên cột ssno của bảng đối tượng person_extent:

```
CREATE TYPE location (
 building_no NUMBER,
 city        VARCHAR2(40) );

CREATE TYPE person (
```

```
  ssno        NUMBER,
  name        VARCHAR2(100),
  address     VARCHAR2(100),
  office      location );

CREATE TABLE person_extent OF person (
  ssno        PRIMARY KEY );
```

Bảng của bộ phận trong thí dụ kế tiếp có một cột mà kiểu của nó là vị trí kiểu đối tượng được xác định trong thí dụ trước. Thí dụ xác định các hạn định trên các thuộc tính vô hướng của các đối tượng vị trí xuất hiện trong cột dept_loc của bảng.

```
CREATE TABLE department (
  deptno      CHAR(5) PRIMARY KEY,
  dept_name   CHAR(20),
  dept_mgr    person,
  dept_loc    location,
  CONSTRAINT  dept_loc_cons1
     UNIQUE (dept_loc.building_no, dept_loc.city),
  CONSTRAINT  dept_loc_cons2
      CHECK (dept_loc.city IS NOT NULL) );
```

Các chỉ mục

Bạn có thể xác định các chỉ mục trên bảng đối tượng hoặc trên bảng bộ nhớ cho cột bảng lồng nhau hoặc thuộc tính giống như bạn có thể thực hiện trên các bảng khác.

Bạn có thể xác định các chỉ mục trên những thuộc tính vô hướng mức lá, ngoại trừ bạn chỉ xác định các chỉ mục trên các thuộc tính REF hoặc các cột nếu REF có phạm vi (xem "Các REF Có Phạm Vi").

Thí dụ sau xác định một chỉ mục trên một thuộc tính của cột đối tượng:

```
CREATE TABLE department (
  deptno      CHAR(5) PRIMARY KEY,
  dept_name   CHAR(20),
  dept_addr   address );

CREATE INDEX  i_dept_addr1
        ON  department (dept_addr.city);
```

Mã này tạo một chỉ mục trên thuộc tính về thành phố của định vị department.

Bất kỳ khi nào Oracle có muốn một tên cột trong định nghĩa chỉ mục, bạn còn có thể ấn định thuộc tính vô hướng của cột đối tượng.

Các bộ kích khởi

Bạn có thể xác định các bộ kích khởi trên một bảng đối tượng giống như bạn có thể thực hiện trên các bảng khác. Bạn không thể xác định một bộ kích khởi trên bảng bộ nhớ cho một cột lồng nhau hoặc thuộc tính.

Bạn không thể sửa đổi các giá trị của các tập hợp (hoặc các LOB) trong mã xác định hoạt động của bộ kích khởi. Trái lại không có các giới hạn đặc biệt về các kiểu người dùng xác định với các bộ kích khởi.

Thí dụ sau xác định bộ kích khởi trên bảng person_extent được xác định trong phần trước:

```
CREATE TABLE movement (
    ssno        NUMBER,
    old_office  location,
    new_office  location );

CREATE TRIGGER trig1
```

```
BEFORE UPDATE
        OF  office
        ON  person_extent
FOR EACH ROW
      WHEN  new.office.city = 'REDWOOD SHORES'
BEGIN
  IF :new.office.building_no = 600 THEN
   INSERT INTO movement (ssno, old_office, new_office)
    VALUES (:old.ssno, :old.office, :new.office);
  END IF;
END;
```

Các đặc quyền về các kiểu người dùng xác định và các phương pháp của chúng

Các đặc quyền cho các kiểu người dùng xác định tồn tại ở mức hệ thống và mức đối tượng giản đồ.

Các đặc quyền hệ thống

Oracle xác định các đặc quyền hệ thống cho các kiểu người dùng xác định:

CREATE TYPE cho phép bạn tạo các kiểu người dùng xác định trong giản đồ riêng của mình.

CREATE ANY TYPE cho phép bạn tạo các kiểu người dùng xác định trong giản đồ bất kỳ.

ALTER ANY TYPE cho phép bạn thay đổi các kiểu người dùng xác định trong giản đồ bất kỳ.

DROP ANY TYPE cho phép bạn bỏ các kiểu có tên trong giản đồ bất kỳ.

EXECUTE ANY TYPE cho phép bạn sử dụng và tham chiếu các kiểu có tên trong giản đồ bất kỳ.

Các vai trò CONNECT và RESOURCE có các đặc quyền hệ thống CREATE TYPE. Vai trò DBA có tất cả các đặc quyền trên.

Các đặc quyền đối tượng giản đồ

Chỉ đặc quyền đối tượng giản đồ áp dụng các kiểu người dùng xác định là EXECUTE.

EXECUTE trên kiểu người dùng xác định cho phép bạn sử dụng kiểu ấy để:
Xác định một bảng.
Xác định một cột trong bảng quan hệ.
Khai báo một biến hoặc tham số của kiểu có tên.

EXECUTE cho bạn viện dẫn các phương pháp của kiểu có bộ cấu tạo.

Sự tiến hành phương pháp và các phép được kết hợp giống như đối với các thủ tục PL/SQL.

Sử dụng các kiểu trong các kiểu mới hoặc các bảng

Bổ sung vào các phép được trình bày chi tiết ở các phần trước, bạn cần có các đặc quyền riêng để:
Tạo các kiểu hoặc các bảng sử dụng các kiểu được người dùng khác tạo.
Cho sử dụng các kiểu mới của bạn hoặc các bảng cho các người dùng khác.

Bạn phải có đặc quyền hệ thống EXECUTE ANY TYPE hoặc bạn phải có đặc quyền đối tượng EXECUTE đối với kiểu bất kỳ bạn sử dụng để xác định một kiểu hoặc bảng mới. Bạn phải nhận được các đặc quyền này cụ thể, không phải qua các vai trò.

Nếu bạn định cho những người dùng khác truy xuất kiểu hoặc bảng mới của mình, bạn phải có các đặc quyền đối tượng EXECUTE cần thiết với tùy chọn GRANT hoặc đặc quyền hệ thống EXECUTE ANY TYPE với tùy chọn WITH ADMIN OPTION. Bạn phải nhận được các đặc quyền cụ thể, không qua các vai trò.

Thí dụ

Giả sử là ba người dùng tồn tại có các vai trò CONNECT và RESOURCE: user1, user2, and user3

User1 thực hiện DDL sau trong giản đồ user1:
```
CREATE TYPE type1 AS OBJECT ( attr1 NUMBER );
CREATE TYPE type2 AS OBJECT ( attr2 NUMBER );
GRANT EXECUTE ON type1 TO user2;
GRANT EXECUTE ON type2 TO user2 WITH GRANT OPTION;
```

User2 thực hiện DDL sau trong giản đồ user2:
```
CREATE TABLE tab1 OF user1.type1;
CREATE TYPE type3 AS OBJECT ( attr3 user1.type2 );
CREATE TABLE tab2 (col1 user1.type2 );
```

Các câu lệnh sau thành công, vì user2 có EXECUTE về type2 của user1's với tùy chọn GRANT:
```
GRANT EXECUTE ON type3 TO user3;
GRANT SELECT on tab2 TO user3;
```

Tuy nhiên, sự cho phép sau không thực hiện được, vì user2 không có EXECUTE về user1.type1 với tùy chọn GRANT:
```
GRANT SELECT ON tab1 TO user3;
```

User3 có thể thực hiện thành công các hoạt động sau:
```
CREATE TYPE type4 AS OBJECT (attr4 user2.type3);
CREATE TABLE tab3 OF type4;
```

Các đặc quyền về truy xuất kiểu và truy xuất đối tượng

Các đặc quyền quy định sử dụng các bảng áp dụng các đối tượng bằng nhau:
SELECT cho bạn truy xuất đối tượng và các thuộc tính của nó từ bảng.
UPDATE cho bạn sửa đổi thuộc tính trong bảng.
INSERT cho bạn bổ sung các đối tượng mới vào bảng.
DELETE cho bạn xóa các đối tượng từ bảng.

Các đặc quyền bảng và cột tương tự quy định sử dụng các cột của các kiểu người dùng xác định.

Truy hồi dữ liệu của các kiểu người dùng xác định không đòi hỏi thông tin về kiểu. Tuy nhiên, diễn dịch dữ liệu đòi hỏi thông tin như thế. Khi Oracle nhận các yêu cầu thông tin kiểu, nó xác minh bộ yêu cầu có đặc quyền EXECUTE về kiểu ấy trước khi cung cấp thông tin được yêu cầu.

Xét giản đồ sau:

```
CREATE TYPE emp_type (
  eno    NUMBER,
  ename  CHAR(31),
  eaddr  addr_t );

CREATE TABLE emp OF emp_type;
```

Và hai truy vấn sau:

```
SELECT VALUE(e) FROM emp e;
SELECT eno, ename FROM emp;
```

Đối với hai truy vấn trên, Oracle kiểm đặc quyền SELECT của người dùng cho bảng emp. Đối với truy vấn thứ nhất, người dùng cần có được thông tin kiểu emp_type type để diễn dịch sử dụng

ấy. Khi truy vấn ấy truy xuất kiểu emp_type, Oracle kiểm đặc quyền EXECUTE của người dùng.

Tuy nhiên, sự tiến hành của truy vấn thứ hai không viện dẫn các kiểu có tên, vì vậy Oracle không kiểm các đặc quyền kiểu.

Hơn nữa, sử dụng giản đồ từ phần trên, user3 có thể thực hiện các truy vấn sau:
SELECT tab1.col1.attr2 from user2.tab1 tab1;
SELECT t.attr4.attr3.attr2 FROM tab3 t;

Lưu ý là trong cả hai lựa chọn của user3, user3 không có các đặc quyền rõ ràng trên các kiểu cơ sở, nhưng câu lệnh thực hiện thành công vì các chủ nhân của kiểu và bảng có các đặc quyền cần thiết với tùy chọn GRANT.

Oracle kiểm các đặc quyền trên các yêu cầu sau, và cho trở lại một lỗi nếu bộ yêu cầu không có đặc quyền cho điều hoạt ấy:
Chốt một đối tượng trong cache đối tượng bằng cách sử dụng giá trị REF làm cho Oracle kiểm đặc quyền SELECT trên bảng chứa đối tượng.
Sửa đổi một đối tượng có sẵn hoặc dọn sạch một đối tượng khỏi cache đối tượng, làm cho Oracle kiểm đặc quyền UPDATE trên bảng đối tượng đích. Dọn sạch một đối tượng mới làm cho Oracle kiểm đặc quyền INSERT trên bảng đối tượng đích.
Xóa một đối tượng làm cho Oracle kiểm đặc quyền DELETE trên bảng đích. Chốt một đối tượng của kiểu có tên làm cho Oracle kiểm đặc quyền EXECUTE trên kiểu đối tượng.
Gọi một phương pháp làm cho Oracle kiểm đặc quyền EXECUTE trên kiểu đối tượng tương ứng.

Oracle không cung cấp các đặc quyền mức cột cho các bảng đối tượng.

Các phụ thuộc và các kiểu không hoàn chỉnh

Các kiểu có phụ thuộc vào các định nghĩa của chúng. Thí dụ, bạn có thể muốn xác định các kiểu đối tượng employee và department sao cho một thuộc tính employee là bộ phận mà nhân viên ấy thuộc về và một thuộc tính department là nhân viên ấy quản lý bộ phận đó.

Các kiểu phụ thuộc vào nhau bằng cách này, trực tiếp hoặc bằng các kiểu trung gian được gọi là phụ thuộc lẫn nhau. Một sơ đồ của các kiểu phụ thuộc lẫn nhau có các mũi tên biểu thị các phụ thuộc, luôn luôn biểu lộ đường của các mũi tên bắt đầu và kết thúc của các kiểu.

Oracle cho phép các phụ thuộc chu kỳ chỉ khi ít nhất một nhánh của chu kỳ sử dụng các REF.

Thí dụ, bạn có thể xác định các kiểu sau:

```
CREATE TYPE department;

CREATE TYPE employee AS OBJECT (
  name    VARCHAR2(30),
  dept    REF department,
  supv    REF employee );

CREATE TYPE emp_list AS TABLE OF employee;

CREATE TYPE department AS OBJECT (
  name    VARCHAR2(30),
  mgr     REF employee,
  staff   emp_list );
```

Đây là một tập hợp hợp lệ của các kiểu phụ thuộc lẫn nhau và chuỗi hợp lệ của các câu lệnh SQL DDL. Oracle biên dịch nó không bị lỗi. Câu lệnh thứ nhất

```
CREATE TYPE department;
```

là tùy chọn. Nó thực hiện tiến trình biên dịch không bị lỗi. Nó thiết lập department là một kiểu đối tượng không hoàn chỉnh. Một REF đối với một kiểu đối tượng không hoàn chỉnh biên dịch không bị lỗi, vì vậy biên dịch của tiến trình employee.

Khi Oracle tới câu lệnh cuối cùng để hoàn tất định nghĩa department, tất cả những thành phần của department biên dịch thành công, vì vậy biên dịch kết thúc không bị lỗi.

Không có khai báo tùy chọn về department là một kiểu không hoàn chỉnh, as employee biên dịch không bị lỗi. Sau đó Oracle tự động bổ sung employee vào thư viện các đối tượng giản đồ của nó là một kiểu đối tượng không hoàn chỉnh. Điều này làm cho các khai báo của emp_list và department biên dịch không bị lỗi. Khi employee được biên dịch lại sau emp_list và department hoàn chỉnh, employee biên dịch không bị lỗi và trở thành kiểu đối tượng hoàn chỉnh.

Hoàn tất các kiểu chưa hoàn chỉnh

Một khi bạn đã khai báo một kiểu đối tượng chưa hoàn chỉnh, bạn phải hoàn tất nó là một kiểu đối tượng. Thí dụ, bạn không thể khai báo nó là kiểu bảng hoặc kiểu mảng. Chỉ một thay thế là bỏ kiểu ấy.

Điều này còn đúng nếu Oracle thực hiện kiểu ấy là một kiểu đối tượng chưa hoàn chỉnh cho bạn - vì nó đã thực hiện khi employee không thực hiện được biên dịch trong phần trước.

Hạn chế này áp dụng dù không có các REF đối với kiểu đối tượng chưa hoàn chỉnh bất kỳ nơi nào trong giản đồ. Theo quy tắc của Oracle là một khi nó đánh dấu cờ một kiểu là đích tiềm năng của các REF, thì kiểu đó phải duy trì đích REF tiềm năng cho tới khi nó bỏ.

Oracle chỉ nhận biết các kiểu đối tượng là các đích REF tiềm năng.

Các phụ thuộc kiểu của các bảng

Nếu một bảng chứa dữ liệu dựa vào một định nghĩa kiểu để truy xuất, bất kỳ thay đổi đối với kiểu ấy làm cho dữ liệu của bảng trở thành không thể truy xuất được. Điều này xảy ra nếu các đặc quyền được đòi hỏi bởi kiểu được gọi hoặc nếu kiểu ấy hoặc kiểu mà nó phụ thuộc vào được bỏ. Sau đó bảng trở thành bất hợp lệ và không thể truy xuất.

Một bảng không hợp lệ vì mất các đặc quyền tự động trở thành hợp lệ và truy xuất được nếu các đặc quyền cần thiết được cho.

Một bảng không hợp lệ vì kiểu mà nó phụ thuộc vào đã được bỏ không bao giờ được truy xuất lại. Chỉ hoạt động được phép để bỏ bảng ấy.
Các lệnh của SQL, REVOKE và DROP TYPE cho trở lại một lỗi và nếu kiểu được tham chiếu trong lệnh ấy có các bảng hoặc các kiểu khác phụ thuộc vào nó.

Tùy chọn FORCE với các lệnh này chạy đè để xử lý. Lệnh ấy thực hiện thành công và các bảng hoặc các kiểu bị ảnh hưởng trở thành bất hợp lệ.

Export/Import của các kiểu người dùng xác định

Các tiện ích Export và Import di chuyển dữ liệu vào và ra khỏi các cơ sở dữ liệu. Chúng còn sao lưu hoặc tàng trữ dữ liệu và trợ giúp di trú tới các bản phát hành khác nhau của Oracle RDBMS.

Export và Import hỗ trợ các kiểu người dùng xác định. Export ghi các định nghĩa kiểu người dùng xác định, các định nghĩa thư

viện chức năng ngoại, các định nghĩa bí danh thư mục và tất cả dữ liệu được kết hợp với tập tin xuất dữ liệu. Sau đó Import tái tạo các mục này từ tập tin xuất dữ liệu.

Thông tin bổ sung:
Xem Các Tiện Ích Của Oracle8 về Export và Import.

Chương 13

Các khung nhìn đối tượng

Chương này mô tả các khung nhìn đối tượng. Nó gồm các phần sau:

- **Giới thiệu**
- **Xác định các khung nhìn đối tượng**
- **Sử dụng các khung nhìn đối tượng**

Lưu ý:
Các đặc trưng được mô tả ở chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với tùy chọn Objects. Bất kỳ khi nào thuật ngữ Oracle server xuất hiện ở chương này, nó liên quan đến Oracle8 Enterprise Edition với Objects Option. Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về các đặc trưng và các tùy chọn có thể sử dụng với Oracle8 Enterprise Edition.

Giới thiệu

Giống như một khung nhìn là một bảng ảo, khung nhìn đối tượng là một bảng đối tượng ảo.

Oracle cung cấp những khung nhìn đối tượng làm phần mở rộng của cơ chế khung nhìn quan hệ cơ sở. Bằng cách sử dụng các khung nhìn đối tượng, bạn có thể tạo các bảng đối tượng ảo từ dữ liệu - của các kiểu lập sẵn hoặc người dùng xác định - được lưu trữ ở các cột của các bảng quan hệ hoặc đối tượng trong cơ sở dữ liệu.

Các khung nhìn đối tượng cung cấp khả năng cung ứng truy xuất chuyên môn hoặc bị giới hạn đối với dữ liệu hoặc các đối tượng trong cơ sở dữ liệu. Thí dụ, bạn có thể sử dụng một khung nhìn đối tượng để cung cấp phiên bản của một bảng đối tượng nhân viên không có những thuộc tính có dữ liệu nhạy cảm và không không có phương pháp xóa.

Các khung nhìn đối tượng cho phép sử dụng dữ liệu quan hệ trong các ứng dụng hướng đối tượng. Chúng để người dùng
Thử sử dụng các kỹ thuật lập trình hướng đối tượng mà không phải chuyển đổi các bảng có sẵn.
Chuyển đổi dữ liệu dần và suốt từ các bảng quan hệ tới các bảng quan hệ đối tượng.
Sử dụng các dữ liệu thừa kế với các ứng dụng hướng đối tượng có sẵn.

Các ưu điểm của những khung nhìn đối tượng

Sử dụng những khung nhìn đối tượng có thể dẫn đến đạt được hiệu suất tốt hơn. Dữ liệu quan hệ thiết lập một hàng của một khung nhìn đối tượng qua mạng bằng một đơn vị, tiết kiệm được đáng kể hành trình nhiều vòng.

Bạn có thể truy nạp dữ liệu quan hệ vào cache đối tượng bên client và ánh xạ nó vào các cấu trúc C hoặc C++ vì vậy các ứng dụng 3GL có thể điều tác nó giống như các cấu trúc ban đầu.

Những khung nhìn đối tượng cung cấp đường dẫn di trú dần dữ liệu thừa kế.

Các khung nhìn đối tượng cung cấp các ứng dụng hướng đối tượng và quan hệ cùng tồn tại. Chúng tạo cho nó dễ dàng đưa ra các ứng dụng hướng đối tượng với dữ liệu quan hệ có sẵn mà không phải thay đổi quá nhiều cách bạn lưu trữ dữ liệu trong cơ sở dữ liệu.

Các khung nhìn đối tượng cung cấp khảo sát dữ liệu đối tượng hoặc quan hệ linh hoạt bằng nhiều cách hơn. Do đó bạn có thể sử dụng các ứng dụng khác nhau mà không phải thay đổi cách bạn lưu trữ dữ liệu trong cơ sở dữ liệu.

Xác định các khung nhìn đối tượng

Về mặt khái niệm, thủ tục để xác định một khung nhìn đối tượng đơn giản:

Xác định kiểu đối tượng để được biểu thị bằng các hàng của khung nhìn đối tượng.

Viết truy vấn xác định dữ liệu nào trong đó các bảng chứa những thuộc tính cho các đối tượng của kiểu ấy.

Xác định dấu định danh dựa trên các thuộc tính của dữ liệu cơ sở để cho phép các REF với các đối tượng (hàng) của khung nhìn đối tượng.

Dấu định danh đối tượng tương ứng với dấu định danh đối tượng mà Oracle tự động phát sinh cho các hàng của bảng đối tượng. Tuy nhiên, trong trường hợp các khung nhìn đối tượng khai báo phải ấn định điều nào đó duy nhất trong dữ liệu cơ sở (thí dụ, một khóa chính)

Nếu khung nhìn đối tượng dựa trên một bảng hoặc một dấu định danh đối tượng, Oracle sử dụng dấu định danh đối tượng ấy của bảng ban đầu hoặc khung nhìn đối tượng.

Nếu bạn muốn có thể cập nhật một khung nhìn đối tượng phức tạp, bạn có thể phải thực hiện một Step khác:

Viết thủ tục bộ kích khởi INSTEAD OF (xem "Cập Nhật Các Khung Nhìn Đối Tượng" về Oracle tiến hành bất kỳ khi nào trình ứng dụng cố cập nhật dữ liệu trong khung nhìn đối tượng.

Sau những Step này bạn có thể sử dụng một khung nhìn đối tượng giống như một bảng đối tượng.

Thí dụ, các câu lệnh SQL sau xác định một khung nhìn đối tượng:

```
CREATE TABLE emp_table (
   empnum   NUMBER (5),
   ename    VARCHAR2 (20),
   salary   NUMBER (9, 2),
   job      VARCHAR2 (20) );

CREATE TYPE employee_t (
   empno    NUMBER (5),
   ename    VARCHAR2 (20),
   salary   NUMBER (9, 2),
   job      VARCHAR2 (20) );

CREATE VIEW emp_view1 OF employee_t
   WITH OBJECT OID (empno) AS
      SELECT   e.empnum, e.ename, e.salary, e.job
      FROM     emp_table e
      WHERE    job = 'Developer';
```

Khung nhìn đối tượng lưu ý người dùng giống như một bảng đối tượng có kiểu cơ sở `employee_t`. Mỗi hàng chứa đối tượng của kiểu `employee_t`. Mỗi hàng có một dấu định danh đối tượng duy nhất.

Oracle cấu tạo dấu định danh đối tượng dựa trên khóa được ấn định. Trong hầu hết các trường hợp, nó là khóa chính của bảng cơ sở. Tuy nhiên, nếu truy vấn xác định khung nhìn đối tượng liên quan các liên kết, bạn phải cung cấp một khóa trên tất cả các bảng liên quan đến các liên kết ấy, để khóa đó vẫn là nhận dạng duy nhất hàng của khung nhìn đối tượng ấy.

Lưu ý:
Các cột trong mệnh đề WITH OBJECT OID - `empno` trong thí dụ - cũng phải là các thuộc tính của kiểu đối tượng cơ sở - `employee_t` ở trong thí dụ. Điều này làm cho các chương trình của bộ kích khởi dễ dàng nhận dạng hàng tương ứng trong bảng cơ sở duy nhất.

Sử dụng các khung nhìn đối tượng

Dữ liệu trong các hàng khung nhìn đối tượng có thể xuất phát từ nhiều bảng, nhưng đối tượng ấy vẫn truyền qua mạng bằng một điều hoạt. Khi môi trường làm việc trong cache đối tượng bên client, nó xuất hiện với lập trình viên bằng cấu trúc C hoặc C++ hay bằng một biến đối tượng PL/SQL. Bạn có thể điều tác nó giống như cấu trúc ban đầu bất kỳ khác.

Bạn có thể tham chiếu các khung nhìn đối tượng bằng các câu lệnh SQL bằng cách tương tự như bạn tham chiếu một bảng đối tượng. Thí dụ, các khung nhìn đối tượng có thể xuất hiện trong danh sách SELECT, trong mệnh đề UPDATE-SET hoặc trong mệnh đề WHERE.

Bạn cũng có thể xác định các khung nhìn đối tượng trên các khung nhìn đối tượng.

Bạn có thể truy xuất dữ liệu khung nhìn đối tượng bên client sử dụng cùng các lệnh gọi OCI mà bạn sử dụng cho các đối tượng từ các bảng đối tượng. Thí dụ, bạn có thể sử dụng OCIObjectPin() để chốt REF và OCIObjectFlush() để xóa sạch đối tượng dịch vụ server. Khi bạn cập nhật hoặc xóa sạch đối tượng đối với server trong khung nhìn đối tượng, Oracle cập nhật khung nhìn đối tượng ấy.

Thông tin bổ sung:
Xem hướng dẫn của Oracle Call Interface Programmer về các lệnh gọi OCI.

Cập nhật các khung nhìn đối tượng

Bạn có thể cập nhật, chèn và xóa dữ liệu trong một khung nhìn đối tượng bằng cách sử dụng cùng SQL DML bạn sử dụng cho các bảng đối tượng. Oracle cập nhật các bảng cơ sở của khung nhìn đối tượng nếu rõ ràng.

Một khung nhìn không cập nhật được nếu truy vấn khung nhìn của nó chứa các liên kết, các toán tử xác lập, các hàm nhóm, GROUP BY hoặc DISTINCT. Nếu một truy vấn khung nhìn chứa các cột giả hoặc các biểu thức, các cột khung nhìn tương ứng không cập nhật được. Các khung nhìn Oracle thường liên quan các liên kết.

Để khắc phục các trở ngại này Oracle cung cấp các bộ kích khởi INSTEAD OF (xem "Các Bộ Kích Khởi Cơ Sở Dữ Liệu"). Chúng được gọi là các bộ kích khởi INSTEAD OF vì Oracle tiến hành thân của bộ kích khởi thay vì câu lệnh DML thực.

Các bộ kích khởi INSTEAD OF cung cấp cách cập nhật các khung nhìn đối tượng hoặc các khung nhìn quan hệ thông suốt. Bạn viết các câu lệnh SQL DML (INSERT, DELETE và UPDATE) tương tự như cho bảng đối tượng. Oracle gọi bộ kích

khởi thích hợp thay vì câu lệnh SQL và cửa sổ hoạt động được ấn định trong thân bộ kích khởi xảy ra.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Application Developer về thí dụ đơn đặt mua hàng/danh mục hàng sử dụng bộ kích khởi INSTEAD OF.

Phần V

TRUY XUẤT DỮ LIỆU

Phần V mô tả cách sử dụng các giao dịch gồm các câu lệnh SQL để truy xuất dữ liệu trong cơ sở dữ liệu Oracle và nó mô tả các cấu trúc ngôn ngữ thủ tục cung cấp tính năng bổ sung để truy xuất dữ liệu. Nó còn mô tả bộ tối ưu chọn cách hiệu quả nhất để tiến hành mỗi lệnh SQL.

Phần V gồm các chương sau:

Chương 14 "SQL và PL/SQL"
Chương 15 "Quản lý giao dịch"
Chương 16 "Truy vấn cao cấp"
Chương 17 "Các bộ phần mềm và các thủ tục"
Chương 18 "Các bộ kích khởi cơ sở dữ liệu"
Chương 19 "Quản lý phụ thuộc Oracle"
Chương 20 "Bộ Tối ưu hóa"

Chương 14
SQL và PL/SQL

Chương này cung cấp tổng quan về SQL, Structured Query Language và PL/SQL, Phần Mở Rộng Thủ Tục Oracle với SQL. Chương này gồm:

- **Ngôn ngữ truy vấn có cấu trúc (SQL)**
- **Xử lý SQL**
- **PL/SQL**

Thông tin bổ sung :
Để có thông tin đầy đủ về PL/SQL, xem Hướng dẫn Người Dùng PL/SQL và Reference.

Ngôn ngữ truy vấn có cấu trúc (SQL)

SQL rất đơn giản, nhưng là ngôn ngữ truy xuất cơ sở dữ liệu rất đơn giản và đầy năng lực. SQL là ngôn ngữ phi thủ tục; các người dùng mô tả bằng SQL những gì họ muốn thực hiện và trình biên dịch ngôn ngữ SQL tự động phát sinh thủ tục để định vị cơ sở dữ liệu và thực hiện tác vụ được yêu cầu.

IBM Research phát triển và xác định SQL và ANSI/ISO đã cải tiến SQL làm ngôn ngữ tiêu chuẩn cho các hệ thống quản lý cơ

sở dữ liệu quan hệ. SQL được công ty Oracle bổ sung Oracle thuận tiện 100 % ở Entry Level với người dùng dữ liệu SQL tiêu chuẩn ANSI/ISO 1992.

Oracle SQL gồm nhiều mở rộng cho ngôn ngữ SQL tiêu chuẩn các công cụ Oracle và các ứng dụng cung cấp các lệnh bổ sung. Các công cụ Oracle SQL*Plus, Oracle Enterprise Manager và Server Manager cho phép bạn thực hiện câu lệnh SQL tiêu chuẩn ANSI/ISO bất kỳ dựa trên cơ sở dữ liệu Oracle cũng như các lệnh bổ sung hoặc các hàm có thể sử dụng cho các công cụ đó.

Dù một số công cụ Oracle và các ứng dụng đơn giản hóa hoặc mặt che sử dụng SQL, tất cả các điều hoạt cơ sở dữ liệu được thực hiện bằng cách sử dụng SQL. Phương pháp truy xuất dữ liệu bất kỳ khác sẽ làm mất đi bảo mật được lập sẵn trong Oracle và sắp xếp bảo mật và toàn vẹn dữ liệu có tiềm năng.

Thông tin bổ sung:
Xem Oracle8 SQL Reference chi tiết về các lệnh SQL và các thành phần khác của SQL (chẳng hạn như các toán tử, các hàm và các mô hình định dạng).

Xem hướng dẫn của Oracle Enterprise Manager Administrator về các lệnh Oracle Enterprise Manager và Server Manager kể cả phân biệt chúng với các lệnh SQL.

Phần này gồm các chủ đề sau:

Các Câu Lệnh SQL
Nhận Dạng SQL Không Theo Tiêu Chuẩn
SQL Tái Quy
Các Con Trỏ
SQL Dùng Chung
Phân Ngữ

Các câu lệnh ngôn ngữ điều tác dữ liệu - Data Manipulation Language (DML)

Các câu lệnh DML truy vấn hoặc điều tác dữ liệu trong các đối tượng giản đồ có sẵn. Chúng cho phép bạn

Truy hồi dữ liệu từ một hoặc nhiều bảng hoặc khung nhìn (SELECT)

Bổ sung các hàng dữ liệu mới vào bảng hoặc khung nhìn (INSERT)

Thay đổi những giá trị cột trong các hàng có sẵn của bảng hoặc khung nhìn (UPDATE)

Xóa các hàng khỏi các bảng hoặc các khung nhìn (DELETE)

Xem kế hoạch thực hiện đối với câu lệnh SQL (EXPLAIN PLAN)

Khóa bảng hoặc khung nhìn để giới hạn tạm thời truy xuất của người dùng khác (LOCK TABLE)

Các câu lệnh DML là những câu lệnh thường được sử dụng nhất. Một số thí dụ về các câu lệnh DML như sau:

```
SELECT ename, mgr, comm + sal FROM emp;

INSERT INTO emp VALUES
   (1234, 'DAVIS', 'SALESMAN', 7698, '14-FEB-1988', 1600, 500,
30);

DELETE FROM emp WHERE ename IN ('WARD','JONES');
```

Các câu lệnh ngôn ngữ định nghĩa dữ liệu - Data Definition Language (DDL)

Các câu lệnh DDL xác định, thay đổi cấu trúc của các đối tượng giản đồ và bỏ chúng. Các câu lệnh DDL cho phép bạn:

Tạo, thay đổi và bỏ các đối tượng giản đồ và các cấu trúc cơ sở dữ liệu khác, kể cả chính cơ sở dữ liệu và những người dùng cơ sở dữ liệu (CREATE, ALTER, DROP)

Thay đổi các tên của các đối tượng giản đồ (RENAME)

Xóa tất cả dữ liệu trong các đối tượng giản đồ mà không xóa cấu trúc của đối tượng ấy (TRUNCATE)
Thu thập thống kê về các đối tượng giản đồ, thẩm định cấu trúc đối tượng và liệt kê các hàng được lập chuỗi xích trong các đối tượng (ANALYZE)
Cho và gọi các đặc quyền và các vai trò (GRANT, REVOKE)
Bật và tắt các tùy chọn kiểm soát (AUDIT, NOAUDIT)
Bổ sung chú thích vào từ điển dữ liệu (COMMENT)

Các câu lệnh DDL cam kết ngầm định giao dịch tiến hành và bắt đầu một giao dịch mới.

Một số thí dụ về các câu lệnh DDL như sau:

```
CREATE TABLE plants
  (COMMON_NAME VARCHAR2 (15), LATIN_NAME
VARCHAR2 (40));

DROP TABLE plants;

GRANT SELECT ON emp TO scott;

REVOKE DELETE ON emp FROM scott;
```

Đối với thông tin xác định trên các câu lệnh DDL tương ứng với cơ sở dữ liệu và truy xuất dữ liệu, xem "Điều khiển Truy Xuất Cơ Sở Dữ Liệu", "Các Đặc Quyền Và Các Vai Trò".

Các câu lệnh điều khiển giao dịch

Các câu lệnh điều khiển giao dịch quản lý những thay đổi do các câu lệnh DML và các câu lệnh DML nhóm vào các giao dịch. Chúng cho phép bạn:

Tạo các thay đổi của giao dịch vĩnh viễn (COMMIT)
Hoán chuyển các thay đổi trong một giao dịch từ khi giao dịch được bắt đầu hoặc từ điểm lưu (ROLLBACK)

Xác lập điểm mà bạn có thể hồi phục về trạng thái cũ (SAVEPOINT)
Thiết lập các đặc tính cho một giao dịch (SET TRANSACTION)

Các câu lệnh điều khiển phiên

Các câu lệnh điều khiển phiên quản lý các đặc tính của phiên người dùng riêng biệt. Thí dụ, chúng cho phép bạn:

Thay đổi phiên hiện hành bằng cách thực hiện một chức năng chuyên môn, chẳng hạn như cho phép và vô hiệu hóa phương tiện tìm vệt SQL (ALTER SESSION)
Cho phép và vô hiệu hóa các vai trò (các nhóm của các đặc quyền) cho phiên hiện hành (SET ROLE)

Các câu lệnh điều khiển hệ thống

Các câu lệnh điều khiển hệ thống thay đổi các đặc tính của môi trường làm việc của Oracle server.
Chỉ lệnh điều khiển hệ thống là ALTER SYSTEM. Nó cho chép bạn thay đổi các xác lập (chẳng hạn như số server ít nhất dùng chung) để kết thúc một phiên và thực hiện các tác vụ khác.

Các câu lệnh SQL nhúng

Các câu lệnh SQL nhúng kết hợp chặt chẽ DDL, DML và các câu lệnh điều khiển giao dịch trong một chương trình ngôn ngữ thủ tục. Chúng được sử dụng với các trình biên dịch trước của Oracle.

Các câu lệnh SQL nhúng cho phép bạn:

Xác định, phân bổ và giải phóng các con trỏ ((DECLARE CURSOR, OPEN, CLOSE)
Ấn định cơ sở dữ liệu và kết nối với Oracle (DECLARE DATABASE, CONNECT)
Gán các tên biến (DECLARE STATEMENT)
Khởi hoạt các bộ mô tả (DESCRIBE)

Ấn định các điều kiện lỗi và cảnh báo được xác định như thế nào (WHENEVER)
Phân ngữ và tiến hành các câu lệnh SQL (PREPARE, EXECUTE, EXECUTE IMMEDIATE)
Truy hồi dữ liệu từ cơ sở dữ liệu (FETCH).

Nhận dạng SQL không theo tiêu chuẩn

Oracle cung cấp các mở rộng SQL tiêu chuẩn "Database Language with Integrity Enhancement". Federal Information Processing Standard đối với SQL (FIPS 127-2) yêu cầu các nhà phân phối cung cấp phương pháp để nhận dạng các câu lệnh SQL sử dụng các mở rộng như thế. Bạn có thể nhận dạng hoặc "đánh dấu cờ" các phần mở rộng Oracle trong SQL tương tác, các trình biên dịch trước của Oracle hoặc SQL*Module bằng cách sử dụng các bộ đánh dấu cờ FIPS.

Nếu bạn quan tâm về tính khả chuyển của các ứng dụng của mình vào các bổ sung khác của SQL, sử dụng các bộ đánh dấu cờ FIPS.

Thông tin bổ sung:
Để có thông tin về cách sử dụng các bộ đánh dấu cờ FIPS, xem hướng dẫn của Pro*C/C++ Precompiler Programmer. Pro*COBOL Precompiler Programmer hoặc SQL*Module for Ada Programmer.

SQL tái quy

Khi câu lệnh DDL được phát hành, Oracle phát hành ngầm định các lệnh SQL tái quy để sửa đổi thông tin của từ điển dữ liệu. Những người dùng không cần quan tâm đến SQL tái quy bên trong được Oracle thực hiện.

Các con trỏ

Con trỏ là một trị điều quản hoặc tên cho một *private SQL area* (vùng SQL riêng) - một vùng trong bộ nhớ mà câu lệnh được phân ngữ và thông tin khác để xử lý câu lệnh ấy được lưu giữ.

Dù phần lớn người dùng Oracle dựa vào con trỏ tự động điều quản các tiện ích của Oracle, nhưng các giao diện lập trình cung ứng cho các nhà thiết kế ứng dụng điều khiển các con trỏ nhiều hơn. Trong phát triển ứng dụng, con trỏ là tài nguyên có tên có thể sử dụng đối với một chương trình và có thể được sử dụng đặc biệt để phân ngữ các câu lệnh SQL được nhúng trong ứng dụng đó.

Mỗi phiên người dùng có thể mở nhiều con trỏ tới mức giới hạn được xác lập bởi tham số khởi hoạt OPEN_CURSORS. Tuy nhiên, các ứng dụng phải đóng các con trỏ không cần thiết để dành riêng bộ nhớ hệ thống. Nếu con trỏ không thể được mở do bị giới hạn vào số con trỏ, người điều hành cơ sở dữ liệu có thể thay đổi tham số khởi hoạt OPEN_CURSORS.

Một số câu lệnh (chủ yếu là các câu lệnh DDL) đòi hỏi Oracle phát hành ngầm định các câu lệnh SQL tái quy, cũng đòi hỏi các con trỏ tái quy. Thí dụ, câu lệnh CREATE TABLE tạo ra nhiều cập nhật vào các bảng từ điển dữ liệu khác nhau để ghi bảng và các cột mới. Các lệnh gọi tái quy được thực hiện cho các con trỏ tái quy đó; một con trỏ có thể tiến hành nhiều lệnh gọi tái quy. Những con trỏ tái quy này còn sử dụng các vùng SQL dùng chung.

SQL dùng chung

Oracle tự động thông báo khi các ứng dụng gửi các câu lệnh SQL đồng nhất tới cơ sở dữ liệu. Vùng SQL được sử dụng để xử lý sự kiện thứ nhất của câu lệnh được sử dụng chung - nghĩa là được sử dụng để xử lý các sự kiện xảy ra liên tiếp của cùng câu lệnh

đó. Do đó, chỉ một vùng SQL dùng chung tồn tại cho một câu lệnh duy nhất. Vì các vùng SQL dùng chung là các vùng bộ nhớ dùng chung, tiến trình Oracle bất kỳ có thể sử dụng một vùng SQL dùng chung. Sử dụng chung các vùng SQL làm giảm sử dụng bộ nhớ trên các server cơ sở dữ liệu, từ đó làm tăng hiệu suất hệ thống.

Để thẩm định xem các câu lệnh có đồng nhất không, Oracle xét các câu lệnh SQL được phát hành trực tiếp bởi các người dùng và các ứng dụng cũng như các câu lệnh SQL tái quy được phát hành bên trong bởi một câu lệnh DDL.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Application Developer về SQL dùng chung.

Phân ngữ

Phân ngữ là một giai đoạn trong xử lý câu lệnh SQL. Khi ứng dụng phát hành một câu lệnh SQL, thì ứng dụng ấy đưa ra một lệnh gọi phân ngữ đối với Oracle. Trong khi lệnh gọi phân ngữ, Oracle:

Kiểm câu lệnh để thẩm định cú pháp và ngữ nghĩa
Xác định xem xử lý đưa ra câu lệnh có các đặc quyền để tiến hành nó
Phân bổ vùng SQL riêng tư cho câu lệnh ấy

Oracle còn xác định xem có vùng SQL dùng chung chứa biểu thị câu lệnh được phân ngữ trong cache thư viện không. Nếu có, xử lý của người dùng sử dụng biểu thị được phân ngữ này và tiến hành câu lệnh ngay. Nếu không, Oracle phát sinh biểu thị được phân ngữ của câu lệnh ấy và xử lý của người dùng phân bổ vùng SQL dùng chung cho câu lệnh trong cache thư viện và lưu trữ biểu thị được phân ngữ ở đó.

Lưu ý sự khác biệt giữa ứng dụng đưa ra lệnh gọi phân ngữ câu lệnh SQL và Oracle thực sự phân ngữ câu lệnh ấy. Lệnh gọi phân ngữ bằng ứng dụng kết hợp câu lệnh SQL với vùng SQL riêng tư. Một khi câu lệnh đã được kết hợp với vùng SQL riêng tư, nó có thể được tiến hành lặp lại mà không phải phân ngữ lại.

Cả các lệnh gọi phân ngữ và phân ngữ có thể tiến hành khá phí tốn, vì vậy ít khi được yêu cầu thực hiện chúng.

Điều đề cập này còn áp dụng với phân ngữ của các khối PL/SQL và phân bổ các vùng PL/SQL. (Xem PL/SQL). Các thủ tục định sẵn, các hàm và các bộ phần mềm và các bộ kích khởi được gán các vùng PL/SQL. Oracle cũng gán mỗi câu lệnh SQL trong khối PL/SQL một vùng SQL dùng chung và riêng tư.

Xử lý SQL

Phần này giới thiệu các cơ sở về xử lý SQL; gồm các chủ đề:

- **Tổng Quan Về Sự Tiến hành Câu Lệnh SQL**
- **Xử Lý Câu Lệnh DML**
- **Xử Lý Câu Lệnh DDL**
- **Điều Khiển Các Giao Dịch**

Tổng quan về sự tiến hành câu lệnh SQL

Hình 14-1 trình bày các giai đoạn thường được sử dụng để xử lý và tiến hành một câu lệnh SQL. Trong một số trường hợp, Oracle có thể tiến hành các giai đoạn này theo thứ tự hơi khác nhau. Thí dụ, giai đoạn DEFINE có thể xảy ra ngay trước giai đoạn FETCH tùy theo bạn đã viết mã của mình như thế nào.

Nhiều công cụ Oracle, nhiều giai đoạn được thực hiện tự động. Phần lớn người dùng không cần quan tâm hoặc biết về mức độ chi tiết này. Tuy nhiên, bạn có thể tìm thấy thông tin hữu dụng khi viết các ứng dụng Oracle.

Xử lý câu lệnh DML

Phần này cung cấp một thí dụ đơn giản về những gì xảy ra khi tiến hành một câu lệnh SQL, trong mỗi giai đoạn xử lý câu lệnh DML.

Giả sử là bạn đang sử dụng Pro*C để tăng mức lương cho tất cả nhân viên trong một bộ phận. Ngoài ra, giả sử là chương trình bạn đang sử dụng đã kết nối với Oracle và bạn được kết nối với giản đồ thích hợp để cập nhật bảng EMP. Bạn có thể nhúng câu lệnh SQL sau vào chương trình của mình:

```
EXEC SQL UPDATE emp SET sal = 1.10 * sal
   WHERE deptno = :dept_number;
```

DEPT_NUMBER là một chương trình khả biến chứa giá trị cho số hiệu bộ phận. Khi câu lệnh SQL được tiến hành, giá trị của DEPT_NUMBER được sử dụng, như được phân phối bởi trình ứng dụng ấy.

Các giai đoạn sau cần thiết cho mỗi kiểu xử lý câu lệnh:

Giai đoạn 1: Tạo một con trỏ
Giai đoạn 2: Phân ngữ câu lệnh
Giai đoạn 5: Kết gán các biến bất kỳ
Giai đoạn 7: Tiến hành các câu lệnh
Giai đoạn 9: Đóng con trỏ

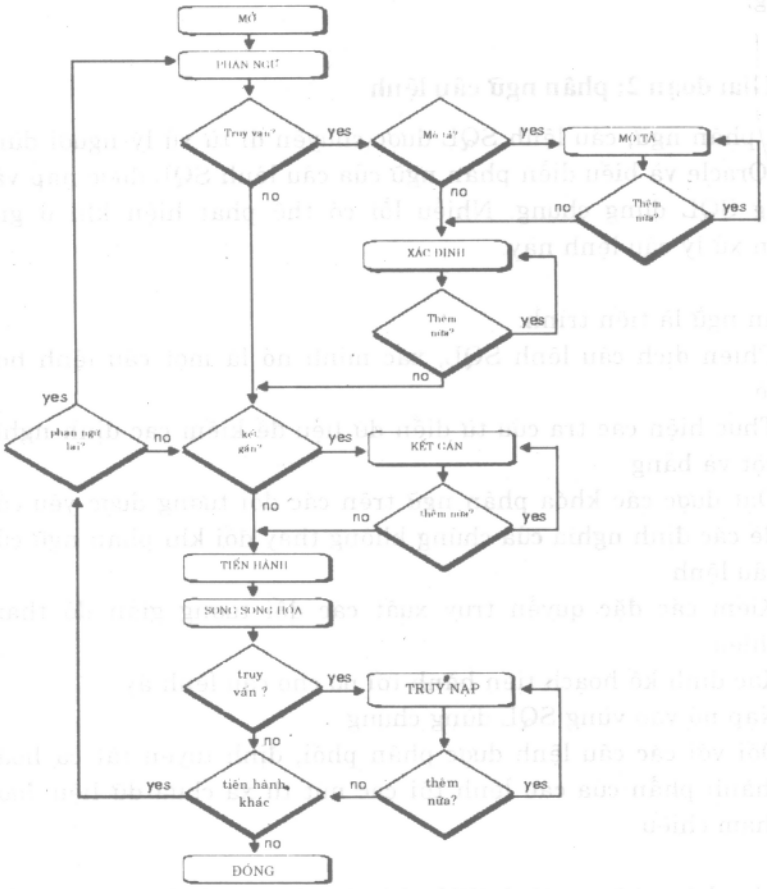
Bạn có thể tùy chọn có cả một giai đoạn khác:

Giai đoạn 6: Song song hóa câu lệnh

Các truy vấn (SELECTs) đòi hỏi nhiều giai đoạn bổ sung, như được trình bày ở "Xử Lý Truy Vấn" ở Hình 14-1:

Giai đoạn 3: Mô tả các kết quả của truy vấn
Giai đoạn 4: Xác định kết xuất của truy vấn
Giai đoạn 8: Truy nạp các hàng của truy vấn
Giai đoạn 9: Đóng con trỏ

Hình 14-1 Các giai đoạn trong xử lý câu lệnh SQL



Xem thêm "Xử Lý Truy Vấn"

Giai đoạn 1: tạo con trỏ

Một lệnh gọi giao diện có thể tạo một con trỏ. Con trỏ được tạo độc lập với câu lệnh SQL bất kỳ; nó được tạo trong khả năng của bất kỳ câu lệnh SQL. Trong phần lớn các ứng dụng, tạo con trỏ

tự động. Tuy nhiên, trong các chương trình biên dịch trước, tạo con trỏ có thể xảy ra ngầm định hoặc được khai báo trước rõ ràng.

Giai đoạn 2: phân ngữ câu lệnh

Khi phân ngữ, câu lệnh SQL được chuyển đi từ xử lý người dùng tới Oracle và biểu diễn phân ngữ của câu lệnh SQL được nạp vào vùng SQL dùng chung. Nhiều lỗi có thể phát hiện khi ở giai đoạn xử lý câu lệnh này.

Phân ngữ là tiến trình:

Phiên dịch câu lệnh SQL, xác minh nó là một câu lệnh hợp lệ.

Thực hiện các tra cứu từ điển dữ liệu để kiểm các định nghĩa cột và bảng

Đạt được các khóa phân ngữ trên các đối tượng được yêu cầu để các định nghĩa của chúng không thay đổi khi phân ngữ của câu lệnh

Kiểm các đặc quyền truy xuất các đối tượng giản đồ tham chiếu

Xác định kế hoạch tiến hành tối ưu cho câu lệnh ấy

Nạp nó vào vùng SQL dùng chung

Đối với các câu lệnh được phân phối, định tuyến tất cả hoặc thành phần của câu lệnh tới các nút từ xa chứa dữ liệu được tham chiếu

Oracle phân ngữ câu lệnh SQL chỉ nếu vùng SQL dùng chung cho câu lệnh SQL đồng nhất không tồn tại ở vùng chia dùng chung. Trong trường hợp này, vùng SQL dùng chung mới được phân bổ và câu lệnh được phân ngữ. (Xem "SQL Dùng Chung").

Giai đoạn phân ngữ có các đòi hỏi xử lý phải được thực hiện chỉ một lần bất kể câu lệnh ấy được tiến hành bao nhiêu lần. Oracle

phiên dịch mỗi câu lệnh SQL chỉ một lần, thực hiện lại câu lệnh được phân ngữ khi các tham chiếu theo sau câu lệnh ấy.

Dù phân ngữ câu lệnh SQL thẩm định câu lệnh đó, nhưng phân ngữ chỉ nhận dạng các lỗi có thể được tìm thấy trước khi câu lệnh được tiến hành.
Do đó, một số lỗi không thể phát hiện được bằng phân ngữ. Thí dụ, các lỗi trong chuyển đổi dữ liệu hoặc các lỗi trong dữ liệu (chẳng hạn như cố nhập các giá trị sao chép ở khóa chính) và các bế tắc đều là các lỗi hoặc các trường hợp có thể bị gặp phải và chỉ được báo cáo vào giai đoạn tiến hành.

Xử lý truy vấn

Các truy vấn khác với các kiểu câu lệnh SQL bởi vì, nếu thành công, chúng cho trở lại làm các kết quả. Trong khi các câu lệnh khác chỉ cho biết thành công hoặc thất bại, nhưng truy vấn có thể cho trở lại một hàng hoặc hàng ngàn hàng. Các kết quả của truy vấn luôn luôn bằng *tabular format* (định dạng xếp thành bảng) và các hàng của kết quả được truy nạp (truy hồi), mỗi lần một hàng hoặc bằng các nhóm.

Nhiều vấn đề chỉ liên quan đến xử lý truy vấn. Các truy vấn không chỉ có các câu lệnh SELECT rõ ràng mà còn có các truy vấn ngầm định (các truy vấn con) trong các câu lệnh khác. Thí dụ, mỗi câu lệnh sau đòi hỏi một truy vấn làm thành phần của sự tiến hành của nó:
INSERT INTO table SELECT...

UPDATE table SET x = y WHERE...

DELETE FROM table WHERE...

CREATE table AS SELECT...

Đặc biệt, các truy vấn:

Đòi hỏi đọc nhất quán

Có thể sử dụng các đoạn tạm thời để xử lý trung gian

Có thể đòi hỏi mô tả, xác định và truy nạp các giai đoạn của xử lý câu lệnh SQL.

Giai đoạn 3: mô tả các kết quả của truy vấn

Giai đoạn mô tả chỉ cần thiết nếu các đặc tính của kết quả truy vấn không biết rõ; thí dụ, khi truy vấn được nhập tương tác bởi người dùng.

Trong trường hợp này, giai đoạn mô tả xác định các đặc tính (các kiểu dữ liệu, các chiều dài và các tên) của kết quả truy vấn.

Giai đoạn 4: xác định kết xuất của truy vấn

Trong giai đoạn xác định các truy vấn, bạn ấn định vị trí, kích thước và kiểu dữ liệu của các biến được xác định để nhận mỗi giá trị được truy nạp. Oracle thực hiện chuyển đổi kiểu dữ liệu nếu cần thiết.

Giai đoạn 5: kết gán các biến bất kỳ

Ở điểm này, Oracle biết ý nghĩa của câu lệnh SQL nhưng vẫn không có đủ thông tin để tiến hành câu lệnh. Oracle cần những giá trị cho bất kỳ các biến được liệt kê trong câu lệnh ấy; trong thí dụ, Oracle cần một giá trị cho DEPT NUMBER. Tiến trình đạt được những giá trị này được gọi là các biến kết gán.

Một chương trình phải ấn định vị trí (địa chỉ bộ nhớ), nơi giá trị có thể được tìm thấy. Các người dùng cuối các ứng dụng có thể không biết họ đang kết gán các biến, vì tiện ích Oracle có thể chỉ nhắc họ về giá trị mới.

Vì bạn ấn định vị trí (kết gán bằng tham chiếu), bạn không cần thiết lập lại biến ấy trước khi thực hiện lại. Bạn có thể thay đổi

giá trị và Oracle tra cứu giá trị mỗi lần thực hiện bằng cách sử dụng địa chỉ bộ nhớ.

Bạn cũng phải ấn định kiểu dữ liệu và chiều dài cho mỗi giá trị (trừ khi chúng được ám chỉ hoặc mặc định) nếu Oracle cần thực hiện chuyển đổi kiểu dữ liệu.

Thông tin bổ sung:
Để có thêm thông tin về ấn định kiểu dữ liệu và chiều dài cho một giá trị, tham chiếu các tài liệu sau:
Hướng dẫn của Oracle Call Interface Programmer
Hướng dẫn của Pro*C/C++ Programmer (xem "Dynamic SQL Method 4")

Giai đoạn 6: song song hóa câu lệnh

Oracle có thể song song hóa các truy vấn (SELECTs), INSERTs, UPDATEs, DELETEs và một số điều hoạt DDL chẳng hạn như tạo chỉ mục, tạo bảng với truy vấn con và các điều hoạt về phân hoạch. Sự song song hóa tạo cho nhiều xử lý server thực hiện hoạt động của câu lệnh SQL để nó có thể hoàn tất nhanh.

Xem "Sự Tiến Hành Song Song" về SQL song song.

Giai đoạn 7: tiến hành câu lệnh

Ở điểm này, Oracle có tất cả những thông tin cần thiết và các tài nguyên để câu lệnh ấy được tiến hành. Nếu câu lệnh là một truy vấn hoặc một câu lệnh INSERT, không hàng nào cần phải khóa vì dữ liệu không bị thay đổi. Nếu câu lệnh là một câu lệnh UPDATE hoặc DELETE, tuy nhiên tất cả các hàng mà các tác động câu lệnh được khóa không cho người dùng khác sử dụng cơ sở dữ liệu cho tới câu lệnh kế tiếp COMMIT, ROLLBACK hoặc SAVEPOINT giao dịch ấy. Điều này bảo đảm tính toàn vẹn dữ liệu.

Đối với một số câu lệnh bạn có thể ấn định số lệnh tiến hành để thực hiện. Điều này được gọi là *array processing* (xử lý mảng). *n* lệnh tiến hành cho trước, các vị trí kết gán và xác định được coi là bắt đầu kích thước *n* mảng.

Giai đoạn 8: truy nạp các hàng truy vấn

Trình ứng dụng giai đoạn truy nạp, các hàng được lựa và có thứ tự (nếu được truy vấn yêu cầu) và mỗi truy nạp thành công truy hồi một hàng kết quả khác cho tới khi hàng cuối cùng đã được truy nạp.

Giai đoạn 9: đóng con trỏ

Giai đoạn cuối cùng xử lý câu lệnh SQL là đóng con trỏ.

Xử lý câu lệnh DDL

Sự tiến hành của các câu lệnh DDL khác với sự tiến hành của các câu lệnh DML và các truy vấn vì sự thành công của câu lệnh DDL đòi hỏi ghi truy xuất vào từ điển dữ liệu. Đối với các câu lệnh, phân ngữ (Giai đoạn 2) thực ra có phân ngữ, tra cứu từ điển và sự tiến hành lệnh.

Quản lý giao dịch, quản lý phiên và các câu lệnh SQL quản lý hệ thống được xử lý bằng cách sử dụng phân ngữ và tiến hành các giai đoạn. Để tiến hành lại chúng, chỉ cần thực hiện một sự tiến hành khác.

Điều khiển các giao dịch

Nói chung, chỉ các người thiết kế ứng dụng sử dụng các giao diện lập trình với Oracle quan tâm đến các kiểu hoạt động tạo nhóm với nhau là một giao dịch. Các giao dịch phải được xác định chính xác để công việc được thực hiện bằng các đơn vị logic và

dữ liệu được lưu giữ nhất quán. Giao dịch phải có tất cả các thành phần cần thiết cho một đơn vị logic hoạt động, không hơn không kém.
Dữ liệu trong các bảng tham chiếu sẽ ở trạng thái nhất quán trước khi giao dịch bắt đầu và sau khi nó kết thúc.
Các giao dịch phải gồm chỉ các câu lệnh SQL tạo một sự thay đổi nhất quán đối với đối tượng ấy.

Thí dụ, một cam kết các quỹ giữa hai tài khoản (giao dịch hoặc đơn vị hoạt động logic) sẽ gồm bên nợ vào một tài khoản (một câu lệnh SQL) và bên có vào một tài khoản khác. Cả hai hoạt động thành công hoặc thất bại cùng với nhau là một đơn vị hoạt động; bên có sẽ không xảy ra nếu không có bên nợ. Các hoạt động không liên quan khác, chẳng hạn như một khoản tiền gửi khác vào một tài khoản, sẽ không có trong cam kết giao dịch của các quỹ.

Hơn nữa xác định những kiểu nào của các hoạt động tạo thành một giao dịch, khi bạn thiết kế một ứng dụng bạn còn phải xác định khi nó cần thiết sử dụng thủ tục BEGIN_DISCRETE_TRANSACTION để tăng hiệu suất của các giao dịch ngắn, không được phân phối. Xem thêm "Quản Lý Giao Dịch Thận Trọng".

PL/SQL

PL/SQL là ngôn ngữ thủ tục của Oracle mở rộng thành SQL. PL/SQL cho phép bạn trộn các câu lệnh SQL với các cấu trúc thủ tục. Với PL/SQL, bạn có thể xác định và tiến hành các đơn vị chương trình PL/SQL chẳng hạn như các thủ tục, các hàm và các bộ phần mềm.

Các đơn vị chương trình PL/SQL thường được phân loại bằng các khối nặc danh và các thủ tục định sẵn.

Một khối nặc danh là một khối PL/SQL xuất hiện trong ứng dụng của bạn và nó không được đặt tên hoặc lưu trữ trong cơ sở dữ liệu. Trong nhiều ứng dụng, các khối PL/SQL có thể xuất hiện bất kỳ khi nào các câu lệnh SQL có thể xuất hiện.

Một thủ tục định sẵn là khối PL/SQL mà Oracle lưu trữ trong cơ sở dữ liệu và có thể được gọi bằng tên từ một ứng dụng. Khi bạn tạo một thủ tục định sẵn, Oracle phân ngữ thủ tục và lưu trữ sự biểu thị của nó trong cơ sở dữ liệu. Oracle còn cho phép bạn tạo và lưu trữ các hàm (giống như các thủ tục) và các bộ phần mềm (là các nhóm thủ tục và các hàm).

Để có thêm thông tin về các thủ tục định sẵn, các hàm, các bộ phần mềm và các bộ kích khởi cơ sở dữ liệu, xem "Các Thủ Tục Và Các Bộ Phần Mềm" và "Các Bộ Kích Khởi Cơ Sở Dữ Liệu".

PL/SQL tiến hành như thế nào

Động cơ PL/SQL xử lý các đơn vị chương trình PL/SQL là thành phần đặc biệt của nhiều sản phẩm Oracle kể cả Oracle server.

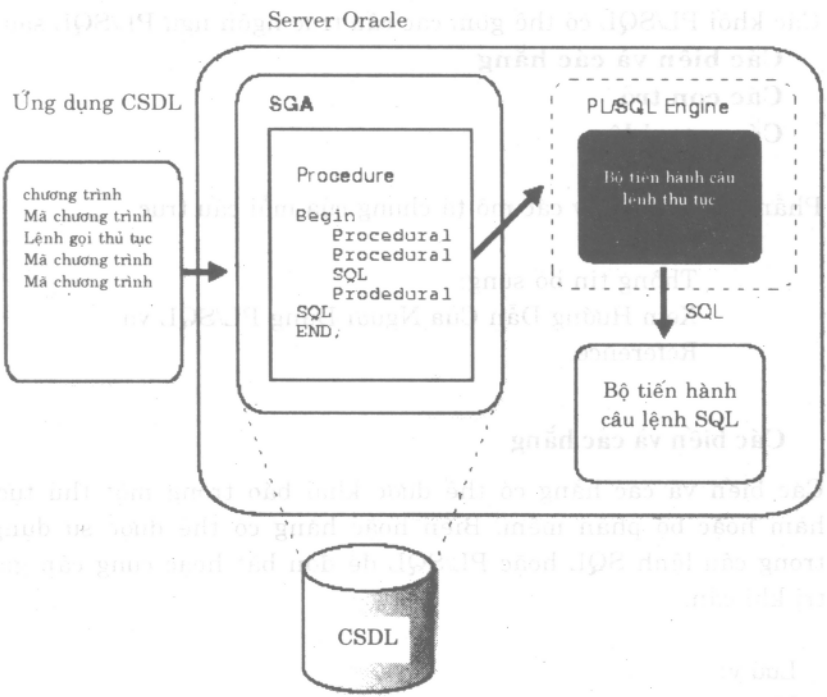
Hình 14-2 minh họa động cơ PL/SQL trong Oracle server.

Thủ tục (hoặc bộ phần mềm) được lưu trữ trong cơ sở dữ liệu. Khi một ứng dụng gọi một thủ tục định sẵn trong cơ sở dữ liệu, thì Oracle nạp thủ tục được biên dịch (hoặc bộ phần mềm) vào vùng chia dùng chung trong vùng toàn cục hệ thống (SGA) và các bộ tiến hành câu lệnh PL/SQL và SQL hoạt động cùng với nhau để xử lý các câu lệnh trong thủ tục ấy.

Các sản phẩm Oracle sau có động cơ PL/SQL:
Oracle Server
Oracle Forms(Version 3 và về sau)
SQL*Menu (Version 5 và về sau)
Oracle Reports (Version 2 và về sau)

Oracle Graphics (Version 2 và về sau)

Hình 14-2 Động cơ PL/SQL và Oracle Server



Bạn có thể gọi một thủ tục định sẵn từ một khối PL/SQL khác. Thí dụ, bạn có thể gọi thủ tục dữ liệu từ các Oracle Forms (Version 3 và về sau).

Hơn nữa, bạn có thể chuyển các khối nặc danh vào Oracle từ các ứng dụng được phát triển với các công cụ này:
Các trình biên dịch trước của Oracle (kể cả các tồn tại của người dùng)
Oracle Call Interfaces (OCIs)
SQL*Plus
Server Manager

Oracle Enterprise Manager

Các cấu trúc ngôn ngữ cho PL/SQL

Các khối PL/SQL có thể gồm các cấu trúc ngôn ngữ PL/SQL sau:

Các biến và các hằng
Các con trỏ
Các ngoại lệ

Phần này trình bày các mô tả chung của mỗi cấu trúc.

> Thông tin bổ sung:
> Xem Hướng Dẫn Của Người Dùng PL/SQL và Reference.

Các biến và các hằng

Các biến và các hằng có thể được khai báo trong một thủ tục, hàm hoặc bộ phần mềm. Biến hoặc hằng có thể được sử dụng trong câu lệnh SQL hoặc PL/SQL để đón bắt hoặc cung cấp giá trị khi cần.

> Lưu ý:
> Một số công cụ tương tác, chẳng hạn như Server Manager cho phép bạn xác định các biến trong phiên hiện hành của mình. Bạn có thể sử dụng các biến như thế giống như bạn sử dụng các biến được khai báo trong các thủ tục hoặc các bộ phần mềm.

Các con trỏ

Các con trỏ có thể được khai báo rõ ràng trong một thủ tục, hàm hoặc bộ phần mềm để tạo thuận tiện xử lý hướng bản ghi dữ liệu của Oracle. Các con trỏ còn có thể được khai báo ngầm định (để

hỗ trợ các hoạt động điều tác dữ liệu khác) bằng động cơ PL/SQL.

Các ngoại lệ

PL/SQL cho phép bạn điều quản rõ ràng các điều kiện lỗi người dùng xác định và điều kiện bên trong được gọi là các ngoại lệ, phát sinh khi xử lý mã PL/SQL. Các ngoại lệ bên trong do các hoạt động bất hợp lệ gây ra, chẳng hạn như chia cho zêrô hoặc các lỗi Oracle được cho trở lại từ mã PL/SQL. Các ngoại lệ người dùng xác định được xác định rõ ràng và được phát tín hiệu trong khối PL/SQL để điều khiển xử lý các lỗi được ấn định với ứng dụng ấy (thí dụ, ghi nợ một tài khoản và để số dư âm).

Khi một ngoại trừ được phát sinh (được phát tín hiệu), sự tiến hành bình thường của mã PL/SQL dừng và một thủ tục con được gọi một trị quản ngoại lệ được viện dẫn. Các trị quản ngoại lệ riêng biệt có thể được viết để điều quản bất kỳ ngoại lệ bên trong hoặc được người dùng xác định.

Các thủ tục định sẵn

Oracle còn cho phép bạn tạo và lưu trữ các thủ tục. Nếu ứng dụng của bạn gọi một thủ tục định sẵn, thì biểu thị được phân ngữ của thủ tục ấy được truy hồi từ cơ sở dữ liệu và được động cơ PL/SQL trong Oracle xác định.

Lưu ý:
Dù nhiều sản phẩm Oracle có các thành phần PL/SQL, nhưng tài liệu này chỉ đề cập riêng đến các thủ tục và các bộ phần mềm có thể được lưu trữ trong cơ sở dữ liệu Oracle và được xử lý bằng cách sử dụng động cơ PL/SQL của Oracle server.

Thông tin bổ sung :
Các khả năng của PL/SQL của mỗi công cụ Oracle được mô tả trong phần hướng dẫn người dùng công cụ thích hợp.

Bạn có thể gọi các thủ tục định sẵn từ các ứng dụng được phát triển bằng cách sử dụng các công cụ sau:

Các trình biên dịch trước của Oracle precompilers (các tồn tại của người dùng)
Oracle Call Interfaces (OCIs)
SQL*Module
SQL*Plus
Server Manager
Oracle Enterprise Manager

Bạn còn có thể gọi thủ tục định sẵn từ một khối PL/SQL khác là khối nặc danh hoặc thủ tục định sẵn khác. Xem thêm "Các Thủ Tục Và Các Bộ Phần Mềm".

Thông tin bổ sung:
Để có thông tin về cách gọi các thủ tục định sẵn từ mỗi kiểu ứng dụng, xem tài liệu về công cụ ứng dụng riêng, chẳng hạn như Pro*C/C++ Precompiler Programmer's Guide hoặc Pro*COBOL Precompiler Programmer's Guide..

SQL động trong PL/SQL

Bạn có thể viết các thủ tục định sẵn và các khối PL/SQL nặc danh có SQL động bằng cách sử dụng bộ phần mềm DBMS_SQL. Các câu lệnh SQL động không được nhúng trong chương trình nguồn của mình; đúng hơn chúng được lưu trữ trong các chuỗi ký tự được nhập vào hoặc được thiết lập bởi chương trình vào thời gian thực hiện.

Điều này cho phép bạn tạo các thủ tục có mục đích tổng quát hơn. Thí dụ, sử dụng SQL động cho phép bạn tạo thủ tục điều hoạt trên bảng có tên không biết rõ cho tới thời gian thực hiện.

Hơn nữa, bạn có thể phân ngữ ngôn ngữ điều tác dữ liệu (DML) bất kỳ hoặc câu lệnh ngôn ngữ định nghĩa dữ liệu (DDL) sử dụng bộ phần mềm DBMS SQL. Điều này giúp giải quyết vấn đề không có thể phân ngữ các câu lệnh DDL bằng cách sử dụng trực tiếp PL/SQL. Thí dụ, bây giờ bạn có thể chọn để phát hành câu lệnh DROP TABLE từ trong thủ tục định sẵn bằng cách sử dụng thủ tục PARSE được bộ phần mềm DBMS_SQL cung cấp.

Thông tin bổ sung:
Để có thêm thông tin về SQL động, xem Oracle8 Application Developer's Guide.

Các thủ tục bên ngoài

Thủ tục PL/SQL tiến hành trên một Oracle server có thể gọi một thủ tục bên ngoài hoặc hàm được viết bằng ngôn ngữ lập trình C và lưu trữ trong thư viện dùng chung. Thủ tục con C tiến hành trong không gian địa chỉ tách riêng từ không gian địa chỉ của Oracle server.

Thông tin bổ sung:
Xem Hướng dẫn người dùng PL/SQL và Reference thông tin chi tiết về các thủ tục bên ngoài.

Chương 15

Quản lý giao dịch

Chương này xác định giao dịch và mô tả bạn có thể quản lý công việc của bạn sử dụng các giao dịch như thế nào. Nó gồm:

- **Giới thiệu về các giao dịch**
- **Oracle và quản lý giao dịch**
- **Quản lý giao dịch thận trọng**

Giới thiệu các giao dịch

Giao dịch là một đơn vị hoạt động logic chứa một hoặc nhiều câu lệnh SQL. Giao dịch là một đơn vị nguyên tử; các kết quả của tất cả các câu lệnh SQL trong giao dịch đều có thể được cam kết (được áp dụng vào cơ sở dữ liệu) hoặc hồi phục về trạng thái cũ (được giải tỏa khỏi cơ sở dữ liệu).

Giao dịch bắt đầu với câu lệnh SQL khả thi đầu tiên. Giao dịch kết thúc khi nó được cam kết hoặc hồi phục về trạng thái cũ rõ ràng (với câu lệnh COMMIT hoặc ROLLBACK statement) hoặc ngầm định (khi câu lệnh DDL được đưa ra).

Để minh họa khái niệm của một giao dịch, xét một cơ sở dữ liệu ngân hàng. Khi khách hàng của ngân hàng chuyển tiền từ một

tài khoản tiết kiệm sang tài khoản tiền gửi, giao dịch ấy có thể gồm các điều hoạt tách riêng: giảm tài khoản tiết kiệm, tăng tài khoản tiền gửi và các bút toán của nghiệp vụ ấy phát sinh trong sổ nhật ký giao dịch (sổ phụ tài khoản).

Oracle phải cho phép hai trường hợp. Nếu tất cả ba câu lệnh SQL có thể được thực hiện để duy trì các tài khoản có số dư chính xác, các kết quả của giao dịch có thể được áp dụng vào cơ sở dữ liệu. Tuy nhiên, nếu điều gì đó xảy ra (chẳng hạn như không đủ tiền, số hiệu tài khoản không hợp lệ hoặc bị sự cố phần cứng) ngăn cản một hoặc hai câu lệnh trong giao dịch không được hoàn tất, thì toàn bộ giao dịch phải được hồi phục về trạng thái cũ để số dư của tất cả tài khoản chính xác.

Hình 15-1 minh họa thí dụ giao dịch ngân hàng.

Sự thi hành câu lệnh và điều khiển giao dịch

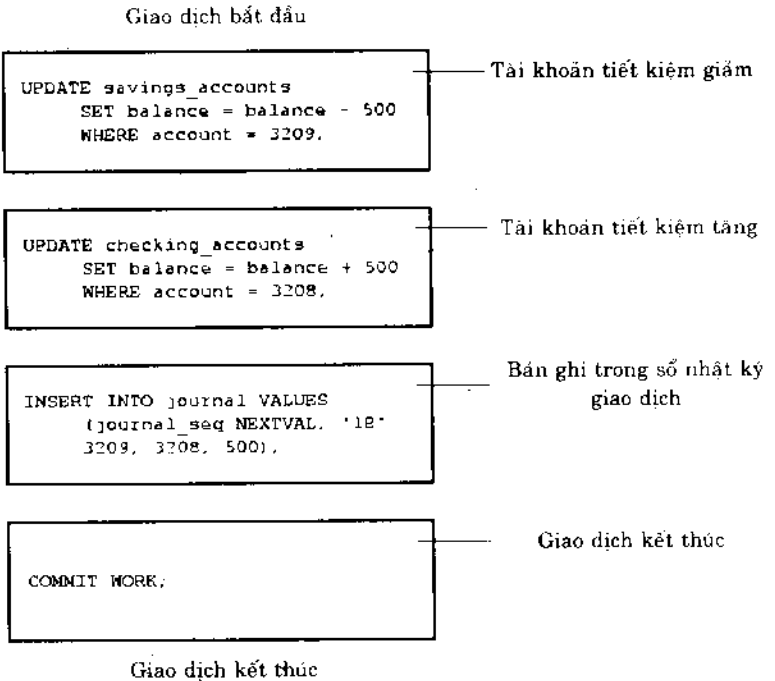
Một câu lệnh SQL "tiến hành thành công" khác với giao dịch "được cam kết".

Tiến hành thành công nghĩa là một câu lệnh đơn đã được phân ngữ và được thấy là một cấu trúc SQL hợp lệ, toàn bộ câu lệnh được tiến hành không bị lỗi là một đơn vị nguyên tử (thí dụ, tất cả các hàng của cập nhật đa hàng bị thay đổi). Tuy nhiên, cho tới khi giao dịch có câu lệnh được cam kết, thì câu lệnh ấy có thể được hồi phục về trạng thái cũ và tất cả những thay đổi của câu lệnh ấy có thể được giải tỏa. Một câu lệnh thực hiện thành công thay vì một giao dịch.

Cam kết có nghĩa là người dùng đã khai báo rõ ràng hoặc ngầm định "tạo các thay đổi trong giao dịch này vĩnh viễn). Những thay đổi xảy ra do (các) câu lệnh của giao dịch trở thành vĩnh viễn và những người dùng khác chỉ thấy được sau khi giao dịch

của bạn đã được cam kết. Chỉ những người dùng khác đã bắt đầu sau giao dịch của bạn sẽ thấy các thay đổi được cam kết.

Hình 15-1 Một giao dịch ngân hàng



Hồi phục mức câu lệnh

Nếu vào thời điểm nào đó khi sự tiến hành một câu lệnh SQL gây ra một lỗi, thì tất cả kết quả của câu lệnh được hồi phục về trạng thái cũ. Kết quả của hồi phục ấy như là câu lệnh đó chưa bao giờ được tiến hành. Đây là hồi phục mức câu lệnh.

Các lỗi được phát hiện khi sự tiến hành câu lệnh SQL làm cho xảy ra các hồi phục mức câu lệnh. (thí dụ, một lỗi như thế cố chèn một giá trị sao chép vào khóa chính). Các lỗi được phát

hiện khi phân ngữ câu lệnh SQL (chẳng hạn như một lỗi cú pháp) chưa được tiến hành, vì vậy không gây ra hồi phục mức câu lệnh. Các câu lệnh SQL đơn liên quan đến một bế tắc (cạnh tranh cùng dữ liệu) còn có thể gây ra hồi phục mức câu lệnh. Xem "Các Bế Tắc".

Một câu lệnh SQL không thành công chỉ làm mất bất kỳ công việc nào chính nó đã thực hiện; nó không làm mất công việc nào đã tiến hành trước giao dịch hiện hành. Nếu câu lệnh ấy là câu lệnh DDL, cam kết ngầm định đã tiến hành ngay trước đó không được giải tỏa.

Lưu ý:
Những người dùng không tham chiếu trực tiếp các điểm lưu ngầm định trong các câu lệnh hồi phục.

Oracle và quản lý giao dịch

Giao dịch trong Oracle bắt đầu khi câu lệnh SQL khả thi gặp được. Một câu lệnh SQL khả thi là câu lệnh SQL phát sinh các lệnh gọi một môi trường làm việc, kể cả các câu lệnh DML và DDL.

Khi một giao dịch bắt đầu, Oracle gán giao dịch ấy vào một đoạn hồi phục có thể xảy ra để ghi các mục nhập hồi phục cho giao dịch mới. Xem " Các Đoạn Hồi Phục Và Các Giao Dịch" về chủ đề này.

Một giao dịch kết thúc khi bất kỳ những điều sau xảy ra:

- Bạn đưa ra một câu lệnh COMMIT hoặc ROLLBACK (không có mệnh đề SAVEPOINT).
- Bạn tiến hành một câu lệnh DDL (chẳng hạn như CREATE, DROP, RENAME, ALTER). Nếu giao dịch hiện hành có bất kỳ câu lệnh DML nào, trước hết Oracle cam kết giao dịch ấy, rồi sau đó tiến hành và cam kết câu lệnh DDL là một giao dịch đơn mới.

- Người dùng giải kết nối từ Oracle. (giao dịch hiện hành được cam kết)
- Tiến trình người dùng kết thúc bất thường. (giao dịch hiện hành được phục hồi về trạng thái cũ).

Sau khi một giao dịch kết thúc, câu lệnh SQL khả thi kế tiếp tự động bắt đầu giao dịch theo sau.

Lưu ý:
Các ứng dụng sẽ luôn luôn cam kết rõ ràng hoặc hồi phục các giao dịch trước khi chương trình kết thúc.

Cam kết các giao dịch

Cam kết một giao dịch nghĩa là tạo ra các thay đổi vĩnh viễn được thực hiện bằng các câu lệnh SQL trong giao dịch ấy.

Trước khi một giao dịch sửa đổi dữ liệu được cam kết, những điều sau đã xảy ra:
Oracle đã phát sinh các bản ghi đoạn hồi phục trong các vùng đệm hồi phục của vùng toàn cục hệ thống (SGA). Thông tin hồi phục có các giá trị dữ liệu cũ bị thay đổi do các câu lệnh SQL của giao dịch ấy.
Oracle đã phát sinh các mục ghi redo log trong vùng đệm redo log của SGA. Những thay đổi này có thể tới đĩa trước khi giao dịch được cam kết.
Các thay đổi đã xảy ra với các vùng đệm cơ sở dữ liệu của SGA. Những thay đổi này có thể đi tới đĩa trước khi một giao dịch thực sự được cam kết.

Lưu ý:
Dữ liệu thay đổi đối với một giao dịch được cam kết, được lưu

trữ trong các vùng đệm cơ sở dữ liệu của SGA, không nhất thiết được ghi trực tiếp vào các tập tin dữ liệu bằng xử lý nền của database writer (DBWn). Hoạt động ghi này xảy ra khi nó hầu như có đủ hiệu lực để thực hiện điều đó. Nó có thể xảy ra trước khi giao dịch cam kết hoặc thay vào đó, nó có thể xảy ra vào lúc nào đó sau khi giao dịch ấy được cam kết.

Khi một giao dịch được cam kết, những điều sau xảy ra:

- Bảng giao dịch bên trong cho các bản ghi đoạn hồi phục được kết hợp mà giao dịch đã cam kết và số hiệu thay đổi hệ thống duy nhất (SCN) của giao dịch ấy được gán và được ghi vào bảng.
- Tiến trình log writer (LGWR) ghi các mục nhập redo log trong các vùng đệm redo log của SGA vào tập tin redo log trực tuyến; nó còn ghi SCN của giao dịch ấy vào tập tin redo log. Sự kiện nguyên tử này cấu tạo cam kết của giao dịch ấy.
- Oracle giải phóng các khóa được lưu trữ trên các hàng và các bảng.
- Oracle đánh dấu giao dịch "hoàn tất".

Xem "Các Xử Lý Của Oracle" về các xử lý nền LGWR và DBWR.

Các giao dịch hồi phục

Rolling back (hồi phục về trạng thái cũ) nghĩa là hoán chuyển bất kỳ thay đổi nào thành dữ liệu địa chỉ được các câu lệnh SQL thực hiện trong một giao dịch không được cam kết.

Oracle cho phép bạn hồi phục toàn bộ giao dịch không được cam kết. Hoặc bạn có thể hồi phục phần cuối của một giao dịch không được cam kết thành một dấu được gọi là điểm lưu; xem "Các Điểm Lưu" để có giải thích đầy đủ.
Tất cả các loại hồi phục sử dụng cùng các thủ tục:

- Hồi phục mức câu lệnh (do câu lệnh hoặc lỗi tiến hành bế tắc)
- Hồi phục thành một điểm lưu
- Hồi phục giao dịch do yêu cầu của người dùng
- Hồi phục giao dịch do kết thúc tiến trình bất thường
- Hồi phục tất cả các giao dịch chưa giải quyết khi môi trường làm việc kết thúc bất thường
- Hồi phục các giao dịch không hoàn toàn khi phục hồi

Hồi phục toàn bộ một giao dịch mà có tham chiếu bất kỳ điểm lưu, những điều sau xảy ra:

Oracle giải phóng tất cả các thay đổi do tất cả các lệnh SQL trong giao dịch ấy bằng cách sử dụng các đoạn hồi phục tương ứng.

Oracle giải phóng tất cả các khóa dữ liệu của giao dịch ấy (xem "Cơ Chế Khóa" để cập đến các khóa).

Giao dịch kết thúc.

Trong hồi phục một giao dịch tới một điểm lưu, những điều sau xảy ra:

Oracle chỉ hồi phục các câu lệnh được tiến hành sau điểm lưu.

Điểm lưu ấn định được duy trì, nhưng tất cả các điểm lưu đã được thiết lập sau khi được ấn định sẽ bị mất.

Oracle giải phóng tất cả các khóa hàng và bảng cần thiết cho từ điểm lưu đó, nhưng vẫn duy trì tất cả các khóa dữ liệu cần thiết trước điểm lưu ấy.

Giao dịch vẫn duy trì hoạt động và có thể được tiếp tục.

Các điểm lưu

Bạn có thể khai báo các dấu trung gian được gọi là các điểm lưu trong ngữ cảnh của giao dịch. Các điểm lưu chia giao dịch dài thành các phần nhỏ hơn.

Sử dụng các điểm lưu, bạn có thể đánh dấu tùy ý công việc của mình vào bất kỳ thời điểm nào trong một giao dịch dài. Sau đó

bạn có được tùy chọn sau này về hồi phục công việc được thực hiện trước điểm hiện hành trong giao dịch ấy (kết thúc của giao dịch) nhưng sau khi điểm lưu được khai báo trong giao dịch ấy. Thí dụ, bạn có thể sử dụng các điểm lưu trong hàng loạt các cập nhật phức tạp dài để nếu bạn gây ra một lỗi, bạn không cần phải cam kết lại mọi câu lệnh.

Tương tự các điểm lưu hữu dụng trong các trình ứng dụng theo phương thức tương tự. Nếu thủ tục có nhiều hàm, bạn có thể tạo một điểm lưu trước mỗi hàm bắt đầu. Sau đó, nếu một hàm không thực hiện thành công để dễ trở về dữ liệu trạng thái của nó trước khi hàm ấy đã bắt đầu và sau đó tiến hành lại hàm ấy với các tham số được sửa đổi hoặc thực hiện một hoạt động phục hồi.

Sau một hồi phục về một điểm lưu, Oracle giải phóng các khóa dữ liệu mà các câu lệnh hồi phục đạt được. Các giao dịch khác đã chờ những tài nguyên được khóa trước đó có thể tiến hành. Các giao dịch khác muốn cập nhật các hàng được khóa trước có thể thực hiện như vậy.

Cơ chế cam kết hai pha

Trong cơ sở dữ liệu được phân phối, Oracle phải phối hợp điều khiển giao dịch trên mạng và duy trì dữ liệu nhất quán, dù mạng hoặc hệ thống xảy ra sự cố.

Cơ chế cam kết hai pha bảo đảm tất cả các server cơ sở dữ liệu tham gia vào giao dịch được phân phối, cam kết tất cả hoặc hồi phục tất cả các câu lệnh trong giao dịch ấy. Cơ chế cam kết hai pha còn bảo vệ các điều hoạt DML ngầm định bằng các hạn định tính nguyên vẹn, các lệnh gọi thủ tục từ xa và các bộ kích khởi.

Cơ chế cam kết hai pha của Oracle truyền thông hoàn toàn tới những người dùng đưa ra các giao dịch được phân phối. Thực ra,

những người dùng thậm chí không cần biết giao dịch được phân phối. Câu lệnh COMMIT biểu thị kết thúc của giao dịch kích khởi tự động cơ chế cam kết hai pha để cam kết giao dịch ấy; không cần có mã hóa hoặc cú pháp câu lệnh phức tạp để có các giao dịch được phân phối trong thân của ứng dụng cơ sở dữ liệu.

Xử lý nền bộ phục hồi (RECO) tự động giải quyết kết quả của các giao dịch được phục hồi không chắc chắn - các giao dịch được phân phối cam kết đã bị ngăn chặn bởi bất kỳ sự cố mạng hoặc hệ thống. Sau khi sự cố được sửa chữa và truyền thông được tái thiết lập, RECO của mỗi Oracle server cục bộ tự động cam kết hoặc hồi phục bất kỳ các giao dịch không chắc chắn nhất quán trên tất cả các nút có liên quan.

Trong trường hợp sự cố kéo dài, Oracle cho phép mỗi người điều hành cam kết hoặc hồi phục thủ công bất kỳ giao dịch được phân phối không chắc chắn do bị sự cố. Tùy chọn này cho phép người điều hành cơ sở dữ liệu cục bộ giải phóng các tài nguyên bị khóa có thể được lưu giữ không hạn định vì sự cố kéo dài.

Nếu cơ sở dữ liệu phải được phục hồi tới một thời điểm trong quá khứ, các phương tiện phục hồi của Oracle cũng cho phép người điều hành cơ sở dữ liệu ở các vùng khác nhau đưa các cơ sở dữ liệu của họ về ngày tháng của thời điểm ban đầu. Điều này bảo đảm là cơ sở dữ liệu toàn cục vẫn duy trì nhất quán.

Quản lý giao dịch thận trọng

Các Developer ứng dụng có thể tăng hiệu suất của các giao dịch ngắn, không được phân phối bằng cách sử dụng thủ tục BEGIN_DISCRETE_TRANSACTION. Thủ tục này sắp xếp xử lý giao dịch hợp lý hơn vì vậy các giao dịch ngắn có thể tiến hành nhanh hơn.

Khi một giao dịch thận trọng, tham chiếu các thay đổi xảy ra với bất kỳ dữ liệu được trì hoãn cho tới khi giao dịch ấy cam kết. Dĩ

nhiên, các giao dịch đồng thời khác không thể thấy các thay đổi không được cam kết của một giao dịch có thận trọng hay không.

Oracle phát sinh thông tin thực hiện lại, nhưng lưu trữ nó trong vị trí tách riêng ở bộ nhớ. Khi giao dịch ấy đưa ra yêu cầu cam kết, hoặc ghi thông tin thực hiện lại vào tập tin redo log (cùng với các cam kết nhóm khác) và áp dụng các thay đổi vào khối cơ sở dữ liệu trực tiếp đối với khối ấy. Hoặc đưa điều khiển trở lại ứng dụng khi cam kết ấy hoàn tất. Điều này kết thúc nhu cầu phát sinh thông tin giải phóng (xả lệnh), vì khối ấy thực sự không được sửa đổi cho tới khi giao dịch được cam kết và thông tin thực hiện lại được lưu trữ ở các vùng đệm redo log.

Thông tin bổ sung:
Để có thêm thông tin về các giao dịch thận trọng, xem Oracle8 Tuning.

Chương 16

Sắp hàng trước

Chương này mô tả đặc trưng Oracle Advanced Queuing (Oracle AQ); gồm:

- **Giới thiệu sắp hàng thông điệp**
- **Sắp hàng trước của Oracle**
- **Sắp hàng các thực thể**
- **Các đặc trưng của sắp hàng trước**

Lưu ý:

Nếu bạn đã có sản phẩm Oracle8, bạn không thể sử dụng Oracle AQ.

Nếu bạn đã có sản phẩm Oracle8 Enterprise Edition mà không có Objects Option, bạn có thể sử dụng Oracle AQ với các hàng sắp chỉ có kiểu dữ liệu RAW.

Nếu bạn đã có sản phẩm Oracle8 Enterprise Edition có Objects Option, bạn có thể sử dụng đầy đủ tính năng của Oracle AQ.

Xem Getting to Know Oracle8 và Oracle Enterprise Edition về các đặc trưng và các tùy chọn có thể sử dụng với Oracle8

Enterprise Edition.

Thông tin bổ sung:
Để có thêm thông tin về Oracle AQ, xem Oracle8 Application Developer's Guide.

Giới thiệu sắp hàng thông điệp

Truyền thông giữa các chương trình được phân loại thành một trong hai kiểu:

- Truyền thông đồng bộ (trực tuyến hoặc mô hình được kết nối)
- Truyền thông không đồng bộ (giải kết nối hoặc mô hình bị trì hoãn)

Truyền thông đồng bộ

Truyền thông đồng bộ dựa trên mô hình yêu cầu/trả lời một chương trình gửi yêu cầu tới một chương trình khác và chờ cho tới khi nhận được trả lời.

Mô hình truyền thông này (còn được gọi là trực tuyến hoặc được kết nối) thích hợp cho các chương trình cần nhận trả lời trước khi chúng có thể tiến hành hoạt động của chúng. Các kiến trúc client/server truyền thống dựa trên mô hình này.

Trở ngại chính của mô hình đồng bộ của truyền thông là các chương trình phải được sử dụng và tiến hành ứng dụng ấy để hoạt động. Trong trường hợp mạng hoặc máy bị sự cố, các chương trình ngưng thực hiện chức năng.

Truyền thông không đồng bộ

Trong các chương trình có mô hình được *disconnected* (giải kết nối) hoặc *deferred* (trì hoãn) truyền thông không đồng bộ, đưa yêu cầu vào hàng sắp và sau đó tiến hành công việc của chúng.

Thí dụ, một ứng dụng có thể cần mục nhập dữ liệu hoặc sự tiến hành của một hoạt động vào thời gian sau đó, sau khi các điều kiện được đáp ứng. Chương trình người nhận truy hồi yêu cầu từ hàng sắp ấy và tác động vào nó. Mô hình này thích hợp với các ứng dụng có thể tiếp tục với công việc của chúng sau khi đưa yêu cầu vào hàng đợi - chúng không bị trở ngại phải chờ trả lời.

Đối với sự tiến hành bị trì hoãn để hoạt động chính xác ngay cả trong trường hợp sự cố mạng, máy và ứng dụng, các yêu cầu phải được lưu trữ liên tục và mỗi lần được xử lý chính xác. Điều này có thể đạt được bằng cách kết hợp sắp hàng liên tục với sự bảo vệ giao dịch.

Xử lý mỗi yêu cầu client/server chính xác một lần thường quan trọng để duy trì tính nguyên vẹn hoặc luồng truyền của một giao dịch. Thí dụ, nếu yêu cầu là một mệnh lệnh về số cổ phiếu với giá đặc biệt, thì sự tiến hành của yêu cầu zêrô hoặc hai lần không thể chấp nhận được dù sự cố mạng hoặc hệ thống xảy ra khi truyền, nhận hoặc tiến hành yêu cầu.

Oracle Advanced Queuing

Oracle Advanced Queuing (Oracle AQ) tích hợp hệ thống sắp hàng thông điệp với cơ sở dữ liệu Oracle. Điều này cho phép bạn lưu trữ các thông điệp vào các hàng sắp đối với truy hồi bị trì hoãn và xử lý bởi Oracle server.

Ứng dụng có thể truy xuất tính năng sắp hàng qua giao diện PL/SQL. Điều này cung cấp hệ thống sắp hàng hiệu quả và đáng tin cậy mà không có phần mềm bổ sung giống như các trình

giám sát xử lý giao dịch (TP) hoặc Message Oriented Middleware.

Oracle AQ cung ứng tính năng sau:

- Ưu tiên và xếp thứ tự các phần tử hàng sắp
- Khả năng ấn định cửa sổ tiến hành mỗi phần tử hàng sắp
- Khả năng truy vấn các hàng sắp sử dụng SQL tiêu chuẩn
- Các giao dịch được tích hợp để đơn giản hóa quản lý và phát triển ứng dụng
- Khả năng giải hàng sắp nhiều phần tử hàng sắp bằng một bó
- Khả năng ấn định nhiều người nhận
- Khả năng truyền những thông điệp tới các hàng sắp trong các cơ sở dữ liệu cục bộ hoặc từ xa
- Sắp hàng liên tục
- Thống kê các thông điệp trong các hàng sắp

Các hàng sắp Oracle AQ được bổ sung vào các bảng cơ sở dữ liệu, để tất cả lợi điểm điều hoạt khả dụng nhất, khả năng vô hướng và khả năng tin cậy được áp dụng vào dữ liệu hàng sắp. Hơn nữa, các công cụ và phát triển cơ sở dữ liệu có thể được sử dụng với các hàng sắp.

Các thực thể sắp hàng

Oracle AQ có 5 thực thể căn bản: các thông điệp, các hàng sắp, các bảng hàng sắp, các tác nhân và bộ giám sát hàng sắp.

Các thông điệp

Thông điệp là đơn vị nhỏ nhất được chèn vào hàng sắp và được truy hồi từ hàng sắp. Một thông điệp gồm thông tin điều khiển và dữ liệu trọng tải hữu hình. Thông tin điều khiển biểu thị các đặc tính thông điệp được Oracle AQ sử dụng để quản lý các

thông điệp. Dữ liệu trọng tải hữu hình là thông tin được lưu trữ trong hàng sắp và được truyền suốt tới Oracle AQ. Kiểu dữ liệu của trọng tải hữu hình có thể là RAW hoặc kiểu đối tượng.

Thông điệp có thể cư trú chỉ trong một hàng sắp. Thông điệp được tạo ra do lệnh gọi sắp hàng và được tiêu thụ bởi lệnh gọi giải hàng sắp. Các lệnh gọi sắp hàng và giải hàng sắp là thành phần của bộ phần mềm DBMS_AQ.

Các hàng sắp

Hàng sắp là kho chứa những thông điệp. Có hai kiểu hàng sắp: các hàng sắp người dùng (tiêu chuẩn) và các hàng sắp ngoại lệ. Hàng sắp người dùng dành để xử lý thông điệp tiêu chuẩn. Tất cả thông điệp trong hàng sắp phải có cùng kiểu dữ liệu. Các thông điệp được cam kết tới một hàng sắp ngoại lệ nếu chúng không thể được truy hồi và xử lý vì một lý do nào đó.

Các hàng sắp có thể được tạo, thay đổi, khởi động, ngưng và giảm bằng cách sử dụng bộ phần mềm DBMS_AQADM.

Các bảng hàng sắp

Các hàng sắp được lưu trữ trong các bảng hàng sắp. Mỗi bảng hàng sắp là một bảng cơ sở dữ liệu và chứa một hoặc nhiều hàng sắp. Mỗi hàng sắp chứa một hàng sắp ngoại lệ mặc định.

Tạo một bảng hàng sắp để tạo một bảng cơ sở dữ liệu có khoảng 25 cột. Các cột này lưu trữ meta dữ liệu của Oracle AQ và trọng tải hữu hình người dùng xác định.

Khung nhìn và hai chỉ mục được tạo trên bảng hàng sắp. Khung nhìn cho phép bạn truy vấn dữ liệu thông điệp. Các chỉ mục được sử dụng để tăng tốc độ truy xuất dữ liệu thông điệp.

Các tác nhân

Một tác nhân là một người dùng hàng sắp. Có hai kiểu tác nhân: các nhà sản xuất đưa những thông điệp vào hàng sắp (sắp hàng) và các khách hàng truy hồi những thông điệp (giải hàng sắp). Bất kỳ số nhà sản xuất và khách hàng có thể truy xuất hàng sắp vào thời gian ấn định trước.

Một tác nhân được nhận dạng bằng tên, địa chỉ và giao thức của họ. Đối với một tác nhân trên cơ sở dữ liệu từ xa, chỉ giao thức hiện hỗ trợ là liên kết cơ sở dữ liệu Oracle sử dụng địa chỉ của dạng queue_name@dblink.

Bộ giám sát hàng sắp

Bộ giám sát hàng sắp là một xử lý nền tùy chọn để giám sát các thông điệp trong hàng sắp ấy. Nó cung cấp cơ chế về thời hạn của thông điệp, thử lại và trì hoãn (xem "Windows of Execution") và cho phép bạn tập hợp thống kê thời khoảng, (xem "Queuing Statistics").

Bộ giám sát hàng sắp khác với hầu hết các xử lý nền Oracle khác trong đó sai sót của xử lý không gây ra cho môi trường làm việc không thực hiện được.

Tham số khởi hoạt AQ_TM_PROCESS ấn định việc tạo một hoặc nhiều xử lý của bộ giám sát hàng sắp vào lúc môi trường làm việc khởi động.

Các đặc trưng sắp hàng trước

Phần này mô tả các đặc trưng chính của Oracle Advanced Queuing.

Trọng tải hữu hình có cấu trúc

Bạn có thể sử dụng các kiểu đối tượng để cấu trúc và quản lý trọng tải hữu hình. (Kiểu dữ liệu RAW có thể được sử dụng cho các trọng tải hữu hình có cấu trúc.)

Hỗ trợ điều hoạt cơ sở dữ liệu tích hợp

Oracle AQ lưu trữ các thông điệp trong các bảng. Tất cả các đặc trưng cơ sở dữ liệu, chẳng hạn như phục hồi, khởi động lại và Oracle Enterprise Manager được hỗ trợ.

Truy xuất SQL

Các thông điệp được lưu trữ bằng các bản ghi cơ sở dữ liệu. Bạn có thể sử dụng SQL để truy xuất các đặc tính thông điệp, lịch sử thông điệp và payload. Tất cả công nghệ SQL có thể sử dụng, chẳng hạn như các chỉ mục có thể được sử dụng để tối ưu hóa truy xuất các thông điệp.
Vai trò AQ_ADMINISTRATOR cung cấp truy xuất thông tin về các hàng sắp.

Các cửa sổ tiến hành

Bạn có thể ấn định sự tiêu thụ thông điệp phải xảy ra trong cửa sổ có thời gian ấn định. Thông điệp có thể được đánh dấu để chỉ có thể sử dụng xử lý sau thời gian ấn định trôi qua (thời gian trì hoãn) và phải được tiêu thụ trước khi hết thời hạn ấn định.

Tham số khởi hoạt của AQ_TM_PROCESS cho phép giám sát thời gian trên các thông điệp hàng sắp được sử dụng cho các thông điệp để ấn định các đặc tính thời hạn và độ trễ. Giám sát thời gian còn phải được cho phép nếu bạn muốn tập hợp các thống kê thời khoảng (xem "Sắp Hàng Thống Kê").

Nếu tham số này được xác lập là 1, Oracle tạo một tiến trình giám sát hàng sắp (QMN0) bằng một xử lý nền để giám sát các thông điệp. Nếu nó được xác lập là 2 tới 10, Oracle tạo số xử lý QMNn; nếu tham số không được ấn định hoặc được xác lập là 0, thì các xử lý của bộ giám sát hàng sắp không được tạo. Các thủ tục trong bộ phần mềm DBMS_AQADM để bắt đầu và ngưng các điều hoạt giám sát hàng sắp chỉ hợp lệ nếu ít nhất một tiến trình giám sát hàng sắp đã được bắt đầu với tham số này làm thành phần của môi trường làm việc khởi động.

Nhiều khách hàng trên mỗi thông điệp

Một thông điệp đơn có thể được tiêu thụ bằng nhiều khách hàng.

Định vị

Bạn có nhiều tùy chọn để lựa thông điệp từ một hàng sắp. Bạn có thể lựa thông điệp đầu tiên hoặc một khi bạn đã lựa một thông điệp và thiết lập một ảnh chụp nhanh đọc nhất quán, bạn có thể truy hồi thông điệp kế tiếp dựa trên ảnh chụp nhanh hiện hành. Bạn sẽ đạt được ảnh chụp nhanh đọc nhất quán mỗi lần bạn lựa thông điệp đầu tiên từ hàng sắp.

Bạn còn có thể truy hồi một thông điệp ấn định sử dụng dấu định danh tương quan của thông điệp.

Sắp xếp thứ tự các thông điệp

Bạn có ba tùy chọn để ấn định thứ tự trong đó các thông điệp được tiêu thụ: thứ tự sắp xếp ấn định các đặc tính được sử dụng để sắp xếp thứ tự tất cả thông điệp trong một hàng sắp, quyền ưu tiên có thể được gán cho mỗi thông điệp và độ lệch chuỗi cho phép bạn định vị thông điệp liên quan đến các thông điệp khác.

Nếu nhiều khách hàng tác động cùng hàng sắp, thì khách hàng nhận được thông điệp thứ nhất có thể sử dụng để tiêu thụ ngay.

Thông điệp trong tiến trình đang được tiêu thụ bởi một khách hàng khác, thông điệp đó sẽ bị bỏ qua.

Các chế độ giải hàng sắp

Yêu cầu DEQUEUE có thể duyệt hoặc xóa một thông điệp. Nếu thông điệp được duyệt nó vẫn có thể sử dụng để xử lý thêm. Nếu thông điệp bị xóa, nó không có thể sử dụng bất kỳ yêu cầu DEQUEUE nào nữa. Tùy theo các đặc tính hàng sắp có thể được duy trì trong bảng hàng sắp.

Chờ các thông điệp đến

DEQUEUE có thể được phát hành đối với một hàng sắp rỗng. Bạn có thể ấn định nếu và trong bao lâu yêu cầu ấy được phép chờ thông điệp đến.

Các lần thử lại có độ trễ

Thông điệp phải được tiêu thụ chính xác một lần. Nếu một cố gắng giải hàng sắp một thông điệp không thành công và giao dịch ấy được hồi phục, thông điệp ấy sẽ được thực hiện để có thể sử dụng xử lý lại sau thời gian trôi qua được người dùng ấn định độ trễ. Xử lý lại sẽ cố thực hiện tới mức giới hạn được ấn định.

Các hàng sắp ngoại lệ

Thông điệp không thể được tiêu thụ trong các hạn định cho trước, nghĩa là trong cửa sổ tiến hành hoặc trong các giới hạn của các lần thử lại. Nếu tình trạng như thế phát sinh, thông điệp sẽ được di chuyển tới hàng sắp ngoại lệ được người dùng ấn định.

Tính năng hiển thị

Các yêu cầu ENQUEUE/DEQUEUE thường là tập hợp của một giao dịch chứa các yêu cầu. Điều này cung cấp xử lý giao dịch

được yêu cầu. Tuy nhiên, bạn có thể ấn định yêu cầu bằng chính giao dịch ấy để kết quả của yêu cầu có thể hiển thị ngay so với các giao dịch khác.

Tạo nhóm tài nguyên

Các thông điệp thuộc một hàng sắp có thể được tạo nhóm để thành lập một tập hợp chỉ có thể được một người dùng tiêu thụ một lần. Điều này đòi hỏi hàng sắp được tạo trong một bảng hàng sắp được cho phép tạo nhóm thông điệp.

Tất cả thông điệp thuộc nhóm phải được tạo trong cùng giao dịch. Đặc trưng này cho phép bạn phân đoạn các thông điệp phức tạp thành các thông điệp đơn giản, thí dụ, các thông điệp được định hướng tới một hàng sắp chứa các hóa đơn có thể được cấu tạo bằng một nhóm bắt đầu với phần đầu thông điệp được theo sau bằng các thông điệp biểu thị chi tiết được theo sau bởi phần cuối thông điệp.

Duy trì

Bạn có thể ấn định các thông điệp được giữ lại sau khi tiêu thụ. Điều này cho phép bạn duy trì lịch sử của các thông điệp thích hợp. Lịch sử có thể được sử dụng để truy tìm kho dữ liệu và hoạt động khai thác dữ liệu.

Lịch sử thông điệp

Oracle AQ lưu trữ thông tin về lịch sử của mỗi thông điệp. Thông tin chứa thông tin ENQUEUE/DEQUEUE và định danh của giao dịch đã tiến hành mỗi yêu cầu.

Tìm vệt

Nếu các thông điệp được giữ lại chúng có thể liên quan với nhau; thí dụ, nếu thông điệp m2 được tạo do sự tiêu thụ thông điệp m1.

mq có liên quan đến m2. Điều này cho phép bạn tìm vệt các chuỗi của những thông điệp có liên quan. Những thông điệp này biểu thị "các nhật ký sự kiện" thường được các ứng dụng tạo thành. Oracle AQ được thiết kế để các ứng dụng tạo các nhật ký sự kiện tự động.

Truyền các thông điệp với các cơ sở dữ liệu khác

Các thông điệp được sắp hàng trong một cơ sở dữ liệu có thể được truyền tới các hàng sắp trên cơ sở dữ liệu khác. Các kiểu dữ liệu của các hàng sắp nguồn và đích phải so khớp với nhau.

Truyền thông điệp cho phép truyền thông với nhau mà không phải kết nối với cùng cơ sở dữ liệu hoặc cùng hàng sắp. Truyền sử dụng các liên kết cơ sở dữ liệu và Net8 giữa các cơ sở dữ liệu cục bộ hoặc từ xa, cả hai phải được Oracle AQ cho phép.

Bạn có thể lập lịch trình (hoặc không lập lịch trình) truyền thông điệp và ấn định thời gian bắt đầu, cửa sổ truyền và hàm ngày tháng cho các cửa sổ truyền sau này trong các lịch trình theo chu kỳ. Khung nhìn từ điển dữ liệu DBA_QUEUE_SCHEDULES mô tả các lịch trình hiện hành để truyền các thông điệp.

Các xử lý nền hàng sắp tác vụ (SNPn) điều quản truyền thông điệp. Để cho phép truyền, ít nhất bạn phải bắt đầu một tiến trình hàng sắp tác vụ với tham số khởi hoạt JOB_QUEUE PROCESSES.

Sắp hàng thống kê

Oracle AQ lưu trữ thống kê về trạng thái hiện hành của hệ thống sắp hàng cũng như thống kê thời khoảng trong bảng động GV$AQ.

Thống kê về trạng thái hiện hành của hệ thông sắp hàng gồm các số thông điệp sẵn sàng, đang chờ và hết hạn.

Một hoặc nhiều xử lý của bộ giám sát hàng sắp phải được khởi động (xem "Windows of Execution") để duy trì các thống kê thời khoảng gồm:

- Số thông điệp trong mỗi trạng thái (sẵn sàng, chờ và hết thời hạn)
- Thời gian chờ trung bình của các thông điệp chờ
- Tổng thời gian chờ của các thông điệp chờ

Import/Export

Import /Export của các hàng sắp cấu tạo Import/Export của các bảng hàng sắp cơ sở và các bảng từ điển liên quan. Import và Export các hàng sắp chỉ có thể được thực hiện mức độ chi tiết của bảng hàng sắp.
Khi một bảng hàng sắp được Export, các thông tin định nghĩa bảng và dữ liệu hàng sắp được Export. Khi một bảng hàng sắp được Import, các thủ tục hoạt động Export duy trì từ điển hàng sắp. Vì dư liệu bảng hàng sắp cũng được Export, người dùng chịu trách nhiệm duy trì tính nguyên vẹn dữ liệu mức ứng dụng khi dữ liệu bảng hàng sắp được truyền tải.

Đối với mọi bảng hàng sắp hỗ trợ nhiều người nhận, có một bảng được tổ chức chỉ mục chứa meta dữ liệu hàng sắp quan trọng. Meta dữ liệu này chủ yếu đối với các hoạt động của hàng sắp, vì vậy bạn phải Export và Import bảng được tổ chức chỉ mục cũng như bảng hàng sắp cho các hàng sắp trình ứng dụng bảng này để hoạt động sau khi Import.

Khi giản đồ chứa bảng hàng sắp được Export, bảng được tổ chức chỉ mục còn tự động được Export. Xử lý tương tự vào thời gian Import. Vì bảng meta dữ liệu chứa các ROWID của một số hàng trong bảng hàng sắp, lệnh Import đưa ra thông báo về các

ROWID quá hạn khi Import bảng meta dữ liệu. Thông báo này có thể được bỏ qua, vì hệ thống hàng sắp tự động điều chỉnh các ROWID hết hạn làm thành phần của tiến trình Import. Tuy nhiên, nếu gặp phải một vấn đề khác khi đang Import (chẳng hạn như cạn không gian đoạn hồi phục về trạng thái cũ), vấn đề ấy sẽ được điều chỉnh và Import sẽ được tiến hành lại.

Dấu định danh tương quan

Bạn có thể gán một dấu định danh vào mỗi thông điệp. Dấu định danh này có thể được sử dụng để truy hồi các thông điệp xác định.

Oracle Enterprise Manager

Oracle Enterprise Manager cung cấp giao diện đồ họa (GUI) cho một số chức năng quản trị hàng sắp, kể cả khởi động và dừng hàng sắp, truyền lập lịch trình và không lập lịch trình và hiển thị các đặc tính hàng sắp là thành phần của trình quản lý giản đồ.

Thông tin bổ sung:
Để có thông tin chi tiết về Oracle AQ, xem Oracle8 Application Developr's Guide.

Chương 17

Các thủ tục và các bộ phần mềm

Chương này đề cập đến các khả năng thủ tục của Oracle. Nó gồm:

- **Giới thiệu các thủ tục định sẵn và các bộ phần mềm**
- **Các thủ tục và các hàm**
- **Các bộ phần mềm**
- **Oracle lưu trữ thủ tục và các bộ phần mềm như thế nào**
- **Oracle tiến hành các thủ tục và các bộ phần mềm như thế nào**

Để có thông tin về các phụ thuộc các thủ tục, các hàm, bộ phần mềm và Oracle quản lý các phụ thuộc này như thế nào, xem "Oracle Dependency Manager".

Thông tin bổ sung:
Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle documentation.

Giới thiệu các thủ tục định sẵn và các bộ phần mềm

Oracle cho phép bạn truy xuất và điều tác thông tin cơ sở dữ liệu sử dụng các đối tượng giản đồ thủ tục được gọi là các đơn vị chương trình PL/SQL. Các thủ tục, các hàm và các bộ phần mềm đều là thí dụ của những đơn vị chương trình PL/SQL.

PL/SQL là mở rộng ngôn ngữ thủ tục SQL của Oracle. Nó mở rộng SQL với điều khiển luồng truyền và các câu lệnh khác tạo cho nó có thể viết các chương trình phức tạp trong nó. Động cơ PL/SQL là công cụ bạn sử dụng để xác định, biên dịch và tiến hành các đơn vị chương trình PL/SQL. Động cơ này là thành phần đặc biệt của các sản phẩm Oracle, kể cả Oracle server.

Khi nhiều sản phẩm Oracle có các thành phần PL/SQL, chương này đặc biệt đề cập đến các thủ tục và các bộ phần mềm có thể được lưu trữ trong cơ sở dữ liệu Oracle và tiến hành bằng cách sử dụng động cơ PL/SQL. Các khả năng của PL/SQL của mỗi công cụ Oracle được mô tả trong tài liệu của công cụ thích hợp. Xem thêm "PL/SQL".

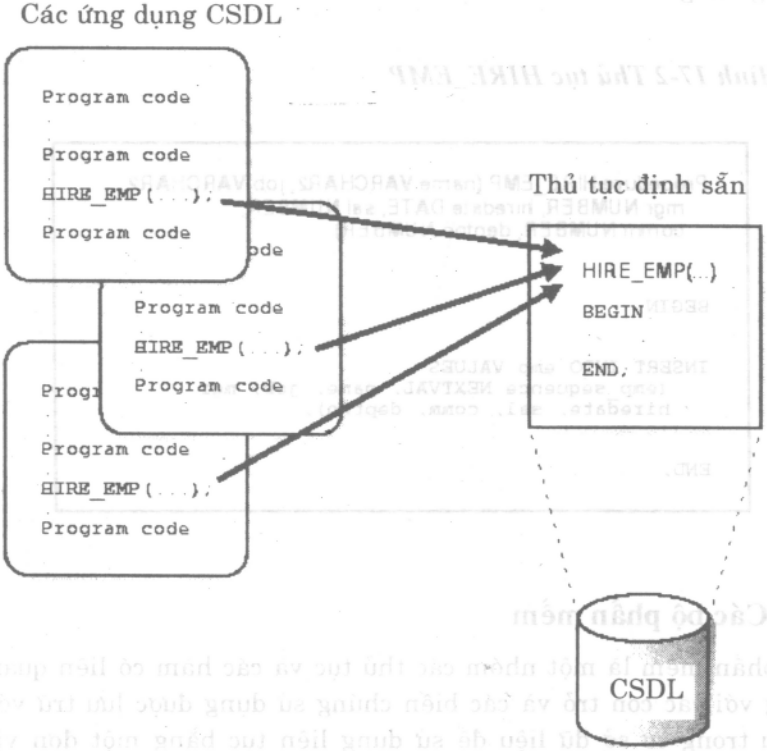
Các thủ tục định sẵn và các hàm

Các thủ tục và các hàm là các đối tượng giản đồ tạo nhóm logic một tập hợp SQL và các câu lệnh ngôn ngữ lập trình PL/SQL khác cùng thực hiện tác vụ ấn định. Các thủ tục và các hàm được tạo trong giản đồ của người dùng và được lưu trữ trong cơ sở dữ liệu để sử dụng tiếp tục. Bạn có thể tiến hành thủ tục hoặc hàm sử dụng công cụ Oracle tương tác, chẳng hạn như SQL*Plus hoặc gọi nó rõ ràng bằng mã của ứng dụng cơ sở dữ liệu, chẳng hạn như Oracle Forms hoặc ứng dụng Precompiler hay bằng mã của một thủ tục hoặc bộ kích khởi khác.

Hình 17-1 minh họa một thủ tục đơn giản được lưu trữ trong cơ sở dữ liệu và được gọi bởi nhiều ứng dụng cơ sở dữ liệu khác nhau.

Các thủ tục và các hàm đồng nhất, ngoại trừ các hàm luôn luôn cho một giá trị đơn trở lại bộ gọi, trong khi các thủ tục không thực hiện điều đó. Để đơn giản hóa, thuật ngữ "thủ tục" được sử dụng trong phần còn lại của chương có nghĩa là "thủ tục hoặc hàm".

Hình 17-1 Một thủ tục định sẵn



Thủ tục định sẵn trong Hình 17-1 chèn một bản ghi nhân viên vào bảng EMP, được trình bày ở Hình 17-2.

Tất cả các ứng dụng cơ sở dữ liệu trong Hình 17-1 gọi thủ tục HIRE_EMP. Hoặc người dùng có đặc quyền có thể sử dụng Oracle Enterprise Manager hoặc Server Manager để tiến hành thủ tục HIRE_EMP sử dụng câu lệnh sau:

```
EXECUTE hire_emp ('TSMITH', 'CLERK', 1037, SYSDATE, \
                  500, NULL, 20);
```

Câu lệnh này đặt bản ghi nhân viên mới cho bảng TSMITH trong bảng EMP.

Hình 17-2 Thủ tục HIRE_EMP

```
Procedure HIRE_EMP (name VARCHAR2, job VARCHAR2,
  mgr NUMBER, hiredate DATE, sal NUMBER,
  comm NUMBER, deptno NUMBER)


BEGIN

INSERT INTO emp VALUES
   (emp_sequence.NEXTVAL, name, job, mgr
   hiredate, sal, comm, deptno),

END;
```

Các bộ phần mềm

Bộ phần mềm là một nhóm các thủ tục và các hàm có liên quan cùng với các con trỏ và các biến chúng sử dụng được lưu trữ với nhau trong cơ sở dữ liệu để sử dụng liên tục bằng một đơn vị. Tương tự với các thủ tục và các hàm được đóng gói độc lập có thể được gọi rõ ràng bằng các ứng dụng hoặc các người dùng.

Hình 17-3 minh họa bộ phần mềm gói riêng một số những thủ tục được sử dụng để quản lý cơ sở dữ liệu của nhân viên.

Các ứng dụng cơ sở dữ liệu gọi rõ ràng các thủ tục được đóng gói khi cần thiết. Sau khi được cho các đặc quyền về bộ phần mềm EMP_MGMT, người dùng có thể tiến hành rõ ràng bất kỳ thủ tục nào được chứa trong nó. Thí dụ, câu lệnh sau có thể được phát hành bằng cách sử dụng Oracle Enterprise Manager hoặc Server Manager để tiến hành thủ tục bộ phần mềm HIRE_EMP:

```
EXECUTE emp_mgmt.hire_emp ('TSMITH', 'CLERK', 1037, \
    SYSDATE, 500, NULL, 20);
```

Các bộ phần mềm cung ứng nhiều ưu điểm về hiệu suất và phát triển trên các thủ tục định sẵn độc lập (xem "Packages").

Các thủ tục và các hàm

Thủ tục hoặc hàm là đối tượng giản đồ gồm một bộ các câu lệnh SQL và cấu trúc PL/SQL khác được tạo nhóm với nhau, được lưu trữ trong cơ sở dữ liệu và được tiến hành bằng một đơn vị để giải quyết vấn đề cụ thể hoặc thực hiện một tập hợp các tác vụ liên quan. Các thủ tục và các hàm cho phép bộ gọi cung cấp các tham số có thể chỉ được nhập liệu hoặc kết xuất hoặc các giá trị nhập và xuất. Các thủ tục và các hàm cho phép bạn kết hợp sự dễ dàng và tính linh hoạt của SQL với tính năng thủ tục của ngôn ngữ lập trình có cấu trúc.

Thí dụ, câu lệnh sau tạo thủ tục CREDIT_ACCOUNT để ghi có một khoản tiền vào tài khoản ngân hàng:

```
CREATE PROCEDURE credit_account
    (acct NUMBER, credit NUMBER) AS
/* This procedure accepts two arguments: an account
  number and an amount of money to credit to the specified
  account. If the specified account does not exist, a
  new account is created. */

    old_balance  NUMBER;
```

```
    new_balance  NUMBER;
  BEGIN
    SELECT balance INTO old_balance FROM accounts
      WHERE acct_id = acct
      FOR UPDATE OF balance;

    new_balance := old_balance + credit;
    UPDATE accounts SET balance = new_balance
      WHERE acct_id = acct;
    COMMIT;

    EXCEPTION
      WHEN NO_DATA_FOUND THEN
        INSERT INTO accounts (acct_id, balance)
          VALUES(acct, credit);
      WHEN OTHERS THEN
    ROLLBACK;
END credit_account;
```

Lưu ý thủ tục mẫu này gồm cả câu lệnh SQL và PL/SQL.

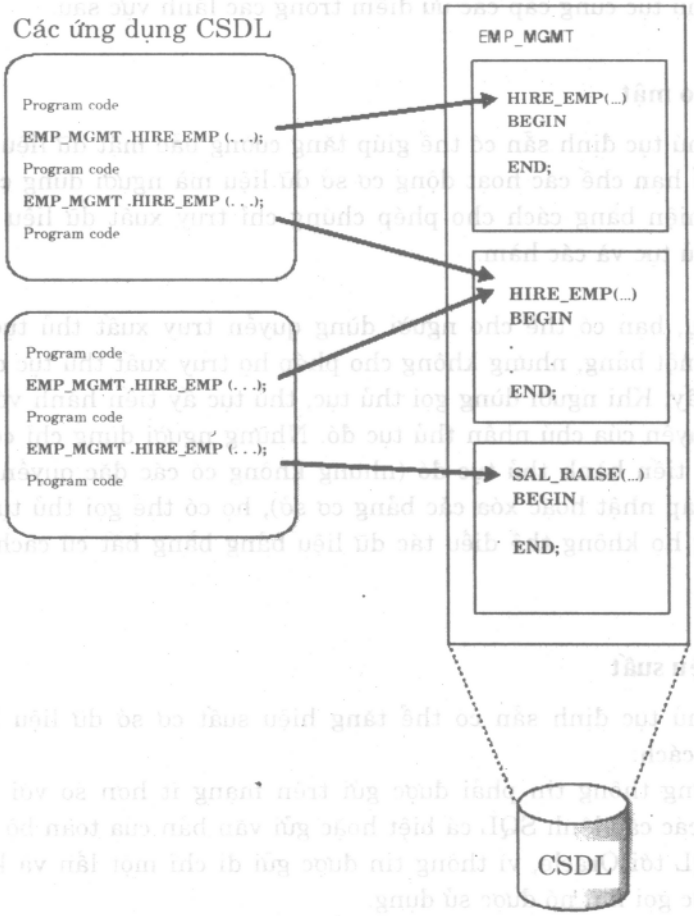
Các nguyên tắc chủ đạo thủ tục

Sử dụng các nguyên tắc chủ đạo thủ tục sau để thiết kế và sử dụng tất cả các thủ tục định sẵn:

Xác định các thủ tục để hoàn tất tác vụ tập trung và đơn giản. Không xác định các thủ tục dài với nhiều tác vụ phụ riêng biệt, vì các tác vụ phụ phổ biến với nhiều thủ tục có thể được sao chép không cần thiết bằng mã của nhiều thủ tục.

Không xác định những thủ tục sao chép tính năng đã được các đặc trưng khác của Oracle cung cấp. Thí dụ, không xác định các thủ tục để tăng cường những nguyên tắc toàn vẹn dữ liệu đơn giản mà bạn có thể dễ dàng tăng cường bằng cách sử dụng các hạn định tính toàn vẹn được khai báo.

Hình 17-3 Bộ phần mềm được lưu trữ



Các ưu điểm của các thủ tục

Các thủ tục cung cấp các ưu điểm trong các lãnh vực sau.

Bảo mật

Các thủ tục định sẵn có thể giúp tăng cường bảo mật dữ liệu. Bạn có thể hạn chế các hoạt động cơ sở dữ liệu mà người dùng có thể thực hiện bằng cách cho phép chúng chỉ truy xuất dữ liệu bằng các thủ tục và các hàm.

Thí dụ, bạn có thể cho người dùng quyền truy xuất thủ tục cập nhật một bảng, nhưng không cho phép họ truy xuất thủ tục chính bảng ấy. Khi người dùng gọi thủ tục, thủ tục ấy tiến hành với các đặc quyền của chủ nhân thủ tục đó. Những người dùng chỉ có đặc quyền tiến hành thủ tục đó (nhưng không có các đặc quyền truy vấn, cập nhật hoặc xóa các bảng cơ sở), họ có thể gọi thủ tục ấy, nhưng họ không thể điều tác dữ liệu bảng bằng bất cứ cách nào khác.

Hiệu suất

Các thủ tục định sẵn có thể tăng hiệu suất cơ sở dữ liệu bằng nhiều cách:
Số lượng thông tin phải được gửi trên mạng ít hơn so với phát hành các câu lệnh SQL cá biệt hoặc gửi văn bản của toàn bộ khối PL/SQL tới Oracle, vì thông tin được gửi đi chỉ một lần và kể từ đó được gọi khi nó được sử dụng.
Biểu mẫu được biên dịch của thủ tục có thể dễ dàng sử dụng trong cơ sở dữ liệu, vì vậy không cần biên dịch vào thời gian tiến hành. Nếu thủ tục đã hiện hữu trong vùng chia dùng chung của SGA rồi, thì cần thiết truy hồi từ đĩa và có thể bắt đầu tiến hành ngay.

Phân bổ bộ nhớ

Bởi vì các thủ tục định sẵn khai thác các khả năng bộ nhớ dùng chung của Oracle, chỉ một bản sao đơn của thủ tục cần được nạp vào bộ nhớ để nhiều người dùng tiến hành. Sử dụng chung cùng mã giữa nhiều người dùng dẫn đến làm giảm các đòi hỏi bộ nhớ của Oracle cho các ứng dụng đáng kể.

Năng suất

Các thủ tục định sẵn tăng năng suất phát triển. Bằng cách thiết kế các ứng dụng khoảng một bộ những thủ tục thông thường, bạn có thể tránh được viết mã không cần thiết và tăng năng suất của bạn.

Thí dụ, các thủ tục có thể được viết để chèn, cập nhật hoặc xóa các hàng khỏi bảng EMP. Các thủ tục này sau đó có thể viết lại các câu lệnh SQL cần thiết để hoàn tất những tác vụ này. Nếu các phương pháp quản lý dữ liệu thay đổi, chỉ các thủ tục cần được sửa đổi, không phải tất cả các ứng dụng sử dụng các thủ tục.

Tính nguyên vẹn

Các thủ tục định sẵn để tăng tính nguyên vẹn và tính nhất quán của các ứng dụng của bạn. Bằng cách phát triển tất cả các ứng dụng của bạn khoảng một nhóm các thủ tục chung, bạn có thể giảm khả năng cam kết các lỗi viết mã.

Thí dụ, bạn có thể trắc nghiệm thủ tục hoặc hàm để bảo đảm nó cho một kết quả chuẩn xác và một khi nó được xác minh, sử dụng lại nó bằng bất kỳ số ứng dụng nào mà không cần trắc nghiệm nó lại. Nếu các cấu trúc dữ liệu được tham chiếu bởi thủ tục được thay bằng bất kỳ cách nào, chỉ thông tin ấy cần được biên dịch lại; các ứng dụng gọi thủ tục ấy không nhất thiết đòi hỏi bất kỳ sửa đổi nào.

Các khối PL/SQL nặc danh so với các thủ tục định sẵn

Thông tin lưu trữ được tạo và được lưu trữ trong cơ sở dữ liệu bằng một đối tượng giản đồ. Một khi được tạo và được biên dịch, nó là một đối tượng có tên có thể được tiến hành mà không phải biên dịch lại. Ngoài ra, thông tin phụ thuộc được lưu trữ trong từ điển dữ liệu để bảo đảm tính hợp lệ của mỗi thủ tục định sẵn.

Một sự lựa chọn khác đối với thủ tục định sẵn, bạn có thể tạo khối PL/SQL nặc danh bằng cách gửi khối PL/SQL không có tên tới Oracle server từ một công cụ Oracle hoặc ứng dụng. Oracle biên dịch khối PL/SQL và đưa phiên bản được biên dịch vào vùng chia dùng chung của SGA, nhưng không lưu trữ mã nguồn hoặc phiên bản được biên dịch trong cơ sở dữ liệu để sử dụng lại ngoài môi trường làm việc hiện hành. SQL dùng chung cho phép các khối PL/SQL nặc danh trong vùng chia dùng chung để sử dụng lại và được sử dụng chung cho tới khi chúng được xóa sạch khỏi vùng chia dùng chung.

Trong hai trường hợp, di chuyển các khối PL/SQL ra ngoài ứng dụng cơ sở dữ liệu và vào các thủ tục cơ sở dữ liệu được lưu trữ trong cơ sở dữ liệu hoặc bộ nhớ, bạn tránh để Oracle biên dịch lại thủ tục không cần thiết vào thời gian tiến hành để tăng hiệu suất tổng quát của ứng dụng và Oracle.

Các thủ tục độc lập

Các thủ tục định sẵn không xác định trong ngữ cảnh của bộ phần mềm được gọi là các thủ tục độc lập. Các thủ tục được xác định trong bộ phần mềm được coi là thành phần của bộ phần mềm ấy. (xem "Packages" về các ưu điểm của các bộ phần mềm).

Sự phụ thuộc truy tìm các thủ tục định sẵn

Thủ tục định sẵn phụ thuộc vào các đối tượng được tham chiếu trong thân của nó. Oracle tự động truy tìm và quản lý các phụ thuộc. Thí dụ, nếu bạn thay đổi định nghĩa của bảng được một thủ tục tham chiếu, thủ tục ấy phải được biên dịch lại để thẩm định nó sẽ tiếp tục hoạt động như được thiết kế. Thường Oracle tự động quản trị sự quản lý phụ thuộc như thế.

Xem "Oracle Dependency Management" về truy tìm sự phụ thuộc.

Các thủ tục bên ngoài

Thủ tục PL/SQL tiến hành trên Oracle server có thể gọi một thủ tục bên ngoài hoặc hàm được viết bằng ngôn ngữ lập trình C và được lưu trữ trong thư viện dùng chung. Thủ tục con C tiến hành trong không gian địa chỉ tách ra từ không gian của Oracle server.

Thông tin bổ sung:
Xem PL/SQL User's và Reference về các thủ tục bên ngoài.

Các bộ phần mềm

Các bộ phần mềm gói riêng các thủ tục, các hàm liên quan, các con trỏ được kết hợp và các biến với nhau bằng một đơn vị trong cơ sở dữ liệu.

Bạn tạo một bộ phần mềm bằng hai phần: định chuẩn và thân. Định chuẩn của bộ phần mềm khai báo tất cả các cấu trúc chung của bộ phần mềm và thân xác định tất cả cấu trúc (chung và riêng) của bộ phần mềm. Sự tách ra hai phần này cung cấp các ưu điểm sau:
Developer có tính linh hoạt hơn trong chu kỳ phát triển. Bạn có thể tạo các định chuẩn và tham chiếu các thủ tục chung mà không thực sự tạo thân bộ phần mềm.

Bạn có thể thay thế các thân của thủ tục được chứa trong thân của bộ phần mềm tách ra từ các định chuẩn được khai báo chung trong định chuẩn bộ phần mềm. Bao lâu định chuẩn thủ tục không thay đổi, các đối tượng tham chiếu các thủ tục được thay thế của bộ phần mềm không bao giờ đánh dấu không hợp lệ; nghĩa là chúng không bao giờ được đánh dấu cần biên dịch lại. (Xem thêm thông tin về các tính phụ thuộc, "Oracle Dependency Management").

Thí dụ sau tạo định chuẩn và thân cho bộ phần mềm chứa nhiều thủ tục và các hàm xử lý các giao dịch ngân hàng.

```
CREATE PACKAGE bank_transactions (null) AS
 minimum_balance  CONSTANT NUMBER := 100.00;
 PROCEDURE apply_transactions;
 PROCEDURE enter_transaction (acct   NUMBER,
                              kind   CHAR,
                              amount NUMBER);
END bank_transactions;

CREATE PACKAGE BODY bank_transactions AS

/*  Package to input bank transactions */

  new_status  CHAR(20); /* Global variable to record status
                           of transaction being applied. Used
                           for update in APPLY_TRANSACTIONS. */

  PROCEDURE do_journal_entry (acct NUMBER,
                              kind CHAR) IS

/*  Records a journal entry for each bank transaction applied
    by the APPLY_TRANSACTIONS procedure. */

  BEGIN
      INSERT INTO journal
        VALUES (acct, kind, sysdate);
      IF kind = 'D' THEN
        new_status := 'Debit applied';
      ELSIF kind = 'C' THEN
        new_status := 'Credit applied';
      ELSE
```

```
      new_status := 'New account';
    END IF;
  END do_journal_entry;

  PROCEDURE credit_account (acct NUMBER, credit NUMBER) IS

/* Credits a bank account the specified amount. If the account
   does not exist, the procedure creates a new account first. */

    old_balance  NUMBER;
    new_balance  NUMBER;

  BEGIN
    SELECT balance INTO old_balance FROM accounts
      WHERE acct_id = acct
      FOR UPDATE OF balance; /* Locks account for credit update */

    new_balance := old_balance + credit;
    UPDATE accounts SET balance = new_balance
      WHERE acct_id = acct;
    do_journal_entry(acct, 'C');

  EXCEPTION
    WHEN NO_DATA_FOUND THEN  /* Create new account if not found */
      INSERT INTO accounts (acct_id, balance)
        VALUES(acct, credit);
      do_journal_entry(acct, 'N');
    WHEN OTHERS THEN  /* Return other errors to application */
      new_status := 'Error: ' || SQLERRM(SQLCODE);
  END credit_account;

  PROCEDURE debit_account (acct  NUMBER, debit NUMBER) IS

/* Debits an existing account if result is greater than the
   allowed minimum balance. */

    old_balance        NUMBER;
    new_balance        NUMBER;
    insufficient_funds  EXCEPTION;

  BEGIN
    SELECT balance INTO old_balance FROM accounts
      WHERE acct_id = acct
      FOR UPDATE OF balance;
```

```
    new_balance := old_balance - debit;
    IF new_balance >= minimum_balance THEN
      UPDATE accounts SET balance = new_balance

        WHERE acct_id = acct;
    do_journal_entry(acct, 'D');
    ELSE
      RAISE insufficient_funds;
    END IF;

  EXCEPTION
    WHEN NO_DATA_FOUND THEN
      new_status := 'Nonexistent account';
    WHEN insufficient_funds THEN
      new_status := 'Insufficient funds';
    WHEN OTHERS THEN  /* Returns other errors to application */
      new_status  = 'Error: ' || SQLERRM(SQLCODE);
  END debit_account;

  PROCEDURE apply_transactions IS

/* Applies pending transactions in the table TRANSACTIONS to the
   ACCOUNTS table. Used at regular intervals to update bank
   accounts without interfering with input of new transactions. */

/* Cursor fetches and locks all rows from the TRANSACTIONS
   table with a status of 'Pending'. Locks released after all
       pending transactions have been applied. */

      CURSOR trans_cursor IS
       SELECT acct_id, kind, amount FROM transactions
         WHERE status = 'Pending'
         ORDER BY time_tag
         FOR UPDATE OF status;

    BEGIN
      FOR trans IN trans_cursor LOOP   /* implicit open and fetch */
       IF trans.kind = 'D' THEN
         debit_account(trans.acct_id, trans.amount);
       ELSIF trans.kind = 'C' THEN
         credit_account(trans.acct_id, trans.amount);
       ELSE
         new_status := 'Rejected';
       END IF;
```

```
        /* Update TRANSACTIONS table to return result of applying
           this transaction. */
        UPDATE transactions SET status = new_status
          WHERE CURRENT OF trans_cursor;
      END LOOP;
      COMMIT;  /* Release row locks in TRANSACTIONS table. */
  END apply_transactions;
  PROCEDURE enter_transaction (acct   NUMBER,
                               kind   CHAR,
                               amount NUMBER) IS

/* Enters a bank transaction into the TRANSACTIONS table. A new
   transaction is always put into this 'queue' before being
   applied to the specified account by the APPLY_TRANSACTIONS
   procedure. Therefore, many transactions can be simultaneously
   input without interference. */

  BEGIN
      INSERT INTO transactions
        VALUES (acct, kind, amount, 'Pending', sysdate);
      COMMIT;
  END enter_transaction;

END bank_transactions;
```

Các bộ phần mềm cho phép người điều hành cơ sở dữ liệu hoặc Developer ứng dụng tổ chức các trình tương tự. Chúng còn cung cấp tính năng và hiệu suất cơ sở dữ liệu tăng lên.

Các ưu điểm của các bộ phần mềm

Các bộ phần mềm được sử dụng để xác định các thủ tục, các biến và các con trỏ có liên quan thường được bổ sung để cung cấp các ưu điểm trong các lãnh vực sau:

Gói riêng các thủ tục và các biến có liên quan

Khai báo các thủ tục chung và riêng, các biến, các hằng và các con trỏ

Hiệu suất cao hơn

Gói riêng

Các bộ phần mềm được lưu trữ cho phép bạn gói riêng - *encapsulate* - (tạo nhóm) những thủ tục định sẵn có liên quan, các biến, các kiểu dữ liệu, vân vân . . . bằng đơn vị được định sẵn có tên đơn trong cơ sở dữ liệu. Điều này cung cấp tổ chức tốt hơn khi tiến trình phát triển.

Gói riêng các cấu trúc thủ tục trong một bộ phần mềm còn tạo cho quản lý đặc quyền dễ dàng hơn. Cho đặc quyền để sử dụng một bộ phần mềm tạo cho tất cả các cấu trúc của bộ phần mềm người được cho đặc quyền có thể truy xuất.

Dữ liệu chung và riêng và các thủ tục

Các phương pháp của định nghĩa bộ phần mềm cho phép bạn xác định các biến, các con trỏ và các thủ tục.

Chung	Người dùng truy xuất trực tiếp bộ phần mềm.
Riêng	Ẩn khỏi bộ phần mềm của người dùng.

Thí dụ, một bộ phần mềm có thể chứa 10 thủ tục. Bạn có thể xác định bộ phần mềm để chỉ 3 thủ tục là chung và do đó người dùng có thể tiến hành bộ phần mềm ấy; phần còn lại của các thủ tục là riêng và chỉ có thể được truy xuất bằng các thủ tục trong bộ phần mềm ấy.

Không lầm lẫn các biến của bộ phần mềm chung và riêng với các đặc quyền là PUBLIC được mô tả ở "Điều khiển Truy Xuất Cơ Sở Dữ Liệu".

Cải tiến hiệu suất

Toàn bộ một bộ phần mềm được nạp vào bộ nhớ khi thông tin trong bộ phần mềm ấy được gọi lần đầu tiên. Lần nạp này được hoàn tất bằng điều hoạt, trái lại các lần nạp cần các thủ tục độc

lập. Do đó khi gọi các thủ tục được đóng gói có liên quan xảy ra, không cần đĩa I/O để tiến hành mã được biên dịch sẵn trong bộ nhớ.

Thân bộ phần mềm có thể được thay thế và được biên dịch lại mà không ảnh hưởng đến định chuẩn. Vì, các đối tượng giản đồ tham chiếu các cấu trúc của bộ phần mềm (luôn luôn bằng định chuẩn ấy) không phải biên dịch lại trừ khi định chuẩn bộ phần mềm cũng được thay thế. Bằng cách sử dụng các bộ phần mềm, giảm tối đa các biên dịch lại không cần thiết dẫn đến ít ảnh hưởng đến hiệu suất cơ sở dữ liệu tổng quát.

Truy tìm tính phụ thuộc đối với các bộ phần mềm

Bộ phần mềm phụ thuộc vào các đối tượng được tham chiếu bằng các thủ tục và các hàm được xác định trong thân của nó. Oracle tự động truy tìm và quản lý các sự phụ thuộc như thế. Xem "Quản lý tính phụ thuộc của Oracle" về truy tìm tính phụ thuộc.

Oracle lưu trữ các thủ tục và các bộ phần mềm như thế nào

Khi bạn tạo một thủ tục hoặc bộ phần mềm, Oracle
Biên dịch thủ tục hoặc bộ phần mềm
Lưu trữ mã được biên dịch trong bộ nhớ
Lưu trữ thủ tục hoặc bộ phần mềm trong cơ sở dữ liệu

Biên dịch các thủ tục và các bộ phần mềm

Trình biên dịch PL/SQL biên dịch mã nguồn. Trình biên dịch PL/SQL là thành phần của động cơ PL/SQL được chứa trong Oracle. Nếu xảy ra lỗi khi biên dịch cho trở lại thông báo.

Thông tin bổ sung:
Thông tin về nhận dạng các lỗi biên dịch có trong Oracle8 Application Developer's Guide.

Lưu trữ mã được biên dịch trong bộ nhớ

Oracle trữ thủ tục được biên dịch hoặc bộ phần mềm trong vùng chia dùng chung của vùng toàn cục hệ thống (SGA). Điều này cho phép mã ấy được tiến hành nhanh và được dùng chung giữa nhiều người dùng. Phiên bản của thủ tục được biên dịch hoặc bộ phần mềm vẫn còn trong vùng chia dùng chung tương ứng với thuật toán Least-Recently-Used sửa đổi được vùng chia dùng chung sử dụng, dù người gọi ban đầu của thủ tục kết thúc phiên của người ấy. Xem "Vùng chia dùng chung" về vùng đệm vùng chia dùng chung.

Lưu trữ các thủ tục hoặc các bộ phần mềm trong cơ sở dữ liệu

Vào thời gian tạo và biên dịch, Oracle tự động lưu trữ các thông tin sau về thủ tục hoặc bộ phần mềm vào cơ sở dữ liệu:

Tên đối tượng giản đồ	Tên này nhận dạng thủ tục hoặc bộ phần mềm, bạn ấn định tên này trong CREATE PROCEDURE, CREATE FUNCTION, CREATE PACKAGE hoặc câu lệnh CREATE PACKAGE BODY.
Mã nguồn và cây phân ngữ	Trình biên dịch PL/SQL phân ngữ mã nguồn và tạo biểu thị phân ngữ của mã nguồn được gọi là cây phân ngữ.
Mã giả (mã P)	Trình biên dịch PL/SQL phát sinh mã giả hoặc mã P dựa trên mã được phân ngữ. Động cơ PL/SQL tiến hành điều này khi thủ tục hoặc bộ phần mềm được gọi.
Các thông điệp lỗi	Oracle có thể phát sinh các lỗi khi biên dịch một thủ tục hoặc bộ phần mềm.

Để tránh biên dịch lại thủ tục hoặc bộ phần mềm không cần thiết, cả cây phân ngữ và mã P của đối tượng được lưu trữ trong

cơ sở dữ liệu. Điều này cho phép động cơ PL/SQL đọc phiên bản được biên dịch của thủ tục hoặc bộ phần mềm vào vùng đệm vùng chia dùng chung của SGA khi nó được gọi và hiện không ở trong SGA. Cây phân ngữ được sử dụng khi mã gọi thủ tục được biên dịch.

Tất cả những phần của các thủ tục cơ sở dữ liệu được lưu trữ trong từ điển dữ liệu (ở trong không gian bảng SYSTEM) của cơ sở dữ liệu tương ứng. Khi dự tính kích thước của không gian bảng SYSTEM, người điều hành cơ sở dữ liệu sẽ ghi nhớ tất cả những thủ tục định sẵn đòi hỏi không gian trong không gian bảng này.

Oracle tiến hành các thủ tục và các bộ phần mềm như thế nào

Khi bạn gọi một thủ tục độc lập hoặc được đóng gói, Oracle xác minh truy xuất người dùng, xác minh tính hợp lệ của thủ tục và tiến hành thủ tục ấy.

Xác minh truy xuất người dùng

Oracle xác minh người gọi có đặc quyền EXECUTE hoặc sở hữu nó trên thủ tục hoặc bộ phần mềm gói riêng. Người dùng tiến hành thủ tục không cần phải truy xuất bất kỳ thủ tục hoặc các đối tượng được tham chiếu trong thủ tục ấy; chỉ bộ tạo thủ tục hoặc bộ phần mềm cần có các đặc quyền để truy xuất những đối tượng giản đồ được tham chiếu.

Xác minh tính hợp lệ thủ tục

Oracle kiểm từ điển dữ liệu để xác định xem trạng thái của thủ tục hoặc bộ phần mềm hợp lệ hoặc không. Thủ tục hoặc bộ phần mềm không hợp lệ khi một trong những điều sau đã xảy ra khi thủ tục hoặc bộ phần mềm đã được biên dịch lần cuối:

Một hoặc nhiều đối tượng giản đồ được tham chiếu trong thủ tục hoặc bộ phần mềm (chẳng hạn như, các bảng, các khung nhìn và

các thủ tục khác) đã được thay đổi hoặc bỏ (thí dụ, nếu người dùng đã bổ sung một cột vào một bảng).
Đặc quyền hệ thống mà bộ phần mềm hoặc thủ tục cần thiết đã được gọi từ PUBLIC hoặc từ chủ nhân của thủ tục hoặc bộ phần mềm.
Đặc quyền đối tượng giản đồ cần thiết cho một hoặc nhiều đối tượng giản đồ được tham chiếu bằng một thủ tục hoặc bộ phần mềm đã được thu hồi từ PUBLIC hoặc từ chủ nhân của thủ tục hoặc bộ phần mềm.

Thủ tục hợp lệ nếu nó không bị bất hợp lệ bởi một trong những điều hoạt trên. Nếu thủ tục được đóng gói hoặc tách riêng hợp lệ được gọi, thì mã được biên dịch được tiến hành. Nếu thủ tục được đóng gói hoặc bất hợp lệ được gọi, nó tự động biên dịch lại trước khi được tiến hành.

Để có đầy đủ thông tin về các thủ tục và các bộ phần mềm hợp lệ và bất hợp lệ, biên dịch lại các thủ tục và các vấn đề phụ thuộc, xem "Quản Lý Tính Phụ Thuộc Của Oracle".

Tiến hành một thủ tục

Động cơ PL/SQL tiến hành thủ tục hoặc bộ phần mềm sử dụng các Step khác nhau phụ thuộc vào từng trường hợp:
Nếu thủ tục hợp lệ và hiện hữu trong bộ nhớ, thì động cơ PL/SQL tiến hành mã P một cách đơn giản.
Nếu thủ tục hợp lệ và không hiện hữu trong bộ nhớ, thì động cơ PL/SQL nạp mã P được biên dịch từ đĩa vào bộ nhớ và tiến hành nó. Đối với các bộ phần mềm, tất cả cấu trúc của bộ phần mềm ấy (tất cả các thủ tục, các biến, vân vân .. được biên dịch bằng một phân đoạn mã khả thi) được nạp bằng một đơn vị.

Động cơ PL/SQL xử lý từng câu lệnh thủ tục, tự nó điều quản tất cả các câu lệnh thủ tục và gửi các câu lệnh SQL tới bộ tiến hành câu lệnh SQL như được minh họa ở Hình 14-1.

Chương 18

Các bộ kích khởi cơ sở dữ liệu

Chương này đề cập đến các bộ kích khởi cơ sở dữ liệu; nghĩa là các thủ tục định sẵn trong cơ sở dữ liệu và được tiến hành ngầm định khi một bảng được sửa đổi. Chương này gồm:

- **Giới thiệu các bộ kích khởi**
- **Các thành phần của bộ kích khởi**
- **Các kiểu bộ kích khởi**
- **Sự tiến hành của bộ kích khởi**

Thông tin bổ sung:
Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle.

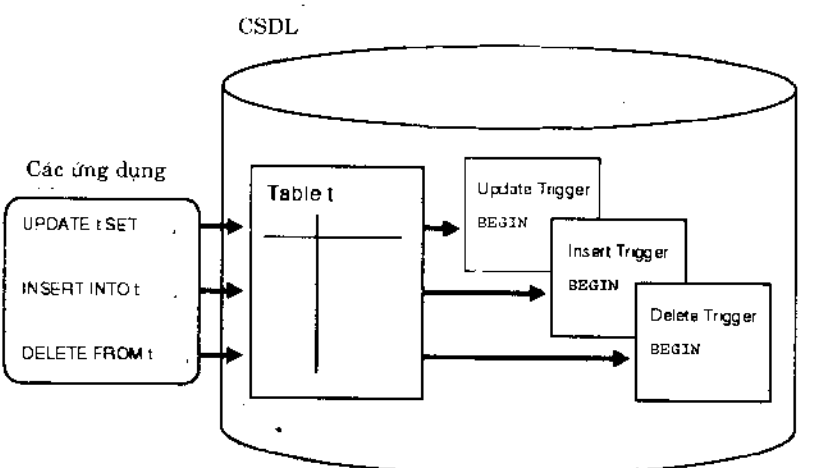
Giới thiệu các bộ kích khởi

Oracle cho phép bạn xác định các thủ tục được tiến hành ngầm định khi một câu lệnh INSERT, UPDATE hoặc DELETE được phát hành dựa trên bảng được kết hợp. Các thủ tục này được gọi là các bộ kích khởi cơ sở dữ liệu.

Các bộ kích khởi giống như các thông tin được lưu trữ đã đề cập ở "Các Thủ Tục Và Các Bộ Phần Mềm". Một bộ kích khởi có thể có các câu lệnh SQL và PL/SQL để tiến hành bằng một đơn vị và

có thể gọi các thủ tục định sẵn. Tuy nhiên, các thủ tục và các bộ kích khởi được gọi bằng các phương thức khác nhau. Thủ tục được tiến hành rõ ràng bởi người dùng, ứng dụng hoặc bộ kích khởi. Các bộ kích khởi (một hoặc nhiều) được tiến hành ngầm định bởi Oracle khi một câu lệnh kích khởi INSERT, UPDATE hoặc DELETE được phát hành, bất kể người dùng nào được kết nối hoặc ứng dụng nào đang được sử dụng.
Hình 18-1 trình bày ứng dụng cơ sở dữ liệu với một số câu lệnh SQL tiến hành nhiều bộ kích khởi được lưu trữ trong cơ sở dữ liệu.

Hình 18-1 Các bộ kích khởi



Lưu ý các bộ kích khởi được lưu trữ trong cơ sở dữ liệu tách khỏi các bảng được kết hợp của chúng.

Các bộ kích khởi có thể chỉ được xác định trên các bảng, không phải trên các khung nhìn. Tuy nhiên, các bộ kích khởi trên (các) bảng cơ sở của khung nhìn được tiến hành nếu câu lệnh INSERT, UPDATE hoặc DELETE được phát hành dựa trên khung nhìn.

Các bộ kích khởi được sử dụng như thế nào

Các bộ kích khởi có thể bổ sung các khả năng tiêu chuẩn của Oracle để cung cấp hệ thống quản lý cơ sở dữ liệu tùy biến cao. Thí dụ, một bộ kích khởi có thể hạn chế các hoạt động DML xảy ra dựa trên một bảng chỉ vào những giờ hoạt động bình thường. Các khả năng khác sử dụng các bộ kích khởi để

- tự động phát sinh các giá trị cột được phái sinh
- ngăn cản các giao dịch bất hợp lệ
- tăng cường các quyền hạn bảo mật phức tạp
- tăng cường tính nguyên vẹn tham chiếu các nút bằng một cơ sở dữ liệu được phân phối
- tăng cường những nguyên tắc hoạt động kinh doanh phức tạp
- cung cấp ghi sổ sự kiện rõ ràng
- cung cấp kiểm soát tinh vi
- duy trì sao chép bảng đồng bộ
- tập hợp thống kê trên truy xuất bảng

Thông tin bổ sung:
Các thí dụ về nhiều cách sử dụng bộ kích khởi này được đề cập ở Oracle8 Application Developer's Guide.

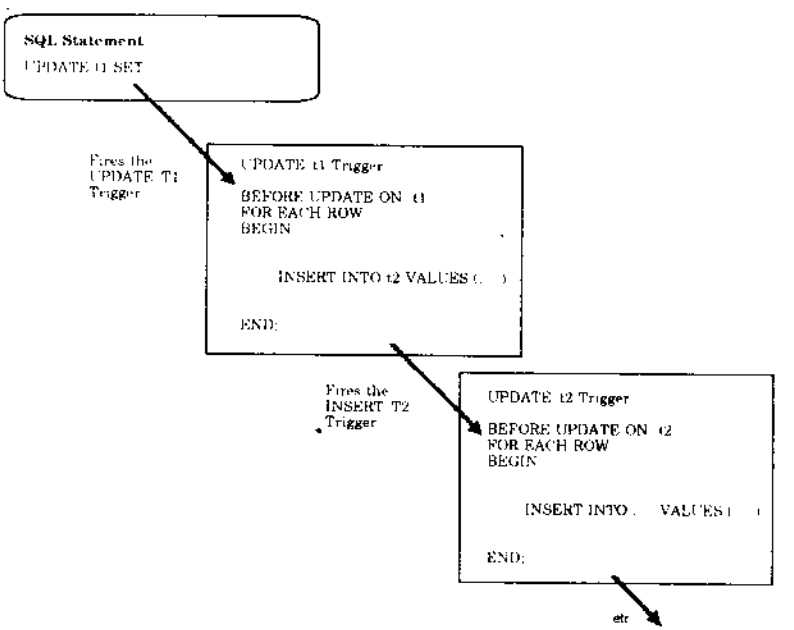
Một số cảnh báo về các bộ kích khởi

Các bộ kích khởi hữu dụng để tùy biến cơ sở dữ liệu. Tuy nhiên, bạn nên sử dụng các bộ kích khởi chỉ khi cần thiết. Sử dụng quá mức các bộ kích khởi có thể dẫn đến bị các sự phụ thuộc hỗ tương phức tạp, có thể khó duy trì trong một ứng dụng lớn. Thí dụ, khi một bộ kích khởi được tiến hành, câu lệnh SQL trong hoạt động của bộ kích khởi của nó có thể kích hoạt các bộ kích khởi khác, như được minh họa ở Hình 18-2.

Khi một câu lệnh trong thân bộ kích khởi gây cho một bộ kích khởi khác được tiến hành, các bộ kích khởi được coi là *cascading* (xếp chuỗi).

Lưu ý:
Oracle Forms có thể xác định, lưu trữ và thực hiện các bộ kích khởi của một sắp xếp khác. Tuy nhiên, không lầm lẫn các bộ kích khởi của Oracle Forms với các bộ kích khởi cơ sở dữ liệu được đề cập ở chương này.

Hình 18-2 Các bộ kích khởi xếp chuỗi



Các bộ kích khởi so với các hạn định tính nguyên vẹn khai báo

Bạn có thể sử dụng cả các bộ kích khởi cơ sở dữ liệu và các hạn định tính nguyên vẹn để xác định và tăng cường bất kỳ kiểu

nguyên tắc về tính nguyên vẹn nào. Tuy nhiên, công ty Oracle đề nghị bạn sử dụng các bộ kích khởi cơ sở dữ liệu để hạn định nhập liệu chỉ trong các trường hợp sau:

- khi nguyên tắc tính nguyên vẹn tham chiếu cần thiết không được tăng cường, sử dụng các hạn định tính nguyên vẹn sau:
  - khóa NOT NULL, UNIQUE
  - PRIMARY KEY
  - FOREIGN KEY
  - CHECK
  - cập nhật CASCADE
  - cập nhật và xóa SET NULL
  - cập nhật và xóa SET DEFAULT
- để tăng cường tính nguyên vẹn tham chiếu khi các bảng con và cha trên các nút khác nhau của cơ sở dữ liệu được phân phối
- để tăng cường các nguyên tắc hoạt động kinh doanh không xác định được, sử dụng các hạn định tính nguyên vẹn

Để có thêm thông tin về các hạn định tính nguyên vẹn, xem "Oracle Tăng Cường Tính Nguyên Vẹn Dữ Liệu Như Thế Nào".

Các thành phần của bộ kích khởi

Một bộ kích khởi có các thành phần căn bản sau:
câu lệnh hoặc sự kiện kích khởi
hạn chế bộ kích khởi
hoạt động của bộ kích khởi

Hình 18-3 biểu thị mỗi thành phần này của bộ kích khởi và không có nghĩa là trình bày cú pháp chính xác. Các phần sau giải thích từng thành phần của bộ kích khởi chi tiết hơn.

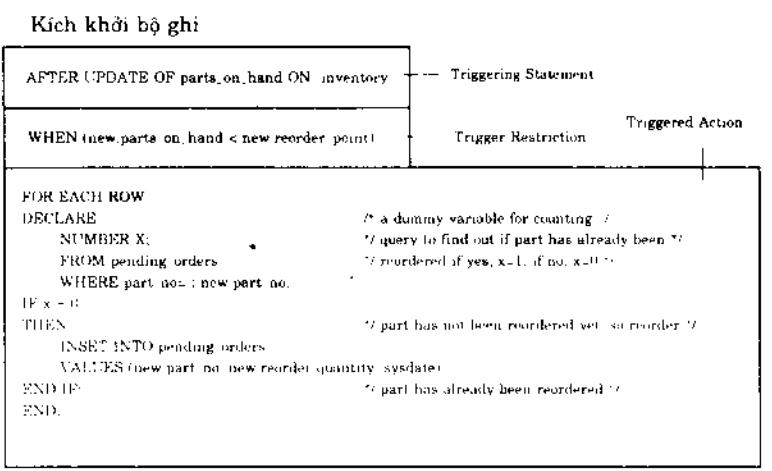
Sự kiện kích khởi hoặc câu lệnh

Sự kiện kích khởi hoặc câu lệnh là câu lệnh SQL làm cho bộ kích khởi được tiến hành. Sự kiện kích khởi có thể là câu lệnh INSERT, UPDATE hoặc DELETE trên bảng. Thí dụ, ở Hình 18-3, câu lệnh kích khởi là

. . . UPDATE OF parts_on_hand ON inventory . . .

nghĩa là: khi cột PARTS_ON_HAND của hàng trong bảng INVENTORY được cập nhật, tiến hành bộ kích khởi ấy. Lưu ý là khi sự kiện kích khởi là một câu lệnh UPDATE, bạn có thể có một danh sách cột để nhận dạng cột nào phải được cập nhật để tiến hành bộ kích khởi ấy. Bạn không thể ấn định danh sách cột về các câu lệnh INSERT và DELETE, vì chúng ảnh hưởng đến toàn bộ các hàng thông tin.

Hình 18-3 Bộ kích khởi REORDER



Sự kiện kích khởi có thể ấn định nhiều câu lệnh DML như trong

. . . INSERT OR UPDATE OR DELETE OF inventory . . .

nghĩa là: khi câu lệnh INSERT, UPDATE hoặc DELETE được phát hành dựa trên bảng INVENTORY, tiến hành bộ kích khởi

ấy. Khi nhiều kiểu câu lệnh DML có thể tiến hành bộ kích khởi, bạn có thể sử dụng các thuộc tính có điều kiện để phát hiện kiểu câu lệnh kích khởi. Bằng cách này, bạn có thể tạo một bộ kích khởi đơn để tiến hành mã khác nhau dựa trên kiểu câu lệnh tiến hành bộ kích khởi.

Sự hạn chế bộ kích khởi

Sự hạn chế bộ kích khởi xác định biểu thức Boolean (logic) phải là TRUE cho bộ kích khởi để tiến hành. Hoạt động của bộ kích khởi không được tiến hành nếu sự hạn chế bộ kích khởi thẩm định là FALSE hoặc UNKNOWN. Trong thí dụ, sự hạn chế bộ kích khởi là

new.parts_on_hand < new.reorder_point

Hoạt động của bộ kích khởi

Hoạt động của bộ kích khởi là thủ tục (khối PL/SQL) chứa các câu lệnh SQL và mã PL/SQL để được tiến hành khi câu lệnh kích khởi được phát hành và sự hạn chế của bộ kích khởi thẩm định là TRUE.

Giống như các thủ tục định sẵn, hoạt động của bộ kích khởi có thể chứa các câu lệnh SQL và PL/SQL, xác định các cấu trúc ngôn ngữ PL/SQL (các biến, các hằng, các con trỏ, các ngoại lệ, vân vân) và gọi các thủ tục định sẵn. Hơn nữa, đối với các bộ kích khởi hàng (được mô tả trong phần kế tiếp), các câu lệnh trong hoạt động của bộ kích khởi được truy xuất các giá trị cột (mới và cũ) của hàng hiện hành được bộ kích khởi xác định. Hai tên tương quan cung cấp truy xuất các giá trị cũ và mới cho mỗi cột.

Các kiểu của bộ kích khởi

Phần này mô tả các kiểu bộ kích khởi khác nhau:
các bộ kích khởi câu lệnh và hàng

các bộ kích khởi BEFORE và AFTER
các bộ kích khởi INSTEAD OF

Các bộ kích khởi hàng và các bộ kích khởi câu lệnh

Khi bạn xác định một bộ kích khởi, bạn có thể ấn định số lần hoạt động của bộ kích khởi ấy được tiến hành: mỗi lần cho mỗi hàng được tác động bởi câu lệnh kích khởi (chẳng hạn như có thể được câu lệnh UPDATE tiến hành để cập nhật nhiều hàng) hoặc một lần cho câu lệnh kích khởi, bất kể bao nhiêu hàng mà nó tác động.

Các bộ kích khởi hàng

Một bộ kích khởi hàng được tiến hành mỗi lần bảng được tác động bằng cậu lệnh kích khởi. Thí dụ, nếu câu lệnh UPDATE cập nhật nhiều hàng của một bảng, bộ kích khởi dữ liệu được tiến hành một lần cho mỗi hàng được câu lệnh UPDATE tác động. Nếu câu lệnh kích khởi không tác động hàng nào, thì bộ kích khởi hàng không tiến hành.

Các bộ kích khởi hàng hữu dụng nếu mã trong hoạt động bộ kích khởi phụ thuộc vào dữ liệu được câu lệnh kích khởi cung cấp hoặc các hàng được tác động. Thí dụ, Hình 18-3 minh họa bộ kích khởi hàng sử dụng những giá trị của mỗi hàng được câu lệnh kích khởi tác động.

Các bộ kích khởi câu lệnh

Statement Trigger (bộ kích khởi câu lệnh) được tiến hành một lần thay cho câu lệnh kích khởi, bất kể số hàng trong bảng mà câu lệnh kích khởi tác động (dù không có hàng nào được tác động). Thí dụ, nếu câu lệnh DELETE xóa nhiều hàng khỏi bảng, bộ kích hoạt DELETE mức câu lệnh chỉ tiến hành một lần, bất kể bao nhiêu hàng được xóa khỏi bảng.

Các bộ kích khởi câu lệnh hữu dụng nếu mã trong hoạt động bộ kích khởi không phụ thuộc vào dữ liệu được câu lệnh kích khởi cung cấp hoặc các hàng được tác động. Thí dụ, nếu bộ kích khởi tạo một kiểm bảo mật phức tạp về thời gian hiện hành hoặc về người dùng hay nếu bộ kích khởi phát sinh một bản ghi kiểm soát đơn về kiểu của câu lệnh kích khởi, một bộ kích khởi được sử dụng.

Các bộ kích khởi before và after

Khi xác định một bộ kích khởi, bạn có thể ấn định thời gian bộ kích khởi xem hoạt động bộ kích khởi phải được tiến hành trước hoặc sau câu lệnh kích khởi. BEFORE và AFTER áp dụng vào cả các bộ kích khởi câu lệnh và hàng. (một kiểu khác của bộ kích khởi được mô tả ở "Các Bộ Kích Khởi INSTEAD OF").

Các bộ kích khởi BEFORE

Các bộ kích khởi BEFORE tiến hành hoạt động bộ kích khởi trước khi câu lệnh kích khởi được tiến hành. Kiểu bộ kích khởi này thường được sử dụng trong các trường hợp sau:

Khi hoạt động bộ kích khởi phải xác định xem câu lệnh kích khởi có được cho hoàn tất không, sử dụng bộ kích khởi BEFORE cho mục đích này, bạn có thể loại trừ tiến trình không cần thiết của câu lệnh kích khởi và hồi phục cuối cùng của nó trong trường hợp nơi mà một ngoại lệ phát sinh trong hoạt động của bộ kích khởi.
Để phái sinh những giá trị cột riêng biệt trước khi hoàn tất một câu lệnh kích khởi INSERT hoặc UPDATE.

Các bộ kích khởi AFTER

Các bộ kích khởi AFTER tiến hành hoạt động của bộ kích khởi sau khi câu lệnh kích khởi được tiến hành. Bộ kích khởi AFTER được sử dụng trong các trường hợp sau:

Khi bạn muốn có câu lệnh kích khởi để hoàn tất trước khi tiến hành hoạt động của bộ kích khởi.
Nếu bộ kích khởi BEFORE hiện hữu rồi, một bộ kích khởi AFTER có thể thực hiện các hoạt động khác nhau trên cùng câu lệnh kích khởi.

Các tổ hợp bộ kích khởi

Sử dụng các tùy chọn được liệt kê ở trên, bạn có thể tạo bốn kiểu bộ kích khởi sau:

Bộ kích khởi câu lệnh BEFORE
: Trước khi tiến hành câu lệnh kích khởi, hoạt động bộ kích khởi được tiến hành.

Bộ kích khởi hàng BEFORE
: Trước khi sửa đổi mỗi hàng được câu lệnh kích khởi tác động và trước khi kiểm các hạn định tính nguyên vẹn thích hợp, hoạt động bộ kích khởi được tiến hành với điều kiện hạn chế bộ kích khởi không bị xâm phạm.

Bộ kích khởi câu lệnh AFTER
: Sau khi tiến hành câu lệnh kích khởi và áp dụng bất kỳ hạn định tính nguyên vẹn bị trì hoãn, hoạt động bộ kích khởi được tiến hành.

Bộ kích khởi hàng AFTER
: Sau khi sửa đổi mỗi hàng được tác động bởi câu lệnh kích khởi và áp dụng các hạn định tính nguyên vẹn thích hợp có thể được, hoạt động bộ kích khởi được tiến hành cho hàng hiện hành với điều kiện hạn chế bộ kích khởi không bị xâm phạm.
: Không giống như bộ kích khởi hàng BEFORE, các bộ kích khởi hàng AFTER khóa các hàng.

Bạn có thể có nhiều bộ kích khởi cùng kiểu dữ liệu cho câu lệnh tương tự đối với bảng cho trước bất kỳ, thí dụ, bạn có thể có hai bộ kích khởi câu lệnh BEFORE cho các câu lệnh UPDATE trên bảng EMP. Nhiều bộ kích khởi cùng kiểu cho phép cài đặt môđun các ứng dụng có các bộ kích khởi trên cùng bảng. Hơn nữa, các sổ nhật ký ảnh chụp nhanh sử dụng các bộ kích khởi hàng AFTER, vì vậy bạn có thể thiết kế bộ kích khởi hàng AFTER riêng để bổ sung vào bộ kích khởi hàng AFTER Oracle xác định.

Bạn có thể tạo nhiều bộ kích khởi của các kiểu khác nhau tiến hành các kiểu khác nhau khi bạn cần cho mỗi kiểu câu lệnh DML (INSERT, UPDATE hoặc DELETE).

Thí dụ, giả sử bạn có một bảng, SAL và bạn muốn biết khi nào bảng ấy sẽ được xử lý và các kiểu truy vấn sẽ được phát hành. Thí dụ bên dưới có bộ phần mềm và bộ kích khởi mẫu truy tìm thông tin bằng giờ và kiểu hoạt động (thí dụ, UPDATE, DELETE hoặc INSERT) trên bảng SAL. Biến phiên toàn cục, STAT.ROWCNT được khởi hoạt thành zêrô do bộ kích khởi câu lệnh BEFORE. Sau đó nó được tăng lên mỗi lần bộ kích khởi hàng được tiến hành. Cuối cùng thông tin thống kê được lưu trữ trong bảng STAT_TAB bằng bộ kích khởi câu lệnh AFTER.

Bộ phần mềm và bộ kích khởi mẫu cho bảng SAL

```
DROP TABLE stat_tab;
CREATE TABLE stat_tab(utype CHAR(8),
    rowcnt INTEGER, uhour INTEGER);

CREATE OR REPLACE PACKAGE stat IS
 rowcnt INTEGER;
END;
```

```
CREATE TRIGGER bt BEFORE UPDATE OR DELETE OR
INSERT ON sal
BEGIN
 stat.rowcnt := 0;
END;
/

CREATE TRIGGER rt BEFORE UPDATE OR DELETE OR
INSERT ON sal
FOR EACH ROW BEGIN
 stat.rowcnt := stat.rowcnt + 1;
END;
/

CREATE TRIGGER at AFTER UPDATE OR DELETE OR
INSERT ON sal
DECLARE
 typ  CHAR(8);
 hour NUMBER;
BEGIN
 IF updating
 THEN typ := 'update'; END IF;
 IF deleting  THEN typ := 'delete'; END IF;
 IF inserting THEN typ := 'insert'; END IF;

 hour := TRUNC((SYSDATE - TRUNC(SYSDATE)) * 24);
 UPDATE stat_tab
   SET rowcnt = rowcnt + stat.rowcnt
  WHERE utype = typ
    AND uhour = hour;
 IF SQL%ROWCOUNT = 0 THEN
  INSERT INTO stat_tab VALUES (typ, stat.rowcnt, hour);
 END IF;

 EXCEPTION
```

```
 WHEN dup_val_on_index THEN
   UPDATE stat_tab
      SET rowcnt = rowcnt + stat.rowcnt
     WHERE utype = typ
       AND uhour = hour;
END;
```

Bộ kích khởi INSTEAD OF

Các bộ kích khởi INSTEAD OF cung cấp phương thức rõ ràng để sửa đổi các khung nhìn trực tiếp bằng các câu lệnh SQL DML (INSERT, UPDATE và DELETE). Các bộ kích khởi này được gọi là các bộ kích khởi INSTEAD OF vì, không giống như các kiểu bộ kích khởi khác, Oracle tiến hành bộ kích khởi thay vì tiến hành câu lệnh kích khởi. Bộ kích khởi cập nhật, chèn hoặc xóa trực tiếp trên các bảng cơ sở.

Bạn có thể viết các câu lệnh chuẩn INSERT, DELETE và UPDATE dựa trên khung nhìn và bộ kích khởi INSTEAD OF hoạt động ẩn trong nền để tạo các hoạt động chính xác xảy ra. Theo mặc định, các bộ kích khởi INSTEAD OF được kích hoạt cho mỗi hàng.

Sửa đổi các khung nhìn

Sửa đổi các khung nhìn có nhiều vấn đề vốn không rõ ràng.
Xóa hàng trong một khung nhìn có thể có nghĩa là xóa nó khỏi bảng cơ sở hoặc cập nhật một số giá trị cột để nó sẽ không còn phải lựa bằng khung nhìn.
Chèn một hàng vào khung nhìn có thể có nghĩa là chèn một hàng mới vào bảng cơ sở hoặc cập nhật một hàng có sẵn để nó sẽ được trình bày bằng khung nhìn.

Cập nhật một cột trong khung nhìn liên quan đến các liên kết có thể thay đổi ngữ nghĩa của các cột khác không được trình bày bằng khung nhìn.

Các khung nhìn đối tượng biểu thị thêm các vấn đề khác (xem "Các Khung Nhìn Đối Tượng"). Thí dụ, sử dụng khóa của các khung nhìn đối tượng để biểu thị các mối quan hệ chính/chi tiết. Điều này liên quan đến các liên kết, nhưng sửa đổi các liên kết vốn không rõ ràng.

Do những điều không rõ ràng này, có nhiều hạn chế mà các khung nhìn có thể sửa đổi được (xem phần kế tiếp). Bộ kích khởi INSTEAD OF có thể được sử dụng trên các khung nhìn đối tượng cũng như các khung nhìn quan hệ, trái lại chúng được sửa đổi.

Cơ chế của các bộ kích khởi INSTEAD OF còn cho phép bạn sửa đổi các môi trường làm việc khung nhìn đối tượng bên phía client tới OCI. Để sửa đổi một đối tượng được cụ thể hóa bằng một khung nhìn đối tượng trong cache đối tượng bên client và dọn nó vào vùng lưu trữ liên tục, bạn phải ấn định các bộ kích khởi INSTEAD OF, trừ khi khung nhìn đối tượng sửa đổi được. Tuy nhiên, nếu đối tượng là chỉ đọc, không cần phải xác định các bộ kích khởi để chốt nó.

Thông tin bổ sung:
Xem Oracle Call Interface Programmer's Guide.

Các khung nhìn không sửa đổi được

Một khung nhìn vốn sửa đổi được nếu nó không được chèn, cập nhật hoặc xóa mà không sử dụng bộ kích khởi INSTEAD OF và nếu nó thích hợp với các hạn chế được liệt kê ở dưới. Nếu truy vấn khung nhìn chứa bất kỳ cấu trúc sau, thì khung nhìn vốn không sửa đổi được và do đó bạn không thể thực hiện các lệnh chèn, cập nhật hoặc xóa trên khung nhìn:

Các toán tử tập hợp
Các hàm nhóm
Các mệnh đề GROUP BY, CONNECT BY hoặc START WITH
Toán tử DISTINCT
Các liên kết (tuy nhiên, một tập hợp con những khung nhìn liên kết có thể cập nhật - xem "Các Khung Nhìn Liên Kết Không Cập Nhật Được").

Nếu một khung nhìn có các cột giả hoặc các biểu thức, bạn chỉ có thể cập nhật với câu lệnh UPDATE không tham chiếu với bất kỳ một cột giả hoặc biểu thức nào.

Thí dụ về bộ kích khởi INSTEAD OF

Thí dụ sau trình bày bộ kích khởi INSTEAD OF để chèn cửa sổ hàng vào khung nhìn MANAGER_INFO.

```
CREATE VIEW manager_info AS
 SELECT e.name, e.empno, d.dept_type, d.deptno, p.level,
     p.projno
  FROM   emp e, dept d, project p
  WHERE  e.empno =  d.mgr_no
  AND    d.deptno = p.resp_dept;

CREATE TRIGGER manager_info_insert
 INSTEAD OF INSERT ON manager_info
 REFERENCING NEW AS n      -- new manager information

 FOR EACH ROW
 BEGIN
  IF NOT EXISTS SELECT * FROM emp
      WHERE emp.empno = :n.empno
  THEN
      INSERT INTO emp
         VALUES(:n.empno, :n.name);
  ELSE
      UPDATE emp SET emp.name = :n.name
         WHERE emp.empno = :n.empno;
   END IF;

   IF NOT EXISTS SELECT * FROM dept
```

```
      WHERE dept.deptno = :n.deptno
   THEN
      INSERT INTO dept
         VALUES(:n.deptno, :n.dept_type);
   ELSE
      UPDATE dept SET dept.dept_type = :n.dept_type
         WHERE dept.deptno = :n.deptno;
   END IF;

   IF NOT EXISTS SELECT * FROM project
      WHERE project.projno = :n.projno
   THEN
      INSERT INTO project
         VALUES(:n.projno, :n.project_level);
   ELSE
      UPDATE project SET project.level = :n.level
         WHERE project.projno = :n.projno;
   END IF;
END;
```

Các hoạt động được trình bày cho các hàng được chèn vào trắc nghiệm đầu tiên khung nhìn MANAGER_INFO để xem các hàng thích hợp có tồn tại trong các bảng của MANAGER_INFO phải sinh không. Sau đó các hoạt động chèn các hàng mới hoặc cập nhật các hàng có sẵn cho thích hợp. Các bộ kích khởi tương tự có thể xác định các hoạt động thích hợp cho UPDATE và DELETE.

Thông tin bổ sung:
Xem câu lệnh CREATE TRIGGER trong Oracle8 SQL Reference về các bộ kích khởi INSTEAD OF.

Sự tiến hành của bộ kích khởi

Bộ kích khởi có thể ở một trong hai chế độ riêng biệt:

Được cho phép	Bộ kích khởi được cho phép tiến hành hoạt động bộ kích khởi của nó nếu câu lệnh kích khởi được phát hành và hạn chế bộ kích khởi (nếu có) thẩm định là TRUE.

Vô hiệu hóa	Bộ kích khởi được vô hiệu hóa không tiến hành hoạt động bộ kích khởi của nó, dù câu lệnh kích khởi được phát hành và hạn chế bộ kích khởi (nếu có) sẽ thẩm định là TRUE.

Đối với các bộ kích khởi, Oracle tự động:

- tiến hành các bộ kích khởi của mỗi kiểu trong chuỗi tiến hành được lập kế hoạch khi nhiều bộ kích khởi được tiến hành bằng một câu lệnh SQL đơn

- thực hiện hạn định tính nguyên vẹn dữ liệu bằng cách kiểm giờ và ngày tháng xác lập liên quan đến các kiểu của các bộ kích khởi khác và bảo đảm các bộ kích khởi không thể làm ảnh hưởng đến các hạn định tính nguyên vẹn dữ liệu

- cung cấp các khung nhìn đọc nhất quán cho các truy vấn và hạn định

- quản lý các phụ thuộc giữa các bộ kích khởi và các đối tượng giản đồ được tham chiếu bằng mã của hoạt động bộ kích khởi

- sử dụng cam kết hai pha nếu bộ kích khởi cập nhật các bảng từ xa trong một cơ sở dữ liệu được phân phối

- tiến hành nhiều bộ kích khởi theo thứ tự không được ấn định, nếu nhiều bộ kích khởi của cùng kiểu tồn tại cho một câu lệnh cho trước.

Mô hình tiến hành cho các bộ kích khởi và kiểm hạn định tính nguyên vẹn dữ liệu

Câu lệnh SQL có thể tiến hành tối đa bốn kiểu bộ kích khởi tiềm năng: các bộ kích khởi hàng BEFORE, bộ kích khởi câu lệnh BEFORE, các bộ kích khởi hàng AFTER và các bộ kích khởi câu lệnh AFTER. Câu lệnh kích khởi hoặc một câu lệnh trong một bộ kích khởi tạo cho một hoặc nhiều hạn định tính nguyên vẹn dữ liệu được kiểm. Ngoài ra, các bộ kích khởi có thể chứa các câu lệnh tạo cho các bộ kích khởi khác tiến hành (các bộ kích khởi xếp tầng).

Oracle sử dụng mô hình tiến hành sau để duy trì chuỗi tiến hành thích hợp của nhiều bộ kích khởi và kiểm hạn định:

1. Tiến hành tất cả các bộ kích khởi câu lệnh BEFORE để áp dụng vào câu lệnh ấy.
2. Vòng lặp cho mỗi hàng được câu lệnh SQL tác động.
    a. Tiến hành tất cả các bộ kích khởi hàng BEFORE để áp dụng vào câu lệnh ấy.
    b. Khóa và thay đổi hàng, thực hiện kiểm hạn định tính nguyên vẹn dữ liệu. (khóa không được giải phóng cho tới khi giao dịch được cam kết).
    c. Tiến hành tất cả các bộ kích khởi hàng AFTER để áp dụng vào câu lệnh ấy.
3. Hoàn tất kiểm hạn định tính nguyên vẹn dữ liệu bị trì hoãn.
4. Tiến hành các bộ kích khởi câu lệnh AFTER để áp dụng vào câu lệnh ấy.

Định nghĩa của mô hình tiến hành là tái quy. Thí dụ, một câu lệnh SQL có thể tạo ra một bộ kích khởi hàng BEFORE để tiến hành và một kiểm hạn định tính nguyên vẹn dữ liệu được kiểm. Bộ kích khởi hàng BEFORE đó tới lượt có thể thực hiện một cập nhật để tạo một hạn định tính nguyên vẹn được kiểm và bộ kích khởi câu lệnh AFTER được tiến hành. Bộ kích khởi câu lệnh

AFTER tạo ra một hạn định tính nguyên vẹn dữ liệu được kiểm. Trong trường hợp này, mô hình tiến hành tiến hành các Step tái quy như sau:

1 Original SQL statement issued.

2. BEFORE row triggers fired.

3. AFTER statement triggers fired by UPDATE in BEFORE row trigger.

4. Statements of AFTER statement triggers executed.

5. Integrity constraint checked on tables changed by AFTER statement triggers.

6. Statements of BEFORE row triggers executed.

7. Integrity constraint checked on tables changed by BEFORE row triggers.

8. SQL statement executed.

9 Integrity constraint from SQL statement checked.

Một tính chất quan trọng của mô hình tiến hành là tất cả những hoạt động và kiểm tra được thực hiện do câu lệnh SQL phải thành công. Nếu một ngoại lệ được phát sinh trong bộ kích khởi và ngoại lệ không được điều quản rõ ràng, thì tất cả hoạt động được thực hiện do câu lệnh SQL ban đầu, kể cả những hoạt động được thực hiện bởi các bộ kích khởi được tiến hành hồi phục về trạng thái cũ. Do đó, các hạn định tính nguyên vẹn không thể được dàn xếp bởi các bộ kích khởi. Mô hình tiến hành chú ý tới các hạn định tính nguyên vẹn dữ liệu và không cho phép các bộ kích khởi xâm phạm các hạn định tính nguyên vẹn khai báo.

Thí dụ, trong viễn tưởng được phác thảo trước, giả sử các Step 1 tới 8 thành công; tuy nhiên, trong Step 9 hạn định tính nguyên vẹn bị xâm phạm. Do sự xâm phạm này, tất cả các thay đổi xảy

ra do câu lệnh SQL (trong Step 8), bộ kích khởi hàng BEFORE được tiến hành (trong Step 6) và bộ kích khởi câu lệnh AFTER (trong Step 4) được hồi phục về trạng thái cũ.

Lưu ý :
Nhận biết các bộ kích khởi của các kiểu khác nhau được tiến hành theo thứ tự riêng. Tuy nhiên, các bộ kích khởi của cùng kiểu dữ liệu đối với câu lệnh tương tự không được bảo đảm tiến hành theo bất kỳ thứ tự riêng biệt nào. Thí dụ, tất cả các bộ kích khởi hàng BEFORE đối với câu lệnh UPDATE đơn không thể luôn luôn tiến hành theo cùng thứ tự. Thiết kế các ứng dụng, vì vậy chúng không dựa vào thứ tự tiến hành nhiều bộ kích khởi của cùng kiểu.

Truy xuất dữ liệu đối với các bộ kích khởi

Khi một bộ kích khởi được tiến hành, các bảng được tham chiếu trong hoạt động bộ kích khởi có thể hiện đang trải qua các thay đổi bởi các câu lệnh SQL trong các giao dịch của những người dùng khác. Trong tất cả các trường hợp, các câu lệnh SQL được tiến hành trong các bộ kích khởi theo sau các nguyên tắc chung được sử dụng cho các câu lệnh SQL độc lập. Đặc biệt, nếu một giao dịch không được cam kết đã sửa đổi những giá trị mà bộ kích khởi đang được tiến hành các yêu cầu đọc (truy vấn) hoặc ghi (cập nhật), các câu lệnh SQL trong thân của bộ kích khởi đang được tiến hành sử dụng các nguyên tắc chủ đạo sau:

- Các truy vấn nhận biết ảnh chụp nhanh hiện được đọc nhất quán của các bảng được tham chiếu và bất kỳ dữ liệu thay đổi trong cùng giao dịch.
- Các cập nhật chờ các khóa dữ liệu có sẵn để được giải phóng trước khi tiến hành.

Các thí dụ sau minh họa các điểm này.

Thí dụ:

Giả sử bộ kích khởi (thân) SALARY_CHECK có câu lệnh SELECT sau:

```
SELECT minsal, maxsal INTO minsal, maxsal
  FROM salgrade
  WHERE job classification = :new.job_classification;
```

Đối với thí dụ này, giả sử giao dịch T1 có một cập nhật cột MAXSAL của bảng SALGRADE. Ở điểm này, bộ kích khởi SALARY_CHECK được tiến hành bởi câu lệnh trong giao dịch T2. Câu lệnh SELECT trong bộ kích khởi được tiến hành (phát xuất từ T2) không hiểu cập nhật ấy bởi giao dịch T1 không được cam kết và truy vấn trong bộ kích khởi ấy cho trở lại giá trị MAXSAL cũ thuộc điểm đọc nhất quán đối với giao dịch T2.

Thí dụ:

Giả sử định nghĩa sau của bộ kích khởi TOTAL_SALARY, bộ kích khởi duy trì cột được phái sinh lưu trữ tổng số lương của tất cả nhân viên trong bộ phận:

```
CREATE TRIGGER total_salary
AFTER DELETE OR INSERT OR UPDATE OF deptno, sal ON emp
 FOR EACH ROW BEGIN
  /* assume that DEPTNO and SAL are non-null fields */
   IF DELETING OR (UPDATING AND :old.deptno !:new.deptno)
   THEN UPDATE dept
  SET total_sal = total_sal - :old.sal
  WHERE deptno = :old.deptno;
   END IF;
   IF INSERTING OR (UPDATING AND :old.deptno != :new.deptno)
   THEN UPDATE dept
    SET total_sal = total_sal + :new.sal
    WHERE deptno = :new.deptno;
   END IF;
   IF (UPDATING AND :old.deptno = :new.deptno AND
    :old.sal != :new.sal )
   THEN UPDATE dept
    SET total_sal = total_sal - :old.sal + :new.sal
   WHERE deptno = :new.deptno;
```

```
END IF;
END;
```

Đối với thí dụ này, giả sử giao dịch không được cam kết của một người dùng có cập nhật cột TOTAL_SAL của hàng trong bảng DEPT. Ở điểm này, bộ kích khởi TOTAL_SALARY được tiến hành bởi một câu lệnh SQL của người dùng thứ hai. Bởi vì giao dịch không được cam kết của người dùng thứ nhất có cập nhật giá trị thích hợp được tiến hành trong cột TOTAL_SAL (nói cách khác, một khóa hàng đang được lưu giữ), các cập nhật được thực hiện bởi bộ kích khởi TOTAL_SALARY không được tiến hành cho tới khi giao dịch lưu giữ khóa hàng ấy được cam kết hoặc hồi phục về trạng thái cũ. Do đó, người dùng thứ hai chờ cho tới điểm cam kết hoặc hồi phục giao dịch của người dùng thứ nhất.

Kho lưu trữ các bộ kích khởi

Oracle lưu trữ các bộ kích khởi ở biểu mẫu được biên dịch của chúng, giống như các thủ tục định sẵn. Khi câu lệnh CREATE TRIGGER cam kết, mã PL/SQL được gọi là mã P (đối với mã giả) được lưu trữ trong cơ sở dữ liệu và mã nguồn của bộ kích khởi được xóa sạch khỏi vùng chia dùng chung.

Để có thêm thông tin về biên dịch và lưu trữ mã PL/SQL, xem "Oracle Lưu Trữ Các Thủ Tục Và Các Bộ Phần Mềm Như Thế Nào".

Sự tiến hành của các bộ kích khởi

Oracle tiến hành bộ kích khởi bên trong bằng cách sử dụng các Step tương tự được sử dụng cho sự tiến hành thủ tục. Chỉ có sự khác biệt tinh vi là người dùng có quyền tiến hành bộ kích khởi nếu họ có đặc quyền tiến hành câu lệnh kích khởi. Thay vì điều này, các bộ kích khởi được thẩm định và được tiến hành bằng cách tương tự như các thủ tục định sẵn.

Xem thêm "Oracle Lưu Trữ Các Thủ Tục Và Các Bộ Phần Mềm Như Thế Nào".

Duy trì tính phụ thuộc cho các bộ kích khởi

Giống như các thủ tục, các bộ kích khởi phụ thuộc vào các đối tượng được tham chiếu. Oracle tự động quản lý các phụ thuộc của bộ kích khởi trên các đối tượng giản đồ được tham chiếu trong hoạt động bộ kích khởi của nó. Các vấn đề phụ thuộc đối với các bộ kích khởi giống như những phụ thuộc đối với các thủ tục định sẵn. Các bộ kích khởi được coi giống như các thủ tục định sẵn; chúng được chèn vào từ điển dữ liệu.

Chương 19

Quản lý sự phụ thuộc Oracle

Các định nghĩa của một số đối tượng, kể cả các khung nhìn và các thủ tục, tham chiếu các đối tượng khác, chẳng hạn như các bảng. Do đó các đối tượng được xác định phụ thuộc vào các đối tượng được tham chiếu bằng các định nghĩa của chúng. Chương này đề cập đến các sự phụ thuộc giữa các đối tượng giản đồ và Oracle tự động truy tìm và quản lý các sự phụ thuộc này như thế nào. Nó gồm:

- **Giới thiệu các vấn đề phụ thuộc**
- **Giải quyết các sự phụ thuộc đối tượng giản đồ**
- **Quản lý sự phụ thuộc và các đối tượng giản đồ không tồn tại**
- **Quản lý sự phụ thuộc SQL dùng chung**

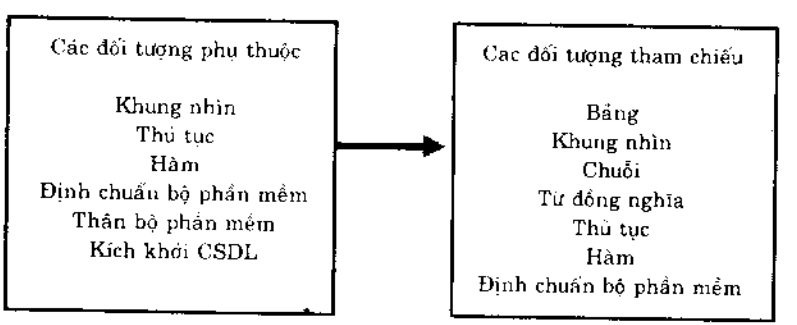
Thông tin bổ sung:

Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle về những phụ thuộc đối tượng giản đồ trong môi trường đó.

Giới thiệu các vấn đề phụ thuộc

Một số kiểu đối tượng giản đồ có thể tham chiếu các đối tượng khác là thành phần của định nghĩa của chúng. Thí dụ, một khung nhìn được xác định bằng một truy vấn tham chiếu các bảng hoặc các khung nhìn khác, một thân của thủ tục có thể có các câu lệnh SQL tham chiếu các đối tượng cơ sở dữ liệu khác. Một đối tượng tham chiếu một đối tượng khác là thành phần của định nghĩa của nó được gọi là đối tượng phụ thuộc, khi đối tượng ấy đang được tham chiếu là một đối tượng tham chiếu. Hình 19-1 minh họa các kiểu khác nhau của những đối tượng phụ thuộc và tham chiếu.

Hình 19-1 Các kiểu đối tượng giản đồ có thể phụ thuộc và tham chiếu



Nếu bạn thay đổi định nghĩa của đối tượng tham chiếu, các đối tượng phụ thuộc có thể hoặc không thể tiếp tục thực hiện chức năng mà không bị lỗi, phụ thuộc vào kiểu thay thế. Thí dụ, nếu bạn bỏ một bảng, khung nhìn không dựa vào bảng được bỏ có thể được sử dụng.

Oracle tự động ghi các sự phụ thuộc giữa các đối tượng để bỏ bớt tác vụ phức tạp quản lý phụ thuộc cho người điều hành cơ sở dữ liệu và các người dùng. Thí dụ, nếu bạn thay đổi một bảng mà phụ thuộc nhiều thủ tục định sẵn thì Oracle tự động biên dịch lại

các thủ tục phụ thuộc, lần kế tiếp các thủ tục ấy được tham chiếu (được tiến hành hoặc được biên dịch dựa vào đó).

Để quản lý những phụ thuộc giữa các đối tượng giản đồ, tất cả những đối tượng giản đồ trong cơ sở dữ liệu có trạng thái:

VALID	Đối tượng đã được biên dịch và có thể sử dụng ngay khi được tham chiếu. Đối tượng phải được biên dịch trước khi nó có thể được sử dụng. Trong trường hợp các thủ tục, các hàm và các bộ phần mềm, điều này có nghĩa biên dịch đối tượng ấy. Trong trường hợp của các khung nhìn, điều này có nghĩa khung nhìn phải được phân ngữ lại, sử dụng định nghĩa hiện hành trong từ điển dữ liệu. Chỉ các đối tượng phụ thuộc có thể bất hợp lệ; các bảng, các chuỗi và các từ đồng nghĩa luôn luôn hợp lệ.
INVALID	Nếu khung nhìn, thủ tục, hàm hoặc bộ phần mềm bất hợp lệ, Oracle có thể đã cố biên dịch nó, nhưng các lỗi liên quan đến đối tượng ấy đã xảy ra. Thí dụ, khi biên dịch một khung nhìn, một trong các bảng cơ sở của nó không tồn tại hoặc các đặc quyền hiện hành cho bảng cơ sở ấy không có thể hiện hữu. Khi biên dịch một bộ phần mềm, có thể là lỗi cú pháp PL/SQL hoặc SQL hay các đặc quyền điều chỉnh đối tượng được tham chiếu có thể không hiệu hữu. Các đối tượng có các vấn đề như thế luôn luôn bất hợp lệ.

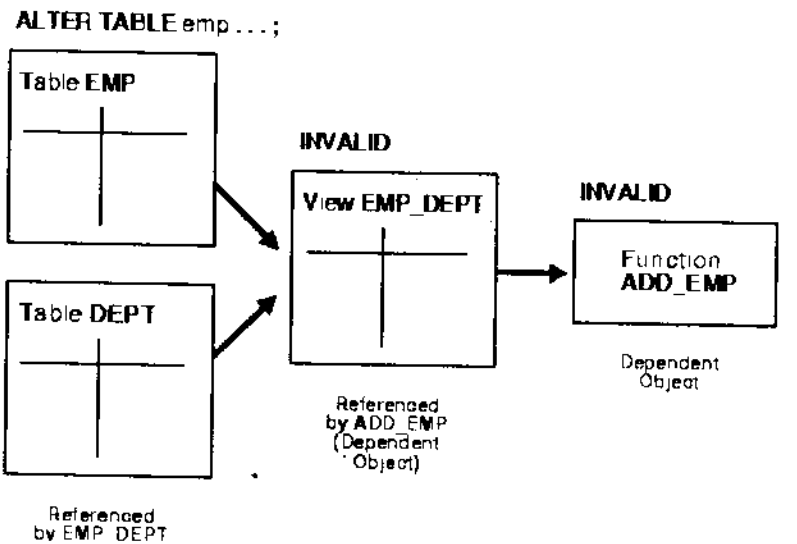
Oracle tự động truy tìm các thay đổi riêng biệt trong cơ sở dữ liệu và ghi trạng thái thích hợp cho các đối tượng liên quan trong từ điển dữ liệu.

Ghi trạng thái là một tiến trình tái quy; bất kỳ thay đổi trong trạng thái của đối tượng được tham chiếu không những thay đổi

trạng thái ấy đối với các đối tượng phụ thuộc trực tiếp mà còn đối với các đối tượng phụ thuộc gián tiếp.

Thí dụ, xét thủ tục định sẵn tham chiếu trực tiếp một khung nhìn. Thực vậy, thủ tục định sẵn tham chiếu gián tiếp (các) bảng cơ sở của khung nhìn đó. Vì thế, nếu bạn thay đổi một bảng cơ sở, thì khung nhìn bị mất hiệu lực, sau đó làm mất hiệu lực thông tin được lưu trữ. Hình 19-2 minh họa điều này.

Hình 19-2 Các phụ thuộc gián tiếp



Giải quyết những phụ thuộc đối tượng giản đồ

Khi một đối tượng giản đồ được tham chiếu (trực tiếp trong câu lệnh SQL hoặc gián tiếp bằng một tham chiếu đối tượng phụ thuộc), Oracle kiểm trạng thái của đối tượng được xác định rõ ràng trong câu lệnh SQL và các đối tượng tham chiếu bất kỳ nếu cần thiết. Hoạt động của Oracle phụ thuộc vào trạng thái của các đối tượng được tham chiếu trực tiếp và gián tiếp trong câu lệnh SQL

Nếu mọi đối tượng được tham chiếu hợp lệ, Oracle tiến hành câu lệnh SQL ngay mà không có hoạt động nào bổ sung.
Nếu khung nhìn được tham chiếu hoặc thủ tục (kể cả các hàm và các bộ phần mềm) không hợp lệ, Oracle tự động cố biên dịch đối tượng ấy.

Nếu tất cả đối tượng tham chiếu không hợp lệ có thể được biên dịch thành công, thì các đối tượng ấy được biên dịch và câu lệnh SQL tiến hành thành công.

Nếu đối tượng không hợp lệ không thể được biên dịch thành công, thì đối tượng ấy vẫn không hợp lệ, được cho lại một lỗi và giao dịch có câu lệnh SQL được hồi phục về trạng thái cũ.

Lưu ý:
Oracle cố biên dịch lại một đối tượng không hợp lệ một cách linh động chỉ nếu nó đã được thay thế cho tới khi nó được phát hiện là không hợp lệ. Sự tối ưu hóa này loại trừ những biên dịch lại không cần thiết.

Biên dịch các khung nhìn và các đơn vị chương trình PL/SQL

Khung nhìn hoặc đơn vị chương trình PL/SQL có thể được biên dịch và được thực hiện hợp lệ nếu thỏa mãn các điều kiện sau:

Định nghĩa của khung nhìn hoặc đơn vị chương trình phải chính xác; tất cả các câu lệnh SQL và PL/SQL phải có cấu trúc thích hợp.

Tất cả các đối tượng được tham chiếu phải hiện hữu và thuộc cấu trúc được yêu cầu. Thí dụ, nếu truy vấn xác định của một khung nhìn có cột và cột ấy phải hiện hữu trong bảng cơ sở.

Owner (chủ nhân) của khung nhìn và đơn vị chương trình phải có các đặc quyền cần thiết về các đối tượng tham chiếu. Thí dụ, nếu câu lệnh SQL trong một thủ tục chèn một hàng vào bảng, thì chủ nhân của thủ tục ấy phải có đặc quyền INSERT privilege cho bảng được tham chiếu đó.

Các khung nhìn và các bảng cơ sở

Khung nhìn phụ thuộc vào các bảng cơ sở (hoặc các khung nhìn) được tham chiếu trong truy vấn xác định của nó. Nếu truy vấn xác định của một khung nhìn không rõ ràng những cột nào được tham chiếu, thí dụ, bảng SELECT * FROM, truy vấn xác định được mở rộng khi được lưu trữ trong từ điển dữ liệu để có tất cả các cột trong bảng cơ sở tham chiếu vào thời gian đó.

Nếu một bảng cơ sở (hoặc khung nhìn) thuộc một khung nhìn bị thay đổi, đặt tên lại hoặc bị bỏ, thì khung nhìn không hợp lệ nhưng định nghĩa của nó vẫn ở trong từ điển dữ liệu cùng với các đặc quyền, các từ đồng nghĩa, các đối tượng khác và các khung nhìn khác tham chiếu khung nhìn không hợp lệ.

Một cố gắng sử dụng khung nhìn không hợp lệ tự động làm cho Oracle biên dịch lại khung nhìn ấy một cách linh động. Sau khi thay thế khung nhìn, khung nhìn ấy có thể hợp lệ hoặc không hợp lệ, tùy theo điều kiện sau:

Tất cả các bảng cơ sở được truy vấn xác định của một khung nhìn tham chiếu phải tồn tại. Nếu bảng cơ sở của một khung nhìn được đặt tên lại hoặc bỏ, thì khung nhìn ấy không hợp lệ và không thể được sử dụng. Các tham chiếu những khung nhìn không hợp lệ làm cho câu lệnh tham chiếu không thực hiện được. Khung nhìn ấy có thể được biên dịch lại chỉ nếu bảng cơ sở được đặt tên lại thành tên ban đầu hoặc bảng cơ sở được tạo lại.

Nếu một bảng cơ sở bị thay đổi hoặc được tạo lại với các cột giống nhau, nhưng kiểu dữ liệu của một hoặc nhiều cột trong bảng cơ sở bị thay đổi, bất kỳ khung nhìn phụ thuộc có thể được biên dịch lại thành công.

Nếu bảng cơ sở của khung nhìn bị thay đổi hoặc được tạo lại có ít nhất các tập hợp cột tương tự, thì khung nhìn ấy có thể hợp lệ. Khung nhìn ấy không thể hợp lệ nếu bảng cơ sở được tạo lại có các cột mới và khung nhìn tham chiếu các cột không

còn được chứa trong bảng được tạo lại. Điểm thứ hai đặc biệt thích hợp trong trường hợp các khung nhìn được xác định với truy vấn bảng SELECT * FROM, vì truy vấn xác định được mở rộng vào thời gian tạo khung nhìn và được lưu trữ vĩnh viễn trong từ điển dữ liệu.

Các đơn vị chương trình và các đối tượng tham chiếu

Oracle tự động làm mất hiệu lực một đơn vị chương trình khi định nghĩa đối tượng được tham chiếu bị thay đổi. Thí dụ, giả sử thủ tục độc lập có nhiều câu lệnh tham chiếu một bảng, khung nhìn, thủ tục độc lập khác và một thủ tục bộ phần mềm chung. Trong trường hợp đó, các điều kiện sau lưu giữ:

- Nếu bảng được tham chiếu bị thay đổi, thì thủ tục phụ thuộc không hợp lệ.
- Nếu bảng cơ sở của khung nhìn được tham chiếu bị thay thế, thì khung nhìn và thủ tục phụ thuộc không hợp lệ.
- Nếu thủ tục độc lập được tham chiếu bị thay thế, thì thủ tục phụ thuộc không hợp lệ.
- Nếu thân của bộ phần mềm được tham chiếu bị thay thế, thì thủ tục phụ thuộc không bị ảnh hưởng. Tuy nhiên, nếu định chuẩn của bộ phần mềm được tham chiếu được thay thế, thì thủ tục phụ thuộc không hợp lệ.

Trường hợp cuối cùng phát hiện cơ chế giảm thiểu tối đa các phụ thuộc trong các thủ tục và các đối tượng được tham chiếu bằng cách sử dụng các bộ phần mềm.

Trạng thái phiên và các bộ phần mềm được tham chiếu

Mỗi phiên tham chiếu một cấu trúc bộ phần mềm có riêng môi trường làm việc của bộ phần mềm đó, kể cả trạng thái liên tục của các biến, con trỏ và các hằng chung và riêng. Tất cả môi trường làm việc bộ phần mềm của phiên (kể cả trạng thái) có

thể bị mất nếu bất kỳ các bộ phần mềm được lập môi trường làm việc của phiên (định chuẩn hoặc thân) kế tiếp bất hợp lệ và được biên dịch lại.

Các quyền hạn bảo mật

Oracle thông báo cho biết khi một đối tượng DML hoặc đặc quyền hệ thống được cho hoặc được gọi từ một người dùng hay PUBLIC tự động vô hiệu hóa tất cả các đối tượng phụ thuộc của chủ nhân. Oracle vô hiệu hóa các đối tượng phụ thuộc để xác minh chủ nhân của đối tượng phụ thuộc tiếp tục có các đặc quyền cần thiết về tất cả những đối tượng được tham chiếu. Về nội bộ, Oracle lưu ý các đối tượng như thế không phải "biên dịch lại"; chỉ các quyền hạn bảo mật cần được hữu hiệu hóa, không phải cấu trúc của những đối tượng bất kỳ. Sự tối ưu hóa này loại trừ những biên dịch không cần thiết để tránh phải thay đổi dấu thời gian của đối tượng phụ thuộc.

Thông tin bổ sung:
Để có thông tin biên dịch lại bắt buộc của một khung nhìn không hợp lệ hoặc đơn vị chương trình, xem Oracle8 Application Developer's Guide. Nếu bạn đang sử dụng Trusted Oracle, xem thêm tài liệu Trusted Oracle.

Quản lý phụ thuộc và các đối tượng không phụ thuộc

Khi một đối tượng phụ thuộc được tạo, Oracle cố giải quyết tất cả các tham chiếu trước hết tìm thấy trong giản đồ hiện hành. Nếu đối tượng được tham chiếu không tìm thấy trong giản đồ hiện hành ấy, Oracle cố gắng giải quyết tham chiếu ấy bằng cách tìm từ đồng nghĩa riêng trong cùng giản đồ. Nếu từ đồng nghĩa riêng không tìm thấy, Oracle di tiếp tục tìm một từ đồng nghĩa chung. Nếu từ đồng nghĩa chung không tìm thấy, Oracle tìm một tên giản đồ để so khớp phần đầu tiên của tên đối tượng. Nếu tên

gián đồ so khớp được tìm thấy, Oracle cố tìm đối tượng trong gián đồ đó. Nếu không tìm thấy gián đồ, được cho lại một lỗi.

Bởi vì Oracle giải quyết các tham chiếu, một đối tượng có thể phụ thuộc vào sự không tồn tại của các đối tượng khác. Điều này xảy ra khi đối tượng phụ thuộc sử dụng tham chiếu sẽ được diễn dịch khác nhau là một đối tượng hiện hữu khác. Thí dụ, giả sử như sau:

Vào thời điểm của ngày tháng hiện hành, gián đồ COMPANY có một bảng có tên EMP.

Từ đồng nghĩa PUBLIC có tên EMP được tạo cho COMPANY.EMP và đặc quyền SELECT đối với COMPANY.EMP được cho vai trò PUBLIC.

Gián đồ JWARD không có bảng hoặc từ đồng nghĩa riêng có tên EMP.

Người dùng JWARD tạo một khung nhìn trong gián đồ của anh ta với câu lệnh sau:

```
CREATE VIEW dept_salaries AS
  SELECT deptno, MIN(sal), AVG(sal), MAX(sal) FROM
emp
  GROUP BY deptno
  ORDER BY deptno;
```

Khi JWARD tạo khung nhìn DEPT_SALARIES, thì tham chiếu EMP được giải quyết, trước hết tìm JWARD.EMP là một bảng, khung nhìn hoặc từ đồng nghĩa riêng không được tìm thấy, thì từ đồng nghĩa chung có tên EMP được tìm thấy. Do đó, Oracle lưu ý JWARD.DEPT_SALARIES phụ thuộc vào sự không tồn tại của JWARD.EMP và sự tồn tại của PUBLIC.EMP.

Bây giờ giả sử JWARD quyết định tạo một khung nhìn mới có tên EMP trong gián đồ của anh ta sử dụng câu lệnh sau:

```
CREATE VIEW emp AS
  SELECT empno, ename, mgr, deptno
  FROM company.emp;
```

Lưu ý JWARD.EMP không có cùng cấu trúc như COMPANY.EMP.

Vì cố giải quyết các tham chiếu bằng những định nghĩa đối tượng, bên trong Oracle lưu ý các phụ thuộc đối tượng phụ thuộc mới có những đối tượng "không tồn tại" - các đối tượng giản đồ, nếu chúng tồn tại, sẽ thay đổi sự diễn dịch của định nghĩa đối tượng ấy. Những phụ thuộc như thế phải được lưu ý trong trường hợp đối tượng không tồn tại được tạo ra về sau, tất cả đối tượng phụ thuộc phải được vô hiệu hóa để các đối tượng phụ thuộc có thể được biên dịch lại và được xác minh.

Do đó, trong thí dụ trên, khi JWARD.EMP được tạo, JWARD.DEPT_SALARIES được vô hiệu hóa vì nó phụ thuộc vào JWARD.EMP. Sau đó khi JWARD.DEPT_SALARIES được sử dụng, Oracle cố biên dịch lại khung nhìn ấy. Khi Oracle giải quyết tham chiếu với EMP, nó tìm thấy JWARD.EMP (PUBLIC.EMP không còn là đối tượng được tham chiếu). Bởi vì JWARD.EMP không có cột SAL, Oracle tìm thấy các lỗi khi thay thế khung nhìn để bất hợp lệ.

Nói tóm lại, những sự phụ thuộc vào các đối tượng không tồn tại được kiểm khi giải quyết đối tượng phải được quản lý trong trường hợp đối tượng không tồn tại được tạo ra sau.

Quản lý sự phụ thuộc SQL dùng chung

Hơn nữa để quản lý các sự phụ thuộc trong các đối tượng giản đồ, Oracle còn quản lý các sự phụ thuộc của mỗi vùng SQL dùng chung trong vùng chia dùng chung. Nếu bảng, khung nhìn, từ đồng nghĩa hoặc chuỗi được tạo, thay thế hoặc bỏ, hay một thủ tục hoặc định chuẩn phần mềm được biên dịch lại, thì tất cả những vùng SQL dùng chung phụ thuộc được vô hiệu hóa. Vào lúc tiến hành tiếp theo của con trỏ tương ứng với vùng SQL dùng

chung không hợp lệ, Oracle phân ngữ lại câu lệnh SQL để tái phát sinh vùng SQL dùng chung.

Quản lý sự phụ thuộc cục bộ và từ xa

Truy tìm các sự phụ thuộc và hoàn tất các biên dịch lại cần thiết được Oracle tự động thực hiện. Trong trường hợp đơn giản nhất, Oracle phải quản lý những sự phụ thuộc giữa các đối tượng trong một cơ sở dữ liệu đơn (quản lý sự phụ thuộc cục bộ). Thí dụ, một câu lệnh trong một thủ tục có thể tham chiếu một bảng trong cùng cơ sở dữ liệu. Trong những hệ thống phức tạp hơn, Oracle phải quản lý các sự phụ thuộc trong những môi trường được phân phối qua mạng (quản lý sự phụ thuộc từ xa). Thí dụ, một bộ kích khởi Oracle Forms có thể phụ thuộc vào đối tượng giản đồ trong cơ sở dữ liệu. Trong cơ sở dữ liệu được phân phối, truy vấn xác định của khung nhìn cục bộ có thể tham chiếu một bảng từ xa.

Quản lý các sự phụ thuộc cục bộ

Oracle quản lý tất cả các sự phụ thuộc cục bộ sử dụng bảng "phụ thuộc" của cơ sở dữ liệu để theo dõi các đối tượng phụ thuộc của mỗi đối tượng giản đồ. Khi đối tượng được tham chiếu bị sửa đổi, Oracle sử dụng bảng phụ thuộc để nhận dạng các đối tượng phụ thuộc, sau đó chúng được vô hiệu hóa. Thí dụ, giả sử thủ tục định sẵn UPDATE_SAL tham chiếu bảng JWARD.EMP. Nếu định nghĩa của bảng bị thay thế bằng cách nào đó, thì trạng thái của mọi đối tượng tham chiếu JWARD.EMP được đổi thành INVALID, kể cả thủ tục định sẵn UPDATE_SAL. Do đó, thủ tục không thể được tiến hành cho tới khi nó được biên dịch lại và hợp lệ. Tương tự, khi đặc quyền DML được thu hồi từ một người dùng, mọi đối tượng phụ thuộc trong giản đồ của người dùng được vô hiệu hóa. Tuy nhiên đối tượng bất hợp lệ vì quyền hạn được thu hồi có thể được thẩm định lại bằng "cho quyền hạn lại" trong trường hợp nó không đòi hỏi biên dịch lại toàn bộ.

Quản lý các sự phụ thuộc từ xa

Các phụ thuộc ứng dụng cơ sở dữ liệu và cơ sở dữ liệu phân phối cũng phải được quản lý. Thí dụ, một ứng dụng Oracle Forms có thể có bộ kích khởi tham chiếu bảng hoặc thủ tục định sẵn cục bộ có thể gọi một thủ tục từ xa trong hệ thống cơ sở dữ liệu được phân phối. Hệ thống cơ sở dữ liệu phải giải thích các sự phụ thuộc trong các đối tượng như thế. Oracle sử dụng các cơ chế khác nhau để quản lý những phụ thuộc từ xa tùy theo các đối tượng có liên quan.

Các sự phụ thuộc trong các thủ tục cơ sở dữ liệu cục bộ và từ xa

Các phụ thuộc trong các thủ tục định sẵn (kể cả các hàm, các bộ phần mềm và bộ kích khởi) trong hệ thống cơ sở dữ liệu được phân phối được quản lý sử dụng kiểm dấu thời gian và kiểm chữ ký.

Tham số khởi hoạt động REMOTE_DEPENDENCIES_MODE xác định xem các dấu thời gian hoặc các chữ ký chi phối các sự phụ thuộc.

Thông tin bổ sung:
Xem Oracle8 Application Developer's Guide về quản lý các sự phụ thuộc từ xa bằng các dấu thời gian hoặc chữ ký.

Kiểm dấu thời gian

Trong các dấu thời gian kiểm mô hình phụ thuộc, bất kỳ khi nào một thủ tục được biên dịch hoặc biên dịch lại dấu thời gian của nó (thời gian nó được tạo, thay thế hoặc thay đổi) được ghi trong từ điển dữ liệu. Hơn nữa, phiên bản được biên dịch thủ tục chứa thông tin về giản đồ của thủ tục từ xa, tên bộ phần mềm và dấu thời gian.

Khi thủ tục phụ thuộc được sử dụng, Oracle so sánh các dấu thời gian từ xa được ghi vào thời gian biên dịch với các dấu thời gian

hiện hành của các thủ tục được tham chiếu từ xa. Tùy theo kết quả của so sánh này, hai trường hợp có thể xảy ra:

- Các thủ tục cục bộ và từ xa tiến hành không phải biên dịch nếu các dấu thời gian so khớp.
- Thủ tục cục bộ được vô hiệu hóa nếu bất kỳ các dấu thời gian của những thủ tục được tham chiếu từ xa không so khớp và một lỗi được cho trở lại môi trường gọi. Hơn nữa, tất cả các thủ tục cục bộ khác có dấu thời gian mới cũng được vô hiệu hóa. Thí dụ, giả sử nhiều thủ tục gọi một thủ tục từ xa và thủ tục từ xa ấy được biên dịch lại. Khi một trong các thủ tục cục bộ được tiến hành và thông báo dấu thời gian khác nhau của thủ tục từ xa, mọi thủ tục cục bộ phụ thuộc vào thủ tục từ xa được vô hiệu hóa.

So sánh dấu thời gian thực xảy ra khi một câu lệnh trong thân của thủ tục cục bộ tiến hành một thủ tục từ xa, chỉ vào lúc này các dấu thời gian được so sánh bằng liên kết truyền thông của cơ sở dữ liệu được phân phối. Do đó, tất cả các câu lệnh trong thủ tục cục bộ đi trước lệnh gọi thủ tục không hợp lệ có thể tiến hành thành công. Các câu lệnh theo sau lệnh gọi thủ tục bất hợp lệ không tiến hành (cần được biên dịch). Tuy nhiên, bất kỳ câu lệnh DML được tiến hành trước lệnh gọi thủ tục không hợp lệ được hồi phục trở về trạng thái cũ.

Kiểm chữ ký

Oracle cung cấp khả năng bổ sung về những phụ thuộc từ xa sử dụng các chữ ký. Tính năng chữ ký chỉ ảnh hưởng các chữ ký từ xa. Các phụ thuộc cục bộ (cùng server) không bị ảnh hưởng, khi biên dịch lại luôn luôn có thể tồn tại trong môi trường này.

Chữ ký của thủ tục chứa những thông tin về

- tên của bộ phần cứng, thủ tục hoặc hàm
- kiểu cơ sở của các tham số
- các chế độ của những tham số (in, out và in out)

Lưu ý:
Chỉ những kiểu và các chế độ của những tham số là quan trọng. Tên của tham số không ảnh hưởng đến chữ ký.

Nếu mô hình phụ thuộc chữ ký bị ảnh hưởng, thì sự phụ thuộc vào đơn vị chương trình từ xa (bộ phần cứng, thủ tục định sẵn, hàm được lưu trữ hoặc bộ kích khởi) làm cho sự vô hiệu hóa của đơn vị phụ thuộc nếu đơn vị phụ thuộc chứa lệnh gọi thủ tục trong đơn vị cha và chữ ký của thủ tục này đã bị thay đổi theo phương thức không tương thích.

Các sự phụ thuộc trong các đối tượng giản đồ từ xa khác

Oracle không quản lý các sự phụ thuộc trong những đối tượng giản đồ từ xa thay vì các phụ thuộc thủ tục cục bộ đối với thủ tục từ xa.

Thí dụ, giả sử một khung nhìn cục bộ được tạo và được xác định bằng một truy vấn tham chiếu một bảng từ xa. Ngoài ra, giả sử là thủ tục cục bộ có câu lệnh SQL tham chiếu cùng bảng từ xa. Về sau, định nghĩa của bảng bị thay đổi.

Do đó, khung nhìn và thủ tục cục bộ được sử dụng sau khi bảng bị thay đổi và dù khung nhìn hoặc thủ tục bây giờ cho trở lại các lỗi khi được sử dụng (trong trường hợp, khung nhìn hoặc thủ tục phải được thay đổi thủ công để không cho lại các lỗi). Trong trường hợp như thế, không có quản lý phụ thuộc thích hợp đối với các biên dịch lại không cần thiết của các đối tượng phụ thuộc.

Các sự phụ thuộc của các ứng dụng

Mà trong các ứng dụng cơ sở dữ liệu có thể tham chiếu những đối tượng trong cơ sở dữ liệu được kết nối. Thí dụ, OCI, Precompiler,

và các ứng dụng SQL*Module có thể đệ trình các khối PL/SQL nặc danh; các bộ kích khởi trong những ứng dụng Oracle Forms có thể tham chiếu một đối tượng giản đồ.

Các ứng dụng như thế phụ thuộc vào những đối tượng giản đồ chúng tham chiếu. Các kỹ thuật quản lý phụ thuộc khác nhau tùy theo môi trường phát triển. Tham chiếu các tài liệu thích hợp về các công cụ phát triển ứng dụng và hệ điều hành của bạn về cách quản lý các sự phụ thuộc từ xa trong các ứng dụng cơ sở dữ liệu.

Chương 20

Bộ tối ưu hóa

Chương này đề cập bộ tối ưu hóa của Oracle chọn cách tiến hành các câu lệnh SQL như thế nào. Nó gồm:

- **Sự tối ưu hóa là gì?**
  - **Các kế hoạch tiến hành**
  - **Thứ tự tiến hành**
- **Sự tối ưu hóa dựa trên nguyên tắc và phí tổn**
- **Tổng quan về các điều hoạt của bộ tối ưu hóa**
  - **Sự thẩm định của các biểu thức và các điều kiện**
  - **Biến thể và tối ưu hóa các câu lệnh**
  - **Chọn cách tiếp cận và mục đích tối ưu hóa**
  - **Chọn các đường dẫn truy xuất**
  - **Tối ưu hóa các câu lệnh liên kết**
  - **Tối ưu hóa các các liên kết tương phản và các bán liên kết**
  - **Tối ưu hóa các truy vấn "Sao"**

Thông tin bổ sung:
Để có thêm thông tin về bộ tối ưu hóa của Oracle, xem Oracle8 Tuning.

Sự tối ưu hóa là gì?

Sự tối ưu hóa là tiến trình chọn cách tiến hành một câu lệnh SQL hiệu quả nhất. Đây là bước quan trọng trong xử lý ngôn ngữ điều tác dữ liệu - Data Manipulation Language (DML): SELECT, INSERT, UPDATE hoặc DELETE. Hiện có nhiều cách khác nhau để tiến hành một câu lệnh SQL, thí dụ, thay đổi thứ tự mà các bảng hoặc các chỉ mục được truy xuất. Thủ tục Oracle sử dụng để tiến hành một câu lệnh có thể rất ảnh hưởng đến tốc độ câu lệnh được tiến hành.

Một thành phần của Oracle được gọi là bộ tối ưu hóa chọn những gì nó tin là cách hiệu quả nhất. Bộ tối ưu hóa thẩm định một số yếu tố để lựa trong các đường dẫn truy xuất thay thế. Đôi khi người thiết kế ứng dụng có nhiều thông tin về dữ liệu của ứng dụng cá biệt hơn một bộ tối ưu hóa có thể có, họ có thể chọn nhiều cách hiệu quả hơn để tiến hành một câu lệnh SQL. Người thiết kế ứng dụng có thể sử dụng các gợi ý trong các câu lệnh SQL để ấn định câu lệnh ấy sẽ được tiến hành như thế nào.

Lưu ý:
Bộ tối ưu hóa không thể đưa ra các quyết định từ một phiên bản của Oracle tới phiên bản kế tiếp. Trong nhiều phiên bản mới đây, bộ tối ưu hóa có thể đưa ra các quyết định khác nhau dựa trên thông tin tinh vi hơn và tốt hơn so với nó.

Thông tin bổ sung:
Xem Oracle8 Tuning về cách sử dụng các gợi ý trong các câu lệnh SQL.

Các kế hoạch tiến hành

Để tiến hành một câu lệnh DML, Oracle có thể phải thực hiện nhiều bước. Mỗi Step này truy hồi các hàng của dữ liệu về mặt vật lý từ cơ sở dữ liệu hoặc chuẩn bị chúng bằng cách nào đó để người dùng phát hành câu lệnh ấy. Tổ hợp các bước Oracle sử dụng để tiến hành câu lệnh được gọi là kế hoạch tiến hành.

Hình 20-1 trình bày biểu thị đồ họa của kế hoạch tiến hành cho câu lệnh SQL lựa tên, tác vụ, lương và tên của bộ phận cho tất cả những nhân viên có mức lương không rơi vào mức lương được đề nghị:

```
SELECT ename, job, sal, dname
  FROM emp, dept
 WHERE emp.deptno = dept.deptno
  AND NOT EXISTS
   (SELECT *
      FROM salgrade
     WHERE emp.sal BETWEEN losal AND hisal);
```

Các Step của kế hoạch tiến hành

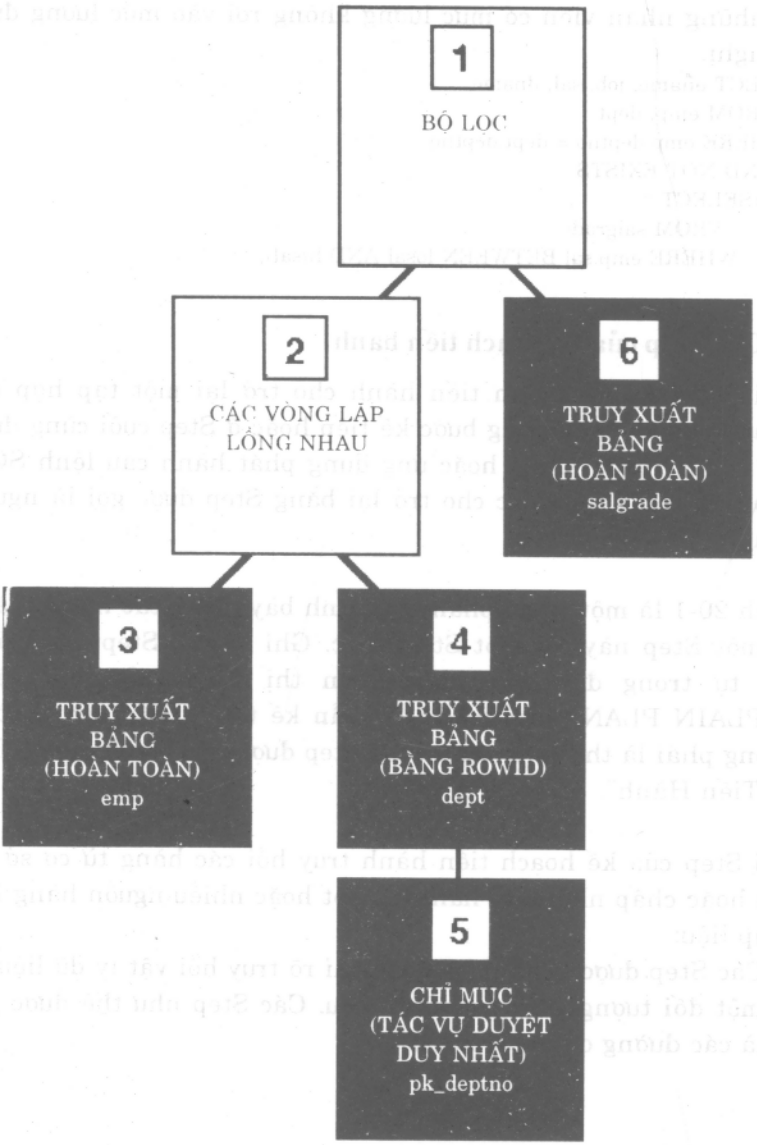
Mỗi Step của kế hoạch tiến hành cho trở lại một tập hợp các hàng được sử dụng bằng bước kế tiếp hoặc ở Step cuối cùng được cho trở lại người dùng hoặc ứng dụng phát hành câu lệnh SQL. Tập hợp các hàng được cho trở lại bằng Step được gọi là nguồn hàng.

Hình 20-1 là một sơ đồ phân cấp trình bày luồng các nguồn hàng từ một Step này tới một Step khác. Ghi số các Step phản ánh thứ tự trong đó chúng được hiển thị để trả lời cho lệnh EXPLAIN PLAN (được mô tả ở phần kế tiếp). Điều này thường không phải là thứ tự trong đó các Step được tiến hành (xem "Thứ Tự Tiến Hành".

Mỗi Step của kế hoạch tiến hành truy hồi các hàng từ cơ sở dữ liệu hoặc chấp nhận các hàng từ một hoặc nhiều nguồn hàng khi nhập liệu:

Các Step được các hộp có bóng chỉ rõ truy hồi vật lý dữ liệu từ một đối tượng trong cơ sở dữ liệu. Các Step như thế được gọi là các đường dẫn truy xuất:

Hình 20-1 Một kế hoạch tiến hành



Step 3 và 6 đọc tất cả các hàng của các bảng EMP và SALGRADE tương ứng.

Step 5 tra cứu trong chỉ mục PK_DEPTNO mỗi giá trị DEPTNO được cho trở lại bởi Step 3. Ở đó nó tìm các ROWID của các hàng được kết hợp trong bảng DEPT.

Step 4 truy hồi từ bảng DEPT mà các hàng có các ROWID đã được Step 5 cho trở lại.

Các Step được các hộp xóa chỉ rõ để điều tác trên các nguồn hàng:

Step 2 thực hiện các điều hoạt vòng lặp lồng nhau để chấp nhận các nguồn hàng từ các Step 3 và 4, liên kết mỗi hàng từ nguồn Step 3 tới hàng tương ứng của nó trong Step 4 và cho trở lại Step 1 các hàng kết quả.

Step 1 thực hiện một điều hoạt bộ lọc. Nó chấp nhận các nguồn hàng từ Step 2 và 6, loại trừ các hàng từ Step 2 có hàng tương ứng trong Step 6 và cho trở lại các hàng còn lại từ Step 2 tới người dùng hoặc ứng dụng phát hành câu lệnh ấy.

Các đường dẫn truy xuất được đề cập thêm ở phần "Chọn Các Đường Dẫn Truy Xuất". Các phương pháp được Oracle liên kết các nguồn hàng được đề cập ở "Các Điều Hoạt Liên Kết".

Lệnh EXPLAIN PLAN

Bạn có thể khảo sát kế hoạch tiến hành được bộ tối ưu hóa chọn cho câu lệnh SQL bằng cách sử dụng lệnh EXPLAIN PLAN làm cho bộ tối ưu hóa chọn kế hoạch tiến hành và sau đó chèn dữ liệu mô tả kế hoạch ấy vào bảng cơ sở dữ liệu.

Thí dụ, bảng kết xuất sau là một mô tả như thế để khảo sát câu lệnh trong phần trước:

```
ID  OPERATION          OPTIONS      OBJECT NAME
--  -----------------  -----------  -----------
0   SELECT STATEMENT
```

```
1    FILTER
2     NESTED LOOPS
3      TABLE ACCESS  FULL          EMP
4      TABLE ACCESS  BY ROWID       DEPT
5       INDEX        UNIQUE SCAN   PK_DEPTNO
6      TABLE ACCESS  FULL          SALGRADE
```

Mỗi hộp trong Hình 20-1 và mỗi hàng trong bảng kết xuất tương ứng với một bước đơn trong kế hoạch tiến hành. Đối với mỗi hàng trong danh sách đó, giá trị trong cột ID là giá trị được trình bày ở hộp tương ứng ở Hình 20-1.

Bạn có thể đạt được một danh sách như thế bằng cách sử dụng lệnh EXPLAIN PLAN và sau đó truy vấn bảng kết xuất.

Thông tin bổ sung:
Xem Oracle8 Tuning về cách sử dụng EXPLAIN PLAN và cho kết quả và diễn dịch kết xuất của nó.

Thứ tự tiến hành

Các bước của kế hoạch tiến hành không được thực hiện theo thứ tự chúng được ghi số. Đúng hơn, trước hết Oracle thực hiện các bước xuất hiện là các nút lá trong biểu diễn đồ họa có cấu trúc cây của kế hoạch tiến hành (các Step 3, 5 và 6 ở Hình 20-1). Các hàng được mỗi bước cho trở lại trở thành các nguồn hàng của bước cha của nó. Sau đó Oracle thực hiện các bước cha.

Để tiến hành câu lệnh đối với Hình 20-1, thí dụ, Oracle thực hiện các bước theo thứ tự này:
Thứ nhất, Oracle thực hiện Step 3 và cho trở lại các hàng kết quả, từng cặp một tới Step 2.
Đối với mỗi hàng được Step 3 cho trở lại, Oracle thực hiện các bước sau:

Oracle thực hiện Step 5 và cho trở lại ROWID kết quả cho Step 4.

Step 3 và 6 đọc tất cả các hàng của các bảng EMP và SALGRADE tương ứng.
Step 5 tra cứu trong chỉ mục PK_DEPTNO mỗi giá trị DEPTNO được cho trở lại bởi Step 3. Ở đó nó tìm các ROWID của các hàng được kết hợp trong bảng DEPT.
Step 4 truy hồi từ bảng DEPT mà các hàng có các ROWID đã được Step 5 cho trở lại.

Các Step được các hộp xóa chỉ rõ để điều tác trên các nguồn hàng:

Step 2 thực hiện các điều hoạt vòng lặp lồng nhau để chấp nhận các nguồn hàng từ các Step 3 và 4, liên kết mỗi hàng từ nguồn Step 3 tới hàng tương ứng của nó trong Step 4 và cho trở lại Step 1 các hàng kết quả.
Step 1 thực hiện một điều hoạt bộ lọc. Nó chấp nhận các nguồn hàng từ Step 2 và 6, loại trừ các hàng từ Step 2 có hàng tương ứng trong Step 6 và cho trở lại các hàng còn lại từ Step 2 tới người dùng hoặc ứng dụng phát hành câu lệnh ấy.

Các đường dẫn truy xuất được đề cập thêm ở phần "Chọn Các Đường Dẫn Truy Xuất". Các phương pháp được Oracle liên kết các nguồn hàng được đề cập ở "Các Điều Hoạt Liên Kết".

Lệnh EXPLAIN PLAN

Bạn có thể khảo sát kế hoạch tiến hành được bộ tối ưu hóa chọn cho câu lệnh SQL bằng cách sử dụng lệnh EXPLAIN PLAN làm cho bộ tối ưu hóa chọn kế hoạch tiến hành và sau đó chèn dữ liệu mô tả kế hoạch ấy vào bảng cơ sở dữ liệu.

Thí dụ, bảng kết xuất sau là một mô tả như thế để khảo sát câu lệnh trong phần trước:

```
ID  OPERATION          OPTIONS      OBJECT NAME
--- ---------- ---------- ---------- ----------
0   SELECT STATEMENT
```

Oracle thực hiện Step 4 và cho trở lại hàng kết quả Step 2.
Oracle thực hiện Step 2, liên kết hàng đơn từ Step 3 với hàng đơn từ Step 4 và cho trở lại hàng đơn cho Step 1.
Oracle thực hiện Step 6 và cho trở lại hàng kết quả Step 1.
Oracle thực hiện Step 1. Nếu hàng không cho trở lại từ Step 6, Oracle cho trở lại hàng từ Step 2 tới người dùng phát hành câu lệnh SQL.

Lưu ý Oracle thực hiện các Step 5, 4, 2, 6 và 1 một lần cho mỗi hàng được Step 3 cho trở lại. Nếu bước cha chỉ đòi hỏi một hàng đơn từ bước con trước khi nó có thể được tiến hành, Oracle thực hiện bước cha (và có thể là phần còn lại của kế hoạch tiến hành) ngay khi hàng đơn được cho trở lại từ bước con. Nếu cha của bước cha đó cũng có thể được kích hoạt bằng cách cho trở lại của một hàng đơn, thì nó cũng được tiến hành.

Do đó sự tiến hành có thể xếp tầng lên cây ấy, có thể hoàn thành phần còn lại của kế hoạch thực hiện. Oracle thực hiện bước cha và tất cả các bước xếp tầng một lần cho mỗi hàng đến lượt được truy hồi bằng bước con. Các bước cha được kích khởi cho mỗi hàng được bước con cho trở lại có các truy xuất bảng, các truy xuất chỉ mục, các liên kết vòng lặp được lồng nhau và các bộ lọc.

Nếu một bước cha đòi hỏi tất cả các hàng từ bước con trước khi nó có thể được tiến hành, Oracle không thể thực hiện bước cha cho tới khi tất cả các hàng đã được cho trở lại từ bước con. Các bước cha như thế có các sắp xếp, các liên kết hợp nhất sắp xếp, các hàm nhóm và các khối tập kết.

Sự phụ thuộc vào nguyên tắc và phí tổn

Để chọn một kế hoạch tiến hành cho một câu lệnh SQL, bộ tối ưu hóa sử dụng một trong hai cách giải quyết: dựa vào nguyên tắc hoặc dựa vào phí tổn.

Cách giải quyết dựa trên phí tổn

Sử dụng cách giải quyết dựa trên phí tổn, bộ tối ưu hóa quyết định kế hoạch tiến hành nào hiệu quả nhất bằng cách xét các đường dẫn truy xuất có thể sử dụng và các yếu tố trong thông tin dựa trên thống kê trong từ điển dữ liệu cho những đối tượng giản đồ (các bảng, các cụm liên kết hoặc các chỉ mục) được câu lệnh truy xuất. Cách giải quyết dựa trên phí tổn còn xét các gợi ý hoặc các đề nghị tối ưu hóa được đặt trong Comment của câu lệnh đó.

Về mặt khái niệm, cách giải quyết dựa trên phí tổn gồm các bước sau:

Bộ tối ưu hóa phát sinh tập hợp các kế hoạch tiến hành tiềm năng cho câu lệnh dựa trên các đường dẫn có thể truy xuất và các gợi ý.

Bộ tối ưu hóa dự đoán phí tổn của mỗi kế hoạch tiến hành dựa trên thống kê đặc tính phân phối và bộ nhớ cho các bảng, các cụm liên kết và các chỉ mục trong từ điển dữ liệu.

> Phí tổn là giá trị được dự đoán tỷ lệ với sử dụng tài nguyên được yêu cầu cần thiết để tiến hành câu lệnh sử dụng kế hoạch tiến hành ấy. Bộ tối ưu hóa tính toán phí tổn dựa trên các tài nguyên máy tính dự đoán, kể cả (nhưng không giá trị) thời gian I/O, CPU và bộ nhớ cần thiết để tiến hành câu lệnh sử dụng kế hoạch ấy.
>
> Các kế hoạch tiến hành nối tiếp chịu phí tổn hơn do mất thời thời gian tiến hành hơn các kế hoạch mất ít thời gian. Tuy nhiên, khi sử dụng một kế hoạch song song, sử dụng tài nguyên không quản lý trực tiếp với thời gian trôi qua.

Bộ tối ưu hóa so sánh các phí tổn của các kế hoạch tiến hành và chọn một kế hoạch ít phí tổn nhất.

Mục đích của cách giải quyết

Dựa trên phí tổn

Theo mặc định, mục đích của cách giải quyết dựa trên phí tổn là hiệu suất cao nhất hoặc sử dụng ít tài nguyên cần thiết nhất để tiến hành tất cả các hàng câu lệnh truy xuất.

Oracle cũng có thể tối ưu hóa câu lệnh với mục đích có thời gian trả lời nhanh nhất hoặc giảm tối đa tài nguyên cần thiết sử dụng để xử lý hàng thứ nhất được câu lệnh SQL truy xuất. Để có thêm thông tin về bộ tối ưu hóa chọn một cách giải quyết và mục đích tối ưu như thế nào, xem "Chọn Cách Giải Quyết Và Mục Đích Tối Ưu Hóa".

Lưu ý:
Đối với tiến hành song song, bộ tối ưu hóa có thể chọn để giảm tối đa thời gian phải trôi với chi phí tiêu thụ tài nguyên. Sử dụng tham số khởi hoạt OPTIMIZER_PERCENT_PARALLEL để ấn định bộ tối ưu hóa cố thực hiện song song bao nhiêu.

Thông tin bổ sung:
Xem Oracle8 về cách sử dụng tham số OPTIMIZER_PERCENT_PARALLEL.

Thông tin thống kê cho cách giải quyết dựa trên phí tổn

Cách giải quyết dựa trên phí tổn sử dụng thông tin thống kê để dự toán phí tổn của mỗi kế hoạch tiến hành. Thông tin thống kê này hội chuẩn sự phân phối dữ liệu và các đặc tính bộ nhớ của các bảng, các cột, các chỉ mục và các phân hoạch. Bạn có thể phát sinh các thông tin thống kê này bằng lệnh ANALYZE. Bộ tối ưu hóa sử dụng các thông tin thống kê này để dự đoán bao nhiêu thời gian I/O, CPU và bộ nhớ cần thiết để tiến hành câu lệnh SQL sử dụng một kế hoạch tiến hành cá biệt.

Bạn có thể xem thông tin thống kê với các khung nhìn từ điển dữ liệu này:

USER_TABLES, ALL_TABLES, and DBA_TABLES
USER_TAB_COLUMNS, ALL_TAB_COLUMNS, and DBA_TAB_COLUMNS
USER_INDEXES, ALL_INDEXES, and DBA_INDEXES
USER_CLUSTERS and DBA_CLUSTERS
USER_TAB_PARTITIONS, ALL_TAB_PARTITIONS, and DBA_TAB_PARTITIONS
USER_IND_PARTITIONS, ALL_IND_PARTITIONS, and DBA_IND_PARTITIONS
USER_PART_COL_STATISTICS, ALL_PART_COL_STATISTICS, and DBA_PART_COL_STATISTICS

Thông tin bổ sung:
Để có thông tin về những thông tin thống kê này, xem Oracle8 Reference.

Các biểu đồ

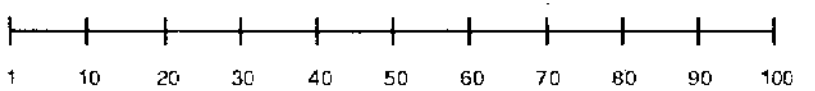
Bộ tối ưu hóa dựa trên phí tổi của Oracle sử dụng các biểu đồ giá trị dữ liệu để có được dự đoán chính xác sự phân phối dữ liệu cột. Các biểu đồ cung cấp các dự đoán nhạy bén được cải tiến về sự hiệu hữu của độ lệch dữ liệu dẫn đến các kế hoạch tiến hành tùy chọn với các phân phối dữ liệu không đồng nhất. Bạn có thể phát sinh các biểu đồ bằng cách sử dụng lệnh ANALYZE.

Một trong những khả năng nền tảng của bộ tối ưu hóa dựa trên phí tổn sẽ xác định tính chọn lọc thuộc tính xuất hiện trong các truy vấn. Các dựa đoán có tính chọnlọc được sử dụng để giải quyết khi sử dụng một chỉ mục và thứ tự để liên kết các bảng. Các vùng tối ưu hóa nhất (các cột của báng) không được phân phối đồng nhất. Bộ tối ưu hóa dựa trên phí tổn sử dụng các biểu đồ cân bằng độ cao trên các thuộc tính ấn định để mô tả các phân phối của những vùng không đồng nhất.

Các thí dụ biểu đồ

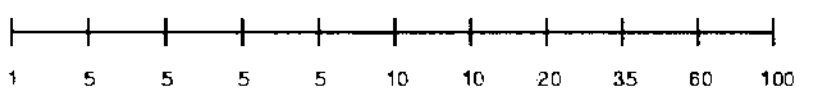
Xét một cột C với những giá trị từ 1 tới 100 và một biểu đồ với 10 cụm (bucket). Nếu dữ liệu trong C được phân phối đồng nhất, thì biểu đồ này nhìn giống như ở đây, trong đó các số là những giá trị điểm cuối.

Số hàng trong mỗi cụm là một phần mười tổng số hàng trong bảng. Bốn phần mười của các hàng có những giá trị từ 60 tới 100 trong thí dụ này về phân phối đồng nhất.



Nếu dữ liệu không được phân phối đồng nhất, thì biểu đồ ấy có thể nhìn như sau:

Trong trường hợp này, phần lớn các hàng có giá trị 5 cho cột. Trong thí dụ này, chỉ 1/10 của các hàng có các giá trị từ 60 tới 100.



Các biểu đồ cân bằng chiều cao

Oracle sử dụng các biểu đồ cân bằng chiều cao (trái lại cân bằng chiều rộng).

Các biểu đồ cân bằng chiều rộng chia dữ liệu thành số cố định các phạm vi chiều rộng bằng nhau và sau đó số những giá trị rơi vào mỗi phạm vi.

Các biểu đồ cân bằng chiều cao đặt cùng số giá trị trong mỗi phạm vi để các điểm cuối của phạm vi được xem bao nhiêu giá trị ở trong phạm vi đó.

Thí dụ, giả sử những giá trị ấy trong một cột đơn của m bảng 1000 hàng từ 10 tới 100 và giả sử bnj muốn có một biểu đồ 10 cụm (các phạm vi trong biểu đồ được gọi là các cụm). Trong biểu đồ cân bằng chiều rộng (1-10, 11-20, 21-30 vân vân) và mỗi cụm

sẽ tính số hàng rơi vào phạm vi của cụm đó. Trong biểu đồ cân bằng chiều cao, mỗi cụm có cung chiều cao (trong trường hợp này là 100 hàng) và các điểm cuối cho mỗi cụm được xác định bằng mật độ của những giá trị riêng biệt trong cột ấy.

Các ưu điểm của những biểu đồ cân bằng chiều cao

Ưu điểm của cách giải quyết cân bằng chiều cao rõ ràng khi dữ liệu có độ lệch cao. Giả sử 800 hàng của bảng 1000 hàng có giá trị 5 và còn lại 200 hàng được phân phối đều từ 1 tới 100. Biểu đồ cân bằng chiều rộng sẽ có 820 hàng trong cụm có nhãn 1-10 và khoảng 20 hàng trong mỗi cụm khác trong mỗi cụm khác. Biểu đồ dựa vào chiều cao sẽ có một cụm có nhãn 5-5, một cụm có nhãn 5-50 và một cụm có nhãn 50-100.

Nếu bạn muốn biết bao nhiêu hàng trong bảng chứa giá trị 5, nó được trình bày từ biểu đồ cân bằng chiều cao khoảng 80% các hàng chứa giá trị này. Tuy nhiên, biểu đồ cân bằng chiều rộng không phương pháp cơ chế để phân biệt giữa giá trị 5 và giá trị 6. Bạn sẽ chỉ tính 8% các hàng có giá trị 5 trong biểu đồ cân bằng chiều rộng. Do đó các biểu đồ cân bằng chiều cao thích hợp hơn để xác định tính chọn lựa của những giá trị cột.

Khi sử dụng các biểu đồ

Đối với nhiều người dùng, sử dụng tùy chọn FOR ALL INDEXED COLUMNS của lệnh ANALYZE thích hợp để tạo các biểu đồ vì các cột có chỉ mục thường là các cột được sử dụng phổ biến trong các mệnh đề WHERE.

Bạn có thể xem các biểu đồ bằng các khung nhìn sau:

USER_HISTOGRAMS, ALL_HISTOGRAMS và DBA_HISTOGRAMS

USER_PART_HISTOGRAMS, ALL_PART_HISTOGRAMS và DBA_PART_HISTOGRAMS

TAB_COLUMNS

Các biểu đồ chỉ hữu dụng khi chúng phản ánh phân phối dữ liệu hiện hành của cột cho trước. Nếu phân phối dữ liệu không tĩnh, thì biểu đồ sẽ được cập nhật thường xuyên. (dữ liệu không cần phải tĩnh bao lâu phân phối ấy vẫn không thay đổi).

Các biểu đồ có thể phản ánh hiệu suất và chỉ được sử dụng khi chúng cải tiến các kế hoạch truy vấn đáng kể. Các biểu đồ không hữu dụng đối với các cột có những đặc tính sau:
Tất cả các thuộc tính trên cột sử dụng các biến kết gán.
Dữ liệu được phân phối đồng nhất.
Cột không được sử dụng trong các mệnh đề WHERE của các truy vấn.
Cột duy nhất và chỉ được sử dụng với các thuộc tính ngang nhau.

Thông tin bổ sung:
Xem Oracle8 Tuning về các biểu đồ.

Khi sử dụng cách giải quyết dựa trên phí tổn

Nói chung, bạn nên sử dụng cách giải quyết dựa trên phí tổn cho tất cả những ứng dụng mới; cách giải quyết dựa trên nguyên tắc được phân phối cho các ứng dụng đã được viết trước khi tối ưu hóa dựa trên phí tổn có thể sử dụng. Sự tối ưu hóa dựa trên phí tổn có thể được sử dụng cho dữ liệu quan hệ và các kiểu đối tượng.

Các đặc trưng sau chỉ có thể sử dụng sự tối ưu hóa dựa trên phí tổn:

Các bảng phân hoạch
Các khung nhìn phân hoạch
Các bảng có tổ chức chỉ mục
Các chỉ mục khóa đảo ngược
Các chỉ mục bitmap
Truy vấn song song và DML song song
Biến thể sao
Liên kết sao

Thông tin bổ sung
Xem Oracle8 Tuning khi sử dụng cách giải quyết dựa trên phí tổn.

Cách giải quyết dựa trên nguyên tắc

Sử dụng cách giải quyết dựa trên nguyên tắc, bộ tối ưu hóa chọn một kế hoạch tiến hành dựa trên các đường dẫn truy xuất khả dụng và các mức của những đường dẫn truy xuất (như được trình bày ở "Các Đường Dẫn Truy Xuất". Bạn có thể sử dụng sự tối ưu hóa dựa trên nguyên tắc để truy xuất cả dữ liệu quan hệ và các kiểu đối tượng.

Định cấp các đường dẫn truy xuất của Oracle là phương pháp suy nghiệm. Nếu có nhiều cách tiến hành câu lệnh SQL, cách giải quyết dựa trên nguyên tắc luôn luôn sử dụng điều hoạt với cấp thấp. Thường các điều hoạt cấp thấp hơn tiến hành nhanh hơn các cấp được kết hợp với các cấu trúc có cấp cao hơn.

Xem thêm "Chọn Trong Các Đường Dẫn Truy Xuất Với Cách Giải Quyết Dựa Trên Nguyên Tắc".

Tổng quan về các điều hoạt của bộ tối ưu hóa

Phần này tóm tắt những điều hoạt bộ tối ưu hóa thực hiện và mô tả các câu lệnh SQL có thể được tối ưu hóa.

Các hoạt động của bộ tối ưu hóa

Đối với câu lệnh SQL bất kỳ được Oracle xử lý, bộ tối ưu hóa thực hiện những điều sau:

Thẩm định những biểu thức và các điều kiện	Trước hết bộ tối ưu hóa thẩm định các biểu thức và các điều kiện có các hằng đầy đủ tối đa. (xem "Sự Thẩm Định Của Các Biểu Thức Và Các

	Điều Kiện")
Sự biến thể câu lệnh	Đối với câu lệnh phức tạp liên quan, thí dụ, các truy vấn con tương quan, bộ tối ưu hóa có thể biến thể câu lệnh ban đầu thành câu lệnh liên kết tương đương. (Xem "Biến thể và tối ưu hóa các câu lệnh").
Hợp nhất khung nhìn	Đối với câu lệnh SQL truy xuất một khung nhìn, bộ tối ưu hóa thường hợp nhất truy vấn trong câu lệnh ấy với câu lệnh trong khung nhìn, sau đó tối ưu hóa kết quả. (Xem "Tối Ưu Hóa Các Câu Lệnh Truy Xuất Những Khung Nhìn").
Chọn cách giải quyết tối ưu hóa	Bộ tối ưu hóa chọn cách giải quyết dựa trên phí tổn hoặc cách giải quyết dựa trên nguyên tắc để tối ưu hóa và xác định mục đích của tối ưu hóa. (Xem "Chọn Cách Giải Quyết Và Mục Đích Tối Ưu Hóa")
Chọn các đường truy xuất	Đối với mỗi bảng được câu lệnh truy xuất, bộ tối ưu hóa chọn một hoặc nhiều đường dẫn truy xuất khả dụng để có được dữ liệu bảng ấy. (Xem "Chọn Các Đường Dẫn Truy Xuất".)
Chọn những thứ tự liên kết	Đối với câu lệnh liên kết, liên kết hơn hai bảng, bộ tối ưu hóa chọn hai bảng được liên kết đầu tiên và sau đó bảng được liên kết với kết quả ấy, vân vân. (Xem "Tối Ưu Hóa Các Câu Lệnh Liên Kết".)
Chọn các điều hoạt liên kết	Đối với câu lệnh liên kết bất kỳ, bộ tối ưu hóa chọn một điều hoạt sử dụng để thực hiện liên kết ấy. (Xem

	"Tối Ưu Hóa Các Câu Lệnh Liên Kết".)

Các kiểu câu lệnh SQL

Oracle tối ưu hóa những kiểu câu lệnh SQL sau:

Câu lệnh đơn	Câu lệnh INSERT, UPDATE, DELETE hoặc SELECT.
Truy vấn đơn	Một tên khác cho câu lệnh SELECT.
Liên kết	Truy vấn lựa dữ liệu từ nhiều bảng. Một liên kết được biểu thị bằng nhiều bảng trong mệnh đề FROM. Oracle ghép cặp từ các bảng này bằng cách sử dụng điều kiện được ấn định trong mệnh đề WHERE và cho trở lại các hàng kết quả. Điều kiện này được gọi là điều kiện liên kết và thường so sánh các cột của tất cả các bảng được liên kết.
Liên kết tương đương	Điều kiện liên kết chứa toán tử tương đương.
Liên kết không tương tương đương	Điều kiện liên kết chứa điều gì đó thay vì toán tử tương đương.
Liên kết bên ngoài	Điều kiện liên kết sử dụng toán tử liên kết bên ngoài (+) với một hoặc nhiều cột của một trong các bảng. Oracle cho trở lại tất cả các hàng đáp ứng điều kiện liên kết. Oracle còn cho trở lại tất cả các hàng từ bảng ấy mà không có toán tử liên kết bên ngoài, không có các hàng so khớp trong bảng với toán tử liên kết bên ngoài.
Tích Đề-các	Liên kết không có các kết quả điều kiện liên kết trong tích Đề-các hoặc tích

	chéo. Tích Đề-các là tập hợp tất cả những tổ hợp có thể của những hàng được rút ra từ mỗi bảng. Nói cách khác, đối với một liên kết của hai bảng, mỗi hàng trong một bảng lần lượt được so khớp với mỗi hàng ở bảng kia. Tích Đề-các đối với nhiều bảng là kết quả của ghép cặp mỗi hàng của một bảng với mọi hàng của tích Đề-các của các bảng còn lại. Tất cả các loại liên kết khác là các tập hợp con của các tích Đề-các được tạo ra một cách hiệu quả bằng cách phái sinh tích Đề-các và sau đó loại trừ các hàng không thực hiện được điều kiện liên kết đó.
Câu lệnh phức tạp	Câu lệnh INSERT, UPDATE, DELETE hoặc SELECT có truy vấn con là biểu mẫu của câu lệnh SELECT trong một câu lệnh khác để tạo ra một tập hợp những giá trị để xử lý thêm trong câu lệnh ấy. Phần bên ngoài của câu lệnh phức tạp chứa truy vấn con được gọi là câu lệnh cha.
Truy vấn ghép	Truy vấn sử dụng các toán tử (UNION, UNION ALL, INTERSECT hoặc MINUS) để kết hợp hai hoặc nhiều câu lệnh đơn hoặc phức tạp. Mỗi câu lệnh đơn hoặc phức tạp trong truy vấn ghép được gọi là truy xuất cấu thành.
Các khung nhìn truy xuất câu lệnh	Câu lệnh đơn, liên kết, phức tạp hoặc ghép truy xuất một hoặc nhiều khung nhìn cũng như các bảng.

Câu lệnh phân phối	Câu lệnh truy xuất dữ liệu trên một cơ sở dữ liệu từ xa.

Thẩm định các biểu thức và các điều kiện

Bộ tối ưu hóa thẩm định toàn bộ các biểu thức bất kỳ khi nào có thể và phiên dịch các cấu trúc cú pháp nào đó thành các cấu trúc tương đương. Lý do trên là Oracle có thể thẩm định nhanh biểu thức kết quả hơn biểu thức ban đầu hoặc biểu thức ban đầu chỉ tương đương với biểu thức kết quả. Các câu trúc SQL khác nhau đôi khi điều hoạt động nhất (thí dụ, = ANY (truy vấn con) và IN (truy vấn con)); Oracle ánh xạ những truy vấn con vào cấu trúc đơn.

Các hằng

Tính toán các hằng chỉ được thực hiện một lần, khi câu lệnh được tối ưu hóa, thay vì mỗi lần một câu lệnh được tiến hành.

Xét các điều kiện sau để kiểm tra các mức lương hàng tháng lớn hơn 2000:
sal > 24000/12

sal > 2000

sal*12 > 24000

Nếu câu lệnh có điều kiện thứ nhất, thì bộ tối ưu hóa đơn giản hóa nó thành điều kiện thứ hai.

Lưu ý bộ tối ưu hóa không đơn giản các biểu thức bằng các toán tử so sánh: trong các thí dụ ở trên, bộ tối ưu hóa không đơn giản hóa biểu thức thứ ba thành biểu thức thứ hai. Vì lý do này, các Developer ứng dụng sẽ viết các điều kiện để so sánh các cột với các hằng bất kỳ khi nào có thể, thay vì các điều kiện với các biểu thức liên quan đến các cột.

Toán tử LIKE

Bộ tối ưu hóa đơn giản hóa các điều kiện sử dụng toán tử so sánh LIKE để so sánh một biểu thức không có các ký tự đại diện thành điều kiện tương đương thay vì sử dụng toán tử tương đương. Thí dụ, bộ tối ưu hóa đơn giản hóa điều kiện thứ nhất bên dưới thành điều kiện thứ hai:
ename LIKE 'SMITH'

ename = 'SMITH'

Bộ tối ưu hóa có thể đơn giản hóa những biểu thức này chỉ khi so sánh liên quan đến các kiểu dữ liệu có chiều dài khả biến. Thí dụ, nếu if ENAME đã thuộc kiểu CHAR(10), bộ tối ưu hóa không thể biến thể phép toán LIKE thành phép toán tương đương do toán tử tương đương theo sau ngữ nghĩa có chêm khoảng trống và LIKE không theo sau ngữ nghĩa chêm khoảng trống.

Toán tử IN

Bộ tối ưu hóa mở rộng điều kiện sử dụng toán tử so sánh IN thành điều kiện tương đương sử dụng các toán tử so sánh và các toán tử logic OR. Thí dụ, bộ tối ưu hóa mở rộng điều kiện thứ nhất bên dưới thành điều kiện thứ hai:
ename IN ('SMITH', 'KING', 'JONES')

ename = 'SMITH' OR ename = 'KING' OR ename = 'JONES'

xem "Thí dụ 2: Truy vấn con IN".

Toán tử ANY hoặc SOME

Bộ tối ưu hóa mở rộng điều kiện sử dụng toán tử so sánh ANY hoặc SOME được theo sau bằng một danh sách trong dấu ngoặc của những giá trị thành điều kiện tương đương mà bộ tối ưu hóa mở rộng điều kiện thứ nhất bên dưới thành điều kiện thứ hai:

sal > ANY (:first_sal, :second_sal)

sal > :first_sal OR sal > :second_sal

Bộ tối ưu hóa biến thể điều kiện sử dụng toán tử ANY hoặc SOME được theo sau bằng một truy vấn con thành điều kiện có toán tử EXISTS và một truy vấn con tương quan. Thí dụ, bộ tối ưu hóa biến thể điều kiện thứ nhất bên dưới thành điều kiện thứ hai:

```
x > ANY (SELECT sal
   FROM emp
   WHERE job = 'ANALYST')

EXISTS (SELECT sal
   FROM emp
   WHERE job = 'ANALYST'
    AND x > sal)
```

Toán tử ALL

Bộ tối ưu hóa mở rộng điều kiện sử dụng toán tử so sánh ALL được theo sau bằng một danh sách trong dấu ngoặc của những giá trị thành điều kiện tương đương sử dụng các toán tử so sánh tương đương và các toán tử logic AND. Thí dụ, bộ tối ưu hóa mở rộng điều kiện thứ nhất bên dưới thành điều kiện thứ hai:

sal > ALL (:first_sal, :second_sal)

sal > :first_sal AND sal > :second_sal

Bộ tối ưu hóa biến thể một điều kiện sử dụng toán tử so sánh ALL được theo sau bằng một truy vấn con thành điều kiện tương đương sử dụng toán tử so sánh ANY và toán tử so sánh bù. Thí dụ, bộ tối ưu hóa biến thể điều kiện thứ nhất bên dưới thành điều kiện thứ hai:

```
x > ALL (SELECT sal
```

```
  FROM emp
  WHERE deptno = 10)

NOT (x <= ANY (SELECT sal
    FROM emp
    WHERE deptno = 10) )
```

Sau đó bộ tối ưu hóa biến thể truy vấn thứ hai thành truy vấn tiếp theo sử dụng nguyên tắc biến thể các điều kiện với toán tư so sánh ANY được theo sau bằng một truy vấn con tương quan.:

```
NOT EXISTS (SELECT sal
    FROM emp
    WHERE deptno = 10
      AND x <= sal)
```

Toán tử BETWEEN

Bộ tối ưu hóa llt thay thế điều kiện sử dụng toán tử so sánh BETWEEN với điều kiện tương đương sử dụng các toán tử so sánh >= và <=. Thí dụ, bộ tối ưu hóa thay thế điều kiện thứ nhất bên dưới bằng điều kiện thứ hai:

```
sal BETWEEN 2000 AND 3000

sal >= 2000 AND sal <= 3000
```

Toán tử NOT

Bộ tối ưu hóa đơn giản hóa điều kiện để loại trừ toán tử logic NOT. Sự đơn giản hóa liên quan đến xóa toán tư logic NOT và thay thế toán so sánh bằng toán tử so sánh đảo của nó. Thí dụ, bộ tối ưu hóa đơn giản hóa điều kiện thứ nhất bên dưới thành điều kiện thứ hai:

```
NOT deptno = (SELECT deptno FROM emp WHERE ename =
'TAYLOR')
```

deptno <> (SELECT deptno FROM emp WHERE ename = 'TAYLOR')

Thường một điều kiện có toán tử logic NOT có thể được viết bằng nhiều cách khác nhau. Bộ tối ưu hóa cố biến thể một điều kiện để các điều kiện phụ được phủ định bằng các NOT được đơn giản tối đa, dù điều kiện kết quả có nhiều NOT. Thí dụ, bộ tối ưu hóa đơn giản hóa điều kiện thứ nhất bên dưới thành điều kiện thứ hai và dhen thành điều kiện thứ ba.
NOT (sal < 1000 OR comm IS NULL)

NOT sal < 1000 AND comm IS NOT NULL

sal >= 1000 AND comm IS NOT NULL

Tính truyền

Nếu hai điều kiện trong mệnh đề WHERE liên quan đến một cột chung, thì bộ tối ưu hóa đôi khi có thể suy ra điều kiện thứ ba sử dụng nguyên tắc truyền. Bộ tối ưu hóa sau đó có thể sử dụng điều kiện được suy ra để tối ưu hóa câu lệnh ấy. Điều kiện suy ra có thể tạo được đường dẫn truy xuất chỉ mục không được cửa sổ điều kiện ban đầu tạo.

> Lưu ý:
> Tính truyền chỉ được sử dụng bằng cách giải quyết dựa trên phí tổn.

Giả sử mệnh đề WHERE có hai điều kiện của các biểu mẫu sau:
WHERE column1 comp_oper constant
AND column1 = column2

Trong trường hợp này, bộ tối ưu hóa suy ra điều kiện:
column2 comp_oper constant

trong đó:

comp_oper	Là một toán tử bất kỳ trong các toán tử so sánh =, !=, ^=, <, <>, >, <=, or >=.
Hằng	Là biểu thức không đổi liên quan đến các toán tử, các hàm SQL, các trực kiện, các biến kết gán và các biến tương quan.

Thí dụ:

Xét truy vấn sau trong mệnh đề WHERE có hai điều kiện, hoặc mỗi điều kiện sử dụng cột EMP.DEPTNO:

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = 20
    AND emp.deptno = dept.deptno;
```

Sử dụng tính truyền, bộ tối ưu hóa suy ra điều kiện sau:
dept.deptno = 20

Nếu chỉ mục tồn tại trên cột DEPT.DEPTNO, điều kiện này tạo đường dẫn truy xuất sử dụng chỉ mục đó.

Lưu ý:
Bộ tối ưu hóa chỉ suy ra các điều kiện liên kết các cột với các biểu thức không thay đổi, thay vì các cột với các cột khác. Giả sử mệnh đề WHERE có hai điều kiện của các biểu mẫu sau:

```
WHERE column1 comp_oper column3
  AND column1 = column2
```

Trong trường hợp này, bộ tối ưu hóa không suy ra điều kiện sau:
column2 comp_oper column3

Biến thể và tối ưu hóa các câu lệnh

SQL là một ngôn ngữ truy vấn linh hoạt; thường có nhiều câu lệnh bạn có thể lập công thức để có được cùng mục đích. Đôi khi bộ tối ưu hóa biến thể một câu lệnh như thế thành một câu lệnh khác để đạt được cùng mục đích nếu câu lệnh thứ hai có thể được tiến hành hiệu quả hơn.

Phần này đề cập đến các chủ đề sau:

Biến Thể Các OR Thành Các Truy Vấn Ghép
Biến Thể Các Câu Lệnh Phức Tạp Thành Các Câu Lệnh Liên Kết
Tối Ưu Hóa Các Câu Lệnh Truy Xuất Các Khung Nhìn
Tối Ưu Hóa Các Truy Vấn Ghép
Tối Ưu Hóa Các Câu Lệnh Được Phân Phối

Để có thêm thông tin về tối ưu hóa các câu lệnh, xem "Tối Ưu Hóa Các Câu Lệnh Liên Kết" và Tối Ưu Hóa Các Truy Vấn "Star".

Biến thể các OR thành các truy vấn ghép

Nếu một truy vấn có mệnh đề WHERE với nhiều điều kiện được kết hợp với các toán tử OR, thì bộ tối ưu hóa biến thể nó thành một truy vấn ghép tương đương sử dụng toán tử tập hợp UNION ALL nếu điều này làm cho tiến hành hiệu quả hơn:

Nếu mỗi điều kiện cá biệt tạo một đường dẫn truy xuất khả dụng, bộ tối ưu hóa có thể tạo biến thể ấy. Sau đó bộ tối ưu hóa chọn một kế hoạch tiến hành cho câu lệnh kết quả để truy xuất bảng nhiều lần bằng cách sử dụng các chỉ mục khác nhau và sau đó kết hợp các kết quả lại với nhau.

Nếu điều kiện bất kỳ đòi hỏi duyệt toàn bộ một bảng vì nó không tạo chỉ mục khả dụ, bộ tối ưu hóa không biến thể câu lệnh ấy. Bộ tối ưu hóa chọn duyệt toàn bộ bảng để tiến hành câu lệnh ấy và Oracle kiểm tra từng hàng trong bảng để xác định xem nó có thỏa đáng bất kỳ điều kiện nào không.

Đối với các câu lệnh sử dụng cách giải quyết dựa trên phí tổn, bộ tối ưu hóa có thể sử dụng các thống kê để xác định nếu có tạo được biến thể bằng cách dự đoán và sau đó so sánh các phí tổn tiến hành câu lệnh ban đầu so với câu lệnh kết quả.
Bộ tối ưu hóa dựa trên phí tổn không sử dụng biến thể OR trong các danh sách hoặc các OR trên cùng cột, thay vào đó, nó sử dụng các toán tử bộ lặp trong danh sách.

Thông tin bổ sung:
Xem thêm Oracle8 Tuning.

Để có thêm thông tin về các đường dẫn truy xuất và các chỉ mục tạo cho chúng khả dụng như thế nào, xem "Các Đường Dẫn Truy Xuất" và các phần tiếp sau nó.

Thí dụ:

Xét truy vấn này với mệnh đề WHERE có hai điều kiện được kết hợp với toán tử OR:

```
SELECT *
  FROM emp
  WHERE job = 'CLERK'
    OR deptno = 10;
```

Nếu có cc chỉ mục trên các cột JOB và DEPTNO, bộ tối ưu hóa có thể biến thể truy vấn này thành truy vấn tương đương bên dưới:

```
SELECT *
  FROM emp
  WHERE job = 'CLERK'
UNION ALL
SELECT *
  FROM emp
  WHERE deptno = 10
    AND job <> 'CLERK';
```

Nếu bạn đang sử dụng cách giải quyết dựa trên phí tổn, thì bộ tối ưu hóa so sánh phí tổn tiến hành truy vấn ban đầu sử dụng duyệt toàn bộ bảng với phí tổn tiến hành truy vấn kết quả khi quyết định xem có tạo được biến thể không.

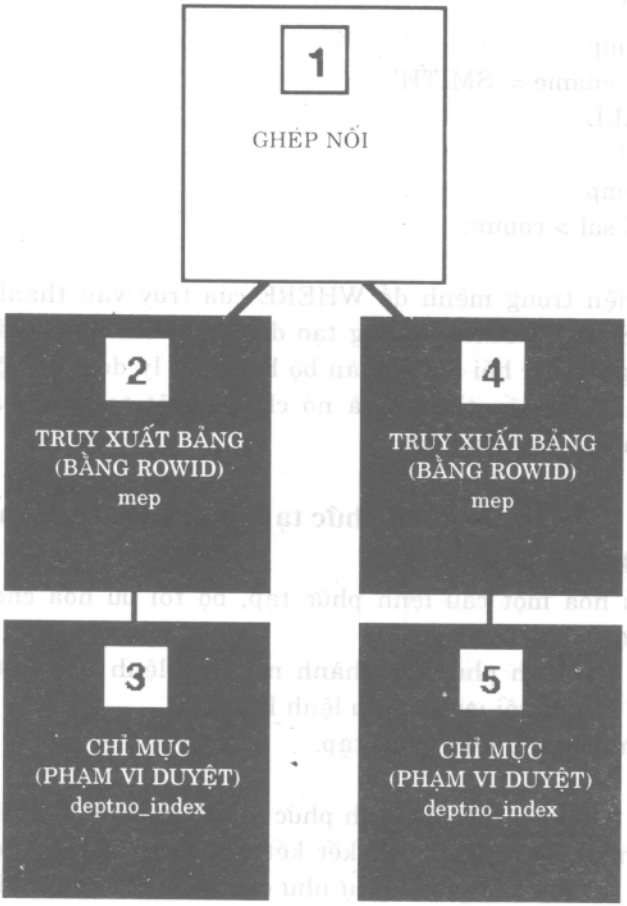
Nếu bạn đang sử dụng cách giải quyết dựa trên nguyên tắc, thì bộ tối ưu hóa tạo biến thể UNION ALL này vì mỗi truy vấn cấu thành truy vấn kết quả ghép có thể được tiến hành bằng cách sử dụng chỉ mục. Cách giải quyết dựa trên nguyên tắc cho là tiến hành truy vấn ghép sử dụng hai lần duyệt chỉ mục nhanh hơn tiến hành truy vấn ban đầu sử dụng duyệt toàn bộ bảng.

Kế hoạch tiến hành đối với câu lệnh được biến thể có thể nhìn giống như được minh họa ở Hình 20-2:

Để tiến hành truy vấn được biến thể, Oracle thực hiện các bước sau:
Các Step 3 và 5 duyệt các chỉ mục trên các cột JOB và DEPTNO sử dụng các điều kiện của các truy vấn thành phần. Những bước này có các ROWID của những hàng thỏa đáng các truy vấn thành phần.
Các Step 2 và 4 sử dụng các ROWID từ các Step 3 và 5 để định vị các hàng thỏa đáng từng truy vấn thành phần.
Step 1 kết hợp các nguồn hàng được Steps 2 và 4 cho trở lại.

Nếu cột JOB hoặc DEPTNO không có chỉ mục, bộ tối ưu hóa thậm chí không xét biến thể ấy, vì truy vấn kết quả ghép sẽ đòi hỏi duyệt toàn bộ bảng để tiến hành một trong những truy vấn thành phần của nó. Tiến hành truy vấn ghép duyệt toàn bộ bảng bổ sung vào duyệt chỉ mục không thể nhanh hơn tiến hành truy vấn ban đầu duyệt toàn bộ bảng.

Hình 20-2 Kế hoạch tiến hành một truy vấn được biến thể có OR



Thí dụ:

Xét truy vấn sau và giả sử là có một chỉ mục trên chỉ trên cột ENAME:

```
SELECT *
  FROM emp
  WHERE ename = 'SMITH'
```

```
  OR sal > comm;
```

Biến thể truy vấn trên sẽ dẫn đến truy vấn ghép bên dưới:

```
SELECT *
  FROM emp
  WHERE ename = 'SMITH'
UNION ALL
SELECT *
  FROM emp
  WHERE sal > comm;
```

Vì điều kiện trong mệnh đề WHERE của truy vấn thành phần thứ hai (SAL > COMM) không tạo được một chỉ mục khả dụng, truy vấn ghép đòi hỏi duyệt toàn bộ bảng. Vì lý do này, bộ tối ưu hóa không tạo biến thể ấy và nó chọn duyệt toàn bộ bảng để tiến hành câu lệnh ban đầu.

Biến thể các câu lệnh phức tạp thành các câu lệnh liên kết

Để tối ưu hóa một câu lệnh phức tạp, bộ tối ưu hóa chọn một trong những thay thế sau:

Biến thể câu lệnh phức tạp thành một câu lệnh liên kết tương đương rồi sau đó tối ưu hóa câu lệnh liên kết.

Nhứ tối ưu hóa câu lệnh phức tạp.

Bộ tối ưu hóa biến thể câu lệnh phức tạp thành câu lệnh liên kết bất kỳ khi nào câu lệnh liên kết kết quả được bảo đảm cho trở lại chính xác các hàng tương tự như câu lệnh phức tạp. Biến thể này cho phép Oracle tiến hành câu lệnh ấy bằng cách khai thác những kỹ thuật tối ưu hóa liên kết được đề cập ở "Tối Ưu Hóa Các Câu Lệnh Liên Kết".

Xét câu lệnh phức tạp lựa tất cả các hàng từ bảng ACCOUNTS có các chủ nhân Xem thêm trong bảng CUSTOMERS:

```
SELECT *
```

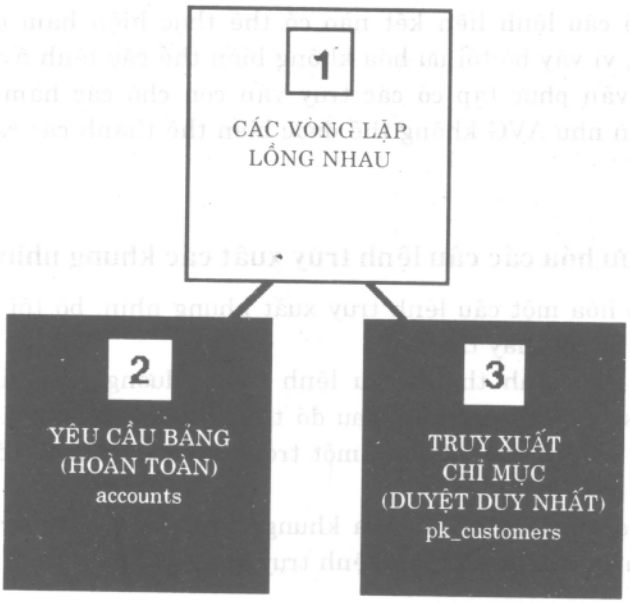
```
FROM accounts
WHERE custno IN
  (SELECT custno FROM customers);
```

Nếu cột CUSTNO là khóa chính hoặc có một hạn định UNIQUE, bộ tối ưu hóa có thể biến thể truy vấn phức tạp thành câu lệnh liên kết được bảo đảm cho trở lại dữ liệu tương tự:

```
SELECT accounts.*
  FROM accounts, customers
  WHERE accounts.custno = customers.custno;
```

Kế hoạch tiến hành đối với câu lệnh này có thể nhìn giống như Hình 20-3.

Hình 20-3 Kế hoạch tiến hành đối với ạm liên kết các vòng lặp lồng nhau



Để tiến hành câu lệnh này, Oracle thực hiện điều hoạt liên kết các vòng lặp lồng nhau. Để có thông tin về các liên kết vòng lặp lồng nhau, xem "Các Điều Hoạt Liên Kết".

Nếu bộ thuộc tính không thể biến thể một câu lệnh phức tạp thành câu lệnh liên kết, bộ tối ưu hóa chọn các kế hoạch tiến hành đối với câu lệnh cah và truy vấn con như chúng là các câu lệnh tách riêng. Sau đó Oracle tiến hành truy vấn con và sử dụng các hàng mà nó cho trở để tiến hành truy vấn cha.

Xét câu lệnh phức tạp này cho trở lại tất cả những hàng từ bảng ACCOUNTS có các số dư lớn hơn số dư tài khoản trung bình:

```
SELECT *
  FROM accounts
  WHERE accounts.balance >
    (SELECT AVG(balance) FROM accounts);
```

Không có câu lệnh liên kết nào có thể thực hiện hàm của câu lệnh này, vì vậy bộ tối ưu hóa không biến thể câu lệnh ấy. Lưu ý các truy vấn phức tạp có các truy vấn con chó các hàm nhóm, chẳng hạn như AVG không thể được biến thể thành các câu lệnh liên kết.

Tối ưu hóa các câu lệnh truy xuất các khung nhìn

Để tối ưu hóa một câu lệnh truy xuất khung nhìn, bộ tối ưu hóa chọn một trong thay thế sau:

Biến thể câu lệnh thành câu lệnh tương đương truy xuất các bảng cơ sở của khung nhìn, sau đó tối ưu hóa câu lệnh kết quả.

Bộ tối ưu hóa có thể sử dụng một trong những kỹ thuật để biến dạng câu lệnh ấy:

- Hợp nhất truy vấn của khung nhìn thành khối truy vấn tham chiếu trong câu lệnh truy xuất.

- Đẩy thuộc tính của khối truy vấn tham chiếu vào bên trong khung nhìn (đối với một khung nhìn không hợp nhất được).

Phát hành truy vấn của khung nhìn, tập hợp tất cả các hàng cho trở lại và sau đó truy xuất tập hợp những hàng này với câu lệnh ban đầu như nó là một bảng. (Xem "Truy Xuất Các Hàng Của Khung Nhìn Với Câu Lệnh Ban Đầu".)

Hợp nhất truy vấn của khung nhìn vào câu lệnh

Để hợp nhất truy vấn của khung nhìn vào một khối truy vấn tham chiếu trong câu lệnh truy xuất, bộ tối ưu hóa thay tên của khung nhìn với các tên bảng cơ sở của nó trong khối truy vấn và bổ sung điều kiện của mệnh đề WHERE của truy vấn khung nhìn vào mệnh đề WHERE của khối truy vấn truy xuất.

Sự tối ưu hóa này áp dụng vào *select-project-join views* (các khung nhìn liên kết phép chiếu lựa chọn) là các khung nhìn chỉ có các lựa chọn, các phép chiếu và các liên kết - nghĩa là các khung nhìn không có các toán tử tập hợp, các hàm nhóm, DISTINCT, GROUP BY, CONNECT BY vân vân (như được đề cập ở "Các Khung Nhìn Hợp Nhất Được Và Không Hợp Nhất Được").

Thí dụ:

Xét khung nhìn của tất cả các nhân viên sau làm việc trong phòng số 10:

```
CREATE VIEW emp_10
  AS SELECT empno, ename, job, mgr, hiredate, sal, comm,
deptno
    FROM emp
    WHERE deptno = 10;
```

Xét truy vấn sau truy vấn khung nhìn ấy. Truy vấn lựa các ID lớn hơn 7800 nhân viên làm việc thuộc phòng 10::

```
SELECT empno
  FROM emp 10
  WHERE empno > 7800;
```

Bộ tối ưu hóa biến thể truy vấn truy xuất bảng cơ sở của khung nhìn ấy:

```
SELECT empno
  FROM emp
  WHERE deptno = 10
    AND empno > 7800;
```

Nếu có các chỉ mục trên các cột DEPTNO hoặc EMPNO, thì mệnh đề kết quả WHERE tạo cho chúng khả dụng.

Các khung nhìn hợp nhất được và không hợp nhất được

Bộ tối ưu hóa có thể hợp nhất một khung nhìn vào khối truy vấn tham chiếu khi khung nhìn có một hoặc nhiều bảng cơ sở, với điều kiện khung nhìn không có:

Các toán tử tập hợp (UNION, UNION ALL, INTERSECT, MINUS)

Mệnh đề CONNECT BY

Cột giả ROWNUM

Các hàm nhóm (AVG, COUNT, MAX, MIN, SUM) trong danh sách lựa.

Khi một khung nhìn có một trong những cấu trúc sau, nó có thể được hợp nhất thành khối truy vấn tham chiếu chỉ nếu hợp nhất khung nhìn phức tạp được cho phép (như được mô tả ở dưới):

Mệnh đề GROUP BY

Toán tử DISTINCT trong danh sách lựa

Hợp nhất khung nhìn không thể xảy ra đối với một khung nhìn có nhiều bảng cơ sở nếu nó ở bên phải của liên kết bên ngoài. Tuy nhiên, nếu một khung nhìn ở bên phải của liên kết bên ngoài chỉ có một bảng cơ sở, thì bộ tối ưu hóa có thể sử dụng hợp nhất khung nhìn phức tạp dù một biểu thức trong khung nhìn có thể cho trở lại đối tượng không rỗng đối với một một NULL. Xem "Cửa Sổ Khung Nhìn Trong Các Liên Kết Bên Ngoài"/.

Hợp nhất khung nhìn phức tạp

Nếu một truy vấn của khung nhìn có mệnh đề GROUP BY hoặc toán tư DISTINCT trong danh sách lựa, thì bộ tối ưu hóa có thể hợp nhất truy vấn của khung nhìn vào câu lệnh truy xuất chỉ nếu hợp nhất khung nhìn phức tạp được cho phép. Hợp nhất phức tạp cũng có thể được sử dụng để hợp nhất một truy vấn con IN vào câu lệnh truy xuất, nếu truy vấn con đó không tương quan (xem Thí dụ 2: Truy vấn con IN").

Hợp nhất phức tạp không dựa vào phía tồn - nó phải được cho phép với tham chiếu khởi hoạt COMPLEX_VIEW MERGING hoặc gợi ý MERGE, nghĩa là tham số
COMPLEX VIEW_MERGING phải được xác lập là TRUE hoặc khối truy vấn truy xuất phải có gợi ý MERGE. Không có gợi ý này hoặc xác lập tham số, bộ tối ưu hóa sử dụng một cách giải quyết khác (xem "Đẩy Thuộc Tính Vào Khung Nhìn").

> Thông tin bổ sung:
> Xem Oracle8 Tuning về các gợi ý MERGE và NO_MERGE.

Thí dụ 1: Khung nhìn với mệnh đề GROUP BY

Xét khung nhìn AVG_SALARY_VIEW có các mức lương trung bình cho mỗi bộ phận:

```
CREATE VIEW avg_salary_view AS
  SELECT deptno, AVG(sal) AS avg_sal_dept,
    FROM emp
```

```
GROUP BY deptno;
```

Nếu hợp nhất khung nhìn phức tạp được cho phép thì bộ tối ưu hóa có thể biến thể truy vấn này để tìm những mức lương trung bình của các bộ phận ở London:

```
SELECT dept.deptloc, avg_sal_dept
  FROM dept, avg_salary_view
  WHERE dept.deptno = avg_salary_view.deptno
    AND dept.deptloc = 'London';
```

Vào truy vấn này:

```
SELECT dept.deptloc, AVG(sal)
  FROM dept, emp
  WHERE dept.deptno = emp.deptno
    AND dept.deptloc = 'London'
  GROUP BY dept.rowid, dept.deptloc;
```

Truy vấn được biến thể truy xuất bảng cơ sở của khung nhìn, chỉ lựa các hàng của những nhân viên làm việc ở London và tạo họ thành nhóm theo bộ phận.

Thí dụ 2: Truy vấn con IN

Hợp nhất phức tạp có thể được sử dụng cho mệnh đề IN với truy vấn con không tương quan, cũng như đối với cửa sổ khung nhìn. Xét khung nhìn MIN_SALARY_VIEW có những mức lương tối thiểu cho mỗi bộ phận:

```
SELECT deptno, MIN(sal)
  FROM emp
  GROUP BY deptno,
```

Nếu hợp nhất phức tạp được phép thì bộ tối ưu hóa biến thể truy vấn này để tìm tất cả những nhân viên có được mức lương tối thiểu đối với bộ phận của họ ở London:

```
SELECT emp.ename, emp.sal
  FROM emp, dept
```

```
  WHERE (emp.deptno, emp.sal) IN min_salary_view
    AND emp.deptno = dept.deptno
    AND dept.deptloc = 'London';
```

Vào truy vấn này (trong đó E1 và E2 biểu thị bảng EMP khi nó được tham chiếu trong khối truy vấn truy xuất và khối truy vấn của khung nhìn ấy một cách tương ứng):

```
SELECT e1.ename, e1.sal
  FROM emp e1, dept, emp e2
  WHERE e1.deptno = dept.deptno
    AND dept.deptloc = 'London'
    AND e1.deptno = e2.deptno
  GROUP BY e1.rowid, dept.rowid, e1.ename, e1.sal
  HAVING e1.sal = MIN(e2.sal);
```

Đẩy thuộc tính vào khung nhìn

Bộ tối ưu hóa có thể biến thể một khối truy vấn để truy xuất một khung nhìn không hợp nhất được bằng cách đẩy những thuộc tính của khối truy vấn ấy vào bên trong truy vấn của khung nhìn đó.

Thí dụ 1:

Xét khung nhìn TWO_EMP_TABLES là sự hợp nhất của hai bảng nhân viên. Khung nhìn ấy được xác định với một truy vấn ghép sử dụng toán tử tập hợp UNION:

```
CREATE VIEW two_emp_tables
  (empno, ename, job, mgr, hiredate, sal, comm, deptno) AS
  SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
    FROM emp1
  UNION
  SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
    FROM emp2;
```

Xét truy vấn này truy xuất khung nhìn ấy. Truy vấn lựa các ID và các tên của tất cả nhân viên trong hai bảng họ làm việc trong phòng 20:

```
SELECT empno, ename
  FROM two_emp_tables
  WHERE deptno = 20;
```

Bởi vì khung nhìn được xác định là một truy vấn ghép, bộ tối ưu hóa không thể hợp nhất truy vấn của khung nhìn vào khối truy vấn truy xuất. Thay vào đó, bộ tối ưu hóa có thể biến thể câu lệnh truy xuất bằng cách đẩy thuộc tính của nó là điều kiện mệnh đề WHERE (DEPTNO = 20) vào truy vấn ghép của khung nhìn.

Câu lệnh kết quả nhìn giống như sau:

```
SELECT empno, ename
  FROM ( SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
           FROM emp1
           WHERE deptno = 20
         UNION
         SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
           FROM emp2
           WHERE deptno = 20 );
```

Nếu có một chỉ mục trên cột DEPTNO, thì các mệnh đề WHERE kết quả tạo cho nó khả dụng.

Hình 20-4 "Truy xuất khung nhìn được xác định với toán tử tập hợp UNION" trình bày kế hoạch tiến hành của câu lệnh kết quả.

Để tiến hành câu lệnh này, Oracle thực hiện những bước sau:

Các Step 5 và 6 thực hiện các lần duyệt toàn bộ của các bảng EMP1 và EMP2.

Step 4 thực hiện phép toán UNION-ALL cho trở lại tất cả các hàng bởi Step 5 hoặc Step 6, kể cả toàn bộ những bản sao chép.

Step 3 sắp xếp kết quả của Step 4 để loại trừ các hàng sao chép.

Step 2 trích các cột yêu cầu từ kết quả của Step 3.
Step 1 chỉ rõ truy vấn của khung nhìn không được hợp nhất vào truy vấn truy xuất.

Thí dụ 2:

Xét khung nhìn EMP_GROUP_BY_DEPTNO có số phòng, mức lương trình bình và mức lương tối đa của tất cả những bộ phận có những nhân viên:

```
CREATE VIEW emp_group by_deptno
  AS SELECT deptno,
        AVG(sal) avg_sal,
        MIN(sal) min_sal,
        MAX(sal) max_sal
    FROM emp
    GROUP BY deptno;
```

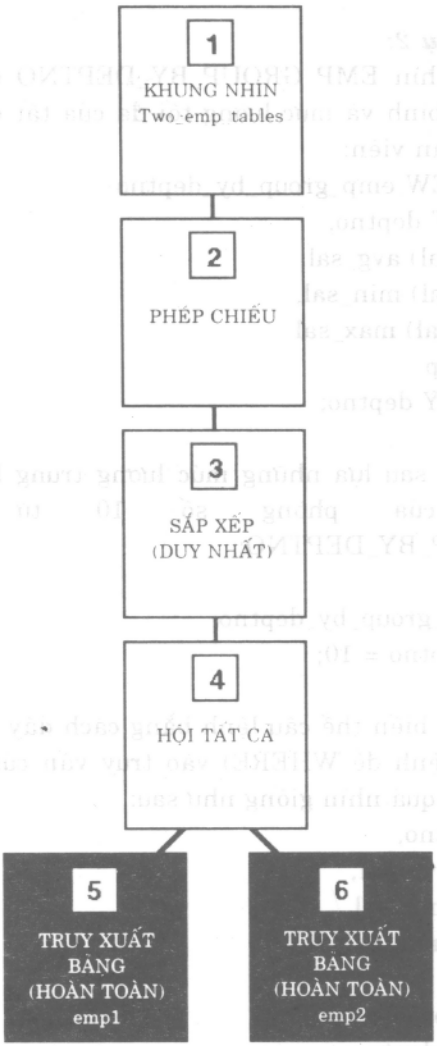
Xét truy vấn sau lựa những mức lương trung bình, tối thiểu và tối đa của phòng số 10 từ khung nhìn EMP GROUP_BY_DEPTNO:

```
SELECT *
  FROM emp_group_by deptno
  WHERE deptno = 10;
```

Bộ tối ưu hóa biến thể câu lệnh bằng cách đẩy thuộc tính của nó (điều kiện mệnh đề WHERE) vào truy vấn của khung nhìn ấy. Câu lệnh kết quả nhìn giống như sau:

```
SELECT deptno,
    AVG(sal) avg_sal,
    MIN(sal) min_sal,
    MAX(sal) max_sal,
  FROM emp
  WHERE deptno = 10
  GROUP BY deptno;
```

Hình 20-4 Truy xuất khung nhìn được xác định với toán tử tập hợp UNION



Nếu có một cập nhật trên cột DEPTNO, thì mệnh đề WHERE kết quả tạo cho nó khả dụng.

Hình 20-5 "Truy xuất khung nhìn được xác định với mệnh đề GROUP BY" trình bày kế hoạch tiến hành đối với câu lệnh kết quả. Kế hoạch tiến hành sử dụng chỉ mục trên cột DEPTNO.

Để tiến hành câu lệnh này, Oracle thực hiện các điều hoạt sau
Step 4 thực hiện duyệt phạm vị trên chỉ mục EMP_DEPTNO_INDEX (chỉ mục trên cột DEPTNO của bảng EMP) để truy hồi các ROWID của tất cả những hàng với DEPTNO giá trị 10.
Step 3 truy xuất bảng EMP sử dụng các ROWID được truy hồi bằng Step 4.
Step 2 sắp xếp các hàng được Step 3 cho trở lại để tính các giá trị SAL trung bình, tối thiểu và tối đa.
Step 1 chỉ rõ truy vấn của khung nhìn ấy đã không được hợp nhất vào truy vấn truy xuất.

Áp dụng một hàm nhóm vào khung nhìn

Bộ tối ưu hóa có thể biến thể truy vấn có hàm chóm (AVG, COUNT, MAX, MIN, SUM) bằng cách áp dụng hàm ấy vào truy vấn của khung nhìn.

Thí dụ:

Xét truy vấn truy xuất khung nhìn EMP_GROUP_BY_DEPTNO được xác định với thí dụ trước. Truy vấn này phái sinh các mức lương trung bình cho phòng có mức lương trung bình, phòng có mức lương tối thiểu và phòng có mức lương tối đa từ bảng nhân viên:

```
SELECT AVG(avg_sal), AVG(min_sal), AVG(max_sal)
   FROM emp_group_by_deptno;
```

Hình 20-5 Truy xuất một khung nhìn được với mệnh đề GROUP BY

Bộ tối ưu hóa biến thể câu lệnh này bằng cách áp dụng hàm nhóm AVG vào danh sách lựa truy vấn của khung nhìn:

```
SELECT AVG(AVG(sal)), AVG(MIN(sal)), AVG(MAX(sal))
  FROM emp
  GROUP BY deptno;
```

Hình 20-6 trình bày kế hoạch tiến hành của câu lệnh kết quả.

Để tiến hành câu lệnh này, Oracle thực hiện những điều hoạt sau:

Step 4 thực hiện duyệt toàn bộ bảng EMP.

Step 3 sắp xếp các hàng được Step 4 cho trở lại vào các nhóm dựa trên những giá trị DEPTNO của chúng và tính giá trị SAL trung bình, tối thiểu và tối đa của mỗi nhóm.

Step 2 chỉ rõ truy vấn của khung nhìn đã không được hợp nhất vào truy vấn truy xuất.

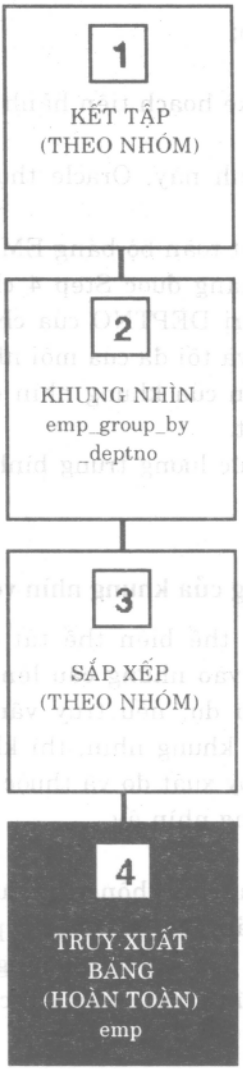
Step 1 tính những mức lương trung bình của những giá trị được Step 2 cho trở lại.

Truy xuất các hàng của khung nhìn với câu lệnh ban đầu

Bộ tối ưu hóa không thể biến thể tất cả những câu lệnh truy xuất các khung nhìn vào những câu lệnh tương đương truy xuất (các) bảng cơ sở. Thí dụ, nếu truy vấn truy xuất một cột giả ROWNUM trong một khung nhìn, thì khung nhìn ấy không thể được hợp nhất vào truy xuất đó và thuộc tính của truy vấn không thể được đẩy vào khung nhìn ấy.

Để tiến hành một câu lệnh không thể được biến thể vào một câu lệnh truy xuất các bảng cơ sở, Oracle phát hành truy vấn của khung nhìn ấy để tập hợp tập hợp những hàng kết quả và sau đó truy xuất tập hợp những hàng này với câu lệnh ban đầu như nó là một bảng.

Hình 20-6 Áp dụng các hàm nhóm vào khung nhìn được xác định với mệnh đề GROUP BY



Thí dụ:

Xét khung nhìn EMP_GROUP_BY_DEPTNO được xác định trong phần trước:

```
CREATE VIEW emp_group_by_deptno
  AS SELECT deptno,
        AVG(sal) avg_sal,
        MIN(sal) min_sal,
        MAX(sal) max_sal
    FROM emp
    GROUP BY deptno;
```

Xét truy vấn sau truy xuất khung nhìn ấy. Truy vấn liên kết những mức lương trung bình, tối thiểu và tối đa từ mỗi bộ phận được biểu thị trong khung nhìn này và vào tên và vị trí của bộ phận trong bảng DEPT:

```
SELECT emp_group_by_deptno.deptno, avg_sal, min_sal,
  max_sal, dname, loc
    FROM emp_group_by_deptno, dept
    WHERE emp_group_by_deptno.deptno = dept.deptno;
```

Vì không có câu lệnh tương đương chỉ truy xuất những bảng cơ sở, bộ tối ưu hóa không thể biến thể câu lệnh này. Thay vào đó, bộ tối ưu hóa chọn một kế hoạch tiến hành để phát hành truy vấn của khung nhìn đó và sau đó sử dụng tập hợp những hàng kết quả như nó là các hàng kết quả từ truy xuất bảng.

Hình 20-7 "Liên Kết Khung Nhìn Được Xác Định Với Mệnh Đề GROUP BY Vào Bảng" trình bày kế hoạch tiến hành cho câu lệnh này. Để có thêm thông tin về Oracle thực hiện các điều hoạt liên kết các vòng lặp lồng nhau như thế nào, xem "Các Điều Hoạt Liên Kết".

Để tiến hành câu lệnh này, Oracle thực hiện những điều hoạt sau:

Step 4 thực hiện duyệt toàn bộ bảng EMP.

Step 3 sắp xếp các kết quả của Step 4 và tính những giá trị trung bình, tối thiểu và tối đa được lựa bằng truy vấn cho khung nhìn EMP GROUP BY_DEPTNO.
Step 2 đa xác định dữ liệu từ hai bước trước cho khung nhìn.
Đối với mỗi hàng được Step 2 cho trở lại, Step 6 sử dụng giá trị DEPTNO để thực hiện duyệt duy nhất chỉ mục PK_DEPT.
Step 5 sử dụng mỗi ROWID được Step 6 cho trở lại để định vị hàng trong bảng DEPTNO với giá trị DEPTNO so khớp.
Oracle kết hợp mỗi hàng được Step 2 cho trở lại với hàng so khớp được Step 5 cho trở lại và cho trở lại kết quả.

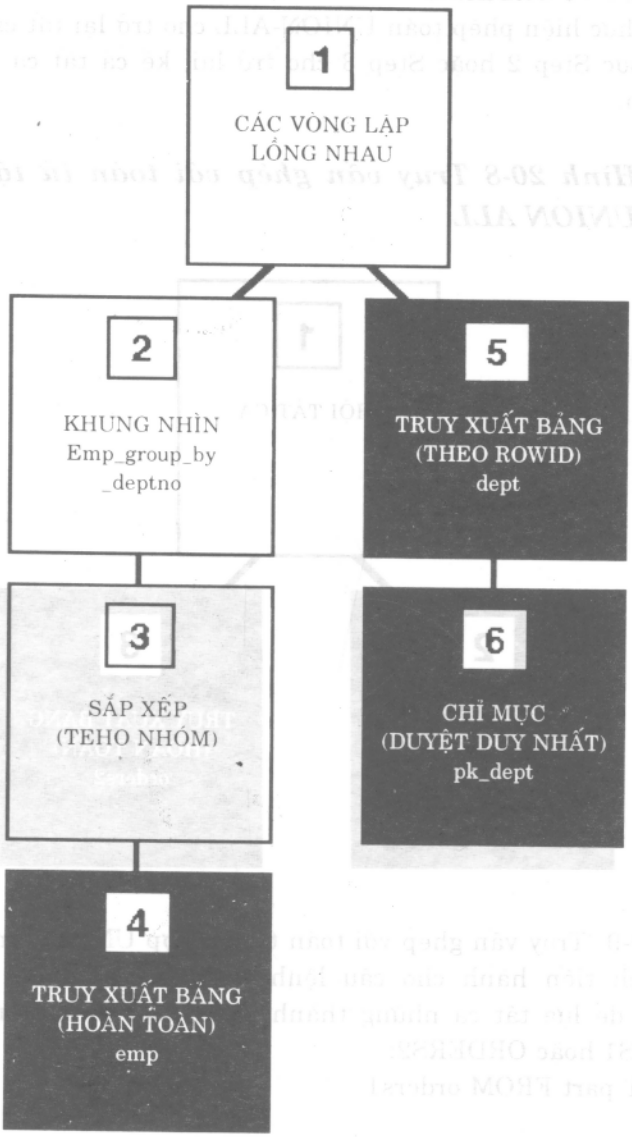
Tối ưu hóa các truy vấn ghép

Để chọn kế hoạch tiến hành đối với truy vấn ghép, bộ tối ưu hóa chọn một kế hoạch tiến hành cho mỗi truy vấn thành phần của nó và sau đó kết hợp các nguồn hàng kết quả với phép toán hội, giao hoặc trừ tùy theo toán tử tập hợp được sử dụng trong truy vấn ghép.

Hình 20-8 "Truy vấn ghép với toán tử tập hợp UNION ALL" trình bày kế hoạch tiến hành cho câu lệnh này, sử dụng toán tử UNION ALL để lựa tất cả các sự kiện của toàn bộ các thành phần trong bảng ORDERS1 hoặc bảng ORDERS2:

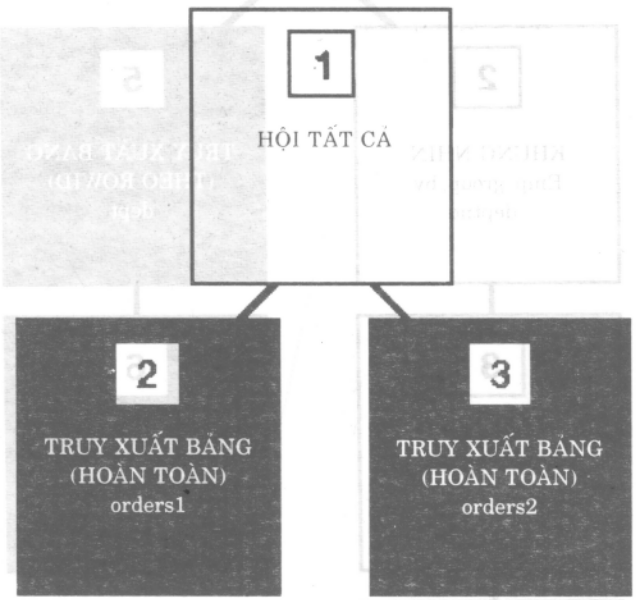
```
SELECT part FROM orders1
UNION ALL
SELECT part FROM orders2;
```

Hình 20-7 Liên kết một khung nhìn được xác định với GROUP BY vào một bảng



Để tiến hành câu lệnh này, Oracle thực hiện những bước sau:
Các Step 2 và 3 thực hiện duyệt toàn bộ bảng trên các bảng ORDERS1 và ORDERS2.
Step 1 thực hiện phép toán UNION-ALL cho trở lại tất cả những hàng được Step 2 hoặc Step 3 cho trở lại, kể cả tất cả các bản sao chép.

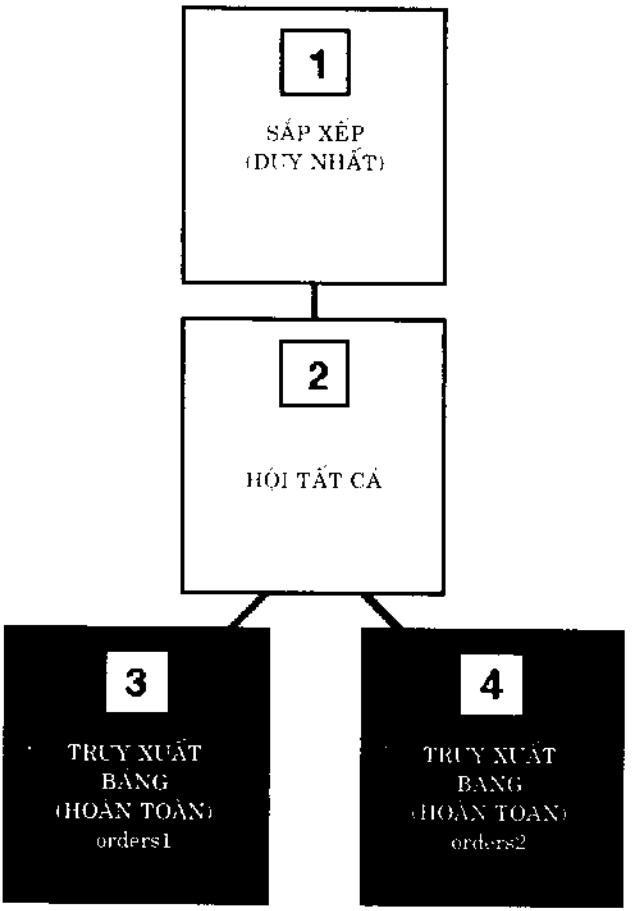
Hình 20-8 Truy vấn ghép với toán tử tập hợp UNION ALL



Hình 20-9 "Truy vấn ghép với toán tử tập hợp UNION" trình bày kế hoạch tiến hành cho câu lệnh theo sau sử dụng toán tử UNION để lựa tất cả những thành phần xuất hiện trong bảng ORDERS1 hoặc ORDERS2:

```
SELECT part FROM orders1
UNION
SELECT part FROM orders2;
```

Hình 20-9 Truy vấn ghép với toán tử tập hợp UNION



Kế hoạch tiến hành này đồng nhất với kế hoạch tiến hành cho toán tử UNION-ALL trình bày ở Hình 20-8 "Truy vấn ghép với toán tử tập hợp UNION ALL", trừ trong trường hợp Oracle sử dụng toán tử SORT để loại trừ các sao chứp được phép toán UNION-ALL cho trở lại.

Hình 20-10 "Truy vấn ghép với toán tử tập hợp INTERSECT" trình bày kế hoạch tiến hành cho câu lệnh này, sử dụng toán tử INTERSECT để chỉ lựa những thành phần xuất hiện trong cả bảng ORDERS1 và ORDERS2:

```
SELECT part FROM orders1
INTERSECT
SELECT part FROM orders2;
```

Để tiến hành câu lệnh này, Oracle thực hiện các bước sau:

Các Step 3 và 5 thực hiện duyệt toàn bộ các bảng ORDERS1 và ORDERS2.

Các Step 2 và 4 sắp xếp các kết quả của các Step 3 và 5 để loại trừ các sao chép trong mỗi nguồn hàng.

Step 1 thực hiện một phép toán INTERSECTION để chỉ được cả Step 2 và 4 cho lại.

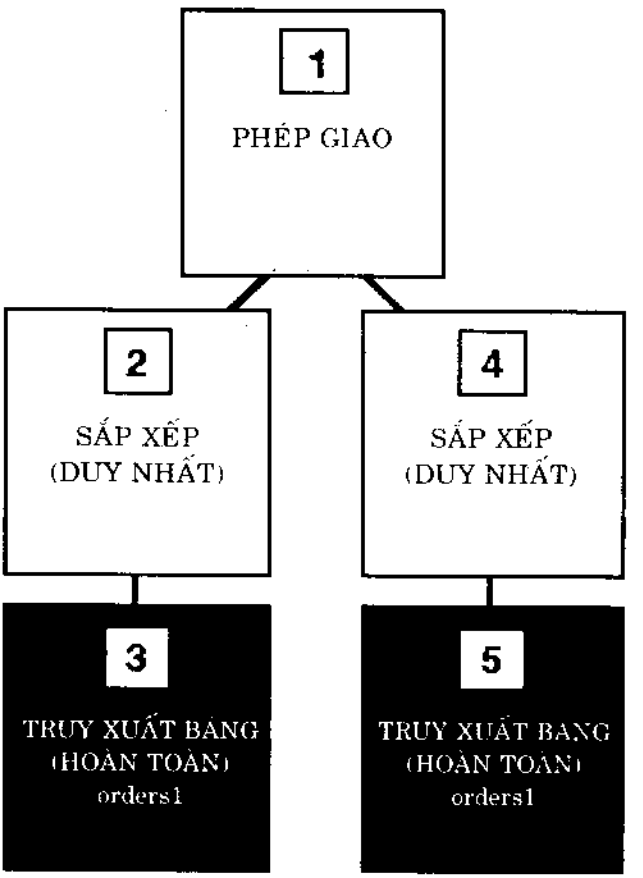
Tối ưu hóa các câu lệnh được phân phối

Bộ tối ưu hóa chọn các kế hoạch tiến hành cho cửa sổ câu lệnh SQL truy xuất dữ liệu trên nhiều cơ sở dữ liệu từ xa theo phương thức nó chọn các tiến hành cho các câu lệnh chỉ truy xuất dữ liệu cục bộ:

Nếu tất cả các bảng được câu lệnh SQL truy xuất được phân bổ trên cùng cơ sở dữ liệu từ xa, Oracle gửi câu lệnh SQL tới cơ sở dữ liệu từ xa đó. Môi trường làm việc Oracle từ xa tiến hành câu lệnh ấy và chỉ gửi các kết quả trở lại cơ sở dữ liệu cục bộ.

Nếu câu lệnh SQL truy xuất các bảng được định vị trên các cơ sở dữ liệu khác nhau, Oracle phân tích câu lệnh ấy thành các phân đoạn cá biệt, mỗi phân đoạn truy xuất các bảng trên một cơ sở dữ liệu đơn. Sau đó Oracle gửi mỗi phân đoạn tới cơ sở dữ liệu mà nó truy xuất. Môi trường làm việc của Oracle từ xa đối với mỗi cơ sở dữ liệu này tiến hành phân đoạn của nó và cho trở lại các kết quả tới cơ sở dữ liệu cục bộ, nơi môi trường làm việc Oracle cục bộ có thể thực hiện bất kỳ xử lý bổ sung câu lệnh cần thiết.

Hình 20-10 Truy vấn ghép với toán tử tập hợp INTERSECT



Khi chọn kế hoạt tiến hành dựa trên phí tổn, bộ tối ưu hóa xét các chỉ mục khả dụng trên những cơ sở dữ liệu từ xa giống như thực hiện các chỉ mục trên cơ sở dữ liệu cục bộ. Bộ tối ưu hóa còn xét các thống kê trên các cơ sở dữ liệu từ xa để tối ưu hóa dựa trên phí tổn. Hơn nữa, bộ tối ưu hóa xét vị trí của dữ liệu khi dự toán phí tổn truy xuất nó. Thí dụ, duyệt toàn bộ một bảng từ xa

có phí tổn dự đoán nhiều hơn duyệt toàn bộ của một bảng cục bộ đồng nhất.

Đối với kế hoạch tiến hành dựa trên nguyên tắc, bộ tối ưu hóa ấy không xét các chỉ mục trên các bảng từ xa.

Chọn cách giải quyết và mục đích tối ưu hóa

Xử lý của bộ tối ưu hóa khi chọn một cách giải quyết và mục đích tối ưu hóa cho câu lệnh SQL bị ảnh hưởng bởi các yếu tố sau:
Tham số khởi hoạt OPTIMIZER_MODE
Các thống kê trong từ điển dữ liệu
Tham số OPTIMIZER_GOAL của lệnh ALTER SESSION
Các gợi ý (các chú thích) trong câu lệnh SQL
Câu lệnh đang được tiến hành trong khối PL/SQL

Tham số khởi hoạt OPTIMIZER_MODE

Tham số khởi hoạt OPTIMIZER_MODE thiết lập xử lý mặc định để chọn một cách giải quyết tối ưu hóa cho môi trường làm việc. Tham số này có thể có những giá trị sau:

CHOOSE	Bộ tối ưu hóa chọn giữa cách giải quyết dựa trên phí tổn và dựa trên nguyên tắc xem các thông kê có thể sử dụng cho cách giải quyết dựa trên phí tổn không. Nếu từ điển dữ liệu có những thông kê cho ít nhất một trong các bảng truy xuất, bộ tối ưu hóa sử dụng cách quản lý dựa trên phí tổn và tối ưu hóa bằng lưu lượng truyền tốt nhất. nếu từ điển dữ liệu không có những thông kê cho bất kỳ bảng được truy xuất. Đây là giá trị mặc định cho tham số ấy.

ALL_ROWS	Bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn cho tất cả các câu lệnh SQL trong phiên làm việc ấy bất kể sự hiện hữu của những thông kê với mục đích lưu lượng truyền tốt nhất (tài nguyên tối thiểu sử dụng để hoàn tất toàn bộ câu lệnh).
FIRST_ROWS	Bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn cho tất cả câu lệnh SQL trong phiên làm việc bất kể sự hiện hữu của những thông kê và tối ưu hóa với mục đích có thời gian trả lời tốt nhất (tài nguyên tối thiểu sử dụng để cho trở lại hàng thứ nhất của tập hợp kết quả).
RULE	Bộ tối ưu hóa chọn cách giải quyết dựa trên nguyên tắc cho tất cả những câu lệnh SQL được phát hành cho môi trường làm việc bất kể sự hiện hữu của những thông kê.

Nếu bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn cho câu lệnh SQL và một số bảng được câu lệnh ấy truy xuất không có thống kê, thì bộ tối ưu hóa sử dụng thông tin nội bộ (chẳng hạn như các khối dữ liệu được phân bổ cho những bảng này) để dựa đoán những thông kê khác cho các bảng này.

Thống kê trong từ điển dữ liệu

Oracle lư thống kê về các cột, các cụm liên kết, các chỉ mục và các phân hoạch trong từ điển dữ liệu để sử dụng bằng bộ tối ưu hóa dựa trên phí tổn (xem "Thống Kê Cho Cách Giải Quyết Dựa Trên Phí Tổn").

Hai tùy chọn của lệnh ANALYZE phát sinh thống kê:
COMPUTE STATISTICS phát sinh thống kê chính xác.

ESTIMATE STATISTICS phát sinh các dự đoán bằng cách lập mẫu dữ liệu.

Thông tin bổ sung:
Xem Oracle Tuning.

Tham số OPTIMIZER_GOAL của lệnh ALTER SESSION

Tham số OPTIMIZER_GOAL của lệnh ALTER SESSION có thể chạy để cách giải quyết tối ưu hóa và mục đích được tham số khởi hoạt OPTIMIZER_MODE thiết lập cho một phiên cá biệt.

Giá trị của tham số tác động đến sự tối ưu hóa của các câu lệnh SQL được phát hành bởi các thủ tục định sẵn và các hàm được gọi trong phiên làm việc ấy, nhưng không ảnh hưởng đến sự tối ưu hóa của các câu lệnh tái quy mà Oracle phát hành trong phiên ấy. Cách giải quyết tối ưu hóa cho các câu lệnh SQL tái quy chỉ tác động bằng giá trị của tham số khởi hoạt OPTIMIZER_MODE.
Tham số OPTIMIZER_GOAL có thể cho những giá trị sau:

CHOOSE	Bộ tối ưu hóa chọn giữa cách giải quyết dựa trên phí tổn và dựa trên nguyên tắc xem thống kê có thể sử dụng cho cách giải quyết dựa trên phí tổn không. Nếu từ điển dữ liệu có thống kê cho ít nhất một trong các bảng được truy xuất, thì bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn và tối ưu hóa với mục đích lưu lượng truyền tốt nhất. Nếu từ điển dữ liệu không có thống kê cho bất kỳ bảng được truy xuất, thì bộ tối ưu hóa sử dụng cách giải quyết dựa trên nguyên tắc.

ALL_ROWS	Bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn cho tất cả các câu lệnh SQL trong phiên làm việc ấy bất kể sự hiện hữu của các thống kê và tối ưu hóa với mục đích lưu lượng truyền tốt nhất (tài nguyên tối thiểu sử dụng để hoàn tất toàn bộ câu lệnh).
FIRST_ROWS	Bộ tối ưu hóa sử dụng cách giải quyết dựa trên phí tổn cho tất cả các câu lệnh SQL trong phiên làm việc bất kể sự hiện hữu của những thống kê và tối ưu hóa với mục đích có thời gian trả lời tốt nhất (tài nguyên tối thiểu sử dụng để cho trở lại hàng thứ nhất của tập hợp kết quả).
RULE	Bộ tối ưu hóa chọn cách giải quyết dựa trên nguyên tắc cho tất cả những câu lệnh SQL tới môi trường làm việc Oracle bất kể sự hiện hữu của những thống kê.

Các gợi ý FIRST_ROWS, ALL_ROWS, CHOOSE và RULE

Gợi ý FIRST_ROWS, ALL_ROWS, CHOOSE hoặc RULE trong một câu lệnh SQL cá biệt có thể chạy đè những tác động của cả hai tham số khởi hoạt OPTIMIZER_MODE
và OPTIMIZER_GOAL của lệnh ALTER SESSION.

Thông tin bổ sung:
Xem Oracle8 Tuning Về Cách Sử Dụng Các Gợi Ý.

PL/SQL và mục đích của bộ tối ưu hóa

Mục đích của bộ tối ưu hóa chỉ áp dụng vào các truy xuất được đệ trình trực tiếp, không phải những truy vấn được đệ trình từ trong PL/SQL.

Câu lệnh ALTER SESSION OPTIMIZER_GOAL không tác động đến SQL được tiến hành từ trong PL/SQL.
PL/SQL bỏ qua tham số khởi hoạt OPTIMIZER_MODE = FIRST_ROWS.

Bạn có thể sử dụng các gợi ý để xác định đường dẫn truy xuất cho các câu lệnh SQL được đệ trình từ trong PL/SQL.

Chọn các đường dẫn truy xuất

Một trong các chọn lựa quan trọng nhất mà bộ tối ưu hóa tạo ra khi lập công thức một kế hoạch tiến hành là cách truy hồi dữ liệu từ cơ sở dữ liệu. Đối với hàng bất kỳ trong bảng bất kỳ được câu lệnh SQL truy xuất, có thể có nhiều đường dẫn truy xuất mà hàng đó có thể định vị và truy hồi. Bộ tối ưu hóa chọn trong số chung.

Phần này đề cập đến:
Ccspp căn bản Oracle có thể truy xuất dữ liệu
Mỗi đường dẫn truy xuất và khi bộ tối ưu hóa có thể sử dụng nó
Bộ tối ưu hóa chọn giữa các đường dẫn truy xuất khả dụng như thế nào

Các phương pháp truy xuất

Phần này mô tả các phương pháp căn bản Oracle có thể truy xuất dữ liệu.

Duyệt toàn bộ bảng

Duyệt toàn bộ bảng truy hồi các hàng từ một bảng. Để thực hiện duyệt toàn bộ bảng, Oracle đọc tất cả các hàng trong bảng ấy, khảo sát mỗi dòng để xác định xem nó có thỏa mãn mệnh đề WHERE của câu lệnh ấy không. Oracle đọc mọi khối dữ liệu được phân bổ vào bảng ấy liên tiếp. Vì vậy duyệt toàn bộ bảng có thể

được thực hiện rất hiệu quả sử dụng đọc dữ liệu đồng thời. Oracle đọc mỗi khối dữ liệu chỉ một lần.

Truy xuất bảng ROWID

Truy xuất bảng bằng ROWID cũng truy xuất từ một bảng. ROWID của hàng xác định kiểu dữ liệu và khối dữ liệu chứa hàng ấy và vị trí của hàng trong khối đó. Định vị hàng bằng ROWID của nó là cách nhanh nhất cho Oracle để tìm một hàng đơn.

Để truy xuất một bảng bằng ROWID, trước hết Oracle có được các ROWID của các hàng được lựa từ mệnh đề WHERE của câệnnnnh hoặc bằng cách duyệt chỉ mục của một hoặc nhiều chỉ mục của bảng. Sau đó Oracle định vị mỗi hàng được lựa trong bảng dựa trên ROWID của nó.

Duyệt cụm liên kết

Từ một bảng được lưu trữ trong cụm liên kết có chỉ mục, duyệt cụm liên kết truy hồi các hàng có cùng giá trị khóa cụm liên kết. Trong cụm liên kết có chỉ mục, tất cả các hàng có cùng giá trị khóa cụm liên kết được lưu trữ trong cùng các khối dữ liệu. Để thực hiện duyệt cụm liên kết, trước hết Oracle có được ROWID của một trong các hàng được lựa bằng cách duyệt chỉ mục cụm liên kết. Sau đó Oracle định vị các hàng dựa trên ROWID này.

Duyệt trị số

Oracle có thể sử dụng duyệt trị số để định vị các hàng trong cụm liên kết trị số dựa trên giá trị số. Trong cụm liên kết trị số, tất cả các hàng có cùng giá trị trị số được lưu trữ trong cùng những khối dữ liệu. Để thực hiện duyệt trị số, trước hết Oracle có được giá trị số ấy bằng cách áp dụng hàm trị số vào giá trị khóa cụm liên kết được câu lệnh ấn định. Sau đó Oracle duyệt các khối dữ liệu có các hàng với giá trị số đó.

Duyệt chỉ mục

Duyệt chỉ mục truy hồi dữ liệu từ một chỉ mục dựa trên giá trị của một hoặc nhiều cột của chỉ mục ấy. Để thực hiện duyệt chỉ mục, Oracle tìm chỉ mục ấy để có các giá trị cột được câu lệnh truy xuất. Nếu câu lệnh ấy chỉ truy xuất các cột của chỉ mục, thì Oracle đọc những giá trị cột có chỉ mục trực tiếp từ chỉ mục ấy thay vì từ bảng.

Chỉ mục không những có giá trị chỉ mục, mà còn các ROWID của các hàng trong bảng có giá trị đó. Vì thế, nếu câu lệnh truy xuất các cột khác bổ sung vào các cột có chỉ mục, Oracle có thể tìm các hàng trong bảng với truy xuất bảng bằng ROWID hoặc duyệt cụm liên kết.

Duyệt chỉ mục có thể là một trong các kiểu sau:

Duyệt duy nhất	Duyệt duy nhất một chỉ mục chỉ cho trở lại một ROWID đơn. Oracle thực hiện duyệt duy nhất chỉ trong trường hợp một ROWID đơn cần thiết, thay vì nhiều ROWID. Thí dụ, Oracle tùy chọn duyệt duy nhất nếu có một hạn định UNIQUE hoặc PRIMARY KEY bảo đảm câu lệnh ấy chỉ truy xuất một hàng đơn.
Duyệt phạm vi	Duyệt phạm vi của một chỉ mục có thể cho trở lại zêrô hoặc nhiều ROWID tùy theo câu lệnh truy xuất bao nhiêu hàng.
Duyệt toàn bộ	Duyệt toàn bộ chỉ mục có thể sử dụng nếu một tối ưu hóa tham chiếu một trong các cột trong chỉ mục ấy. Tối ưu hóa không phải là một trình điều khiển chỉ mục. Duyệt toàn bộ còn có thể sử dụng khi không có thuộc tính nếu tất cả các cột trong bảng được tham chiếu trong truy vấn có trong chỉ mục và ít nhất một trong các cột không thể rỗng. Duyệt toàn

	bộ có thể được sử dụng để loại trừ phép toán sắp xếp. Nó đọc từng khối.
Duyệt toàn bộ nhanh	Duyệt chỉ mục toàn bộ nhanh là thay cho duyệt toàn bộ bảng khi chỉ mục có tất cả các cột cần thiết cho truy vấn và ít nhất một cột trong chỉ mục có hạn định NOT NULL. Qứt toàn bộ nhanh truy xuất dữ liệu trong chính chỉ mục mà không truy xuất bảng. Nó không thể được sử dụng để loại từ phép toán sắp xếp. Nó đọc toàn bộ chỉ mục sử dụng đọc dữ liệu đồng thời (không giống như duyệt toàn bộ chỉ mục) và có thể được song song hóa. Duyệt toàn bộ nhanh chỉ có thể sử dụng với sự tối ưu hóa dựa trên phí tổn. bạn có thể ấn định nó với tham số khởi hoạt FAST_FULL_SCAN_ENABLED hoặc gợi ý INDEX_FFS.
Bitmap	Các chỉ mục bitmap sử dụng bitmap cho các giá trị khóa và ánh xạ hàm chuyển đổi mỗi vị trí bit vào ROWID. Các bitmap hợp nhất các chỉ mục hiệu quả để đáp ứng nhiều điều kiện trong mệnh đề WHERE, sử dụng các phép toán Boolean để giải quyết các điều kiện AND và OR. Truy xuất bitmap chỉ có thể sử dụng với tối ưu hóa dựa trên phí tổn. Xem "Các Chỉ Mục Bitmap".

Lưu ý:
Các chỉ mục bitmap chỉ có thể sử dụng nếu bạn có Oracle Enterprise Edition. Xem "Getting to Know Oracle8 va Oracle8 Enterprise Edition".

Các đường truy xuất

Hình 20-1 liệt kê các đường dẫn truy xuất. Bộ tối ưu hóa chỉ có thể sử dụng đường dẫn truy xuất cá biệt đối với bảng nếu câu

lệnh có điều kiện mệnh đề WHERE hoặc cấu trúc khác để tạo đường dẫn truy xuất khả dụng.
Cách giải quyết dựa trên phí tổn chọn đường dẫn dựa trên tài nguyên sử dụng (xem "Chọn Trong Các Đường Dẫn Truy Xuất Với Cách Giải Quyết Dựa Trên Phí Tổn".
Cách giải quyết dựa trên nguyên tắc sử dụng cấp của mỗi đường dẫn để chọn đường dẫn khi có thể sử dụng nhiều đường dẫn (xem "Chọn Trong Các Đường Dẫn Truy Xuất Với Cách Giải Quyết Dựa Nguyên Tắc").

Bảng 20-1 Các đường dẫn truy xuất

Cấp	Đường dẫn truy xuất
1	Một hàng đơn bằng ROWID
2	Hàng đơn bằng liên kết cụm liên kết
3	Hàng đơn bằng khóa cụm liên kết trị số với khóa duy nhất hoặc khóa chính
4	Hàng đơn bằng khóa duy nhất hoặc khóa chính
5	Liên kết cụm liên kết
6	Khóa cụm liên kết trị số
7	Khóa cụm liên kết có chỉ mục
8	Khóa tổng hợp
9	Các chỉ mục cột đơn
10	Tìm phạm vi biên trên các cột có chỉ mục
11	Tìm phạm vi không có biên trên các cột có chỉ mục
12	Liên kết hợp nhất-sắp xếp
13	Cột có chỉ mục MAX hoặc MIN
14	ORDER BY trên các cột có chỉ mục
15	Duyệt toàn bộ bảng
Các Đường Dẫn Truy Xuất Không Có Cấp	
	Duyệt toàn bộ chỉ mục nhanh (không thể sử dụng với bộ tối ưu hóa dựa trên nguyên tắc): xem

	Oracle8 Tuning
-	Duyệt chỉ mục bitmap (không thể sử dụng với bộ tối ưu hóa dựa trên nguyên tắc): xem "Các Chỉ Mục Bitmap"

Mỗi phần sau mô tả một đường dẫn truy xuất và:
Khi nó có thể sử dụng
Phương pháp Oracle sử dụng để truy xuất dữ liệu với nó
Kết xuất được phát sinh cho nó bằng lệnh EXPLAIN PLAN

Đường dẫn 1: hàng đơn bằng ROWID

Đường dẫn truy xuất này chỉ có thể sử dụng nếu mệnh đề WHERE của câu lệnh nhận dạng các hàng được ROWID lựa hoặc với cú pháp SQL nhúng CURRENT OF CURSOR được các Oracle Precompiler hỗ trợ. Để tiến hành câu lệnh ấy, Oracle truy xuất bảng bằng ROWID.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau:

```
SELECT * FROM emp WHERE ROWID =
'AAAA7bAA5AAAA1UAAA';
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS         BY ROWID      EMP
```

Đường dẫn 2: hàng đơn bằng liên kết cụm liên kết

Đường dẫn này có thể sử dụng cho các câu lệnh liên kết các bảng được lưu trữ trong cùng cụm liên kết nếu cả hai điều kiện sau đúng:

- Mệnh đề WHERE của câu lệnh bằng mỗi cột của khóa cụm liên kết với cột tương ứng trong bảng khác.
- Mệnh đề WHERE của câu lệnh có điều kiện bảo đảm là liên kết chỉ cho trở lại một hàng. Điều kiện như thế có thể là một điều kiện bằng trên (các) cột của một khóa duy nhất hoặc khóa chính.

Những điều kiện này phải được kết hợp với các toán tử AND. Để tiến hành câu lệnh ấy, Oracle thực hiện các phép toán vòng lặp lồng nhau. (Để có thông tin về các phép toán vòng lặp lồng nhau, xem "Các Phép Toán Liên Kết".)

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh sau trong đó các bảng EMP và DEPT được liên kết trên cột DEPTNO và EMPNO là khóa chính của bảng EMP:

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = dept.deptno
    AND emp.empno = 7900;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION          OPTIONS          OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
   NESTED LOOPS
      TABLE ACCESS       BY ROWID       EMP
      INDEX              UNIQUE SCAN    PK_EMP
      TABLE ACCESS       CLUSTER        DEPT
```

PK_EMP là tên của chỉ mục để tăng cường khóa chính.

Đường dẫn 3: hàng đơn bằng khóa cụm liên kết trị số với khóa duy nhất hoặc khóa chính

Đường dẫn này có thể sử dụng nếu cả hai điều kiện sau đúng:

- Mệnh đề WHERE của câu lệnh sử dụng tất cả các cột của khóa cụm liên kết trị số trong các điều kiện bằng nhau. Đối với các khóa cụm tổng hợp, thì các điều kiện bằng nhau phải được kết hợp với các toán tử AND.
- Câu lệnh được bảo đảm chỉ cho lại một hàng vì các cột thiết lập khóa cụm liên kết trị số cũng chỉ thiết lập một khóa duy nhất hoặc khóa chính.

Để tiến hành câu lệnh ấy, Oracle áp dụng hàm trị số của cụm liên kết ấy vào giá trị khóa cụm liên kết trị số được ấn định trong câu lệnh để có được giá trị trị số. Sau đó Oracle sử dụng giá trị số để thực hiện duyệt trị số trên bảng.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau, trong đó các bảng ORDERS và LINE_ITEMS được lưu trữ trong cụm liên kết trị số và cột ORDERNO vừa là khóa cụm liên kết và khóa chính của bảng ORDERS:

```
SELECT *
   FROM orders
   WHERE orderno = 65118968;
```

Kết suất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS      OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
   TABLE ACCESS        HASH         ORDERS
```

Đường dẫn 4: hàng đơn bằng khóa duy nhất hoặc khóa chính

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh sử dụng tất cả các cột của khóa duy nhất hoặc khóa chính trong các điều kiện bằng nhau. Đối với các khóa tổng hợp, các điều kiện bằng nhau phải được liên kết với các phép toán AND. Để tiến hành câu lệnh ấy, Oracle thực hiện duyệt duy nhất trên chỉ mục hoặc khóa chính để truy hồi ROWID đơn và sau đó truy xuất bảng bằng ROWID đó.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau trong đó cột EMPNO là khóa chính của bảng EMP:

```
SELECT *
   FROM emp
   WHERE empno = 7900;
```

Kết suất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION                OPTIONS        OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS           BY ROWID       EMP
    INDEX                UNIQUE SCAN    PK_EMP
```

PK_EMP là tên của chỉ mục để tăng cường khóa cính.

Đường dẫn 5: liên kết cụm liên kết

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh liên kết các bảng trong cùng cụm liên kết nếu mệnh đề WHERE của câu lệnh có các điều kiện bằng với mỗi cột của khóa cụm liên kết trong một bảng với cột tương ứng trong bảng khác. Đối với khóa cụm liên kết tổng hợp, các điều kiện bằng nhau phải được kết hợp với các toán tử AND. Để tiến hành câu lệnh ấy, Oracle thực

hiện các phép toán vòng lặp lồng nhau, xem "Các Phép Toàn Liên Kết").

Thí dụ:

Đường dẫn này có thể sử dụng trong câu lệnh sau, trong đó các bảng EMP và DEPT được liên kết trên cột DEPTNO:

```
SELECT *
   FROM emp, dept
   WHERE emp.deptno = dept.deptno;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
  NESTED LOOPS
    TABLE ACCESS       FULL          DEPT
    TABLE ACCESS       CLUSTER       EMP
```

Đường dẫn 6: khóa cụm liên kết trị số

Đường dẫn này có thể sử dụng nếu mệnh đề WHERE của câu lệnh sử dụng tất cả các cột của khóa cụm liên kết trị số trong các điều kiện bằng nhau. Đối với khóa cụm liên kết tổng hợp, các điều kiện bằng nhau phải được kết hợp với các toán tử AND. Để tiến hành cậệnnnnh ấy, Oracle áp dụng hàm trị số của cụm liên kết vào giá trị khóa cụm liên kết trị số được xác định trong câu lenẹh để có được giá trị số. Sau đó Oracle sử dụng giá trị số này để thực hiện duyệt trị số trên bảng.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh sau tron gdó các bảng ORDERS và LINE_ITEMS được lưu trữ trong cụm liên kết trị số và cột ORDERNO là khóa cụm liên kết:

```
SELECT *
   FROM line items
   WHERE orderno = 65118968;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------------
SELECT STATEMENT
   TABLE ACCESS        HASH          LINE_ITEMS
```

Đường dẫn 7: khóa cụm liên kết có chỉ mục

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh sử dụng tất cả các cột của khóa cụm liên kết có chỉ mục trong các điều kiện bằng nhau. Đối với khóa cụm liên kết tổng hợp, các điều kiện bằng nhau phải được kết hợp với các toán tử AND. Để tiến hành câu lệnh ấy, Oracle thực hiện duyệt duy nhất trên chỉ mục cụm liên kết để truy hồi ROWID của một hàng với giá trị khóa cụm liên kết được xác định. Sau đó Oracle sử dụng ROWID đó để truy xuất bảng với duyệt cụm liên kết. Vì tất cả các hàng có cùng giá trị khóa cụm liên kết được lưu trữ với nhau, cụm ấy có thể chỉ đòi hỏi ROWID đơn để tìm tất cả chúng.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau, trong đó bảng EMP được lưu trữ trong một cụm liên kết có chỉ mục và cột DEPTNO là khóa cụm liên kết:

```
SELECT *  FROM emp
  WHERE deptno = 10;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------------
```

```
SELECT STATEMENT
  TABLE ACCESS          CLUSTER       EMP
    INDEX             UNIQUE SCAN   PERS_INDEX
```

PERS_INDEX là tên tên của chỉ mục cụm liên kết.

Đường dẫn 8: chỉ mục tổng hợp

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh sử dụng tất cả các cột của chỉ mục tổng hợp trong các điều kiện bằng nhau được kết hợp với các toán tử AND. Để tiến hành câu lệnh ấy, Oracle thực hiện duyệt phạm vi trên chỉ mục ấy để truy hồi các ROWID của các hàng được lựa và sau đó truy xuất bảng bằng các ROWID đó.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau trong đó có một chỉ mục tổng hợp trên các cột JOB và DEPTNO:

```
SELECT *
   FROM emp
   WHERE job = 'CLERK'
     AND deptno = 30;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS           BY ROWID      EMP
    INDEX              RANGE SCAN    JOB_DEPTNO_INDEX
```

JOB_DEPTNO_INDEX là tên của chỉ mục tổng hợp trên các cột JOB và DEPTNO.

Đường dẫn 9: các chỉ mục cột đơn

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh sử dụng các cột của một hoặc nhiều chỉ mục cột đơn trong các điều kiện bằng nhau. Đối với nhiều chỉ mục cột đơn, các điều kiện phải được kết hợp với các toán tử AND.

Nếu mệnh đề WHERE sử dụng cột chỉ có một chỉ mục, Oracle tiến hành câu lệnh bằng cách thực hiện duyệt phạm vi trên chỉ mục ấy để truy hồi các ROWID của các hàng được lựa và sau đó xử lý bảng bằng các ROWID này.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau trong đó có một chỉ mục trên cột JOB của bảng EMP:

```
SELECT *
   FROM emp
   WHERE job = 'ANALYST';
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS         BY ROWID      EMP
    INDEX              RANGE SCAN    JOB_INDEX
```

JOB_INDEX là chỉ mục trên EMP.JOB.

Nếu các mệnh đề WHERE sử dụng các cột của nhiều chỉ mục cột đơn, Oracle tiến hành câu lệnh ấy bằng cách thực hiện duyệt phạm vi trên mỗi chỉ mục để truy hồi các ROWID của các hàng thỏa mãn mỗi điều kiện. Sau đó Oracle hợp nhất các tập hợp của các ROWID để có được một tập hợp của các ROWID của các hàng thỏa mãn tất cả các điều kiện. Sau đó Oracle xử lý bảng sử dụng các ROWID này.

Oracle có thể hợp nhất tối đa 5 chỉ mục. Nếu mệnh đề WHERE sử dụng các cột của hơn 5 chỉ mục cột đơn, thì Oracle hợp nhất 5 chỉ mục của chúng, truy xuất bảng bằng ROWID và sau đó kiểm tra các hàng kết quả để xác định xem chúng có thỏa mãn các điều kiện còn lại trước khi cho chúng trở lại.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau, trong đó có các chỉ mục trên cả hai cột JOB và DEPTNO của bảng EMP:

```
SELECT *
   FROM emp
   WHERE job = 'ANALYST'
     AND deptno = 20;
```

Kết xuất EXPLAIN PLAN cho câệnnnnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
 TABLE ACCESS          BY ROWID      EMP
   AND-EQUAL
     INDEX             RANGE SCAN    JOB_INDEX
     INDEX             RANGE SCAN    DEPTNO_INDEX
```

Phép toán AND-EQUAL hợp nhất các ROWID có được bằng cách duyệt JOB INDEX và DEPTNO INDEX cho kết quả trong một tập hợp các ROWID của các hàng thỏa mãn truy vấn ấy.

Đường dẫn 10: tìm phạm vị có biên trên các cột có chỉ mục

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh có điều kiện sử dụng cột của chỉ mục cột đơn hoặc một hoặc nhiều cột thiết lập phần đầu của chỉ mục tổng hợp:

column = expr

column >[=] expr AND column <[=] expr

column BETWEEN expr AND expr

column LIKE 'c%'

Mỗi điều kiện này ấn định một phạm vi biên của những chỉ mục được câu lệnh ấy xử lý. Phạm vi được coi là biên vì các điều kiện ấn định cả giá trị nhỏ nhất và lớn nhất của nó. Để tiến hành một câu lệnh như thế, Oracle thực hiện duyệt phạm vi trên chỉ mục và sau đó xử lý bảng bằng ROWID.

Đường dẫn truy xuất không thể sử dụng nếu biểu thức expr tham chiếu cột có chỉ mục.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh này, trong đó có một cập nhật trên cột SAL của bảng EMP:

```
SELECT *
   FROM emp
   WHERE sal BETWEEN 2000 AND 3000;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
  TABLE ACCESS          BY ROWID      EMP
    INDEX              RANGE SCAN    SAL_INDEX
```

SAL_INDEX là tên của chỉ mục trên EMP.SAL.

Thí dụ:

Đường dẫn truy xuất này còn có thể sử dụng trong câu lệnh sau, trong có một chỉ mục trên cột ENAME của bảng EMP:

```
SELECT *
   FROM emp
   WHERE ename LIKE 'S%';
```

Đường dẫn 11: tìm phạm vi không có biên trên các cột có chỉ mục

Đường dẫn truy xuất này có thể sử dụng nếu mệnh đề WHERE của câu lệnh có một trong những điều kiện sau sử dụng cột của chỉ mục cột đơn hoặc một hoặc nhiều cột của phần đầu của chỉ mục tổng hợp:

```
WHERE column >[=] expr

WHERE column <[=] expr
```

Mỗi điều kiện này ấn định một phạm vi không có biên của những giá trị chỉ mục được câu lệnh truy xuất. Phạm vi được coi là không có biên vì điều kiện ấn định giá trị nhỏ nhất ro lớn nhất của nó, nhưng không phải cả hai. Để tiến hành một câu lệnh như thế, Oracle tiến hành một duyệt phạm trên chỉ mục và sau đó truy xuất bảng bằng ROWID.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau, trong đó có một chỉ mục trên cột SAL của bảng EMP:

```
SELECT *
   FROM emp
   WHERE sal > 2000;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION                OPTIONS        OBJECT_NAME
```

```
---------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS          BY ROWID     EMP
    INDEX               RANGE SCAN   SAL_INDEX
```

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng trong câu lệnh sau, trong đó có một chỉ mục tổng hợp trên các cột ORDER và LINE của bảng LINE_ITEMS:

```
SELECT *
   FROM line_items
   WHERE order > 65118968;
```

Đường dẫn truy xuất có thể sử dụng vì mệnh đề WHERE sử dụng cột ORDER, phần đầu của cm.

Thí dụ:

Đường dẫn truy xuất này không thể sử dụng trong câu lệnh sau, trong đó có một chỉ mục trên các cột ORDER và LINE:

```
SELECT *
   FROM line_items
   WHERE line < 4;
```

Đường dẫn truy xuất ấy không thể sử dụng vì mệnh đề WHERE chỉ sử dụng cột LINE, không phải là phần đầu của chỉ mục.

Đường dẫn 12: liên kết sắp xếp - hợp nhất

Đường dẫn truy xuất có thể sử dụng cho các câu lệnh liên kết các bảng không được lưu trữ cùng với nhau trong một cụm liên kết nếu mệnh đề WHERE của câu lệnh sử dụng các cột từ mỗi bảng trong các điều kiện bằng nhau. Để tiến hành một câu lệnh như thế, Oracle sử dụng phép toán sắp xếp - hợp nhất. Oracle cũng có thể sử dụng phép toán vòng lặp lồng nhau để tiến hành một

câu lệnh liên kết. (Để có thêm thông tin về các phép toán này, xem "Tối Ưu Hóa Các Câu Lệnh Liên Kết".)

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh sau trong đso các bảng EMP và DEPT không được lưu trữ trong cùng cụm liên kết:

```
SELECT *
   FROM emp, dept
   WHERE emp.deptno = dept.deptno;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS      OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
 MERGE JOIN
  SORT                 JOIN
   TABLE ACCESS        FULL         EMP
  SORT                 JOIN
 TABLE ACCESS          FULL         DEPT
```

Đường dẫn 13: cột có chỉ mục MAX hoặc MIN

Đường dẫn này có thể sử dụng cho câu lệnh SELECT khi tất cả các điều kiện đều đúng:

Truy vấn sử dụng hàm MAX hoặc MIN để lựa giá trị cực đại và cực tiểu của cột có chỉ mục cột đơn hoặc cột đầu có chỉ mục tổng hợp. Chỉ mục không thể là một chỉ mục cụm liên kết. Đối số đối với hạm MAX hoặc MIN có thể là một biểu thức bất kỳ liên quan đến cột, hằng số hoặc toán tử cộng (+), phép toán ghép (||) hoặc hàm CONCAT.

Không có các biểu thức trong danh sách lựa.

Câu lệnh không có mệnh đề WHERE hoặc mệnh đề GROUP BY.

Để tiến hành truy vấn ấy, Oracle thực hiện duyệt phạm vi của chỉ mục để tìm giá trị có chỉ mục cực đại và cực tiếp. Vì chỉ giá trị này được lựa, Oracle không cần truy xuất bảng sau khi duyệt chỉ mục ấy.

Thí dụ:

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh sau, trong đó có một chỉ mục trên cột SAL của bảng EMP:

```
SELECT MAX(sal) FROM emp;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-----------------------------------------------
SELECT STATEMENT
  AGGREGATE              GROUP BY
    INDEX              RANGE SCAN    SAL_INDEX
```

Đường dẫn 14: ORDER BY trên cột có chỉ mục

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh SELECT khi tất cả các câu lệnh đều đúng:

Truy vấn có mệnh đề ORDER BY sử dụng cột có chỉ mục cột đơn hoặc phần đầu của chỉ mục tổng hợp. Chỉ mục không thể là một chỉ mục cụm liên kết.

Phải có hạn định tính nguyên vẹn là PRIMARY KEY hoặc NOT NULL để bảo đảm ít nhất một trong các cột có chỉ mục được liệt kê trong mệnh đề ORDER BY có các giá trị rỗng.

Tham số NLS_SORT được xác lập là BINARY.

Để tiến hành truy vấn ấy, Oracle thực hiện duyệt phạm vi của chỉ mục để truy hồi các ROWID của các hàng được lựa theo thứ tự có được sắp xếp. Sau đó Oracle truy xuất bảng bằng các ROWID này.

Thí dụ:

Đường dẫn truy xuất có thể sử dụng cho câu lệnh sau, trong đó có một phím chính trên cột EMPNO của bảng EMP:

```
SELECT *
   FROM emp
   ORDER BY empno;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS        OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS           BY ROWID       EMP
    INDEX              RANGE SCAN     PK_EMP
```

PK_EMP là tên của chỉ mục tăng cường khóa chính. Khóa chính bảo đảm cột ấy không có các giá trị rỗng.

Đường dẫn 15: duyệt toàn bộ bảng

Đường dẫn truy xuất này có thể sử dụng cho câu lệnh SQL bất kỳ, bất kể các điều kiện của mệnh đề WHERE.

Lưu ý những điều kiện này làm cho các đường dẫn chỉ mục sử dụng được:

```
column1 > column2
column1 < column2
column1 >= column2
column1 <= column2
```

trong đó column1 và column2 ở trong cùng bảng.

```
column IS NULL
column IS NOT NULL
column NOT IN
column != expr
column LIKE '%pattern'
```

bất kể xem cột có chỉ mục hay không.
expr = expr2

trong đó expr là một biểu thức điều tác trên cột với một toán tử hoặc một hàm, bất kể cột có chỉ mục hay không.
Truy vấn con NOT EXISTS
Cột giả ROWNUM trong khung nhìn
Điều kiện bất kỳ liên quan đến một cột không có chỉ mục.

Câu lệnh SQL bất kỳ chỉ có các cấu trúc và các câu lệnh khác làm cho các đường dẫn m khả dụng phải sử dụng duyệt toàn bộ bảng.

Thí dụ:

Câu lệnh này sử dụng duyệt toàn bộ bảng để truy xuất bảng EMP:

```
SELECT *
   FROM emp;
```

Kết xuất EXPLAIN PLAN cho câu lệnh này có thể nhìn giống như sau:

```
OPERATION              OPTIONS       OBJECT_NAME
-------------------------------------------------
SELECT STATEMENT
  TABLE ACCESS         FULL          EMP
```

Chọn giữa các đường dẫn

Phần này mô tả bộ tối ưu hóa chọn giữa các đường dẫn khả dụng như thế nào:

- khi sử dụng cách giải quyết dựa trên phí tổn
- khi sử dụng cách giải quyết dựa trên nguyên tắc

Chọn giữa các đường dẫn truy xuất với cách giải quyết dựa trên phí tổn

Với cách giải quyết dựa trên phí tổn, bộ tối ưu hóa chọn một đường dựa trên các yếu tố sau:
các đường dẫn truy xuất khả dụng cho câu lệnh
phí tổn dự đoán tiến hành câu lệnh sử dụng mỗi đường dẫn truy xuất hoặc tổ hợp của các đường dẫn

Để chọn một đường dẫn truy xuất, bộ tối ưu hóa trước hết xác định những đường dẫn truy xuất nào có thể sử dụng bằng cách khảo sát các điều kiện trong mệnh đề WHERE của câu lệnh. Sau đó bộ tối ưu hóa phát sinh một tập hợp các kế hoạch tiến hành khả dụng và dự toán phí tổn của mỗi kế hoạch sử dụng những thống kê cho chỉ mục, các cột và các bảng truy xuất câu lệnh ấy. Sau đó bộ tối ưu hóa chọn kế hoạch tiến hành với phí tổn dự đoán thấp nhất.

Sự chọn lựa của bộ tối ưu hóa giữa các đường dẫn truy xuất khả dụng có thể được chạy đè bằng các gợi ý.

> Thông tin bổ sung:
> Xem Oracle8 Tuning về các gợi ý trong những câu lệnh SQL.

Để chọn giữa các đường dẫn truy xuất khả dụng, bộ tối ưu hóa xét các yếu tố sau:
Selectivity: tính chọn lọc là số phần trăm các hàng trong bảng mà truy vấn lựa. Một truy vấn lựa một số phần trăm nhỏ các hàng của bảng có tính chọn lọc tốt, trong khi truy vấn lựa số phần trăm lớn các hàng có tính chọn lọc kém.

> Bộ tối ưu hóa có thể chọn lựa duyệt chỉ mục so với duyệt toàn bộ bảng đối với một truy vấn có tính chọn lọc tốt hơn đối với một truy vấn có tính chọn lọc kém. Duyệt chỉ mục thường hiệu quả hơn duyệt toàn bộ bảng đối với các truy vấn chỉ truy xuất số phần trăm nhỏ các hàng của

bảng, trong khi duyệt toàn bộ bảng thường nhanh hơn đối với những truy vấn truy xuất số phần trăm lớn hơn.

Để xác định tính chọn lọc của một truy vấn, bộ tối ưu hóa xét các nguồn thông tin sau:

Các toán tử được sử dụng trong mệnh đề WHERE

Các cột có khóa duy nhất và khóa chính được sử dụng trong mệnh đề WHERE

Những thống kê cho bảng

Các thí dụ sau minh họa bộ tối ưu hóa sử dụng tính chọn lọc như thế nào.

DB_FILE_MULTIBLOCK_READ_COUNT: duyệt toàn bộ bảng sử dụng đọc dữ liệu đồng thời, vì vậy phí tổn của duyệt toàn bộ bảng phụ thuộc vào số lần đọc dữ liệu đồng thời cần thiết để đọc toàn bộ bảng, phụ thuộc vào số khối được đọc bằng đa khối đơn, được ấn định bằng tham số khởi hoạt DB_FILE_MULTIBLOCK_READ_COUNT. Vì lý do này, bộ tối ưu hóa rất có thể chọn duyệt toàn bộ bảng khi giá trị của tham số này cao.

Thí dụ:

Xét truy vấn sau sử dụng một điều kiện bằng nhau trong mệnh đề WHERE của nó để lựa tất cả các nhân viên có tên Jackson:

```
SELECT *
  FROM emp
  WHERE ename = 'JACKSON';
```

Nếu cột ENAME là khóa duy nhất hoặc khóa chính, thì bộ tối ưu hóa quết định chỉ có một nhân viên có tên Jackson và truy vấn chỉ cho trở lại một hàng. Trong trường hợp này, truy vấn ấy thực sự có chọn lọc và bộ tối ưu hóa rất có thể truy xuất bảng sử dụng duyệt duy nhất trên chỉ mục để tăng cường khóa duy nhất hoặc khóa chính (đường dẫn truy xuất 4).

Thí dụ:

Xét lại truy vấn trong thí dụ trước. Nếu cột ENAME không phải là khóa duy nhất hoặc khóa chính, bộ tối ưu hóa có thể sử dụng các thống kê này để dự đoán tính chọn lọc của truy vấn ấy:
USER_TAB_COLUMNS.NUM_DISTINCT là số giá trị cho mỗi cột trong bảng.
USER_TABLES.NUM_ROWS là số hàng trong mỗi bảng.

Bằng cách chia số hàng trong bảng EMP cho số giá trị riêng biệt trong cột ENAME, bộ tối ưu hóa dự đoán số phần trăm nhân viên có cùng tên. Bằng cách giả sử những giá trị ENAME được phân phối đồng nhất, thì bộ thuộc tính sử dụng số phần trăm này làm tính chọn lọc dự đoán của truy vấn.

Thí dụ:

Xét truy vấn sau lựa tất cả những nhân viên với số ID của nhân viên nhỏ hơn 7500:

```
SELECT *
  FROM emp
  WHERE empno < 7500;
```

Để dự đoán tính chọn lọc của truy vấn, bộ tối ưu hóa sử dụng giá trị biên 7500 trong điều kiện mệnh đề WHERE và các giá trị thống kê HIGH_VALUE và LOW_VALUE cho cột EMPNO nếu có thể được. Các thống kê này có thể được tìm thấy trong khung nhìn USER_TAB_COLUMNS. Bộ tối ưu hóa giả sử là các giá trị EMPNO được phân phối đồng đều trong phạm vi giữa giá trị thấp nhất và cao nhất, thì bộ tối ưu hóa xác định số phần trăm nào của phạm vi này nhỏ hơn giá trị 7500 và sử dụng giá trị này làm tính chọn lọc dự đoán của truy vấn.

Thí dụ:

Xét truy vấn sau sử dụng biến kết gán thay vì giá trị trực kiện cho giá trị biên trong điều kiện mệnh đề WHERE:

```
SELECT *
```

```
FROM emp
WHERE empno < :e1;
```

Bộ tối ưu hóa không biết giá trị của biến kết gán E1. Thực ra, giá trị của E1 không thể khác nhau đối với mỗi tiến hành truy vấn. Vì lý do này, bộ tối ưu hóa không thể sử dụng các phương tiện được mô tả ở thí dụ trước để xác định tính chọn lọc của truy vấn này. Trong trường hợp này, bộ tối ưu hóa dự đoán bằng cách suy nghiệm giá trị nhỏ đối với tính chọn lọc của cột (vì nó có chỉ mục). Bộ tối ưu hóa tạo ra sự thừa nhận này bất kỳ khi nào biến kết gán được sử dụng làm giá trị biên trong điều kiện với một trong các toán tử <, >, <=, or >=.

Xử lý của bộ tối ưu hóa của các biến kết gán có thể tạo cho nó chọn các kế hoạch tiến hành khác nhau cho các câu lệnh SQL chỉ khác nhau về sử dụng các biến kế gán thay vì các hằng. Trong một trường hợp sự khác nhau này có thể rất rõ ràng, bộ tối ưu hóa có thể chọn các kế hoạch tiến hành cho một câu lệnh SQL nhúng với một biến kết gán trong chương trình Oracle Precompiler và cùng câu lệnh SQL với hằng trong SQL*Plus.

Thí dụ:

Xét truy vấn sau sử dụng các biến kết gán làm những giá trị biên trong điều kiện với toán tử BETWEEN:

```
SELECT *
  FROM emp
  WHERE empno BETWEEN :low_e AND :high_e;
```

Bộ tối ưu hóa phân tích điều kiện BETWEEN thành hai điều kiện sau:

```
empno >= :low_e
empno <= :high_e
```

Bộ tối ưu hóa dự đoán bằng cách suy nghiệm một tính chọn lọc nhỏ cho các cột có chỉ mục để cho phép sử dụng chỉ mục ấy.

Thí dụ:

Xét truy vấn sau sử dụng toán tử BETWEEN để lựa tất cả những nhân viên có số ID từ 7500 tới 7800:

```
SELECT *
  FROM emp
  WHERE empno BETWEEN 7500 AND 7800;
```

Để xác định tính chọn lọc của truy vấn sau, bộ tối ưu hóa phân tích điều kiện mệnh đề WHERE thành hai điều khiển sau:

empno >= 7500

empno <= 7800

Bộ tối ưu hóa dự đoán tính chọn lọc cá biệt của mỗi điều kiện sử dụng phương tiện được mô tả ở thí dụ trước. Sau đó bộ tối ưu hóa sử dụng các chọn lọc này (S1 và S2) và hàm giá trị tuyệt đối (ABS) trong công thức sau để dự đoán tính chọn lọc (S) của điều kiện BETWEEN:

S = ABS(S1 + S2 - 1)

Chọn lựa giữa các đường dẫn truy xuất với cách giải quyết dựa trên nguyên tắc

Với cách giải quyết dự trên nguyên tắc, bộ tối ưu hóa chọn xem sử dụng đường dẫn truy xuất nào dựa trên các yếu tố sau:

các đường dẫn truy xuất khả dụng cho câu lệnh

các cấp của các đường dẫn truy xuất trong bảng 20-1 "Các Đường Dẫn Truy Xuất"

Để chọn một đường dẫn truy xuất, trước hết bộ tối ưu hóa khảo sát các điều kiện trong mệnh đề WHERE của câu lệnh để xác định các đường dẫn nào có thể sử dụng. Sau đó bộ tối ưu hóa chọn đường dẫn khả dụng có cấp cao nhất.

Lưu ý duyệt toàn bộ bảng là đường truy xuất có cấp thấp nhất trên danh sách. Điều này có nghĩa là cách giải quyết dựa trên

nguyên tắc luôn luôn chọn một đường dẫn truy xuất sử dụng chỉ mục nếu đường dẫn có thể sử dụng, dù duyệt toàn bộ bảng có thể tiến hành nhanh.

Thứ tự của các điều kiện trong mệnh đề WHERE thường không tác động đến sự chọn lựa của bộ tối ưu hóa giữa các đường dẫn truy xuất.

Thí dụ:

Xét câu lệnh SQL này lựa số nhân viên của tất cả những nhân viên trong bảng EMP với giá trị ENAME của 'CHUNG' và với giá trị SAL lớn hơn 2000:

```
SELECT empno
  FROM emp
  WHERE ename = 'CHUNG'
    AND sal > 2000;
```

Cũng xét bảng EMP có các hạn định tính nguyên vẹn và các chỉ mục:

Có một hạn định PRIMARY KEY trên cột EMPNO được tăng cường bằng chỉ mục PK_EMPNO.

Có một chỉ mục có tên ENAME_IND trên cột ENAME.

Có một chỉ mục có tên SAL_IND trên cột SAL.

Dựa trên các điều kiện trong mệnh đề WHERE của câu lệnh SQL, các hạn định tính nguyên vẹn và các chỉ mục, các đường dẫn này có thể sử dụng:

Đường dẫn truy xuất chỉ mục cột đơn sử dụng chỉ mục ENAME_IND có thể được thực hiện bằng điều kiện ENAME = 'CHUNG'. Đường dẫn truy xuất này có cấp 9.

Một cấp có biên sử dụng chỉ mục SAL_IND có thể được thực hiện bằng điều kiện SAL > 2000. Đường dẫn truy xuất này có cấp 11.

Duyệt toàn bộ bảng có thể tự động sử dụng cho tất cả những câu lệnh SQL. Đường dẫn truy xuất này có cấp 15.

Lưu ý chỉ mục PK_EMPNO không tạo hàng ơn bằng đường dẫn truy xuất khóa chính được vì cột có chỉ mục không xuất hiện trong điều kiện của mệnh đề WHERE.

Sử dụng các giải quyết dựa trên nguyên tắc, bộ tối ưu hóa chọn đường dẫn truy xuất sử dụng chỉ mục ENAME_IND để tiến hành câu lệnh này. Bộ tối ưu hõa chọn đường dẫn này vì nó có thể là đường dẫn có cấp cao nhất.

Tối ưu hóa các câu lệnh liên kết

Để chọn kế hoạch tiến hành cho câu lệnh liên kết, bộ tối ưu hóa phải đưa ra các quyết định tương quan sau:

Các đường dẫn truy xuất	Vì dịch vụ các câu lệnh đơn, bộ tối ưu hóa phải chọn một đường dẫn truy xuất để truy hồi dữ liệu từ mỗi bảng trong câu lệnh liên kết. (Xem "Chọn Các Đường Dẫn Truy Xuất".)
Các phép toán liên kết	Để liên kết mỗi cặp nguồn hàng, Oracle phải thực hiện một trong các phép toán sau: Các vòng lặp lồng nhau Sắp xếp-hợp nhất Cụm liên kết Liên kết trị số (không thể sử dụng với tối ưu hóa dựa trên nguyên tắc)
Thứ tự liên kết	Để tiến hành một câu lệnh liên kết hơn hai bảng, Oracle liên kết hai trong các bảng và sau đó liên kết nguồn hàng kết quả với bảng kế tiếp. Tiến trình này được tiếp tục cho tới khi tất cả các bảng được liên kết vào kết quả ấy.

Các phép toán liên kết

Bộ tối ưu hóa có thể sử dụng các phép toán sau để liên kết hai nguồn hàng:
Liên kết các vòng lặp lồng nhau
Liên kết sắp xếp-hợp nhất
Liên kết trị số

Liên kết các vòng lặp lồng nhau

Để thực hiện liên kết các vòng lặp lồng nhau, Oracle thực hiện các bước sau:

Bộ tối ưu hóa chọn một trong các bảng làm bảng bên ngoài hoặc bảng điều khiển. Bảng khác được gọi là bảng bên trong.

Đối với mỗi hàng trong bảng bên ngoài, Oracle tìm tất cả các hàng trong bảng bên trong thỏa mãn điều kiện liên kết.

Oracle kết hợp dữ liệu trong mỗi cặp hàng thỏa mãn điều kiện liên kết và cho trở lại các hàng kết quả.

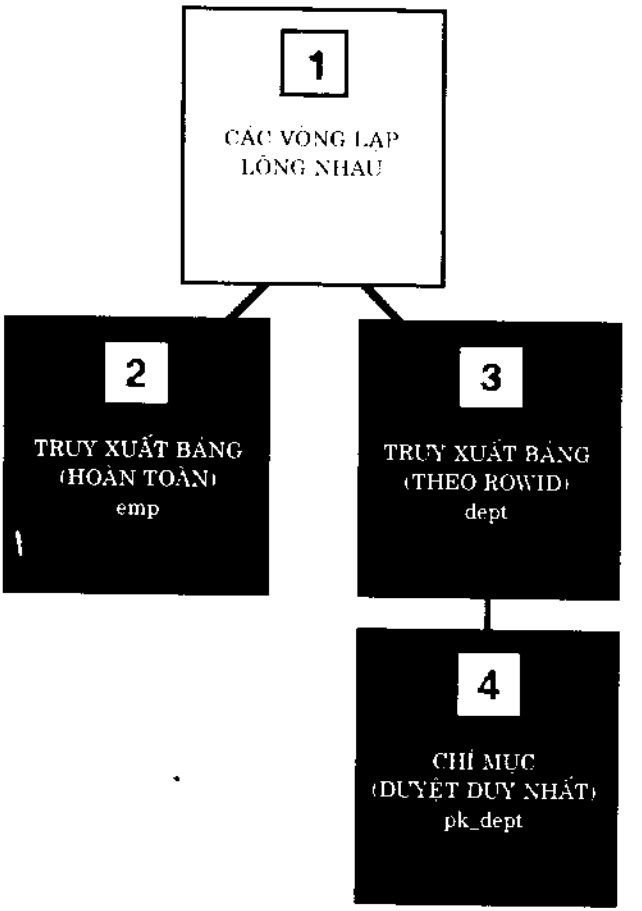
Hình 20-11 trình bày kế hoạch tiến hành cho câu lệnh sau sử dụng liên kết vòng lặp lồng nhau::

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = dept.deptno;
```

Để tiến hành câu lệnh này, Oracle thực hiện các bước sau:

Step 2 truy xuất bảng bên ngoài (EMP) bằng duyệt toàn bộ bảng. Đối với mỗi hàng cho Step 2, Step 4 cho trở lại sử dụng giá trị EMP.DEPTNO để thực hiện duyệt duy nhất trên chỉ mục PK_DEPT.

Hình 20-11 Liên kết các vòng lặp lồng nhau



Step 3 sử dụng ROWID từ Step 4 để định vị hàng so khớp trong bảng bên trong (DEPT).

Oracle kết hợp với mỗi hàng được Step 2 cho trở lại với hàng so khớp được Step 4 cho trở lại và cho trở lại kết quả.

Liên kết sắp xếp-hợp nhất

Oracle chỉ có thể thực hiện liên kết sắp xếp-hợp nhất cho một liên kết tương đương. Để thực hiện liên kết sắp xếp-hợp nhất, Oracle thực hiện các bước sau:

Oracle sắp xếp mỗi nguồn hàng được liên kết nếu chúng không được sắp xếp bằng điều hoạt trước. Các hàng được sắp xếp trên các giá trị của các cột được sử dụng trong điều kiện liên kết.

Oracle hợp nhất hai nguồn để mỗi cặp hàng, mỗi hàng từ mỗi nguồn có các giá trị so khớp cho các cột được sử dụng trong điều kiện liên kết được kết hợp và được cho trở lại bằng nguồng hàng kết quả.

Hình 20-12 trình bày câu lệnh sau sử dụng liên kết sắp xếp-hợp nhất:

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = dept.deptno;
```

Để tiến hành câu lệnh này, Oracle thực hiện các bước sau:

Các Step 3 và 5 thực hiện duyệt toàn bộ bảng của các bảng EMP và DEPT.

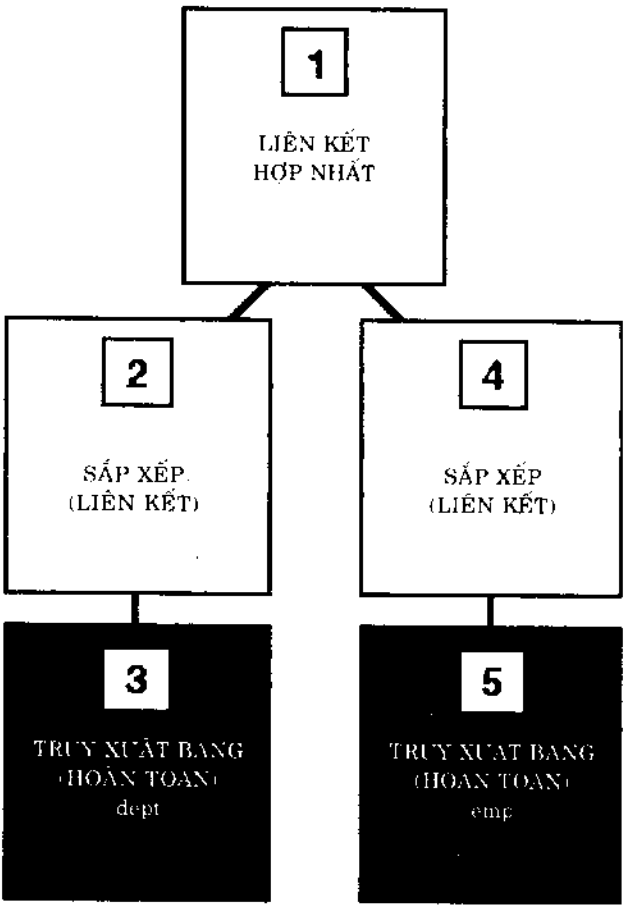
Steps 2 và 4 sắp xếp mỗi nguồn hàng tách riêng.

Step 1 hợp nhất các nguồn từ các Step 2 và 4 với nhau kết hợp mỗi hàng từ Step 2 với mỗi hàng so khớp từ Step 4 và cho trở lại nguồn hàng kết quả.

Liên kết cụm liên kết

Oracle có thể chỉ thực hiện cụm liên kết cho liên kết tương đương cân bằng các cột khóa cụm liên kết của hai bảng trong cùng cụm liên kết. Trong cụm liên kết, các hàng từ hai bảng cùng các giá trị khóa cụm liên kết được lưu trữ trong cùng các khối, vì vậy Oracle chỉ truy xuất các khối này.

Hình 20-12 Liên kết sắp xếp-hợp nhất

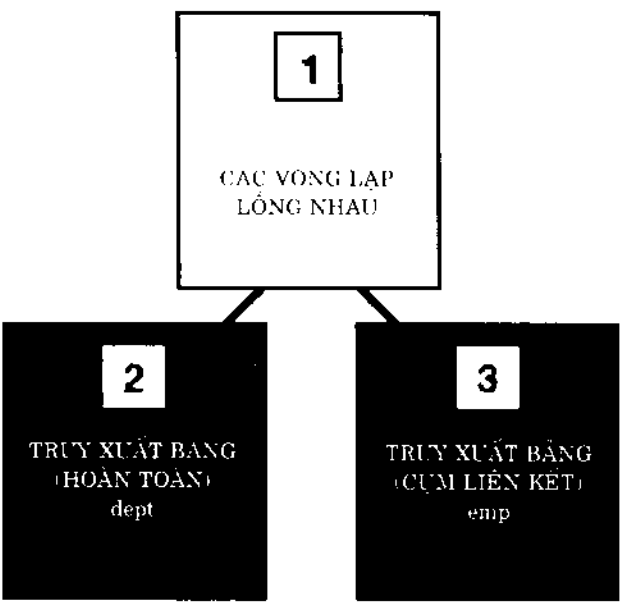


Thông tin bổ sung:
Oracle8 Tuning cung cấp các nguyên tắc chủ đạo để quyết định bảng nào liên kết để có hiệu suất cao nhất.

Hình 20-13 trình bày kế hoạch tiến hành cho câu lệnh sau, trong đó các bảng EMP và DEPT được lưu trữ cùng với nhau trong cùng cụm liên kết:

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = dept.deptno;
```

Hình 20-13 Liên kết cụm liên kết



Để tiến hành câu lệnh này, Oracle thực hiện các bước sau:
Step 2 truy xuất bảng bên ngoài (DEPT) với duyệt toàn bộ bảng. Đối với mỗi hàng được Step 2, Step 3 cho trở lại sử dụng giá trị DEPT.DEPTNO để tìm các hàng so khớp trong bảng bên trong (EMP) với duyệt cụm liên kết.

Liên kết cụm liên kết không gì khác hơn là một liên kết các vòng lặp lồng nhau liên quan đến hai bảng được lưu trữ cùng với

nhau trong cụm liên kết. Vì mỗi hàng từ bảng DEPT được lưu trữ trong cùng các khối dữ liệu bằng các hàng so khớp trong bảng EMP, Oracle có thể truy xuất các hàng so khớp một cách hiệu quả nhất.

Liên kết trị số

Oracle chỉ có thể thực hiện liên kết trị số cho một liên kết tương đương. Liên kết trị số không thể sử dụng với tối ưu hóa dựa trên nguyên tắc. Bạn phải cho phép tối ưu hóa liên kết trị số, bằng cách sử dụng tham số khởi hoạt HASH_JOIN_ENABLED (có thể được xác lập với lệnh ALTER SESSION) hoặc gợi ý USE_HASH.

Để thực hiện một liên kết trị số, Oracle thực hiện các bước sau:

Oracle thực hiện duyệt toàn bộ bảng trên mỗi bảng và tác thành tối đa các phân hoạch có thể dựa trên bộ nhớ có thể sử dụng.

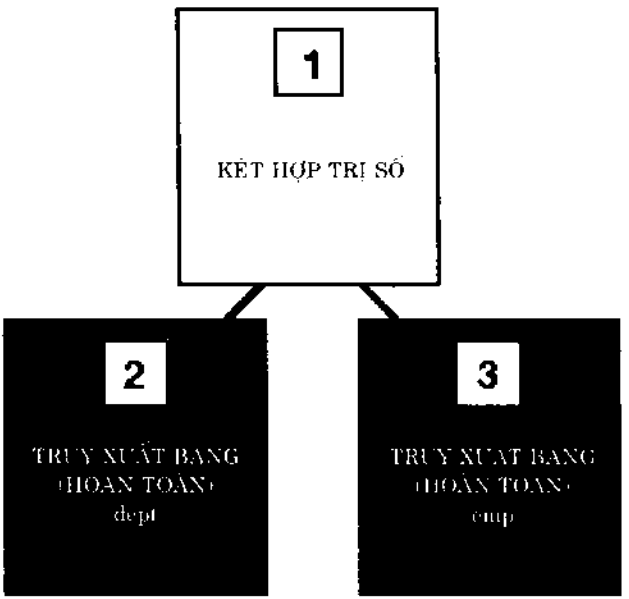
Oracle thiết lập bảng trị số từ một trong các phân hoạch (nếu có thể, Oracle sẽ lựa phân hoạch thích hợp với bộ nhớ có thể sử dụng). Sau đó Oracle sử dụng phân hoạch tương ứng trong bảng khác để thăm dò bảng trị số. Tất cả các cặp phân hoạch không thích hợp với bộ nhớ được đưa vào đĩa.

Đối với mỗi cặp phân hoạch (mỗi phân hoạch từ mỗi bảng), Oracle sử dụng phân hoạch nhỏ hơn để thiết lập bảng trị số và phân hoạch lớn hơn để thăm dò bảng trị số.

Hình 20-14 trình bày kế hoạch tiến hành cho câu lệnh sau sử dụng liên kết trị số:

```
SELECT *
  FROM emp, dept
  WHERE emp.deptno = dept.deptno;
```

Hình 20-14 Liên kết trị số



Để tiến hành câu lệnh này, Oracle thực hiện các bước sau:
Các Step 2 và 3 thực hiện duyệt toàn bộ bảng của các bảng EMP và DEPT.
Step 1 thiết lập một bảng trị số ngoài các hàng phát xuất từ Step 2 và thăm dò nó với mỗi hàng phát xuất từ Step 3.

Tham số khởi hoạt HASH_AREA_SIZE điều khiển dùng lượng bộ nhớ được sử dụng cho điều hoạt liên kết trị số và tham số khởi hoạt HASH_MULTIBLOCK_IO_COUNT điều khiển số khối mà điều hoạt liên kết trị số sẽ đọc và ghi đồng thời.

Thông tin bổ sung:
Xem Oracle8 Tuning về các tham số khởi hoạt này và gợi ý USE_HASH.

Chọn các kế hoạch tiến hành cho các câu lệnh liên kết

Phần này mô tả bộ tối ưu hóa chọn kế hoạch tiến hành cho câu lệnh liên kết như thế nào:
khi sử dụng cách giải quyết dựa trên phí tổn
khi sử dụng cách giải quyết dựa trên nguyên tắc

Lưu ý các nhận xét này áp chung vào các cách giải quyết dựa trên phí tổn và nguyên tắc:
Trước hết bộ tối ưu hóa xác định xem liên kết hai hoặc nhiều bảng có kết quả rõ ràng từ nguồn hàng có tối đa một hàng. Bộ tối ưu hóa nhận biết các trường hợp như thế dựa trên các hạn định UNIQUE và PRIMARY KEY trên các bảng. Nếu một trường hợp như thế tồn tại, thì trước hết bộ tối ưu hóa đưa các bảng này theo thứ tự liên kết. Sau đó bộ tối ưu hóa liên kết của tập hợp các bảng còn lại..
Đối với các câu lệnh liên kết có các điều kiện liên kết bên ngoài, bảng có toán tử liên kết bên ngoài phải xảy ra sau khi bảng khác trong điều kiện theo thứ tự liên kết. Bộ tối ưu hóa không xét các thứ tự xâm phạm nguyên tắc này.

Chọn các kế hoạch tiến hành cho các liên kết với cách giải quyết dựa trên phí tổn

Với cách giải quyết dựa trên phí tổn, bộ tối ưu hóa phát sinh tập hợp các kế hoạch tiến hành theo thứ tự liên kết có thể, các điều hoạt liên kết và các đường dẫn truy xuất có thể sử dụng. Sau đó bộ tối ưu hóa dự toán phí tổn của mỗi kế hoạch và chọn một kế hoạch có phí tổn thấp nhất. Bộ tối ưu hóa dự đoán các phí tổn này bằng các cách sau:
Phí tổn của phép toán các vòng lặp lồng nhau dựa trên phí tổn đọc mỗi hàng được lựa của bảng bên ngoài và mỗi hàng so khớp bảng bên trong của nó vào bộ nhớ. Bộ tối ưu hóa dự đoán các phí tổn này bằng cách sử dụng thống kê trong từ điển dữ liệu.
Phí tổn của liên kết sắp xếp-hợp nhất dựa trên mức độ lớn về phí tổn đọc tất cả những nguồn vào bộ nhớ và sắp xếp chúng.

Bộ tối ưu hóa cũng xét các yếu tố khác khi xác định phí tổn của mỗi điều hoạt. Thí dụ:

Kích thước vùng sắp xếp nhỏ hơn có thể làm tăng phí tổn cho liên kết sắp xếp-hợp nhất vì sắp xếp sử dụng nhiều thời gian của CPU và I/O trong vùng sắp xếp nhỏ hơn. Kích thước vùng sắp xếp được ấn định bằng tham số khởi hoạt SORT_AREA_SIZE.

Đếm số lần đọc dữ liệu đồng thời lớn hơn có thể làm giảm phí tổn cho liên kết sắp xếp-hợp nhất liên quan đến các liên kết vòng lặp lồng nhau. Nếu một số lớn các khối liên tiếp có thể được đọc từ đĩa bằng I/O đơn, một chỉ mục trên bảng bên trong cho liên kết các vòng lặp lồng nhau ít có thể làm tăng hiệu suất so với duyệt toàn bộ bảng. Đếm số lần đọc dữ liệu đồng thời được ấn định bằng tham số khởi hoạt DB_FILE_MULTIBLOCK_READ_COUNT.

Đối với các câu lệnh với các điều kiện liên kết khác, bảng với toán tử liên kết phải xảy ra sau khi bảng khác trong điều kiện theo thứ tự liên kết. Bộ tối ưu hóa không xét các thứ tự liên kết xâm phạm nguyên tắc này

Với cách giải quyết dựa trên phí tổn, sự chọn lựa của bộ tối ưu hóa của các thứ tự liên kết có thể được ghi đè với gợi ý ORDERED. Nếu gợi ý ORDERED ấn định thứ tự liên kết xâm phạm nguyên tắc đối với liên kết bên ngoài, bộ tối ưu hóa bỏ quá gợi ý và chọn thứ tự ấy. Bạn còn có thể chạy để chọn lựa của bộ tối ưu hóa về các điều hoạt có các gợi ý.

Thông tin bổ sung:
Xem Oracle8 Tuning về cách sử dụng các gợi ý.

Chọn các kế hoạch tiến hành cho các liên kết với cách giải quyết dựa trên nguyên tắc

Với cách giải quyết dựa trên nguyên tắc, bộ tối ưu hóa thực hiện các bước sau để chọn một kế hoạch tiến hành cho câu lệnh liên kết các bảng R:

Bộ tối ưu hóa phát sinh tập hợp những thứ tự liên kết R, mỗi liên kết với bảng khác nhau là bảng thứ nhất. Bộ tối ưu hóa phát sinh mỗi thứ tự liên kết tiềm năng sử dụng thuật toán sau:

Điền mỗi vị trí theo thứ tự liên kết, bộ tối ưu hóa chọn bảng có đường dẫn truy xuất có thể sử dụng có cấp cao nhất theo các cấp cho các đường dẫn truy xuất ở Bảng 20-1 "Các Đường Dẫn Truy Xuất". Bộ tối ưu hóa lặp lại bước này để điền mỗi vị trí tiếp theo theo thứ tự liên kết.

Đối với mỗi bảng theo thứ tự liên kết, bộ tối ưu hóa còn chọn điều hoạt để liên kết bảng với bảng trước hoặc nguồn hàng theo thứ tự. Bộ tối ưu hóa thực hiện điều này bằng cách "lập cấp" điều hoạt sắp xếp-hợp nhất bằng đường dẫn 12 và áp dụng những nguyên tắc sau:

Nếu đường dẫn truy xuất đối với bảng được chọn có cấp 11 hoặc hơn, thì bộ tối ưu hóa chọn điều hoạt vòng lặp lồng nhau sử dụng bảng trước hoặc nguồn hàng theo thứ tự liên kết bằng bảng bên ngoài.

Nếu đường dẫn truy xuất đối với bảng có cấp thấp hơn 12 và có điều kiện liên kết tương đương giữa bảng được chọn và bảng trước hoặc nguồn hàng theo thứ tự liên kết, thì bộ tối ưu hóa chọn phép toán sắp xếp-hợp nhất..

Nếu đường dẫn truy xuất đối với bảng được chọn có cấp thấp hơn 12 và không có điều kiện liên kết tương đương, thì bộ tối ưu hóa chọn phép toán vòng lặp lồng nhau với bảng trước hoặc nguồn hàng theo thứ tự liên kết bằng bảng bên ngoài.

Sau đó bộ tối ưu hóa chọn giữa tập hợp kết quả của các kế hoạch tiến hành. Mục đích chọn lựa của bộ tối ưu hóa là để tăng tối đa số phép toán liên kết vòng lặp lồng nhau, trong

đó bảng bên trong được truy xuất sử dụng duyệt chỉ mục. Vì liên kết vòng lặp lồng nhau liên quan đến truy xuất bảng bên trong nhiều lần, chỉ mục trên bảng bên rong có thể tăng hiệu suất của liên kết vòng lặp lồng nhau rất nhiều.

Thường bộ tối ưu hóa không xét thứ tự các bảng xuất hiện trong mệnh đề FROM khi chọn kế hoạch tiến hành. Bộ tối ưu hóa thực hiện chọn lựa này bằng cách áp dụng những nguyên tắc theo thứ tự sau:

Bộ tối ưu hóa chọn kế hoạch tiến hành với các phép toán vòng lặp lồng nhau ít nhất, trong đó in bảng bên trong được truy xuất với duyệt toàn bảng.

Nếu có một liên kết, thì bộ tối ưu hóa chọn kế hoạch tiến hành với các phép toán sắp xếp-hợp nhất.

Nếu vẫn còn liên kết, bộ tối ưu hóa chọn kế hoạch tiến hành để bảng thứ nhất theo thứ tự liên kết có đường dẫn truy xuất có cấp cao nhất:

Nếu có liên kết trong nhiều kế hoạch có các bảng thứ nhất được Xem thêm bằng đường dẫn truy xuất các chỉ mục cột đơn, thì bộ tối ưu hóa chọn kế hoạch có bảng thứ nhất được truy xuất với các chỉ mục hợp nhất nhiều nhất.

Nếu có liên kết giữa nhiều kế hoạch có các bảng thứ nhất được truy xuất bằng duyệt phạm vi có biên, thì bộ tối ưu hóa chọn kế hoạch có bảng thứ nhất được truy xuất với số cao nhất của các cột đầu của chỉ mục tổng hợp.

Nếu vẫn có liên kết, bộ tối ưu hóa chọn kế hoạch tiến hành cho bảng thứ nhất xuất hiện trong mệnh đề FROM của truy vấn sau.

Các khung nhìn trong những liên kết bên ngoài

Đối với một khung nhìn ở bên phải của liên kết bên ngoài, bộ tối ưu hóa có thể sử dụng một trong hai phương pháp, tùy theo bao nhiêu bảng cơ sở mà khung nhìn đó truy xuất:

Nếu khung nhìn chỉ có một bảng cơ sở, thì bộ tối ưu hóa có thể sử dụng hợp nhất khung nhìn.

Nếu khung nhìn có nhiều bảng cơ sở, thì bộ tối ưu hóa đẩy tối ưu hóa liên kết vào khung nhìn đó.

Hợp nhất khung nhìn có bảng cơ sở đơn

Khung nhìn có bảng cơ sở đơn và ở bên phải của liên kết bên ngoài được hợp nhất vào khối truy vấn của câu lệnh truy xuất. (Xem "Hợp Nhất Truy Vấn Của Khung Nhìn Vào Câu Lệnh". Hợp nhất khung nhìn có thể xảy ra dù một biểu thứcc trong khung nhìn có thể cho trở lại giá trị không rỗng dịch vụ một NULL.

Thí dụ:

Xét khung nhìn NAME_VIEW, tập kết các tên gọi và họ từ bảng EMP:

```
CREATE VIEW name_view
  AS SELECT emp.firstname || emp.lastname AS emp fullname,
emp.deptno
      FROM emp;
```

và xét câu lệnh liên kết bên ngoài để tìm các tên của tất cả nhân viên ở London và các bộ phận của chọn, cũng như bộ phận bất kỳ không có nhân viên:

```
SELECT dept.deptno, name_view.emp_fullname
  FROM emp_fullname, dept
  WHERE dept.deptno = name_view.deptno(+)
    AND dept.deptloc = 'London';
```

Bộ tối ưu hóa hợp nhất truy vấn của khung nhìn vào câu lệnh liên kết bên ngoài. Câu lệnh kết quả nhìn giống như sau:

```
SELECT dept.deptno, DECODE(emp.rowid, NULL, NULL,
emp.firstname || emp.lastname)
  FROM emp, dept
  WHERE dept.deptno = emp.deptno(+)
    AND dept.deptloc = 'London';
```

Câu lệnh được biến thể chỉ lựa những nhân viên làm việc ở London.

Đẩy thuộc tính liên kết vào khung nhìn có nhiều bảng cơ sở

Đối với một khung nhìn với nhiều bảng cơ sở ở bên phải của liên kết bên ngoài, bộ tối ưu hóa có thể đẩy thuộc tính liên kết vào khung nhìn ấy (xem Đẩy Thuộc Tính Vào Khung Nhìn", nếu tham số khởi hoạt PUSH JOIN PREDICATE được xác lập là TRUE hoặc truy vấn truy xuất có gợi ý PUSH_JOIN_PRED.

Đẩy tối ưu hóa liên kết là biến thể dựa trên phí tổn có thể cho phép đường dẫn truy xuất và các phương pháp liên kết hiệu quả hơn, chẳng hạn như biến thể các liên kết trị số vào các liên kết vòng lặp lồng nhau và duyệt toàn bộ bảng sang duyệt chỉ mục.

Thông tin bổ sung:
Xem Oracle8 Tuning về các gợi ý của bộ tối ưu hóa.

Thí dụ:

Xét khung nhìn LONDON_EMP lựa những nhân viên làm việc ở London:

```
CREATE VIEW london_emp
  AS SELECT emp.ename
      FROM emp, dept
      WHERE emp.deptno = dept.deptno
        AND dept.deptloc = 'London';
```

và xét câu lệnh liên kết bên ngoài sau, để tìm những kỹ sư và các nhân viên kế toán đang làm việc ở London đã nhận tiền thưởng:

```
SELECT bonus.job, london_emp.ename
  FROM bonus, london_emp
  WHERE bonus.job IN ('engineer', 'accountant')
```

```
  AND bonus.ename = london_emp.ename(+);
```

Bộ tối ưu hóa đẩy tối ưu hóa liên kết bên ngoài vào khung nhìn ấy. Câu lệnh kết quả (không thích hợp với cú pháp SQL chuẩn) nhìn giống nhau sau:

```
SELECT bonus.job, london_emp.ename
  FROM bonus, (SELECT emp.ename FROM emp, dept
                 WHERE bonus.ename = london_emp.ename(+)
                   AND emp.deptno = dept.deptno
                   AND dept.deptloc = 'London')
  WHERE bonus.job IN ('engineer', 'accountant');
```

Tối ưu hóa các liên kết tương phản và các bán liên kết

Liên kết tương phản cho trở lại các hàng từ bên trái của thuộc tính để không có hàng tương ứng ở bên phải của thuộc tính ấy. Nghĩa là nó cho trở lại các hàng không so khớp được (NOT IN) truy vấn con ở bên phải. Thí dụ, liên kết tương phản có thể lựa một danh sách những nhân viên không ở trong một tập hợp ca biệt của các bộ phận::

```
SELECT * FROM emp
  WHERE deptno NOT IN
    (SELECT deptno FROM dept
      WHERE loc = 'HEADQUARTERS');
```

Bộ tối ưu hóa sử dụng các thuật toán vòng lặp lồng nhau của các truy vấn con NOT IN theo mặc định, trừ khi tham số khởi hoạt ALWAYS_ANTI_JOIN được xác lập là MERGE hoặc HASH và các điều kiện cần thiết khác nhau được đáp ứng để cho phép biến thể truy vấn con NOT IN thành sắp xếp-hợp nhất hoặc liên kết tương phản trị số. Bạn có thể đưa gợi ý MERGE_AJ hoặc HASH_AJ vào truy vấn con NOT IN để ấn định thuật toán nào bộ tối ưu hóa sẽ sử dụng..

Bán liên kết cho trở lại các hàng so khớp truy vấn con EXISTS mà không sao chép từ bên trái của thuộc tính khi nhiều hàng ở bên phải thỏa mãn tiêu chuẩn của truy vấn ấy. Thí dụ:

```
SELECT * FROM dept
  WHERE EXISTS
    (SELECT * FROM emp
      WHERE dept.ename = emp.ename
        AND emp.bonus > 5000);
```

Trong truy vấn này, chỉ một hàng cần được cho trở lại từ DEPT dù nhiều hàng trong EMP có thể so khớp truy vấn con. Nếu không có chỉ mục trên cột BONUS trong EMP, bán liên kết có thể được sử dụng để tăng hiệu suất truy vấn.

Bộ tối ưu hóa sử dụng thuật toán vòng lặp lồng nhau cho các truy vấn con theo mặc định, từ khi tham số khởi hoạt ALWAYS_SEMI_JOIN được xác lập là MERGE hoặc HASH và các điều kiện cần thiết khác nhau được đáp ứng. Bạn có thể đưa gợi ý MERGE_SJ hoặc HASH_SJ vào truy vấn con EXISTS để ấn định thuật toán nào bộ tối ưu hóa sẽ sử dụng.

Thông tin bổ sung:
Xem Oracle8 Tuning" về các gợi ý của bộ tối ưu hóa.

Tối ưu hóa các truy vấn "sao"

Một kiểu thiết kế kho dữ liệu tập trung được gọi là giản đồ "sao", biểu thị đặc tính của một hoặc nhiều bảng sự thật rất lớn có thông tin chủ yếu về kho dữ liệu và một số các bảng kích thước nhỏ hơn nhiều (các bảng "tra cứu"), mỗi bảng chứa thông tin về các mục nhập cho một thuộc tính cá biệt trong bảng sự thật.

Truy vấn sao là liên kết giữa bảng sự thật và một số bảng tra cứu. Mỗi bảng tra cứu được liên kết với bảng sự thật sử dụng liên

kết khóa chính với khóa ngoại, nhưng các bảng tra cứu không được liên kết với nhau.

Bộ tối ưu hóa dựa trên phí tổn của Oracle nhận biết những truy vấn sao và phát sinh các kế hoạch tiến hành cho chung. (các truy vấn sao không được bộ tối ưu hóa dựa trên nguyên tắc nhận biết.)

Bảng sự thật tiêu biểu có các khóa và chuẩn đánh giá. Thí dụ, một bảng sự thật đơn có thể có chuẩn đánh giá Sales và các khóa Time, Product và Market. Trong trường hợp này, sẽ có các bảng kích thước tương ứng đối với Time, Product và Market. Thí dụ, bảng kích thước thường có thông tin về mỗi số hiệu sản phẩm xuất hiện trong bảng sự thật.

Liên kết sao là liên kết khóa chính với khóa ngoại của các bảng kích thước với bảng sự thật. Bảng sự thật thường có chỉ mục ghép trên các cột khóa để tạo thuận tiện cho kiểu liên kết này.

Thí dụ truy vấn sao

Phần này để cập các truy vấn sao liên quan đến thí dụ sau:

```
SELECT SUM(dollars)
  FROM facts, time, product, market
  WHERE market.stat = 'New York'
    AND product.brand = 'MyBrand'
    AND time.year = 1995
    AND time.month = 'March'
    /* Joins*/
    AND time.key = facts.tkey
    AND product.pkey = facts.pkey
    AND market.mkey = facts.mkey;
```

Điều hòa các truy vấn sao

Để tiến hành các truy vấn sao một cách hiệu quả, bạn phải sử dụng bộ tối ưu hóa dựa trên phí tổn. Bắt đầu bằng cách sử dụng lệnh ANALYZE để tập hợp những thống kê cho mỗi bảng được truy vấn ấy truy xuất.

Lập chỉ mục

Ở thí dụ trên, bạn sẽ cấu trúc chỉ mục ghép trên các cột tkey, pkey, và mkey. Thứ tự của các cột trong chỉ mục ấy rất quan trọng về hiệu suất. Các cột trong chỉ mục khai thác thứ tự của dữ liệu bất kỳ. Nếu các hàng được bổ sung vào bảng lớn theo thứ tự thời gian, thì tkey sẽ là khóa đầu tiên trong chỉ mục ấy. Khi dữ liệu được trích tinh từ một cơ sở dữ liệu khác, nó đáng để sắp xếp dữ liệu trên các cột hóa trước khi nạp nó.

Nếu tất cả những truy vấn ấn định các thuộc tính trên mỗi bảng nhỏ, một chỉ mục ghép đơn đủ đáp ứng. Nếu các chỉ mục bỏ các cột đầu của chỉ mục ghép thường xuyên, thì các chỉ mục bổ sung có thể hữu dụng. Trong thí dụ này, nếu có những truy vấn thường xuyên bỏ bảng thời gian, một chỉ mục trên PKEY và MKEY có thể được bổ sung.

Các gợi ý

Thường nếu bạn phân tích các bảng mà bộ tối ưu hóa sẽ chọn một kế hoạch sao đủ hiệu lực. Bạn cũng có thể sử dụng các gợi ý để tăng kế hoạch ấy. Phương pháp chính xác nhất là lập thứ tự các bảng trong mệnh đề FROM theo thứ tự của các khóa trong chỉ mục với bảng lớn ở cuối. Sau đó sử dụng các gợi ý sau:

```
/*+ ORDERED USE_NL(facts) INDEX(facts fact_concat) */
```

Một phương pháp phổ biến hơn là sử dụng STAR hint /*+ STAR */

Các giản đồ sao mở rộng

Mỗi bảng nhỏ có thể được thay thế bằng một liên kết của nhiều bảng nhỏ hơn. Thí dụ, bảng sản phẩm có thể được chuẩn hóa theo các bảng về hãng và nhà sản xuất. Sự chuẩn hóa tất cả các bảng nhỏ có thể gây ra vấn đề hiệu suất. Một vấn đề xảy ra do số lần hoán vị tăng lên mà bộ tối ưu hóa phải xét. Vấn đề khác là kết quả của tiến hành nhiều liên kết bảng nhỏ. Bạn có thể giải quyết cả hai vấn đề này bằng cách sử dụng các khung nhìn được giải chuẩn hóa. Thí dụ:

```
CREATE VIEW prodview AS SELECT /*+ NO_MERGE */ *
   FROM brands, mfgrs WHERE brands.mfkey = mfgrs.mfkey;
```

Gợi ý này làm giảm không gian tìm kiếm của Oracle và trữ kết quả của khung nhìn.

Biến thể sao

Biến thể sao là biến thể dựa trên phí tổn có mục đích tiến hành những truy vấn sao một cách hiệu quả. Trong khi tối ưu hóa sao hoạt động tốt đối với các giản đồ có số kích thước nhỏ và các bảng sự thật có mật độ cao, biến thể sao có thể được xét là một chọn lựa thay thế nếu bất kỳ lưu trữ sau đúng:

Số kích thước lớn.

Bảng sự thật thưa.

Có các truy vấn nơi các bảng kích thước không có các thuộc tính hạn định.

Biến thể sao không dựa vào tính toán tích Đề-các của các bảng kích thước để tạo chỏ nó thích hợp hơn với các trường hợp bảng sự thật rải rác và/hoặc số kích thước lớn sẽ dẫn tới tích Đề-các lớn với một ít hàng có so khớp thực sự trong bảng sự thật. Hơn nữa, thay vì dựa vào các chỉ mục ghép, biến thể sao dựa vào các chỉ mục bitmap kết hợp trên các cột bảng sự thật cá biệt.

Vì thế sự biến thể có thể chọn các chỉ mục kết hợp tương ứng chính xác với các kích thước có hạn định. Không cần tạo nhiều

chỉ mục ghép nơi các thứ tự cột khác nhau so khớp cửa sổ mẫu kích thước hạn định khác nhau trong những truy vấn khác nhau.

Lưu ý:
Các chỉ mục bitmap chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition. Trong Oracle8, các chỉ mục bitmap không thể sử dụng và xử lý truy vấn sao sử dụng các chỉ mục B*-tree. Trong Oracle8 Enterprise Edition, thuật toán liên kết chỉ mục bitmap cũng có thể sử dụng để xử lý truy vấn sao.

Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về các đặc trưng có thể sử dụng trong Oracle8 và Oracle8 Enterprise Edition.

Biến thể sao hoạt động bằng cách phát sinh các truy vấn con mới có thể được sử dụng để điều khiển đường truy xuất chỉ mục bitmap cho bảng sự thật.

Xét một trường hợp đơn với ba bảng kích thước "d1", "d2" và "d3" và bảng sự thật "fact". Truy vấn sau:

```
EXPLAIN PLAN FOR
  SELECT * FROM fact, d1, d2, d3
    WHERE fact.c1 = d1.c1 AND fact.c2 = d2.c1 AND fact.c3 =
d3.c1
    AND d1.c2 IN (1, 2, 3, 4)
    AND d2.c2 < 100
    AND d3.c2 = 35
```

có được biến thể bằng cách bổ sung ba truy vấn con:

```
SELECT * FROM fact, d1, d2
  WHERE fact.c1 = d1.c1 AND fact.c2 = d2.c1
    AND d1.c2 IN (1, 2, 3, 4)
    AND d2.c2 < 100
    AND fact.c1 IN (SELECT d1.c1 FROM d1 WHERE d1.c2 IN
(1, 2, 3, 4))
```

```
AND fact.c2 IN (select d2.c1 FROM d2 WHERE d2.c2 < 100)
AND fact.c3 IN (SELECT d3.c1 FROM d3 WHERE d3.c2 = 35)
```

Cho trước các chỉ mục bitmap trên fact.c1, fact.c2 và fact.c3, các truy vấn con mới phát sin có thể được sử dụng để điều khiển đường truy xuất chỉ mục bitmap bằng cách sau. Đối với mỗi giá trị d1.c1 được truy hồi từ truy vấn con thứ nhất, bitmap cho giá trị đó được truy hồi từ chỉ mục fact.c1 và những bitmap này được hợp nhất. kết quả là bitmap cho các hàng đó chính xác, thực ra so khớp điều kiện trên d1 trong truy vấn con ở WHERE-clause.

Tương tự, những giá trị từ truy vấn con thứ hai được sử dụng cùng với chỉ mục bitmap trên fact.c2 để tạo bitmap hợp nhất tương ứng với các hàng thực sự so khớp với điều kiện trên d2 trong truy vấn con thứ hai. Các phép toán tương tự áp chung cho truy vấn con thứ ba. Ba bitmap hợp nhất sau đó có thể là ANDed cho kết quả trong bitmap tương ứng với các hàng thực sự đáp ứng các điều kiện trong tất cả ba truy vấn con đồng thời.

Bitmap này có thể được sử dụng để truy xuất sự kiện và truy hồi các hàng thích hợp. Sau đó các hàng này được liên kết với d1, d2, và d3 để tạo ra trả lời cho truy vấn ấy. Không cần tích Đề-các.

Kế hoạch tiến hành

Kế hoạch tiến hành sau có thể có kết quả từ truy vấn trên:

```
SELECT STATEMENT
 HASH JOIN
  HASH JOIN
   HASH JOIN
    TABLE ACCESS          FACT    BY ROWID
     BITMAP CONVERSION             TO ROWIDS
      BITMAP AND
       BITMAP MERGE
        BITMAP KEY ITERATION
```

```
    TABLE ACCESS        D3      FULL
    BITMAP INDEX        FACT_C3 RANGE SCAN
  BITMAP MERGE
   BITMAP KEY ITERATION
    TABLE ACCESS        D1      FULL
    BITMAP INDEX        FACT_C1 RANGE SCAN
  BITMAP MERGE
   BITMAP KEY ITERATION
    TABLE ACCESS        D2      FULL
    BITMAP INDEX        FACT_C2 RANGE SCAN
 TABLE ACCESS           D1      FULL
TABLE ACCESS            D2      FULL
TABLE ACCESS            D3      FULL
```

Trong kế hoạch này bảng sự thật được truy xuất bằng đường dẫn truy xuất bitmap dựa trên bitmap AND của ba bitmap hợp nhất. ba bitmap ấy được phát sinh bằng nguồn hàng BITMAP MERGE là được bitmap cung cấp từ các cây nguồn hàng bên dưới nó. Mỗi cây nguồn hàng như thế gồm một nguồn hàng BITMAP KEY ITERATION truy nạp những giá trị từ cây nguồn hàng truy vấn con, mà trong thí dụ này chính là truy xuất toàn bảng và đối với mỗi giá trị như thế truy hồi bitmap từ chỉ mục bitmap. Sau các hàng bảng sự thật thích hợp đã được truy hồi bằng cách sử dụng đường dẫn truy xuất này, chúng được liên kết với các bảng kích thước để tạo trả lời cho truy vấn ấy.

Biến thể sao là biến thể dựa trên phí tổn bằng sự nhận biết tiếp theo. Bộ tối ưu hóa phát sinh và lưu kế hoạch tốt nhất mà nó có thể tạo không có sự biến thể. Nếu biến thể được cho phép, sau đó bộ tối ưu hóa cố áp dụng nó vào truy vấn được biến thể. Dựa trên so sánh các dự toán phí tổn giữa các kế hoạch tốt nhất cho hai phiên bản của truy vấn, bộ tối ưu hóa sau đó sẽ quyết định xem sử dụng kế hoạch tốt nhất cho phiên bản được biến thể hoặc không được biến thể.

Nếu truy vấn đòi hỏi truy xuất số phần trăm hàng lớn trong bảng sự thật, nó có thể sử dụng duyệt toàn bộ bảng tốt hơn và không sử dụng biến thể. Tuy nhiên, nếu các thuộc tính hạn định trên các bảng kích thước chọn lọc có hiệu lực, chỉ một phần nhỏ bảng sự thật cần được truy hồi, kế hoạch dựa trên biến thể có thể sẽ đạt kết quả nhất.

Lưu ý bộ tối ưu hóa sẽ phát sinh truy vấn con cho bảng kích thước chỉ nếu nó quyết định thực hiện như vậy dựa trên một số tiêu chuẩn là hợp lý. Không có bảo đảm nào là các truy vấn con sẽ được phát sinh cho tất cả các bảng kích thước. Bộ tối ưu hóa cũng có thể quyết định dựa trên các tính chất của các bảng và truy vấn mà sự biến thể ấy không có giá trị áp dụng vào truy vấn cá biệt. Trong trường hợp này, kế hoạch thông thường nhất sẽ được sử dụng.

Sử dụng biến thể sao

Bạn có thể cho phép biến thể sao bằng cách xác lập giá trị của tham số khởi hoạt STAR_TRANSFORMATION_ENABLED to TRUE. Sử dụng gợi ý STAR_TRANSFORMATION để tạo cho bộ tối ưu hóa sử dụng kế hoạch tốt nhất trong đó biến thể đã được sử dụng.

Các hạn chế về biến thể sao

Biến thể sao không hỗ trợ cho các bảng có bất một trong các đặc điểm sau:

các bảng có gợi ý bảng không tương thích với đường dẫn truy xuất bitmap

các bảng có quá ít chỉ mục bitmap (phải có chỉ mục bitmap trêu bảng sự thật đối với bộ tối ưu hóa để xét phát sinh một truy vấn con cho nó)

các bảng từ xa (tuy nhiên, các bảng kích thước từ xa được cho phép trong các truy vấn con được phát sinh)

các bảng liên kết tương phản

các bảng được sử dụng làm bảng kích thước trong một truy vấn con rồi
các bảng thực sự là các khung nhìn không được hợp nhất, không phải là các phân hoạch khung nhìn
các bảng có đường dẫn truy xuất bảng đơn tốt
các bảng quá nhỏ so không đáng để biến thể

Phần VI

PARALLEL SQL VÀ DIRECT-LOAD INSERT

Chương 21

Direct-Load INSERT

Chương này mô tả đặc trưng Oracle direct-load INSERT cho các lệnh chèn nối tiếp hoặc song song. Nó cũng mô tả đặc trưng NOLOGGING có thể sử dụng cho các câu lệnh nạp trực tiếp và một số câu lệnh DDL, gồm các chủ đề:

- **Giới thiệu về Direct-Load INSERT**
- **Các câu lệnh Direct-Load INSERT khác nhau**
- **INSERT nối tiếp và song song**
- **Chế độ ghi sổ nhật ký**
- **Các nhận xét bổ sung về Direct-Load INSERT**
- **Các giới hạn về Direct-Load INSERT**

Xem chương 22 "Tiến Hành Song Song" về các vấn đề xác định song song.

Lưu ý:
Đặc trưng INSERT nạp trực tiếp song song được mô tả trong chương trình này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition.

Thông tin bổ sung:
Xem Oracle8 Tuning về cách điều hợp INSERT nạp trực tiếp song song.

Giới thiệu về Direct-Load INSERT

Direct-load INSERT tăng cường hiệu suất khi các điều hoạt chèn bằng cách định dạng và ghi dữ liệu trực tiếp vào các tập tin dữ liệu Oracle mà không sử dụng cache vùng đệm. Tính năng này giống như tiện ích Direct Loader utility (SQL*Loader).

Direct-load INSERT ghép nối dữ liệu được chèn sau dữ liệu có sẵn trong table; giải phóng không gian trong dữ liệu có sẵn không được sử dụng. Dữ liệu có thể được chèn vào các bảng được phân hoạch hoặc chưa phân hoạch song song hoặc nối tiếp.

Nhiều tùy chọn của INSERT nạp trực tiếp có sẵn với trạng thái song song, phân hoạch bảng và ghi sổ nhật ký.

Các ưu điểm của Direct-Load INSERT

Ưu điểm chính của INSERT nạp trực tiếp là bạn có thể nạp dữ liệu mà không phải ghi sổ nhật ký hoặc các mục nhập hoán đổi để cải tiến hệ điều hành chèn đáng kể. Cả INSERT nạp trực tiếp nối tiếp và song song có ưu điểm hiệu suất này trên đường dẫn quy ước INSERT.

Trái lại, với đường dẫn quy ước INSERT, giải phóng không gian trong đối tượng được sử dụng lại và tính nguyên vẹn dữ liệu tham chiếu có thể được duy trì. Đường dẫn quy ước đối với các phần giao không thể song song.

So sánh bằng CREATE TABLE ... AS SELECT

Với INSERT nạp trực tiếp, bạn có thể chèn dữ liệu vào các bảng có sẵn thay vì phải tạo các bảng mới. INSERT nạp trực tiếp cập nhật các chỉ mục của bảng ấy, nhưng CREATE TABLE ... AS SELECT chỉ tạo một bảng mới không có bất kỳ cập nhật nào.

Ưu điểm về nạp trực tiếp song song (SQL*Loader)

Với INSERT song song, hóa trị của giao dịch được bảo đảm. Hóa trị không được bảo đảm nếu nhiều lần nạp song song được sử dụng. Ngoài ra, nạp song song có thể để một số chỉ mục bảng ở trạng thái "Unusable" ở cuối lần nạp nếu các lỗi xảy ra khi cập nhật các chỉ mục ấy. Trong so sánh, INSERT song song tự động cập nhật bảng và các chỉ mục (nghĩa là, nó hồi phục về trước câu lệnh nếu lỗi xảy ra khi cập nhật chỉ mục ấy)

Thông tin bổ sung:
Xem Oracle8 Utilities về nạp song song.

Các câu lệnh INSERT ... SELECT

INSERT nạp truyền (nối tiếp hoặc song song) chỉ có thể hỗ trợ cú pháp INSERT ... SELECT của câu lệnh INSERT, không hỗ trợ cú pháp các đối tượng INSERT.... Trạng thái song song đối với INSERT ... SELECT được xác định từ các gợi ý song song hoặc các mệnh đề định nghĩa bảng song song.

Thông tin bổ sung:
Xem Oracle SQL Reference về cú pháp của các câu lệnh INSERT ... SELECT.

Các sự khác nhau của các câu lệnh Direct-Load INSERT

Direct-load INSERT có thể được thực hiện:
nối tiếp hoặc song song
vào các bảng được phân hoạch hoặc chưa được phân hoạch
có hoặc không có dữ liệu ghi sổ nhật ký hoạt động

INSERT nối tiếp và song song

Direct-load INSERT có thể được thực hiện trên các bảng được phân hoạch hoặc chưa được phân hoạch và nó có thể được thực hiện nối tiếp hoặc song song.
INSERT nạp trực tiếp nối tiếp vào bảng được phân hoạch hoặc chưa được phân hoạch. Dữ liệu được chèn ngoài dấu mức nước cao hiện hành của đoạn bảng hoặc mỗi đoạn phân hoạch. Khi một cam kết, mức nước cao được cập nhật giá trị mới để tạo dữ liệu cho tiến trình khác có thể hiển thị.
INSERT nạp trực tiếp song song vào bảng chưa được phân hoạch. Mỗi tiến trình server song song phân bổ đoạn mới tạm thời, khi một cam kết tiến hành, tiến trình bộ điều hợp hợp nhất các đoạn mới tạm thời vào đoạn bảng chính. (Xem thêm "Kiến Trúc Xử Lý Để Tiến Hành Song Song").
INSERT nạp trực tiếp song song vào bảng được phân hoạch. Mỗi tiến trình server song song được gán một hoặc nhiều phân hoạch, không hơn một tiến trình hoạt động trên bất kỳ phân hoạch định trước. Tiến trình server song song chèn dữ liệu trên mức nước cao của (các) đoạn phân hoạch được gán cho nó. Khi một cam kết tiến hành, mức nước cao của mỗi đoạn phân hoạch được tiến hành bằng tiến trình bộ điều hợp tới giá trị mới để tạo dữ liệu cho tiến trình khác có thể hiển thị.

Trong tất cả các trường hợp, sự tăng vọt của mực nước cao hoặc hợp nhất đoạn tạm thời bị trì hoãn cho tới khi cam kết được phát hành, vì hoạt động này tạo dữ liệu cho các tiến trình khác hiển thị ngay (nghĩa là, nó cam kết điều hoạt chèn).

Ấn định INSERT nạp trực tiếp nối tiếp hoặc song song

Gợi ý APPEND cần thiết để sử dụng INSERT nạp trực tiếp song song. INSERT nạp trực tiếp song song đòi hỏi gợi ý PARALLEL trong câu lệnh ấy hoặc mệnh đề PARALLEL trong định nghĩa bảng; gợi ý APPEND được tùy chọn. INSERT nạp trực tiếp song song cũng đòi hỏi DML song song được cho phép bằng câu lệnh ALTER SESSION ENABLE PARALLEL DML cho phép.

Bảng 21-1 tóm tắt các đòi hỏi này và so sánh INSERT nạp trực tiếp với INSERT quy ước.

Bảng 21-1 Tóm tắt các câu lệnh INSERT ... SELECT nối tiếp và song song

Kiểu chèn	Nối tiếp	Song song
INSERT nạp trực tiếp	Có: đòi hỏi gợi ý APPEND trong câu lệnh SQL	Có: đòi hỏi ALTER SESSION ENABLE PARALLEL DML Thuộc tính PARALLEL bảng hoặc gợi ý PARALLEL câu lệnh (APPEND được tùy chọn)
INSERT quy ước	Có (mặc định)	Không

Các thí dụ về INSERT nạp trực tiếp nối tiếp và song song

Bạn có thể ấn định INSERT nối tiếp với gợi ý APPEND, thí dụ:

```
INSERT /*+ APPEND */ INTO emp
```

```
  SELECT * FROM t_emp;
COMMIT;
```

Bạn có thể ấn định INSERT nạp trực tiếp song song bằng cách xác lập thuộc tính PARALLEL của bảng mà các hàng được chèn vào, thí dụ:

```
ALTER BẢNG emp PARALLEL (DEGREE 10);
ALTER SESSION ENABLE PARALLEL DML;
INSERT INTO emp
  SELECT * FROM t_emp;
COMMIT;
```

Bạn cũng có thể ấn định trạng thái song song cho điều hoạt SELECT bằng cách xác lập thuộc tính PARALLEL của bảng mà các hàng được lựa:

```
ALTER TABLE emp PARALLEL (DEGREE 10);
ALTER TABLE t_emp PARALLEL (DEGREE 10);
ALTER SESSION ENABLE PARALLEL DML;
INSERT INTO emp
  SELECT * FROM t_emp;
COMMIT;
```

Gợi ý PARALLEL cho điều hoạt INSERT hoặc SELECT thực hiện trước thuộc tính TABLE's PARALLEL. Thí dụ, mức trạng thái song song trong câu lệnh sau INSERT ... SELECT là 12 bất kể các thuộc tính PARALLEL được xác lập cho EMP và T_EMP TABLEs:

```
ALTER SESSION ENABLE PARALLEL DML;
INSERT /*+ PARALLEL(emp,12) */ INTO emp
  SELECT /*+ PARALLEL(t_emp,12) */ * FROM t_emp;
COMMIT;
```

Để có thêm thông tin về các câu lệnh INSERT song song, xem "Các Nguyên Tắc Song Song Hóa INSERT . . . SELECT".

Chế độ ghi sổ nhật ký

Các điều hoạt INSERT nạp trực tiếp có thể được thực hiện có hoặc không có thông tin ghi sổ nhật ký hoạt động. Bạn có thể ấn định chế độ cho bảng, phân hoạch hoặc chỉ mục mà dữ liệu được chèn vào bằng cách sử dụng lệnh ALTER TABLE, ALTER INDEX hoặc ALTER TABLESPACE.
INSERT nạp trực tiếp có ghi sổ nhật ký. Chế độ này ghi sổ nhật ký hoàn toàn cho môi trường làm việc và phục hồi môi trường. Ghi sổ nhật ký là chế độ mặc định.
INSERT nạp trực tiếp không có ghi sổ nhật ký. Ở chế độ này, dữ liệu được chèn không có ghi sổ nhật ký hoạt động hoặc hoán đổi. (Ghi sổ nhật ký không còn được thực hiện nhiều để đánh dấu các khối lưu riêng mới bất hợp lệ và các thay đổi từ điển luôn luôn được ghi sổ nhật ký hoàn toàn). Khi được áp dụng khi phục hồi môi trường, các bản ghi khối lưu riêng bất hợp lệ đánh dấu phạm vi các khối bị sai sót logic, vì dữ liệu redo không được ghi sổ nhật ký.

Chế độ không ghi sổ nhật ký làm tăng hiệu suất vì nó ít phát sinh ghi sổ nhật ký hơn nhiều. Người dùng chịu trách nhiệm sao lưu dữ liệu sau khi hoạt động chèn không ghi sổ nhật ký để có thể phục hồi môi trường.

Lưu ý:
Chế độ ghi/không ghi sổ nhật ký là một thuộc tính vĩnh viễn của bảng, phân hoạch hoặc chỉ mục. Sau khi đối tượng cơ sở dữ liệu được chèn đã di trú dữ liệu và sao chép, bạn có thể xác lập trạng thái của nó vào chế độ ghi sổ nhật ký để những thay đổi tiếp theo sẽ được ghi sổ nhật ký.

Bảng 21-2 so sánh các chế độ LOGGING và NOLOGGING cho INSERT nạp trực tiếp và quy ước.

Bảng 21-2 Tóm tắt về các tùy chọn LOGGING và NOLOGGING

Kiểu chèn	Ghi sổ nhật ký	Không ghi sổ nhật ký
INSERT nạp trực tiếp	Có: tính năng phục hồi đòi hỏi Chế độ cơ sở dữ liệu ARCHIVELOG	Có: đòi hỏi Thuộc tính NOLOGGING cho không gian bảng, bảng, phân hoạch hoặc chỉ mục
INSERT quy ước	Có (mặc định): tính năng phục hồi đòi hỏi Chế độ cơ sở dữ liệu ARCHIVELOG	Không có

Các thí dụ về chế độ không ghi sổ nhật ký

Bạn có thể ấn định chế độ không ghi sổ nhật ký cho INSERT nạp trực tiếp bằng cách xác lập thuộc tính NOLOGGING của bảng mà các hàng được chèn vào, thí dụ:

```
ALTER TABLE emp NOLOGGING;
ALTER SESSION ENABLE PARALLEL DML;
INSERT /*+ PARALLEL(emp,12) */ INTO emp
  SELECT /*+ PARALLEL(t_emp,12) */ * FROM t_emp;
COMMIT;
```

Bạn cũng có thể xác lập thuộc tính NOLOGGING cho phân hoạch, không gian bảng hoặc chỉ mục; thí dụ:

```
ALTER TABLE emp MODIFY PARTITION emp_lmnop
NOLOGGING;

ALTER TABLESPACE personnel NOLOGGING;
ALTER INDEX emp_ix NOLOGGING;
```

ALTER INDEX emp ix MODIFY PARTITION eix_lmnop NOLOGGING;

Các câu lệnh SQL có thể sử dụng chế độ không ghi sổ nhật ký

Dù bạn có thể xác lập thuộc tính NOLOGGING cho một bảng, phân hoạch, chỉ mục hoặc không gian bảng, chế độ không ghi sổ nhật ký không áp dụng cho mọi điều hoạt được thực hiện trên đối tượng giản đồ mà bạn xác lập thuộc tính NOLOGGING.

Chỉ các điều hoạt sau có thể khai thác chế độ không ghi sổ nhật ký:

direct load (SQL*Loader)
direct-load INSERT
CREATE TABLE ... AS SELECT
CREATE INDEX
ALTER TABLE ... MOVE PARTITION
ALTER TABLE ... SPLIT PARTITION
ALTER INDEX ... SPLIT PARTITION
ALTER INDEX ... REBUILD
ALTER INDEX ... REBUILD PARTITION

Tất cả các câu lệnh SQL có thể được song song hóa (Xem "Tiến Hành Song Song"). Chúng có thể tiến hành ở chế độ ghi sổ nhật ký hoặc không ghi sổ nhật ký cho cả tiến hành nối tiếp và song song.

Các câu lệnh SQL khác (chẳng hạn như UPDATE, DELETE, đường dẫn quy ước INSERT và các câu lệnh DDL không được liệt kê ở trên) không bị ảnh hưởng bởi thuộc tính NOLOGGING của đối tượng giản đồ.

Chế độ ghi sổ nhật ký mặc định

Nếu mệnh đề LOGGING hoặc NOLOGGING không được xác định, thì tối ưu hóa ghi sổ nhật ký của bảng, phân hoạch hoặc chỉ mục được mặc định theo thuộc tính ghi sổ nhật ký của không gian bảng mà nó cư trú.

Định vị các LOB, nếu mệnh đề LOGGING hoặc NOLOGGING được bỏ, thì:
Nếu CACHE được xác định, LOGGING được sử dụng (vì các LOB không thể có CACHE NOLOGGING)
Trái lại, mặc định có được từ không gian bảng trong đó giá trị LOB cư trú.

Các nhận xét bổ sung về INSERT nạp trực tiếp

Phần này mô tả duy trì chỉ mục, phân bổ không gian và các khóa dữ liệu cho INSERT nạp trực tiếp.

Duy trì chỉ mục

Đối với INSERT nạp trực tiếp trên các bảng không được phân hoạch hoặc các bảng được phân hoạch có các chỉ mục cục bộ hoặc toàn cục, duy trì chỉ mục được thực hiện ở cuối hoạt động INSET. Duy trì chỉ mục được thực hiện bằng các tiến trình server song song hoặc bằng tiến trình đơn đối với INSET nạp trực tiếp trên các bảng được phân hoạch hoặc không được phân hoạch.

Nếu INSERT nạp trực tiếp của bạn sửa đổi phần lớn dữ liệu trong một bảng, bạn có thể tránh ảnh hưởng hiệu suất của duy trì chỉ mục bằng cách bỏ chỉ mục ấy trước điều hoạt INSERT, rồi thiết lập lại nó về sau.

Các nhận xét không gian

INSERT nạp trực tiếp đòi hỏi nhiều không gian hơn đường dẫn quy ước INSERT, vì INSERT bỏ qua không gian có sẵn trong các danh sách trống của đoạn ấy. Đối với INSERT nạp trực tiếp song song vào các bảng không được phân hoạch, các khối trống bên dưới mức nước cao của đoạn bảng cũng được bỏ qua. Các đòi hỏi không gian bổ sung phải được xét trước khi sử dụng INSERT nạp trực tiếp.

INSERT nạp trực tiếp song song vào bảng không được phân hoạch tạo các đoạn tạm thời - mỗi đoạn cho một mức trạng thái song song. Thí dụ, nếu bạn sử dụng INSERT song song vào bảng không được phân hoạch với mức trạng thái song song được xác lập là 4, thì 4 đoạn tạm thời được tạo.

Trước hết mỗi tiến trình server song song chèn dữ liệu của nó vào đoạn tạm thời và cuối cùng dữ liệu trong tất cả các đoạn tạm thời nối ghép vào bảng ấy. (Đây là cơ chế giống như CREATE TABLE ... AS SELECT.)

Để cung cấp đủ bộ nhớ cho các đoạn tạm thời, không lãng phí không gian trên các đoạn nhiều hơn bạn cần, bạn phải ấn định các giá trị thích hợp cho các tham số bộ nhớ NEXT và PCTINCREASE đối với một bảng không được phân hoạch mà bạn muốn thực hiện một điều hoạt INSERT song song. Bạn có thể thay đổi những giá trị của các tham số này với tùy chọn STORAGE của câu lệnh ALTER TABLE. Sau khi thực hiện câu lệnh DML song song, bạn có thể thay đổi các tham số bộ nhớ NEXT và PCTINCREASE trở lại các xác lập thích hợp cho các điều hoạt không song song.

Lưu ý:
Tham số bộ nhớ PCTINCREASE có thể tạo ra các đoạn tạm thời rất lớn, trừ khi nó được xác lập là 0. Để tránh phải sử dụng hết không gian khi tùy chọn DML song song, chắc chắn

tham số PCTINCREASE được xác lập là 0.

Tuy nhiên, đối với INSERT song song vào TABLE được phân hoạch, không có đoạn tạm thời nào được tạo. Mỗi tiến trình server song song chỉ chèn dữ liệu của nó vào phân hoạch trên mức nước cao.

Thông tin bổ sung:
Tham chiếu tiến hành song song trong Oracle8 Tuning về quản lý không gian.

Các nhận xét khóa

Trong INSERT nạp trực tiếp, các khóa riêng biệt có được trên bảng (hoặc trên tất cả các phân hoạch của bảng được phân hoạch) loại bỏ bất kỳ điều hoạt chèn, cập nhật hoặc xóa đồng thời trên bảng. Tuy nhiên, các truy vấn đồng thời được hỗ trợ và sẽ chỉ thấy dữ liệu trong bảng ấy trước khi điều hoạt INSERT bắt đầu. Các khóa này cũng ngăn cản các điều hoạt tạo hoặc thiết lập lại chỉ mục. Điều này phải lưu ý trước khi sử dụng INSERT nạp trực tiếp vì nó ảnh hưởng đến tính đồng thời bảng. Xem thêm "Khóa và các tài nguyên hàng đợi cho DML song song".

Các giới hạn về INSERT nạp trực tiếp

Các giới hạn trên INSERT nạp trực tiếp giống như các giới hạn ảnh hưởng về nạp song song đường dẫn trực tiếp với SQL*Loader, vì chúng sử dụng cùng cơ chế cơ sở, các giới hạn DML song song chung cũng áp dụng vào INSERT nạp trực tiếp.

INSERT nạp trực tiếp nối tiếp và song song có các giới hạn sau:
Giao dịch có thể có nhiều câu lệnh INSERT nạp trực tiếp (hoặc cả các câu lệnh INSERT nạp trực tiếp và các câu lệnh UPDATE song song hoặc DELETE), nhưng sau một trong các câu lệnh này

sửa đổi một bảng, không có câu lệnh SQL nào trong giao dịch có thể truy xuất cùng bảng.
Các truy vấn truy xuất cùng bảng được cho phép trước câu lệnh INSERT nạp trực tiếp, mà không sau nó.
Bất kỳ các câu lệnh nối tiếp hoặc song song truy xuất bảng đã được sửa đổi bằng một INSERT nạp trực tiếp (hoặc DML song song) trong cùng giao dịch bị từ chối bằng một thông báo lỗi.
Nếu tham số khởi hoạt ROW_LOCKING = INTENT, thì các điều hoạt chèn không thể được thực hiện bằng đường dẫn nạp trực tiếp.
INSERT nạp trực tiếp không hỗ trợ tính nguyên vẹn tham chiếu.
Các bộ kích khởi không được hỗ trợ cho các điều hoạt INSERT nạp trực tiếp.
Tính năng sao chép không được hỗ trợ cho INSERT nạp trực tiếp.
INSERT nạp trực tiếp không thể xảy ra trên các bảng có các cột đối tượng hoặc các cột LOB hoặc trên các bảng có tổ chức chỉ mục.
Một giao dịch liên quan trong điều hoạt INSERT nạp trực tiếp không thể là giao dịch được phân phối hoặc trở thành được phân phối.
Các bảng liên kết không được hỗ trợ.

Các vi phạm các giới hạn sẽ tạo cho câu lệnh tiến hành nối tiếp, bằng cách sử dụng đường dẫn chèn quy ước, mà không có cảnh báo hoặc các thông báo lỗi. Ngoại trừ giới hạn trên các câu lệnh truy xuất cùng bảng hơn một lần trong một giao dịch, có thể gây ra các thông báo lỗi.

Thí dụ, nếu các bộ kích khởi hoặc tính nguyên vẹn tham chiếu hiện hữu trên bảng ấy, thì gợi ý APPEND sẽ được bỏ qua khi bạn cố sử dụng INSERT nạp trực tiếp (nối tiếp hoặc song song), cũng như gợi ý PARALLEL hoặc mệnh đề, nếu có.

Để có thêm thông tin về các giới hạn chung về DML song song, xem "Các giới hạn về DML song song".

Chương 22

Tiến hành song song

Chương này mô tả tiến hành song song của các câu lệnh SQL, gồm các chủ đề:

- **Tổng quan về tiến hành song song**
- **Kiến trúc xử lý tiến hành song song**
- **Xác lập mức trạng thái song song**
- **DDL song song**
- **DML song song**
- **Quan hệ cấu trúc**

Lưu ý:
Các đặc trưng tiến hành song song trong chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition.

Tiến hành song song không giống như Oracle Parallel Server (tùy chọn Parallel Server). Bạn không cần tùy chọn Parallel Server để thực hiện tiến hành song song các câu lệnh SQL; tuy nhiên, một số khía cạnh tiến hành song song chỉ áp dụng với Oracle Parallel Server.

Tổng quan về tiến hành song song

Khi Oracle không song song hóa tiến hành các câu lệnh SQL, mỗi câu lệnh SQL được tiến hành tiếp theo sau bằng một tiến trình đơn. Tuy nhiên, với tiến hành song song, nhiều tiến trình hoạt động cùng với nhau đồng thời để tiến hành một câu lệnh SQL đơn. Bằng cách chia tác vụ cần thiết để tiến hành câu lệnh trong nhiều tiến trình, Oracle có thể tiến hành câu lệnh ấy nhanh hơn nếu chỉ một tiến hành đơn tiến hành nó.

Tiến trình song song có thể tăng hiệu suất lên rất nhiều cho các điều hoạt dữ liệu mở rộng với các ứng dụng hỗ trợ quyết định hoặc các môi trường CSDL rất lớn. Đa xử lý đối ứng (SMP), các hệ thống liên kết và các hệ thống song song hàng loạt (MPP) đạt được các ưu điểm hiệu suất lớn nhất nhờ vào tiến hành song song vì xử lý câu lệnh có thể tách giữa nhiều CPU trên một hệ thống Oracle đơn.

Tiến hành song song giúp các hệ thống có cùng hiệu suất bằng cách khai thác tối ưu những tài nguyên phần cứng. Nếu các CPU của hệ thống và các bộ điều khiển đĩa được tải nạp quá nhiều rồi, bạn cần làm giảm bớt tải của hệ thống hoặc tăng các tài nguyên phần cứng trước khi sử dụng tiến hành song song để tăng hiệu suất.

Thông tin bổ sung:
Xem Oracle8 Tuning về điều hợp các tập tin tham số và CSDL để khai thác ưu điểm tiến hành song song.

Các điều hoạt có thể được song song hóa

Các Oracle server có thể sử dụng tiến hành song song cho bất kỳ những điều hoạt này:

Tác vụ duyệt bảng
Liên kết vòng lặp lồng nhau

Lựa riêng biệt
Hội và hội tất cả
Kết tập
Các hàm PL/SQL được gọi từ SQL
Theo thứ tự
Tạo bằng lựa chọn
Tạo chỉ mục
Thiết lập lại chỉ mục
Thiết lập lại phân hoạch chỉ mục
Di chuyển phân hoạch
Tách phân hoạch
Cập nhật
Xóa
Chèn . . . lựa
Hạn định cho phép (tác vụ duyệt bảng được song song hóa
Biến thể sao

Oracle song song hóa các điều hoạt như thế nào

Câu lệnh SELECT chỉ gồm một truy vấn. Câu lệnh DML hoặc DDL thường gồm phần truy vấn và phần DML hoặc DDL. Oracle có thể song song hóa cả phần truy vấn và phần DML hoặc DDL của các câu lệnh SQL được liệt kê trong phần trước đó.

Lưu ý:
Dùng ngôn ngữ điều tác dữ liệu (DML) thường có các truy vấn, trong chương này "DML" chỉ tham chiếu các lệnh chèn, cập nhật và xóa.

Oracle chủ yếu song song hóa các câu lệnh SQL bằng cách sau:

- Song song hóa theo các phạm vi khối cho các điều hoạt tác vụ duyệt (các SELECT và các truy vấn con trong những câu lệnh DML và DDL).
- Song song hóa theo các phân hoạch cho các điều hoạt trên các bảng được phân hoạch và các chỉ mục.

Song song hóa theo các tiến trình server song song cho các lệnh chèn chỉ vào các bảng không được phân hoạch.

Song song theo phạm vi khối

Oracle song song hóa một truy vấn linh hoạt vào thời gian tiến hành. Trạng thái song song động chia bảng hoặc chỉ mục thành các phạm vi của những khối CSDL (phạm vi ROWID) và tiến hành điều hoạt song song trên các phạm vi khác nhau. Nếu sự phân phối hoặc vị trí của dữ liệu thay đổi, Oracle tự động thích ứng để tối ưu hóa sự song song hóa cho mỗi tiến hành của phần truy vấn của câu lệnh SQL.

Các tác vụ duyệt song song theo phạm vi khối tách bảng hoặc chỉ mục thành các phân đoạn được phân định bằng những giá trị ROWID cao và thấp. Bảng hoặc chỉ mục có thể không được phân hoạch hoặc được phân hoạch.

Đối với các bảng được phân hoạch và các chỉ mục, liên kết phạm vi ROWID nào có thể duyệt phân hoạch dù một phân hoạch có thể có nhiều phạm vi ROWID. Oracle gửi đi các số phân hoạch với các phạm vi ROWID để tránh phải tra bản đồ phân hoạch. Các thuộc tính thời gian biên dịch và tiến hành trên các cột phân hoạch giới hạn các phạm vi ROWID thích hợp với các phân hoạch, loại bỏ các tác vụ duyệt không cần thiết (dọn sạch phân hoạch).

Điều này có nghĩa là truy vấn song song truy xuất bảng được phân hoạch cùng một tác vụ duyệt thực hiện tác vụ tổng quát hoặc ít hơn cùng truy vấn trên bảng không được phân hoạch. Truy vấn trên bảng được phân hoạch tiến hành trạng thái song song tương đương, dù tổng số đĩa được truy xuất có thể giảm do dọn sạch phân hoạch.

Oracle có thể song song hóa các điều hoạt sau trên các bảng và cửa sổ chỉ mục bằng phạm vi khối (phạm vi ROWID):

- các truy vấn sử dụng các tác vụ duyệt bảng (kể cả những truy vấn trong các câu lệnh DML và DDL)
- di chuyển phân hoạch
- tách phân hoạch
- thiết lập lại phân hoạch chỉ mục
- tạo chỉ mục (chỉ mục không được phân hoạch)
- tạo table . . . nhưng được lựa (bảng không được phân hoạch)

Song song hóa theo phân hoạch

Các phân hoạch là sự phân chia các bảng logic tĩnh và các chỉ mục có thể được sử dụng để tách một số điều hoạt tiến hành thời gian dài vào thành các điều hoạt nhỏ hơn được tiến hành song song trên các phân hoạch cá biệt; không có trạng thái song song trong một phân hoạch.

Các điều hoạt trên các bảng được phân hoạch và các chỉ mục được thực hiện song song bằng cách gán các tiến trình server song song vào các phân hoạch khác nhau của bảng hoặc chỉ mục. Các thuộc tính thời gian biên dịch và tiến hành giới hạn các phân hoạch khi các tham chiếu điều hoạt phân hoạch các cột (dọn sạch phân hoạch). Điều hoạt ấy tiến hành nối tiếp khi những thuộc tính thời gian biên dịch hoặc thời gian tiến hành giới hạn phân hoạch thành một phân hoạch đơn.

Điều hoạt song song có thể sử dụng ít tiến hành server song song hơn số phân hoạch được truy xuất (vì các giới hạn tài nguyên, các gợi ý hoặc các thuộc tính bảng), nhưng mỗi phân hoạch được truy xuất bằng một tiến trình server song song đơn. Tuy nhiên, tiến trình server song song có thể truy xuất nhiều phân hoạch.

Các điều hoạt trên các bảng được phân hoạch và các chỉ mục được thực hiện song song chỉ khi một phân hoạch được truy xuất và khi tính chọn lựa của bảng hoặc chỉ mục sao cho hơn một số bảng ít nhất được xác định trước hoặc các trang chỉ mục sẽ được truy xuất.

Oracle có thể song song hóa các điều hoạt sau trên các bảng được phân hoạch và các chỉ mục theo phân hoạch:

tạo chỉ mục
tạo bảng ... như được lựa
cập nhật
xóa
chèn ... được lựa
thay đổi chỉ mục ... tái thiết lập
các truy vấn sử dụng tác vụ duyệt phạm vi trên chỉ mục được phân hoạch

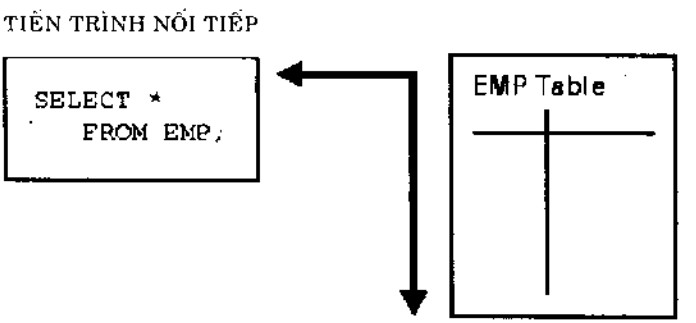
Song song hóa theo các tiến trình Parallel Server

Chỉ đối với các bảng không được phân hoạch, Oracle song song hóa các điều hoạt chèn bằng cách chia truy vấn ấy trong các tiến trình server song song. Vì các hàng mới không có các ROWID, các hàng được phân phối trong các tiến trình server song song để chèn chúng vào không gian trống.

Kiến trúc xử lý để tiến hành song song

Khi tiến hành song song hiện không được được sử dụng, thì một tiến trình server đơn thực hiện tất cả xử lý cần thiết cho tiến hành chuỗi của một câu lệnh SQL. Thí dụ, để thực hiện tác vụ duyệt toàn bảng (chẳng hạn như SELECT * FROM EMP), một tiến trình thực hiện toàn bộ điều hoạt, như được minh họa ở Hình 22-1.

Hình 22-1 Tác vụ duyệt bảng toàn bộ nối tiếp



Tiến hành song song thực hiện các điều hoạt này song song sử dụng nhiều tiến hành song song. Một tiến trình được coi là bộ điều phối song song, phân công sự tiến hành của câu lệnh tới nhiều tiến trình server song song và điều phối các kết quả từ tất cả những tiến trình server để gửi các kết quả trở lại người dùng.

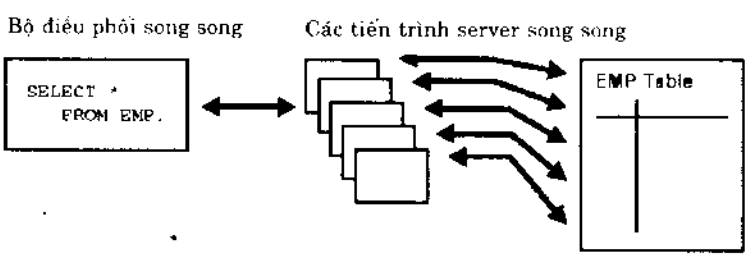
Lưu ý:
Trong ngữ cảnh này, từ ngữ "tiến trình server song song" không có nghĩa là một tiến trình của Oracle Parallel Server, nhưng thay vào đó có nghĩa là một tiến trình thực hiện một điều hoạt song song. (Tuy nhiên, trong Oracle Parallel Server, server song song có thể trải rộng trên các môi trường làm việc). Các tiến trình server song song đôi khi còn được gọi là "các tiến trình phụ thuộc".

Khi một điều hoạt được chia thành các phân đoạn để tiến hành song song bằng cấu hình tiến trình song song hàng loạt (MPP), Oracle gán một phân đoạn cá biệt của điều hoạt vào tiến trình server song song bằng cách lưu ý tới quan hệ cấu trúc của tiến trình đối với phân đoạn của bảng hoặc chỉ mục được sử dụng cho điều hoạt ấy. Bố cục vật lý của các bảng được phân hoạch và các chỉ mục ảnh hưởng quan hệ cấu trúc được sử dụng để gán truy vấn cho những tiến trình server song song.

Hình 22-2 minh họa nhiều tiến trình server song song đồng thời thực hiện tác vụ duyệt từng phần của bảng EMP được chia theo phạm vi khối (phân hoạch động). Các tiến trình server song song gửi các kết quả trở lại tiến trình bộ điều phối song song để tập hợp các phân đoạn vào tác vụ duyệt toàn bảng được yêu cầu.

Bộ điều phối song song tách các chức năng tiến hành thành các phân đoạn song song và sau đó tích hợp các kết quả từng phần được các tiến trình server song song tạo. Số tiến trình server song song được gán vào điều hoạt đơn là mức trạng thái song song đối với một điều hoạt. Nhiều điều hoạt trong cùng câu lệnh SQL đều có cùng mức trạng thái song song (xem "Quyết định mức trạng thái song song cho các điều hoạt").

Hình 22-2 Tác vụ duyệt toàn bảng



Vùng chia Parallel Server

Khi một môi trường làm việc khởi động, Oracle tạo một vùng chia của những tiến trình server song song có thể sử dụng cho bất kỳ điều hoạt nào. Tham số khởi hoạt PARALLEL_MIN_SERVERS ấn định số các tiến trình server song song mà Oracle tạo vào lúc khởi động môi trường làm việc.

Khi tiến hành một điều hoạt song song, bộ điều phối song song có được các tiến trình server song song từ vùng chia và gán chúng vào điều hoạt ấy. Nếu cần, Oracle có thể tạo các tiến trình

server song song bổ sung cho điều hoạt ấy. Những tiến trình server song song này vẫn duy trì với hoạt động ấy trong suốt tiến hành tác vụ, sau đó trở thành có thể sử dụng cho các điều hoạt khác. Sau khi câu lệnh được xử lý hoàn toàn, những tiến trình server song song trở lại vùng chia ấy.

Lưu ý:
Bộ điều phối song song và những tiến trình server song song chỉ có thể phục vụ một câu lệnh một lần. Bộ điều phối song song không thể điều phối, thí dụ, một truy vấn song song và một câu lệnh DML vào cùng thời gian.
Bạn có thể xác lập PARALLEL_MIN_SERVERS thành một giá trị cao hơn nên bạn cần tiến hành câu lệnh song song đồng thời.

Khi một người dùng phát hành một câu lệnh SQL, bộ tối ưu hóa quyết định xem có tiến hành các điều hoạt song song và quyết định mức trạng thái song song cho mỗi điều hoạt không. Bạn có thể ấn định số tiến trình server song song cần thiết cho một điều hoạt bằng các cách khác nhau.

Nếu bộ tối ưu hóa nhắm vào câu lệnh để tiến hành song song, thì chuỗi các sự kiện sau xảy ra:
Tiến trình nền trước của câu lệnh SQL trở thành bộ điều phối song song.
Bộ điều phối song song có được tối đa các tiến trình server song song cần thiết (được xác định bằng mức trạng thái song song) từ vùng chia server hoặc tạo các tiến trình server song song mới cần thiết.
Oracle tiến hành câu lệnh bằng một chuỗi các điều hoạt. Mỗi điều hoạt được thực hiện song song, nếu có thể.
Khi xử lý câu lệnh hoàn tất, bộ điều phối cho tiến trình người dùng phát hành câu lệnh ấy những dữ liệu kết quả bất kỳ và đưa các tiến trình server song song trở về vùng chia server.

Bộ điều phối song song gọi dựa trước các tiến trình server song song khi tiến hành câu lệnh SQL (không phải khi phân ngữ câu lệnh). Do đó, khi tiến hành song song được sử dụng với server đa xâu, tiến trình server xử lý lệnh gọi EXECUTE của câu lệnh người dùng trở thành tiến trình bộ điều phối cho câu lệnh ấy.

Những số tiến trình server song song khác nhau

Nếu số điều hoạt song song được xử lý đồng thời bởi môi trường làm việc thay đổi rất nhiều, Oracle tự động thay đổi số tiến trình server song song trong vùng chia.

Nếu số điều hoạt song song tăng lên, Oracle tạo các tiến trình server song song bổ sung để điều quản các yêu cầu đến. Tuy nhiên, Oracle không bao giờ tạo các tiến trình server song song cho một môi trường làm việc nhiều hơn những gì được ấn định bằng tham số khởi hoạt PARALLEL_MAX_SERVERS.

Nếu số điều hoạt song song giảm, Oracle kết thúc bất kỳ những tiến trình server song song đã không còn hoạt động trong khoảng thời gian được ấn định bằng tham số khởi hoạt PARALLEL_SERVER_IDLE_TIME. Oracle không giảm kích thước của vùng chia xuống dưới giá trị của PARALLEL_MIN_SERVERS bất kể những tiến trình server song song ấy không hoạt động trong bao lâu.

Xử lý không đủ những tiến trình server song song

Oracle có thể xử lý điều hoạt song song ít hơn số tiến trình yêu cầu; xem "số tiến trình server song song ít nhất" về ấn định tiến trình song song ít nhất bằng tham số khởi hoạt PARALLEL_MIN_PERCENT.

Nếu tất cả tiến trình server song song trong vùng chia được chiếm giữ và số tiến trình server song song tối đa đã được khởi động, bộ điều phối song song chuyển sang tiến trình nối tiếp.

> Thông tin bổ sung:
> Xem Oracle8 điều hợp về thông tin giám sát một vùng chia môi trường làm việc của những tiến trình server song song và quyết định những giá trị thích hợp của các tham số khởi hoạt.

Song song hóa các câu lệnh SQL

Mỗi câu lệnh SQL phải trải qua một tiến trình tối ưu hóa và song song hóa khi nó được phân ngữ. Do đó, khi dữ liệu thay đổi, nếu thêm một kế hoạch tiến hành tối ưu hóa hoặc kế hoạch song song hóa trở thành sử dụng được, thì Oracle có thể tự động thích ứng với trường hợp mới. (Xem thêm "Bộ Tối Ưu Hóa".)

Sau khi bộ tối ưu hóa xác định kế hoạch tiến hành của một câu lệnh, bộ điều phối song song quyết định phương pháp song song hóa cho mỗi điều hoạt trong kế hoạch tiến hành (thí dụ, song song hóa một tác vụ duyệt toàn bảng). Bộ điều phối phải quyết định xem một hoạt động có thể được thực hiện song song không, nếu được, bao nhiêu tiến trình server song song phải đăng ký (nghĩa là, mức trạng thái song song).

Xem "Xác lập mức trạng thái song song" và "Các nguyên tắc song song hóa cho các câu lệnh SQL".

Phân chia truy vấn trong các tiến trình Parallel Server

Bộ điều phối song song giám sát những đòi hỏi phân phối lại của mỗi điều hoạt. Một đòi hỏi phân phối lại của điều hoạt là cách trong đó các hàng được điều hoạt bởi điều hoạt phải được chia hoặc được phân phối lại trong các tiến trình server song song.

Sau khi quyết định đòi hỏi phân phối lại cho mỗi điều hoạt trong kế hoạch tiến hành, bộ tối ưu hóa quyết định thứ tự các điều hoạt trong kế hoạch tiến hành phải được thực hiện. Với thông tin này, bộ tối ưu hóa quyết định luồng dữ liệu của câu lệnh ấy.

Hình 22-3 minh họa luồng dữ liệu của truy vấn sau:

```
SELECT dname, MAX(sal), AVG(sal)
  FROM emp, dept
    WHERE emp.deptno = dept.deptno
    GROUP BY dname;
```

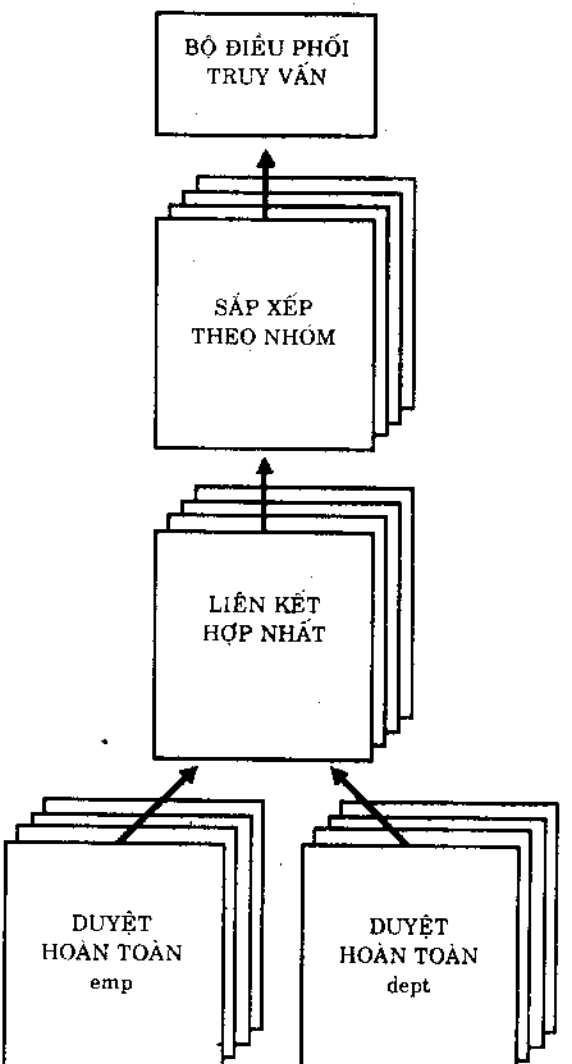
Trạng thái song song trong các điều hoạt

Các điều hoạt đòi hỏi kết xuất của các điều hoạt khác được coi là các điều hoạt cha. Ở Hình 22-3, điều hoạt GROUP BY SORT là cha của điều hoạt MERGE JOIN vì GROUP BY SORT đòi hỏi kết xuất MERGE JOIN.

Các điều hoạt cha có thể bắt đầu tiêu thụ các hàng ngay khi các điều hoạt con đã tạo ra các hàng. Trong thí dụ trước, khi những tiến trình server song song đang tạo các hàng trong điều hoạt FULL SCAN DEPT, một tập hợp khác gồm những tiến trình server song song có thể bắt đầu thực hiện điều hoạt MERGE JOIN để tiêu thụ các hàng.

Một trong hai điều hoạt được thực hiện đồng thời tập hợp những tiến trình server song song được định trước của riêng nó. Do đó, cả các điều hoạt truy vấn và chính cây luồng dữ liệu có trạng thái song song. Trạng thái song song của một điều hoạt cá biệt được gọi là trạng thái điều hoạt bên trong và trạng thái song song giữa các điều hoạt trong một cây luồng dữ liệu được gọi là trạng thái song song liên điều hoạt.

Hình 22-3 Biểu đồ luồng truyền dữ liệu cho một liên kết của các bảng EMP và DEPT



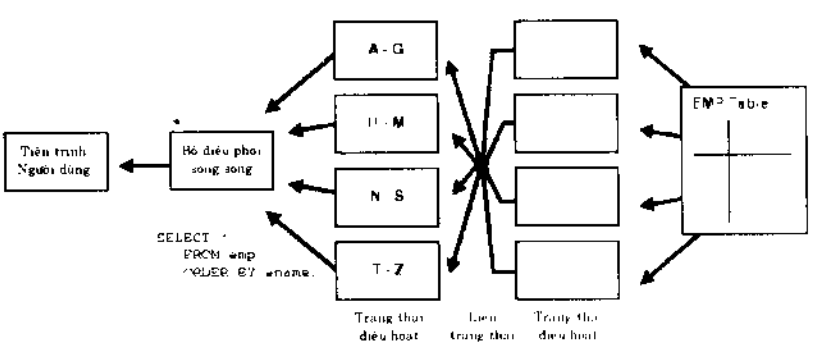
Do tính chất nhà sản xuất/người tiêu thụ của các điều hoạt server Oracle, chỉ hai điều hoạt trong một cây định trước phải được thực hiện đồng thời để giảm tối đa thời gian tiến hành.

Để minh họa trạng thái song song bên trong và trạng thái song song liên điều hoạt, xét câu lệnh sau:
SELECT * FROM emp ORDER BY ename;

Kế hoạch tiến hành gồm tác vụ duyệt toàn bộ của bảng EMP được theo sau bằng một sắp xếp những hàng được truy hồi dựa trên giá trị của cột ENAME. Mục đích của thí dụ này, giả sử cột ENAME không có chỉ mục. Ngoài ra, giả sử mức trạng thái song song đối với truy vấn được xác lập là 4, nghĩa là 4 tiến trình server song song có thể hoạt động cho một điều hoạt định trước bất kỳ.

Hình 22-4 minh họa tiến hành song song của truy vấn thí dụ.

Hình 22-4 Trạng thái song song liên điều hoạt và phân hoạch năng động



Như bạn có thể thấy ở Hình 22-4, có 8 tiến trình server song song liên quan đến truy vấn ấy dù mức trạng thái song song là 4

Điều này xảy ra vì toán tử cha và con có thể được thực hiện cùng thời gian (trạng thái song song liên điều hoạt).

Ngoài ra, thấy tất cả những tiến trình server song song liên quan trong điều hoạt tác vụ duyệt gửi các hàng tới tiến trình server song song thích hợp thực hiện điều hoạt sắp xếp. Nếu một hàng được duyệt bằng một tiến trình server song song có một đối tượng cho cột ENAME giữa A và G, hàng đó được gửi tới tiến trình server song song ORDER BY đầu tiên. Khi tác vụ duyệt ấy hoàn tất, các tiến trình sắp xếp có thể cho bộ điều phối các kết quả sắp xếp và đến lượt nó cho người dùng các kết quả truy vấn hoàn toàn.

Lưu ý:
Khi một tập hợp các tiến trình server song song hoàn tất điều hoạt của nó, nó di chuyển tới các điều hoạt cao hơn trong luồng dữ liệu. Thí dụ, trong biểu đồ trước, nếu đã có một điều hoạt ORDER BY khác sau điều hoạt ORDER BY, thì các tiến trình server song song thực hiện tác vụ duyệt bảng tiến hành điều hoạt ORDER BY thứ hai sau khi hoàn tất tác vụ duyệt bảng ấy.

Xác lập mức trạng thái song song

Bộ điều phối song song có thể đăng ký hai hoặc nhiều tiến trình server song song của môi trường làm việc để xử lý một câu lệnh SQL. Số tiến trình server song song được kết hợp với một điều hoạt đơn được coi là mức trạng thái song song.

Mức trạng thái song song được ấn định ở mức câu lệnh (với các gợi ý hoặc mệnh đề PARALLEL), ở mức bảng hoặc chỉ mục (trong định nghĩa của bảng hoặc chỉ mục) hoặc theo mặc định dựa trên số đĩa hoặc CPU.

Lưu ý mức trạng thái song song chỉ áp dụng trực tiếp vào trạng thái song song điều hoạt bên trong. Nếu trạng thái điều hoạt bên trong có thể xảy ra, thì tổng số tiến trình server song song cho

một câu lệnh có thể gấp đôi mức trạng thái song song được ấn định. Không hơn hai điều hoạt cập nhật được thực hiện đồng thời.

Quyết định mức trạng thái song song cho các điều hoạt

Bộ điều phối song song quyết định mức trạng thái song song bằng cách xét nhiều định chuẩn. Trước hết bộ điều phối kiểm các gợi ý hoặc mệnh đề PARALLEL được xác định trong chính câu lệnh SQL, sau đó nó tìm định nghĩa của bảng hoặc chỉ mục và cuối cùng nó kiểm mức trạng thái song song mặc định (xem "Mức trạng thái song song mặc định"). Khi mức trạng thái song song được tìm thấy một trong các định chuẩn này, nó trở thành mức trạng thái song song cho điều hoạt ấy.

Xem thêm "Các nguyên tắc song song hóa đối với các câu lệnh SQL".

Các gợi ý, các mệnh đề PARALLEL, các định nghĩa bảng hoặc chỉ mục và những giá trị mặc định chỉ quyết định số tiến trình server song song mà bộ điều phối yêu cầu một điều hoạt định trước. Số những tiến trình server song song thực sự được sử dụng phụ thuộc vào bao nhiêu tiến trình có thể sử dụng trong vùng chia server song song (xem "Vùng chia server song song" và xem trạng thái song song điều hoạt bên trong nào có thể xảy ra (xem "Các điều hoạt giữa các trạng thái song song").

Các gợi ý

Bạn có thể ấn định các gợi ý (*Hint*) trong một câu lệnh SQL để xác lập mức trạng thái song song cho một bảng hoặc chỉ mục và hoạt động lưu trữ của điều hoạt ấy.
Gợi ý PARALLEL chỉ được sử dụng cho các điều hoạt trên các bảng. Bạn có thể sử dụng nó để song song hóa những truy vấn và các câu lệnh DML (INSERT, UPDATE và DELETE).

Gợi ý PARALLEL_INDEX song song hóa một tác vụ duyệt phạm vi chỉ mục của một chỉ mục được phân hoạch. (trong một diều hoạt chỉ mục, gợi ý PARALLEL không hợp lệ và bị bỏ qua).

Thông tin bổ sung:
Tham chiếu Oracle8 Tuning về sử dụng các gợi ý trong các câu lệnh SQL và cú pháp ấn định cho các gợi ý PARALLEL, NOPARALLEL, PARALLEL_INDEX, CACHE và NOCACHE.

Các định nghĩa bảng và chỉ mục

Bạn có thể ấn định mức trạng thái song song trong một định nghĩa bảng hoặc chỉ mục. Sử dụng một trong các câu lệnh SQL sau để xác lập mức trạng thái song song cho một giá trị hoặc một chỉ mục: CREATE TABLE, ALTER TABLE, CREATE INDEX hoặc ALTER INDEX.

Thông tin bổ sung:
Tham chiếu Oracle8 SQL Reference về cú pháp đầy đủ của các câu lệnh SQL.

Mức trạng thái song song mặc định

Mức trạng thái song song mặc định được sử dụng khi bạn không ấn định mức trạng thái song song trong một gợi ý hoặc trong định nghĩa của một bảng hoặc chỉ mục. Mức trang thái song song mặc định thích hợp cho phần lớn ứng dụng. Xem thêm Oracle8 Tuning về điều chỉnh mức trạng thái song song.

Mức trạng thái song song mặc định cho một câu lệnh SQL được quyết định bằng các yếu tố sau.

Số CPU trong hệ thống.
Số môi trường làm việc Oracle Parallel Server.
Số đĩa (hoặc tập tin, nếu thông tin quan hệ cấu trúc không sử dụng được) mà bảng hoặc chỉ mục được lưu trữ.

Đối với song song hóa phân hoạch, số những phân hoạch sẽ được truy xuất, dựa trên dọn sạch phân hoạch (nếu thích hợp).

Đối với các điều hoạt DML song song có duy trì chỉ mục toàn cục, số giao dịch trống ít nhất liệt kê trong tất cả những chỉ mục toàn cục để được cập nhật. Số giao dịch trống ít nhất liệt kê cho chỉ mục toàn cục được phân hoạch là số giao dịch ít nhất qua tất cả các phân hoạch chỉ mục. Điều này cần thiết để ngăn cản "tự bế tắc".

Thí dụ, nếu hệ thống của bạn có 20 CPU và bạn phát hành truy vấn song song trên bảng được lưu trữ trên 15 ổ đĩa, thì mức trạng thái song song mặc định cho truy vấn của bạn là 15 server truy vấn.

Lưu ý:
Oracle có được những thông tin về các đĩa và các CPU từ hệ điều hành.

Tuy nhiên, các yếu tố trên quyết định số tiến trình server song song mặc định sử dụng, số tiến trình thực sự được sử dụng bị giới hạn bởi tính khả dụng của chúng trên những môi trường làm việc được yêu cầu trong thời gian tiến hành. Tham số khởi hoạt PARALLEL_MAX_SERVERS xác lập một giới hạn trên về tổng số những tiến trình server song song mà một môi trường làm việc có thể có.

Nếu phần chia nhỏ nhất của những tiến trình server song song yêu cầu không được sử dụng (được ấn định bằng tham số khởi hoạt PARALLEL_MIN_PERCENT), một lỗi người dùng phát sinh. Sau đó người dùng có thể thử lại truy vấn ấy bằng trạng thái song song ít hơn.

Lưu ý :
Các tham số khởi hoạt PARALLEL_DEFAULT_SCANSIZE và

PARALLEL_DEFAULT_MAX_SCANS có thể sử dụng được nữa.

Số tiến trình Parallel Server ít nhất

Oracle có thể thực hiện một điều hoạt song song bao lâu ít nhất hai tiến trình server song song có thể sử dụng. Nếu quá ít tiến trình server song song có thể của, thì câu lệnh SQL của bạn có thể tiến hành chậm hơn mong muốn. Bạn có thể ấn định số phần trăm tối thiểu những tiến trình server song song được yêu cầu phải được sử dụng để điều hoạt ấy tiến hành. Điều này bảo đảm câu lệnh SQL của bạn tiến hành với hiệu suất song song tối thiểu có thể chấp nhận. Nếu số phần trăm những tiến trình server song song được yêu cầu không có thể xảy ra, thì câu lệnh SQL không tiến hành và cho trở lại một lỗi.

Tham số khởi hoạt PARALLEL_MIN_PERCENT xác định số phần trăm nhỏ nhất của những tiến trình server được yêu cầu. Tham số này ảnh hưởng đến các điều hoạt DML và DDL cũng như các truy vấn.

Thí dụ, nếu bạn ấn định cho tham số này là 50, thì ít nhất 50% của các tiến trình server song song được yêu cầu đối với điều hoạt song song bất kỳ phải có thể xảy ra để điều hoạt thành công. Nếu 20 tiến trình server song song được yêu cầu, thì ít nhất 10 tiến trình phải có thể xảy ra hoặc người dùng nhận được một lỗi. Nếu PARALLEL_MIN_PERCENT được xác lập là rỗng, thì tất cả những điều hoạt song song sẽ tiến hành bao lâu ít nhất hai tiến trình server song song có thể xử lý.

Giới hạn số môi trường làm việc có thể sử dụng

Trong Oracle Parallel Server, các nhóm môi trường làm việc có thể được sử dụng để giới hạn số môi trường làm việc tham gia vào điều hoạt song song. Bạn có thể tạo bất kỳ số nhóm môi trường làm việc, mỗi nhóm gồm một hoặc nhiều môi trường làm

việc. Sau đó bạn có thể ấn định nhóm môi trường làm việc nào được sử dụng cho bất kỳ điều hoạt song song hoặc tất cả các điều hoạt song song. Những tiến trình server song song sẽ chỉ được sử dụng trên những môi trường làm việc là các thành viên của nhóm môi trường làm việc được ấn định.

Cân bằng tải tác vụ

Để tối ưu hóa hiệu suất, tất cả những tiến trình server song song phải có các mức tải tác vụ bằng nhau. Đối với các câu lệnh SQL được song song hóa bằng phạm vi khối hoặc bằng các tiến trình server song song, thì tải tác vụ được chia một cách linh hoạt trong các tiến trình server song song. Điều này giảm tối đa độ lệch tải tác vụ xảy ra khi những tiến trình server song song thực hiện nhiều tác vụ hơn các tiến trình khác.

Đối với những câu lệnh SQL được song song hóa bằng các phân hoạch, nếu tải tác vụ được phân phối đều trong các phân hoạch thì bạn có thể tối ưu hóa hiệu suất bằng cách so khớp số những tiến trình server song song với số phân hoạch hoặc bằng cách chọn mức trạng thái song song vì thế số phân hoạch gồm nhiều số tiến trình.

Thí dụ, nếu một bảng có 10 phân hoạch và điều hoạt song song chia đều số tác vụ giữa chúng, bạn có thể sử dụng 10 tiến trình server song song (mức trạng thái song song = 10) để thực hiện tác vụ trong khoảng bằng 1/10 thời gian một tiến trình sẽ thực hiện hoặc bạn có thể sử dụng 5 tiến trình thực hiện tác vụ trong khoảng 1/5 thời gian hoặc 2 tiến trình thực hiện tác vụ ấy bằng nửa số thời gian.

Tuy nhiên, nếu bạn sử dụng 9 tiến trình thực hiện trên 10 phân hoạch, thì tiến trình thứ nhất kết thúc tác vụ của nó trên một phân hoạch, rồi bắt đầu hoạt động trên phân hoạch thứ 10; và khi các tiến trình khác kết thúc tác vụ của chúng, chúng trở

thành không hoạt động. Điều này không tạo ra được hiệu suất tốt khi tác vụ được chia đều trong các phân hoạch. Khi tác vụ không được chia đều, thì hiệu suất khác nhau tùy theo xem phân hoạch còn lại cuối cùng có nhiều hơn hoặc ít hơn các phân hoạch kia.

Tương tự, nếu bạn sử dụng những tiến trình hoạt động trên 10 phân hoạch và tác vụ được chia đều, thì mỗi tiến trình hoạt động trên phân hoạch thứ hai sau khi kết thúc phân hoạch thứ nhất của nó, nhưng chỉ hai trong những tiến trình hoạt động trên phân hoạch thứ ba khi hai tiến trình khác vẫn không hoạt động.

Nói chung, bạn không thể chấp nhận thời gian mất đi để thực hiện một điều hoạt song song trên các phân hoạch N với các tiến trình server song song P sẽ là N/P, vì khả năng một số tiến trình có thể phải chờ các tiến trình khác kết thúc hoạt động trên (các) tiến trình cuối cùng. Tuy nhiên, bằng cách chọn một mức trạng thái song song, bạn có thể giảm tối đa độ lệch tải tác vụ và tối ưu hóa hiệu suất.

Xem thêm về cân bằng tải tác vụ với quan hệ cấu trúc đĩa "Quan hệ cấu trúc và DML song song".

Các nguyên tắc song song hóa đối với các câu lệnh SQL

Câu lệnh SQL có thể được song song hóa nếu nó có một gợi ý song song hoặc nếu bảng hoặc chỉ mục đang được điều hoạt đã là PARALLEL được khai báo với câu lệnh CREATE hoặc ALTER. Hơn nữa, một câu lệnh ngôn ngữ định nghĩa dữ liệu (DDL) có thể được song song hóa bằng cách sử dụng mệnh đề PARALLEL. Tuy nhiên, không phải tất cả những phương pháp này áp dụng cho tất cả những kiểu câu lệnh SQL.

Sự song song hóa có hai thành phần: quyết định song song hóa và mức trạng thái song song. Các thành phần này được quyết

định khác nhau đối với các truy vấn, các điều hoạt DDL và các điều hoạt DML.

Để quyết định mức trạng thái song song, Oracle khảo sát các đối tượng tham chiếu.

Truy vấn song song khảo sát từng bảng và chỉ mục, trong phần có thể truy vấn đang được song song hoá để quyết định là bảng tham chiếu nào. Nguyên tắc căn bản để lựa bảng hoặc chỉ mục với mức trạng thái song song lớn nhất.

DML song song (chèn, cập nhật và xóa), đối tượng tham chiếu quyết định mức trạng thái song song là bảng đang được sửa đổi bằng một hoạt động chèn, cập nhật hoặc xóa. DML song song còn bổ sung một số giới hạn vào mức trạng thái song song để ngăn cản bị bế tắc. Nếu câu lệnh DML song song có một truy vấn con, mức trạng thái song song của truy vấn con giống như điều hoạt DML.

Đối với DDL song song, đối tượng tham chiếu quyết định mức trạng thái song song là bảng, chỉ mục hoặc phân hoạch đang được tạo, thiết kế lại, tách hoặc di chuyển. Nếu câu lệnh DDL song song có truy vấn con, mức trạng thái song song của truy vấn con giống như điều hoạt DDL.

Những nguyên tắc đối với các truy vấn song song hóa

Quyết định song song hóa

Câu lệnh SELECT chỉ có thể được song song hóa nếu các điều kiện sau được thỏa mãn:

- Truy vấn có định chuẩn gợi ý "song song" (PARALLEL hoặc PARALLEL_INDEX) hoặc các đối tượng giản đồ được tham chiếu trong truy vấn có khai báo PARALLEL được kết hợp với chúng.
- Ít nhất một trong các bảng được ấn định trong truy vấn ấy đòi hỏi một trong những điều kiện sau:
  - tác vụ duyệt toàn bảng
  - tác vụ duyệt phạm vi chỉ mục nối nhiều phân hoạch

Mức trạng thái song song

Mức trạng thái song song đối với một truy vấn được xác định bởi các nguyên tắc sau:

Truy vấn sử dụng mức trạng thái tối đa có được từ tất cả những khai báo bảng có liên quan trong truy vấn và tất cả những chỉ mục tiềm năng là ứng viên để thỏa mãn truy vấn ấy (các đối tượng tham chiếu). Nghĩa là bảng hoặc chỉ mục có mức trạng thái song song nhiều nhất quyết định mức trạng thái song song của truy vấn (hướng dẫn truy vấn tối đa).

Nếu một bảng có cả định chuẩn gợi ý "song song" trong truy vấn và khai báo song song trong định chuẩn bảng của nó, định chuẩn gợi ý tiến hành theo định chuẩn song song.

Các nguyên tắc để song song hóa UPDATE và DELETE

Các điều hoạt cập nhật và xóa được song song hóa bằng phân hoạch. Các điều hoạt cập nhật và xóa chỉ có thể được song song hóa trên các bảng được phân hoạch; mức trạng thái cập nhật/xóa không thể xảy ra trong một phân hoạch hoặc không ở bảng không được phân hoạch.

Bạn có hai cách để ấn định các hướng dẫn song song cho cửa sổ điều hoạt UPDATE và DELETE (giả sử là chế độ PARALLEL DML được cho phép):

Mệnh đề song song được ấn định trong định nghĩa của bảng được cập nhật hoặc được xóa (đối tượng tham chiếu).

Cập nhật hoặc xóa gợi ý song song được ấn định ở câu lệnh.

Các gợi ý song song được đặt ngay sau các từ khóa UPDATE hoặc DELETE trong các câu lệnh UPDATE và DELETE. Gợi ý còn áp dụng vào tác vụ duyệt cơ sở của bảng bị thay đổi.

Các mệnh đề song song trong các lệnh CREATE TABLE và ALTER TABLE ấn định trạng thái song song bảng. Nếu mệnh

để song song tồn tại trong định nghĩa bảng, nó quyết định trạng thái song song của những câu lệnh DML cũng như các truy vấn. Nếu câu lệnh DML có các gợi ý rõ ràng cho một bảng, thì các gợi ý đó chạy đè kết quả của các mệnh đề song song cho bảng đó.

Quyết định song song hóa

Nguyên tắc sau quyết định xem điều hoạt cập nhật/xóa có phải song song hóa trong câu lệnh cập nhật/xóa không:
Điều hoạt UPDATE hoặc DELETE sẽ được song song hóa nếu và chỉ nếu bảng được cập nhật/xóa có định chuẩn PARALLEL hoặc gợi ý PARALLEL được ấn định trong câu lệnh DML.

Nếu câu lệnh có các truy vấn con hoặc các khung nhìn cập nhật được, chúng có thể có các gợi ý hoặc mệnh đề song song tách riêng, nhưng những hướng dẫn song song này không ảnh hưởng đến quyết định song song hóa cập nhật hoặc xóa.

Dù gợi ý hoặc mệnh đề trên các bảng được sử dụng bởi cả truy vấn và các phần cập nhật/xóa để quyết định trạng thái song song, nhưng quyết định song song hóa các phần cập nhật/xóa được thực hiện độc lập với phần truy vấn và ngược lại.

Mức trạng thái song song

Mức trạng thái song song được quyết định bởi các nguyên tắc giống nhau đối với các truy vấn. Lưu ý là trong trường hợp các điều hoạt cập nhật và xóa, chỉ một bảng (chỉ một đối tượng tham chiếu) có liên quan là bảng đích được sửa đổi.

Nguyên tắc ưu tiên quyết định mức trạng thái song song cho điều hoạt cập nhật/xóa là định chuẩn gợi ý song song cập nhật hoặc xóa tiến hành trước trên định chuẩn khai báo song song của bảng đích:
Gợi ý Update/Delete > định chuẩn khai báo của bảng đích

Mức trạng thái song song tối đa bạn có thể đạt được bằng với số phân hoạch trong bảng ấy. Tiến trình server song song có thể cập nhật vào hoặc xóa khỏi nhiều phân hoạch, nhưng mỗi phân hoạch chỉ có thể được cập nhật hoặc xóa bằng một tiến trình server song song.

Nếu mức trạng thái nhỏ hơn số phân hoạch, thì tiến trình thứ nhất kết thúc hoạt động trên một phân hoạch, tiếp tục hoạt động trên một phân hoạch khác, vân vân ... cho tới khi hoạt động ấy được kết thúc trên tất cả các phân hoạch. Nếu mức trạng thái song song lớn hơn số phân hoạch có liên quan trong điều hoạt, thì tiến trình server song song vượt hơn không có tác vụ hoạt động.

Thí dụ 1:

```
UPDATE tbl_1 SET c1=c1+1 WHERE c1>100;
```

Nếu TBL_1 là một bảng được phân hoạch và định nghĩa bảng của nó có mệnh đề song song, thì điều hoạt cập nhật sẽ được song song hóa dù tác vụ duyệt trên bảng ấy nối tiếp (chẳng hạn như một tác vụ duyệt chỉ mục), giả sử là bảng có hơn một phân hoạch với C1 lớn hơn 100.

Thí dụ 2:

```
UPDATE /*+ PARALLEL(tbl_2,4) */ tbl_2 SET c1=c1+1;
```

Cả tác vụ duyệt và điều hoạt cập nhật trên TBL_2 sẽ được song song hóa với mức 4.

Các nguyên tắc song song hóa INSERT ... SELECT

Một câu lệnh INSERT ... SELECT song song hóa các điều hoạt INSERT và SELECT của nó độc lập, trừ mức trạng thái song song.

Bạn có thể ấn định gợi ý "song song" sau từ khóa INSERT trong câu lệnh INSERT ... SELECT. Vì các bảng được truy vấn thường không giống như được chèn vào, gợi ý ấy cho phép bạn ấn định các hướng dẫn song song riêng biệt cho điều hoạt chèn.

Bạn có bốn cách để ấn định các hướng dẫn cho câu lệnh INSERT... SELECT (giả sử chế độ PARALLEL DML được cho phép):

(Các) gợi ý song song select được ấn định ở câu lệnh.

(Các) câu lệnh song song được ấn định trong định nghĩa của các bảng được lựa.

Gợi ý song song INSERT được ấn định ở câu lệnh.

Mệnh đề song song được ấn định trong định nghĩa của các bảng được chèn vào.

Quyết định song song hóa

Nguyên tắc sau quyết định xem điều hoạt chèn có phải song song hóa trong câu lệnh INSERT... SELECT không:

Điều hoạt INSERT sẽ được song song hóa nếu và chỉ nếu bảng được chèn vào (đối tượng tham chiếu) có định chuẩn khai báo PARALLEL hoặc gợi ý PARALLEL được ấn định sau INSERT trong câu lệnh DML.

Vì thế quyết định song song hóa điều hoạt chèn được thực hiện độc lập với điều hoạt lựa và ngược lại.

Mức trạng thái song song

Một khi quyết định song song hóa điều hoạt lựa và/hoặc chèn được thực hiện, một hướng dẫn song song được chọn để quyết định mức trạng thái song song của cả câu lệnh sử dụng nguyên tắc ưu tiên sau:

Hướng dẫn Insert Hint > định chuẩn khai báo song song của bảng chèn > hướng dẫn Maximum Query

Trong đó hướng dẫn Maximum Query có nghĩa là nhiều bảng và chỉ mục, bảng hoặc chỉ mục có mức trạng thái song song tối đa quyết định trạng thái song song đối với điều hoạt truy vấn.

Hướng dẫn song song chọn lựa được áp dụng vào cả các điều hoạt lựa và chèn.

Thí dụ:

Trong thí dụ sau, mức trạng thái song song được sử dụng sẽ là 2, là mức được ấn định trong gợi ý Insert:

```
INSERT /*+ PARALLEL(tbl_ins,2) */ INTO tbl_ins
  SELECT /*+ PARALLEL(tbl_sel,4) */ * FROM tbl_sel;
```

Các nguyên tắc song song hóa các câu lệnh DDL

Quyết định song song hóa

Các điều hoạt DDL có thể được song song hóa nếu mệnh đề PARALLEL (khai báo) được ấn định trong cú pháp. Trong trường hợp của CREATE INDEX và ALTER INDEX ... REBUILD hoặc ALTER INDEX ... REBUILD PARTITION, khai báo song song được lưu trữ trong từ điển dữ liệu.

Mức trạng thái song song

Mức trạng thái song song được quyết định bởi định chuẩn trong mệnh đề PARALLEL. Tái thiết lập chỉ mục được phân hoạch không bao giờ được song song hóa.

Các nguyên tắc đối với song song hóa CREATE INDEX, REBUILD INDEX, MERGE/SPLIT PARTITION

CREATE INDEX hoặc ALTER INDEX ... REBUILD song song

Các câu lệnh CREATE INDEX và ALTER INDEX ... REBUILD chỉ có thể được song song hóa bằng mệnh đề PARALLEL.

ALTER INDEX ... REBUILD chỉ có thể được song song cho chỉ mục không được phân hoạch, nhưng ALTER INDEX ... REBUILD PARTITION có thể được song song hóa bằng môi trường PARALLEL.

Tác vụ duyệt cho ALTER INDEX ... REBUILD (không được phân hoạch), ALTER INDEX ... REBUILD PARTITION và CREATE INDEX có cùng trạng thái song song như điều hoạt REBUILD hoặc CREATE và sử dụng cùng mức trạng thái song song. Nếu mức trạng thái song song không được ấn định cho REBUILD hoặc CREATE, mặc định là số CPU.

MOVE PARTITION hoặc SPLIT PARTITION song song

Các câu lệnh ALTER INDEX ... MOVE PARTITION và ALTER INDEX ... SPLIT PARTITION có thể được song song hóa bằng mệnh đề PARALLEL. Các điều hoạt duyệt của chúng có cùng trạng thái như các điều hoạt MOVE/SPLIT tương ứng. Nếu mức trạng thái song song không được ấn định, thì mặc định là số CPU.

Các nguyên tắc song song hóa Create Table as Select

Câu lệnh CREATE TABLE ... AS SELECT có hai phần:
phần CREATE (DDL)
phần SELECT (truy vấn)

Oracle có thể song song hóa cả hai phần của câu lệnh. Phần CREATE thực hiện theo cùng các nguyên tắc như cửa sổ điều hoạt DDL khác.

Quyết định song song hóa (phần Query)

Phần truy vấn của câu lệnh CREATE TABLE ... AS SELECT có thể được song song hóa chỉ nếu các điều kiện sau được thỏa đáng:

Truy vấn có định chuẩn gợi ý "song song" (PARALLEL hoặc PARALLEL_INDEX) hoặc phần CREATE của câu lệnh có định chuẩn mệnh đề PARALLEL hoặc các đối tượng giản

đồ được tham chiếu trong truy vấn ấy có khai báo PARALLEL được kết hợp với chúng.

Ít nhất một trong các bảng được ấn định trong truy vấn đòi hỏi một trong các điều hoạt sau:

Duyệt toàn bảng

Duyệt phạm vi chỉ mục nối nhiều phân hoạch

Mức trạng thái song song (phần Query)

Mức trạng thái song song đối với phần truy vấn của câu lệnh CREATE TABLE ... AS SELECT được quyết định bởi một trong các nguyên tắc sau:

Phần truy vấn sử dụng những giá trị được xác định trong mệnh đề PARALLEL của phần CREATE.

Nếu mệnh đề PARALLEL không được xác định, mức trạng thái song song mặc định là số CPU.

Lưu ý, bất kỳ các giá trị được xác định trong một gợi ý cho trạng thái song song sẽ được bỏ qua.

Quyết định song song (phần Create)

Điều hoạt CREATE của CREATE TABLE ... AS SELECT chỉ có thể được song song hóa bằng mệnh đề PARALLEL.

Khi điều hoạt CREATE của CREATE TABLE ... AS SELECT được song song hóa, Oracle cũng song song hóa điều hoạt duyệt nếu có thể. Điều hoạt duyệt không thể được song song hóa, thí dụ, nếu:

Mệnh đề SELECT có gợi ý NOPARALLEL

Điều hoạt duyệt một chỉ mục của bảng không được phân hoạch

Khị điều hoạt CREATE không được song song hóa, SELECT có thể được song song hóa nếu nó có gợi ý PARALLEL hoặc nếu bảng được lựa (hoặc chỉ mục được phân hoạch) có khai báo song song.

Mức trạng thái song song (phần Create)

Mức trạng thái song song đối với điều hoạt CREATE và đối với điều hoạt SELECT nếu nó được song song hóa, được ấn định bằng mệnh đề PARALLEL của câu lệnh CREATE. Nếu câu lệnh CREATE không ấn định mức trạng thái song song, thì mặc định là số CPU. Lưu ý trạng thái song song bất kỳ được ấn định trong mệnh đề SELECT được bỏ qua.

Tóm tắt về những nguyên tắc song song hóa

Bảng 22-1 trình bày những kiểu câu lệnh SQL khác nhau như có thể được song song hóa như thế nào và trình bày các phương pháp ấn định trạng thái nào được ưu tiên.
Định chuẩn ưu tiên (1) chạy đè ưu tiên (2) và ưu tiên (3).
Định chuẩn ưu tiên (2) chạy đè ưu tiên (3).

Thông tin bổ sung:
Xem thêm các mệnh đề và các gợi ý song song trong các câu lệnh SQL ở Oracle8 SQL Reference.

Bảng 22-1 Các nguyên tắc song song hóa

Hoạt động song song	Được song song hóa bằng mệnh đề, gợi ý hoặc khai báo bảng/chỉ mục cơ sở (ưu tiên: 1, 2, 3)		
	Mệnh đề song song	**Gợi ý song song**	**Khai báo song song**
Duyệt bảng truy vấn song song (bảng được phân hoạch hoặc không được phân hoạch)		(1) PARALLEL	(2) của bảng
Duyệt phạm vi chỉ mục truy vấn song song (chỉ mục được phân hoạch)		(1) PARALLEL_INDEX	(2) của chỉ mục

UPDATE/DELETE song song (chỉ đối với bảng được phân hoạch)		(1) PARALLEL	(2) của bảng được cập nhật hoặc được xóa từ
Điều hoạt chèn INSERT ... SELECT song song (bảng được phân hoạch hoặc không được phân hoạch)		(1) PARALLEL của điều hoạt chèn	(2) thuộc bảng được chèn vào
Điều hoạt lựa INSERT... SELECT song song (bảng được phân hoạch hoặc không được phân hoạch)		(1) PARALLEL của điều hoạt lựa	(2) thuộc bảng lựa
Điều hoạt Create của CREATE TABLE ... AS SELECT song song (bảng được phân hoạch hoặc không được phân hoạch)	(1)	(Lưu ý: Gợi ý trong mệnh đề lựa không ảnh hưởng đến điều hoạt tạo.)	
Điều hoạt lựa của CREATE TABLE ... AS SELECT song song (bảng được phân hoạch hoặc không được phân hoạch)	(2)	(1) PARALLEL/PARALLEL_INDEX	(3) của các bảng truy vấn/các chỉ mục được phân hoạch
CREATE INDEX song song (chỉ mục được phân hoạch hoặc không được phân hoạch)	(1)		
REBUILD INDEX song song (chỉ mục không được phân hoạch)	(1)		
REBUILD INDEX (chỉ mục được phân hoạch)	Chưa bao giờ được song song hóa		

Phân hoạch REBUILD INDEX song song	(1)		
Phân hoạch MOVE/SPLIT song song	(1)		

DDL song song

Phần này gồm các chủ đề về trạng thái song song đối với các câu lệnh người dùng định nghĩa dữ liệu (DDL):
Các câu lệnh DDL có thể được song song hóa
CREATE TABLE . . . AS SELECT song song
Khả năng phục hồi và DDL song song
Quản lý không gian đối với DDL song song

Các câu lệnh DDL có thể được song song hóa

Bạn có thể song song hóa các câu lệnh DDL cho các bảng và những chỉ mục không được phân hoạch hoặc được phân hoạch. Bảng 22-1 tóm tắt các điều hoạt có thể được song song hóa trong các câu lệnh DDL.

Các câu lệnh DDL song song cho các bảng không được phân hoạch và các chỉ mục là:
CREATE INDEX
CREATE TABLE ... AS SELECT
ALTER INDEX ... REBUILD

Các câu lệnh DDL song song cho các bảng được phân hoạch và các chỉ mục là:
CREATE INDEX
CREATE TABLE ... AS SELECT
ALTER TABLE ... MOVE PARTITION
ALTER TABLE ... SPLIT PARTITION
ALTER INDEX ... REBUILD PARTITION

ALTER INDEX ... SPLIT PARTITION - chỉ nếu phân hoạch chỉ mục (toàn cục) được tách ra là Usable .

Tất cả những điều hoạt DDL có thể được thực hiện ở chế độ không ghi sổ nhật ký (xem "Chế độ ghi sổ nhật ký" đối với tiến hành song song và nối tiếp.

Trạng thái song song khác nhau được sử dụng cho các điều hoạt khác nhau (xem Bảng 22-1). Bảng tạo song song (được phân hoạch) được lựa và chỉ mục tạo song song (được phân hoạch) tiến hành với mức trạng thái song song bằng với số các phân hoạch.

Bảng phân tích song song phân hoạch được thực hiện ít cần đến các lệnh ANALYZE {TABLE, INDEX} PARTITION, vì phân tích song song của một bảng được phân hoạch toàn bộ có thể được cấu tạo với nhiều phiên người dùng.

Thông tin bổ sung:
Xem Oracle8 SQL Reference về cú pháp và sử dụng cửa sổ câu lệnh DDL.

CREATE TABLE ... AS SELECT song song

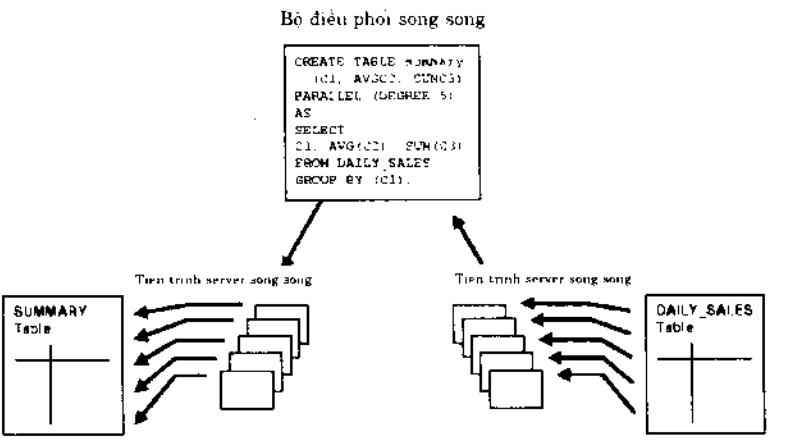
Quyết định hỗ trợ các ứng dụng, vì các lý do hiệu suất, thường đòi hỏi số lượng dữ liệu lớn được tóm tắt hoặc "cuộn lên" vào các bảng nhỏ hơn để sử dụng với các truy vấn hỗ trợ quyết định không định trước. Điều hoạt cuộn lên thường xảy ra (chẳng hạn như hàng đêm hoặc hàng tuần) khi một khoảng thời gian ngắn hệ thống không hoạt động.

Tiến hành song song cho phép bạn song song hóa truy vấn và các điều hoạt tạo và truy vấn để tạo bảng bằng truy vấn con từ một bảng khác hoặc tập hợp các bảng.

Hình 22-5 minh họa tạo một bảng từ một truy vấn con song song.

Lưu ý:
Các bảng liên kết không thể được tạo và cư trú song song.

Hình 22-5 Tạo một bảng tóm tắt song song



Tính năng phục hồi và DDL song song

Khi dữ liệu của bảng tóm tắt được phái sinh từ dữ liệu của các bảng khác, tính năng phục hồi từ sự cố môi trường đối với bảng tóm tắt nhỏ hơn có thể không quan trọng và có thể được đóng lại khi tạo bảng tóm tắt.

Nếu bạn vô hiệu hóa ghi sổ nhật ký khi tạo bảng song song (hoặc điều hoạt DDL bất kỳ khác), bạn nên sao chép không gian bảng có bảng ấy khi bảng được tạo để tránh mất bảng ấy do sự cố môi trường.

Sử dụng mệnh đề NOLOGGING của các câu lệnh CREATE/ALTER TABLE/INDEX để vô hiệu hóa phát sinh số nhật ký undo và redo log. Xem "Chế độ ghi số nhật ký".

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Administrator về tính năng phục hồi của các bảng được tạo song song.

Quản lý không gian cho DDL song song

Tạo một bảng hoặc chỉ mục song song có các ngầm định dữ liệu không gian ảnh hưởng đến cả không gian lưu trữ cần thiết khi điều hoạt song song và không gian trống có thể sử dụng sau khi bảng hoặc chỉ mục đã được tạo.

Không gian lưu trữ cho CREATE TABLE ... AS SELECT và CREATE INDEX

Khi tạo một bảng hoặc chỉ mục song song, mỗi tiến trình server song song sử dụng những giá trị trong mệnh đề STORAGE của câu lệnh CREATE để tạo các đoạn tạm thời để lưu trữ các hàng. Do đó, bảng được tạo với INITIAL 5M và PARALLEL DEGREE 12 tiêu thụ ít nhất 60 megabyte bộ nhớ khi tạo bảng, bởi vì mỗi tiến trình khởi động với khối lưu riêng bằng 5 megabyte. Khi bộ điều phối song song kết hợp các đoạn ấy, một đoạn có thể được cắt bỏ và bảng kết quả có thể nhỏ hơn 60 megabyte được yêu cầu.

Thông tin bổ sung:
Xem Oracle8 SQL Reference về cú pháp của lệnh CREATE TABLE.

Không gian trống và Parallel DDL

Khi bạn tạo các chỉ mục và các bảng song song, mỗi tiến trình server song song phân bổ một khối lưu riêng mới và điền khối

lưu riêng ấy với dữ liệu của bảng hoặc của chỉ mục. Do đó, nếu bạn tạo chỉ mục với mức trạng thái song song bằng 3, sẽ có ít nhất 3 khối lưu riêng cho chỉ mục đầu. (Điều này còn áp dụng với các chỉ mục tái thiết lập song song và các phân hoạch di chuyển, tách hoặc tái thiết lập song song.)

Các điều hoạt nối tiếp đòi hỏi đối tượng giản đồ có ít nhất một khối lưu riêng. Các điều hoạt tạo song song mà các bảng hoặc các chỉ mục có ít nhất bằng với nhiều khối lưu riêng có những tiến trình server song song tạo đối tượng giản đồ.

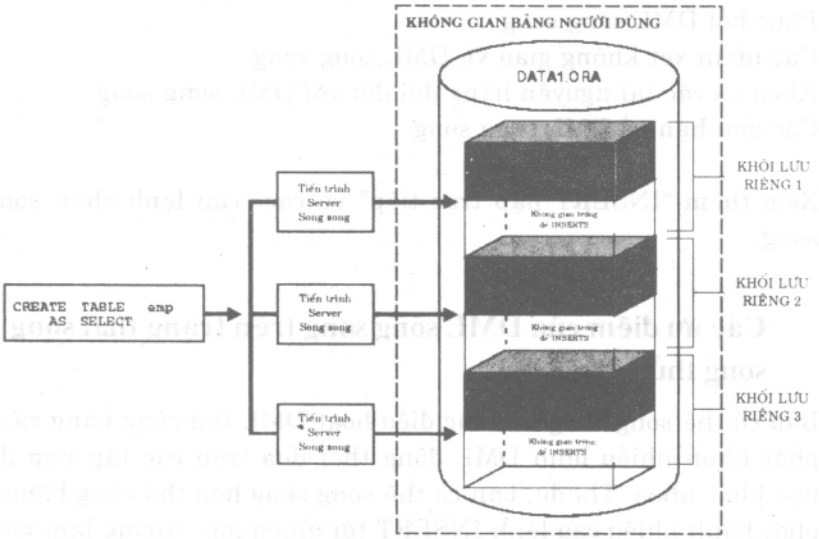
Khi bạn tạo một bảng hoặc chỉ mục song song, nó có thể tạo "gói tin" của không gian trống - phân mảnh bên ngoài hoặc bên trong. Điều này xảy ra khi các đoạn tạm thời được sử dụng bởi các tập tin server song song lớn hơn những gì cần thiết để lưu trữ các hàng. Nếu không gian chưa được sử dụng trong mỗi đoạn tạm thời lớn hơn giá trị của tham số MINIMUM EXTENT được xác lập ở mức không gian bảng, Oracle cắt bớt không gian chưa sử dụng khi hợp nhất những hàng từ tất cả những đoạn tạm thời vào bảng hoặc chỉ mục ấy. Không gian chưa sử dụng được cho trở lại không gian trống của hệ thống và có thể được phân bổ cho các khối lưu riêng mới, nhưng nó không thể hợp lại thành một đoạn lớn hơn vì nó không phải là không gian liên tục (phân mảnh bên ngoài).
Nếu không gian chưa sử dụng trong mỗi đoạn tạm thời nhỏ hơn giá trị của tham số MINIMUM EXTENT, không gian chưa sử dụng không thể được cắt bớt khi các hàng trong các đoạn tạm thời được hợp nhất vào bảng hoặc chỉ mục. Không gian chưa sử dụng này không cho trở về không gian trống của hệ thống ấy; nó trở thành thành phần của bảng hoặc chỉ mục (phân mảnh bên trong) và chỉ có thể sử dụng cho điều hoạt chèn tiếp theo hoặc các cập nhật đòi hỏi không gian bổ sung.

Thí dụ, nếu bạn ấn định mức trạng thái song song là 3 cho một câu lệnh CREATE TABLE ... AS SELECT nhưng chỉ có một tập

tin dữ liệu trong không gian bảng, phân mảnh bên trong được minh họa ở Hình 22-6 có thể phát sinh. "gói tin" của không gian trống trong các khối lưu riêng bảng bên trong của tập tin dữ liệu không thể được kết hợp với không gian trống khác và được phân bổ bằng các khối lưu riêng.

Xem thêm "Các không gian bảng và các tập tin dữ liệu".

Hình 22-6 Không gian trống chưa sử dụng (Phân mảnh bên trong)



Parallel DML

Parallel DML (chèn, cập nhật và xóa song song) sử dụng các cơ chế tiến hành song song để tăng tốc độ hoặc đáp ứng thích hợp các điều hoạt DML lớn dựa trên các bảng và chỉ mục CSDL lớn.

Lưu ý :
Dù ngôn ngữ điều tác dữ liệu (DML) chung có các truy vấn trong chương này với thuật ngữ "DML" chỉ tham chiếu các điều hoạt chèn, cập nhật và xóa.

Phần này đề cập đến những chủ đề DML song song sau:
Các ưu điểm của DML song song trên trạng thái song song thủ công
Khi sử dụng DML song song
Cho phép DML song song
Mô hình giao dịch cho DML song song
Phục hồi DML song song
Các nhận xét không gian về DML song song
Khóa và các tài nguyên hàng đợi đối với DML song song
Các giới hạn về DML song song

Xem thêm "INSERT nạp trực tiếp" về các câu lệnh chèn song song.

Các ưu điểm của DML song song trên trạng thái song song thủ công

Bạn có thể song song hóa các điều hoạt DML thủ công bằng cách phát hành nhiều lệnh DML đồng thời dựa trên các tập hợp dữ liệu khác nhau. Thí dụ, bạn có thể song song hóa thủ công bằng:
phát hành nhiều câu lệnh INSERT tới nhiều môi trường làm việc của Oracle Parallel Server để khai thác không gian trống từ nhiều khối danh sách trống
phát hành nhiều câu lệnh UPDATE và DELETE với các phạm vi giá trị khóa khác nhau hoặc các phạm vi ROWID.

Tuy nhiên, trạng thái thủ công có những bất lợi sau:
Khó sử dụng: bạn phải mở nhiều phiên (có thể trên các môi trường làm việc khác nhau) và phát hành nhiều câu lệnh.

Thiếu các tính chất giao dịch: các câu lệnh DML được phát hành vào những thời điểm khác nhau, do đó những thay đổi được thực hiện với các ảnh chụp nhanh của CSDL không nhất quán. Để được tự trị, cam kết hoặc hồi phục về trước những câu lệnh khác nhau phải được điều phối thủ công (có thể qua các môi trường làm việc).
Sự phức tạp phân chia truy vấn: bạn có thể phải truy vấn bảng tìm ROWID hoặc các phạm vi giá trị khóa để phân chia công việc chính xác.
Thiếu quan hệ cấu trúc và thông tin tài nguyên: bạn phải biết thông tin quan hệ cấu trúc để phát hành câu lệnh DML thích hợp vào đúng môi trường làm việc khi tiến hành Oracle Parallel Server. Bạn còn phải tìm cách sử dụng tài nguyên hiện hành để cân bằng tải tác vụ qua các môi trường làm việc.

Parallel DML xóa bỏ những bất lợi này bằng cách thực hiện các điều hoạt chèn, cập nhật và xóa song song tự động.

Khi sử dụng Parallel DML

Các điều hoạt DML song song chủ yếu được sử dụng để tăng tốc độ các điều hoạt DML dựa trên các đối tượng CSDL lớn. Parallel DML hữu dụng trong môi trường hệ thống hỗ trợ quyết định (DSS) nơi hiệu suất và tính vô hướng truy xuất các đối tượng lớn quan trọng. Parallel DML bổ sung truy vấn song song cung cấp cho bạn cả những khả năng cập nhật các cơ sở dữ liệu DSS.

Tổng xác lập các điều hoạt DML song song không thể đáp ứng cho các giao dịch OLTP ngắn. Tuy nhiên, các điều hoạt DML song song có thể tăng tốc độ các tác vụ lô trong cơ sở dữ liệu OLTP.

Các bảng tô bổ của hệ thống kho dữ liệu

Trong hệ thống kho dữ liệu (data warehouse) các bảng lớn cần được tô bổ (được cập nhật) định kỳ dữ liệu mới hoặc được sửa đổi từ hệ thống tạo sản phẩm. Bạn có thể thực hiện điều này hiệu quả bằng cách sử dụng DML song song được kết hợp với các khung nhìn liên kết cập nhật.

Dữ liệu cần được tô bổ thường được tải vào một bảng tạm thời trước khi bắt đầu tiến trình tô bổ. Bảng này chứa những hàng mới hoặc các hàng đã được cập nhật từ lần tô bổ cuối cùng kho dữ liệu. Bạn có thể sử dụng khung nhìn liên kết cập nhật với UPDATE song song để tô bổ các hàng cập nhật và có thể sử dụng liên kết trị số tương phản với INSERT song song để tô bổ các hàng mới.

Các bảng tóm tắt trung gian

Trong môi trường DSS, nhiều ứng dụng đòi hỏi những tính toán phức tạp phải cấu trúc và điều tác nhiều bảng tóm tắt trung gian lớn. Những bảng tóm tắt này thường tạm thời và thường không cần phải được ghi sổ nhật ký. DML song song có thể tăng tốc độ những điều hoạt dựa trên những bảng trung gian lớn này. Một bảng tóm tắt có ưu điểm là bạn đưa các kết quả tăng lên vào các bảng trung gian và thực hiện các cập nhật song song.

Hơn nữa, các bảng tóm tắt có thể có thông tin so sánh hoặc tích lũy phải duy trì ngoài các phiên ứng dụng; do đó, những bảng tạm thời không thể sử dụng. Các điều hoạt DML song song có thể tăng tốc độ những thay đổi đối với các bảng tóm tắt lớn.

Các bảng ghi nợ

Nhiều ứng dụng DSS ghi nợ những khách hàng định kỳ dựa trên một tập hợp tiêu chuẩn. Số tiền nợ thường xuyên được lưu trữ

trong các bảng DSS lớn. Thí dụ, thông tin tiền nợ sau đó được sử dụng để đưa ra quyết định trong một danh sách thư tín.

Hoạt động ghi nợ này truy vấn và cập nhật rất nhiều hàng trong một bảng lớn. DML song song có thể tăng tốc độ các điều hoạt dựa trên các bảng lớn này.

Các bảng lịch sử

Các bảng lịch sử mô tả các giao dịch kinh doanh của công ty trong khoảng thời gian gần nhất. Định kỳ, DBA xóa tập hợp những hàng cũ nhất và chèn tập hợp các hàng mới vào bảng. Các điều hoạt INSERT... SELECT và DELETE song song có thể tăng tốc độ tác vụ cuộn (rollover) này lên.

Dù bạn còn có thể sử dụng bộ nạp trực tiếp song song (SQL*Loader) để chèn dữ liệu hàng loạt từ nguồn bên ngoài, nhưng INSERT... SELECT song song sẽ chèn dữ liệu có sẵn ở một bảng khác trong CSDL nhanh hơn.

Bỏ một phân hoạch còn có thể được sử dụng để xóa các hàng cũ, nhưng để thực hiện điều này, bảng phải được phân hoạch theo ngày tháng và với khoảng thời gian thích hợp.

Các tác vụ lô

Các tác vụ lô được tiến hành trong cơ sở dữ liệu OLTP khi những giờ không hoạt động có cửa sổ thời gian cố định trong đó các tác vụ phải hoàn tất. Cách hữu hiện để bảo đảm hoàn tất tác vụ đúng lúc phải song song hóa các hoạt động của chúng. Giống như tải tác vụ gia tăng, nhiều tài nguyên máy có thể được bổ sung; tính chất đáp ứng thích hợp các điều hoạt song song bảo đảm hạn định thời gian có thể được đáp ứng.

Cho phép DML song song

Một câu lệnh DML có thể được so sánh hóa chỉ nếu bạn đã cho phép DML song song rõ ràng trong phiên bằng tùy chọn ENABLE PARALLEL DML của câu lệnh ALTER SESSION. Chế độ này cần thiết vì DML song song và DML nối tiếp có khóa, giao dịch và các đòi hỏi không gian đĩa khác nhau. (xem "Nhận xét về DML song song" và "Khóa và những tài nguyên hàng đợi cho DML song song".)

Chế độ mặc định của phiên là DISABLE PARALLEL DML. Khi PARALLEL DML được vô hiệu hóa, không DML nào sẽ được tiến hành dù gợi ý PARALLEL hoặc mệnh đề PARALLEL được sử dụng.

Khi PARALLEL DML được cho phép trong một phiên, tất cả những câu lệnh DML trong phiên này sẽ được xét để tiến hành song song. Tuy nhiên, dù PARALLEL DML được cho phép, điều hoạt DML vẫn có thể tiến hành nối tiếp nếu không có các gợi ý song song hoặc các mệnh đề song song hoặc nếu các giới hạn về các điều hoạt song song bị xâm phạm. (xem "Các giới hạn về DML song song").

Chế độ PARALLEL DML của phiên không ảnh hưởng đến trạng thái song song của các câu lệnh SELECT, DDL và các phần truy vấn của các câu lệnh DML. Do đó, nếu chế độ này không được xác lập, thì điều hoạt DML không được song song hóa nhưng tác vụ duyệt và các điều hoạt liên kết trong câu lệnh DML có thể vẫn được song song hóa.

Các giao dịch với PARALLEL DML được cho phép

Một phiên được cho phép đối với PARALLEL DML có thể đưa các giao dịch vào phiên ở chế độ đặc biệt: Nếu câu lệnh DML bất kỳ trong một giao dịch sửa đổi một bảng song song, thì không có truy vấn song song hoặc nối tiếp theo sau hoặc câu lệnh DML có

thể truy xuất lại cùng bảng trong giao dịch đó. Nghĩa là các kết quả của những sửa đổi song song không thể được thấy trong giao dịch ấy.

Các câu lệnh nối tiếp hoặc song song cố truy xuất một bảng đã bị sửa đổi trong cùng giao dịch bị từ chối bằng một thông báo lỗi.

Nếu thủ tục PL/SQL hoặc khối dữ liệu được tiến hành trong phiên PARALLEL DML được cho phép, thì nguyên tắc này áp dụng vào thủ tục hoặc khối ấy.

Mô hình giao dịch đối với Parallel DML

Để tiến hành một điều hoạt DML song song, tiến trình của bộ điều phối đạt được hoặc phát sinh những tiến trình server song song và mỗi tiến trình server song song tiến hành một phần của công việc dưới giao dịch tiến trình song song riêng của nó.
Mỗi server song song tạo một giao dịch tiến trình song song khác nhau.
Để giảm tranh chấp trên các đoạn hồi phục về trước, chỉ một ít giao dịch tiến trình song song sẽ cư trú trong cùng đoạn hồi phục về trước (xem phần kế tiếp).

Bộ điều phối còn có giao dịch bộ điều phối riêng của nó, có thể có đoạn hồi phục về trước của riêng nó.

Các đoạn hồi phục về trước

Oracle gán những giao diện vào các đoạn hồi phục về trước có ít giao dịch hoạt động nhất. Để tăng tốc độ cả những điều hoạt hoán chuyển và gửi đi, bạn phải tạo và đưa ra đủ các đoạn hồi phục về trước sao cho hai giao dịch tiến trình song song được gán vào một đoạn hồi phục về trước.

Tạo các đoạn hồi phục về trước trong các không gian bảng có đủ không gian cho chúng để mở rộng khi cần và xác lập những tham số lưu trữ MAXEXTENTS cho những đoạn hồi phục về trước thành UNLIMITED.

Cam kết hai pha

Một điều hoạt DML song song được tiến hành bằng hơn một giao dịch tiến trình song song phụ thuộc. Để bảo đảm tính tự trị giao dịch mức người dùng, bộ điều phối sử dụng giao thức cam kết hai pha để cam kết những thay đổi được thực hiện bởi các giao dịch tiến trình song song.

Giao thức cam kết hai pha là một phiên bản đơn giản hóa khai thác kiến trúc đĩa dùng chung để tăng tốc độ khảo sát trạng thái giao dịch, nhất là khi phục hồi giao dịch. Nó không đòi hỏi thư viện Oracle XA. Các giao dịch không chắc chắn, người dùng không bao giờ thấy được.

Phục hồi đối với Parallel DML

Thời gian cần thiết để hồi phục một điều hoạt DML song song về trước gần bằng với thời gian thực hiện điều hoạt gửi đi.

Oracle hỗ trợ phục hồi giao dịch song song (phục hồi "hoán chuyển") khi giao dịch và các sai sót tiến trình thành một khối lưu riêng nhỏ nhất khi có những sự cố môi trường làm việc và hệ thống.

Để tăng tốc độ phục hồi giao dịch, tham số khởi hoạt CLEANUP ROLLBACK_ENTRIES sẽ được xác lập thành giá trị cao gần bằng số mục nhập hồi phục về trước được phát sinh cho điều hoạt gửi đi.

Phục hồi giao dịch

Hồi phục về trước được người dùng phát hành trong một sai sót giao dịch do lỗi câu lệnh được thực hiện song song bởi bộ điều phối song song và những tiến trình server song song. Hồi phục về trước gần bằng số thời gian giao dịch gửi đi.

Phục hồi tiến trình

Phục hồi từ sai sót của một bộ điều phối DML song song hoặc tiến trình server song song được thực hiện bởi tiến trình PMON.
Nếu một tiến trình server song song đơn không thực hiện được, PMON hồi phục về trước công việc của tiến trình đó và tất cả những tiến trình server song song khác hồi phục về trước công việc của riêng chúng.
Nếu nhiều tiến trình server song song không thực hiện được, PMON hồi phục về trước tất cả công việc của chúng nối tiếp.
Nếu tiến trình bộ điều phối không thực hiện được, PMON phục hồi bộ điều phối ấy và tất cả những tiến trình server song song hồi phục về trước công việc của riêng chúng song song.

Do đó, thời gian phục hồi đối với các sai sót tiến trình có thể lâu hơn công việc ban đầu (gửi đi).

Phục hồi hệ thống

Phục hồi từ sự cố hệ thống cần khởi động lại. Phục hồi được thực hiện bằng tiến trình SMON. Các câu lệnh DML song song phục hồi nối tiếp và tất cả những tài nguyên vẫn bị khóa cho tới khi phục hồi được hoàn tất. Do đó, phục hồi lâu hơn nhiều so với giao dịch ban đầu (gửi đi) nếu giao dịch gửi đi đã sử dụng mức trạng thái song song cao và đã thực hiện nhiều công việc.

Một cách để tăng tốc độ phục hồi giao dịch là tiến hành lại câu lệnh DML song song. Khi bộ điều phối mới và những tiến trình server song song gặp những tài nguyên được khóa, chúng kích

hoạt phục hồi giao dịch đồng thời. Sau những tiến trình song song mới kết thúc phục hồi những tài nguyên ấy, bạn có thể cam kết hoặc hồi phục về trước giao dịch ấy.

Phục hồi môi trường làm việc (Oracle Parallel Server)

Phục hồi từ một sai sót môi trường làm việc trong Oracle Parallel Server được thực hiện bởi những tiến trình SMON của các môi trường làm việc sinh động khác. Mỗi tiến trình SMON của các môi trường làm việc sinh động có thể phục hồi bộ điều phối song song và/hoặc những giao dịch tiến trình server song song của môi trường làm việc bị sai sót độc lập. Nếu có nhiều tiến trình server song song các môi trường làm việc sinh động, thì thời gian phục hồi lâu hơn công việc tiến hành của môi trường làm việc bị sai sót.

Các nhận xét về không gian đối với Parallel DML

UPDATE song song sử dụng không gian trong đối tượng có sẵn, trái lại với INSERT nạp trực tiếp nhận những đoạn mới cho các đoạn dữ liệu ấy.

Những đặc tính sử dụng không gian song song có thể khác nhau so với chúng nếu câu lệnh ấy được tiến hành nối tiếp, vì nhiều giao dịch con đồng thời sửa đổi đối tượng ấy.

Xem "Các nhận xét không gian" về INSERT nạp trực tiếp.

Khóa và các tài nguyên hàng đợi đối với Parallel DML

Một khóa của điều hoạt DML song song và các đòi hỏi tài nguyên hàng đợi rất khác n au với các đòi hỏi DML nối tiếp. DML song song lưu giữ nhiều khóa hơn, vì vậy bạn tăng giá trị của tham số ENQUEUE_RESOURCES và DML_LOCKS.

Những tiến trình đối với câu lệnh UPDATE, DELETE hoặc INSERT đạt được những khóa sau:
Bộ đồng thời đạt được:
1 khóa bảng SX
1 khóa phân hoạch X trên mỗi phân hoạch

> Đối với INSERT song song vào bảng phân hoạch, bộ điều phối đạt được các khóa phân hoạch cho tất cả các phân hoạch. Đối với UPDATE song song hoặc DELETE, bộ điều phối đạt được các khóa phân hoạch cho tất cả những phân hoạch, trừ khi mệnh đề WHERE giới hạn các phân hoạch có liên quan ấy.

Mỗi tiến trình server song song đạt được:
1 khóa bảng SX
1 khóa phân hoạch NULL trên mỗi phân hoạch
1 khóa đợi phân hoạch X trên mỗi phân hoạch

tiến trình server song song có thể hoạt động trên một hoặc nhiều phân hoạch, nhưng một phát hiện chỉ có thể được hoạt động bởi một tiến trình server song song.

Thí dụ, đối với một bảng có 600 phân hoạch tiến hành với mức song song 100, một câu lệnh DML song song cần các khóa sau (giả sử tất cả phân hoạch có liên quan trong câu lệnh):
Bộ điều phối đạt được 1 khóa bảng SX và 600 khóa phân hoạch X.
Tổng số tiến trình server song song đạt được 100 khóa bảng SX, 600 khóa phân hoạch NULL và 600 khóa đợi phân hoạch X.

Bảng 22-2 tóm tắt những kiểu khóa bộ điều phối và các tiến trình server song song đạt được đối với những kiểu câu lệnh DML song song khác nhau.

Bảng 22-2 Các khóa đạt được bằng các câu lệnh DML song song

Kiểu câu lệnh	Tiến trình bộ điều phối đạt được:	Mỗi tiến trình server song song đạt được:
UPDATE song song hoặc DELETE vào bảng được phân hoạch; mệnh đề WHERE ấn định phân hoạch ấy	1 khóa bảng SX 1 khóa phân hoạch X trên mỗi phân hoạch	1 khóa bảng SX 1 khóa phân hoạch NULL trên mỗi phân hoạch 1 khóa đợi phân hoạch X trên mỗi phân hoạch
UPDATE song song, DELETE hoặc INSERT vào bảng được phân hoạch	1 khóa bảng SX Các khóa phân hoạch X cho tất cả các phân hoạch	1 khóa bảng SX 1 khóa phân hoạch NULL trên mỗi phân hoạch 1 khóa đợi phân hoạch X trên mỗi phân hoạch
INSERT song song vào bảng không được phân hoạch	1 khóa bảng X	Không

Các giới hạn về Parallel DML

Những giới hạn sau áp dụng vào DML song song (kể cả INSERT nạp trực tiếp):

Các điều hoạt cập nhật và xóa không được song song hóa trên các bảng không được phân hoạch.

Đối với các điều hoạt cập nhật song song, các chỉ mục duy nhất toàn cục không được hỗ trợ. Tất cả những chỉ mục khác được duy trì hoàn toàn bằng các điều hoạt song song.

Một giao dịch có thể có nhiều câu lệnh DML song song sửa đổi các bảng khác nhau, nhưng sau câu lệnh DML sửa đổi một bảng, không có câu lệnh nối tiếp hoặc song song (DML hoặc truy vấn) sau đó có thể truy xuất lại cùng bảng trong giao dịch đó.
Giới hạn này còn tồn tại sau một câu lệnh INSERT nạp trực tiếp: không có câu lệnh SQL (DML hoặc truy vấn) tiếp theo có thể truy xuất bảng được sửa đổi trong giao dịch đó.
Các truy vấn truy xuất cùng bảng được cho phép trước câu lệnh DML song song hoặc INSERT nạp trực tiếp mà không phải sau.
Bất kỳ các câu lệnh nối tiếp hoặc song song cố truy xuất một bảng đã được sửa đổi bằng một UPDATE song song, DELETE song song hoặc INSERT nạp trực tiếp trong cùng giao dịch bị từ chối bằng một thông báo lỗi.
Nếu tham số khởi hoạt ROW LOCKING = INTENT, thì các điều hoạt chèn, cập nhật và xóa không được song song hóa (bất kể chế độ nối tiếp).
Các bộ kích khởi không được hỗ trợ cho các điều hoạt DML song song.
Tính năng sao chép không được hỗ trợ cho DML song song.
DML song song không thể xảy ra khi có hạn định nào đó: tính nguyên vẹn tự tham chiếu, xếp chuỗi xóa và tính nguyên vẹn trì hoãn. Ngoài ra, đối với INSERT nạp trực tiếp không có hỗ trợ cho tính nguyên vẹn tham chiếu bất kỳ.
DML song song không thể xảy ra trên các bảng có những cột đối tượng hoặc các cột LOB hoặc trên các bảng có tổ chức chỉ mục.
Một giao dịch có liên quan trong điều hoạt DML song song không thể là giao dịch được phân phối hoặc trở thành giao dịch được phân phối.
Các bảng liên kết không được hỗ trợ.

Các xâm phạm sẽ làm cho câu lệnh tiến hành nối tiếp mà không có các cảnh báo hoặc các thông báo lỗi (ngoại trừ giới hạn về các câu lệnh truy xuất cùng bảng trong giao dịch, có thể gây ra thông báo lỗi). Thí dụ, chỉ mục sẽ được nối tiếp hóa nếu nó đòi hỏi duy trì chỉ mục duy nhất toàn cục).

Các phần sau đề cập thêm về những giới hạn.

Giới hạn khóa phân hoạch

Bạn chỉ có thể cập nhật khóa phân hoạch của bảng được phân hoạch thành giá trị mới nếu cập nhật ấy không làm cho hàng đó di chuyển tới một phân hoạch mới. Đây là một giới hạn chung về các bảng được phân hoạch.

Các giới hạn tính nguyên vẹn dữ liệu

Phần này mô tả những tương tác về các hạn định tính nguyên vẹn và các câu lệnh DML song song.

NOT NULL và CHECK

Những kiểu hạn định tính nguyên vẹn được cho phép. Chúng không gây ra vấn đề đối với DML song song vì chúng được tăng cường trên mức cột và hàng tương ứng.

UNIQUE và PRIMARY KEY

Cả hai hạn định này được tăng cường với các chỉ mục duy nhất. Một lệnh UPDATE sửa đổi chỉ mục khóa chính hoặc duy nhất được song song hóa chỉ nếu chỉ mục ấy cục bộ.

FOREIGN KEY (tính nguyên vẹn tham chiếu)

Có những giới hạn đối với tính nguyên vẹn tham chiếu bất kỳ khi nào một điều hoạt DML trên một bảng có thể gây cho một điều hoạt DML tái quy trên một bảng khác hoặc để thực hiện kiểm tính nguyên vẹn dữ liệu, nó sẽ cần thấy tất cả những thay đổi đồng thời xảy ra đối với đối tượng được sửa đổi.

Bảng 22-3 liệt kê tất cả những điều hoạt có thể xảy ra trên các bảng liên quan đến các hạn định tính nguyên vẹn tham chiếu

Bảng 22-3 Các giới hạn tính nguyên vẹn tham chiếu

Câu lệnh DML	Được phát hành trên bảng cha	Được phát hành trên bảng con	Tự tham chiếu
INSERT	(Không áp dụng)	Không được song song hóa	Không được song song hóa
UPDATE No Action	Được hỗ trợ	Được hỗ trợ	Không được song song hóa
DELETE No Action	Được hỗ trợ	Được hỗ trợ	Không được song song hóa
DELETE Cascade	Không được song song hóa	(Không được áp dụng)	Không được song song hóa

Xếp chuỗi xóa

Xóa trên các bảng có khóa ngoại bằng xếp chuỗi xóa không được song song hóa vì những tiến trình server song song sẽ cố xóa các hàng khỏi nhiều phân hoạch (các bảng cha và con).

Tính tự tham chiếu

DML trên các bảng có các hạn định tính tự tham chiếu không được song song hóa nếu cửa sổ khóa được tham chiếu (các khóa chính) có liên quan. Đối với DML trên tất cả những cột khác, trạng thái song song có thể xảy ra.

Các hạn định tính nguyên vẹn trì hoãn được

Nếu có bất kỳ những hạn định có thể trì hoãn trên bảng đang được điều hoạt, điều hoạt DML sẽ không được song song hóa.

Các giới hạn bộ kích khởi

Điều hoạt DML sẽ không được song song hóa nếu bất kỳ bộ kích khởi được cho phép trên các bảng bị ảnh hưởng có thể nhận được

kích hoạt do câu lệnh ấy. Điều này ám chỉ là các câu lệnh DML trên các bảng đang được sao chép sẽ không được song song hóa.

Các bộ kích khởi thích hợp phải được vô hiệu hóa để song song hóa DML trên bảng đó. Lưu ý các bộ kích khởi cho phép/không cho phép vô hiệu hóa các con trỏ dung chung phụ thuộc.

Các giới hạn chức năng

Chỉ những chức năng không đọc hoặc ghi trạng thái CSDL hoặc trạng thái bộ phần mềm được cho phép trong các câu lệnh DML song song. Điều hoạt DML sẽ không được song song hóa nếu câu lệnh DML đã nhúng các chức năng đọc hoặc ghi trạng thái CSDL hoặc trạng thái bộ phần mềm.

Thông tin bổ sung:
Xem mô tả về RESTRICT_REFERENCES trong hướng dẫn Oracle8 Application Developer.

Các giới hạn giao dịch được phân phối

Điều hoạt DML không thể được song song hóa nếu nó ở trong một giao dịch được phân phối hoặc nếu DML hay điều hoạt truy vấn dựa trên đối tượng từ xa.

Thí dụ 1:

```
Trong giao dịch được phân phối:
select * from t1@dblink; /* this starts a distributed transaction */
delete /*+ parallel (t2,2) */ from t2; /* not parallelized */
commit;
```

Thí dụ 2:

```
Điều hoạt DML trên một đối tượng từ xa:
delete /*+ parallel *t1, 2) */ from t1@dblink;
/* cannot parallel delete from remote object */
```

Thí dụ 3:

Câu lệnh DML truy vấn một đối tượng từ xa:

```
insert /* append parallel (t3,2) */ into t3 select * from t4@dblink;
/* not parallelized because of reference to remote object */
```

Quan hệ cấu trúc

Trong cụm liên kết đĩa dùng chung hoặc cấu hình xử lý song song hàng loạt (MPP), một môi trường làm việc của Oracle Parallel Server được coi có quan hệ cấu trúc đối với một thiết bị nếu thiết bị ấy được truy xuất trực tiếp từ (các) bộ xử lý mà môi trường làm việc ấy đang tiến hành. Tương tự, môi trường làm việc có quan hệ cấu trúc đối với một tập tin nếu nó có quan hệ cấu trúc đối với (các) thiết bị tập tin ấy lưu trữ.

Lưu ý:
Các đặc trưng được mô tả về phần này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition với Parallel Server Option.

Sự xác định của quan hệ cấu trúc có thể liên quan đến những xác định tùy ý cho các tập tin được lập dải qua nhiều thiết bị. Tùy ý ở mức độ nào đó, môi trường làm việc được coi có quan hệ cấu trúc đối với một không gian bảng (hoặc một phân hoạch của một bảng hoặc chỉ mục trong không gian bảng) nếu môi trường làm việc ấy có quan hệ cấu trúc đối với tập tin thứ nhất trong không gian bảng.

Oracle xét quan hệ cấu trúc khi phân bố công việc đối với các tiến trình server song song. Sử dụng quan hệ cấu trúc đối với sự tiến hành song song của các câu lệnh SQL thông suốt đến người dùng.

Quan hệ cấu trúc và các truy vấn song song

Quan hệ cấu trúc trong những truy vấn song song tăng tốc độ duyệt dữ liệu từ đĩa bằng cách thực hiện các tác vụ duyệt trên một bộ xử lý "gần" dữ liệu ấy. Điều này có thể tăng hiệu suất đáng kể cho các máy không hỗ trợ các đĩa dùng chung vốn có.

Sử dụng thông thường nhất của quan hệ cấu trúc đối với một bảng hoặc phân hoạch chỉ mục để được lưu trữ trong một tập tin trên một thiết bị. Cấu hình này cung cấp tính khả dụng cao nhất bằng cách giới hạn sự thiệt hại do sự cố thiết bị gây ra và khai thác tốt nhất những tác vụ duyệt chỉ mục phân hoạch-song song.

Các khách hàng DSS có thể thích các bảng được lập dải trên nhiều thiết bị (có thể một tập hợp con của tổng số thiết bị). Điều này cho phép một số truy vấn cắt bớt tổng số dữ liệu được truy xuất sử dụng tiêu chuẩn phân hoạch và vẫn có được trạng thái song song qua các tác vụ duyệt bảng song song ROWID-range (phân hoạch). Nếu các thiết bị có cấu hình là RAID, tính khả dụng có thể vẫn rất tốt. Ngay cả khi được sử dụng cho DSS, các chỉ mục có thể sẽ được phân hoạch trên các dịch vụ cá biệt.

Các cấu hình khác (thí dụ, nhiều phân hoạch trong một tập tin được lập dải trên nhiều thiết bị) sẽ mang lại các kết quả truy vấn chính xác, nhưng bạn có thể cần sử dụng các gợi ý hoặc xác lập các thuộc tính đối tượng rõ ràng để lựa mức trạng thái song song chính xác.

Quan hệ cấu trúc và Parallel DML

Đối với DML song song (chèn, cập nhật và xóa), các tăng cường quan hệ cấu trúc tăng hiệu suất cache bằng cách định tuyến điều hoạt DML tới nút có quan hệ cấu trúc đối với phân hoạch ấy.

Quan hệ cấu trúc xác định cách phân phối công việc trong tập hợp các môi trường làm việc và/hoặc các tiến trình server song

song để thực hiện điều hoạt DML song song. Quan hệ cấu trúc có thể tăng hiệu suất của các truy vấn bằng nhiều cách:

Đối với những kiến trúc MPP nào đó, Oracle sử dụng thông tin quan hệ cấu trúc thiết bị-tới-nút để xác định những nút nào phát sinh những tiến trình server song song (*phân bổ tiến trình song song*) và tác vụ kết hạt (các phạm vi ROWID hoặc các phân hoạch) để gửi tới các nút cá biệt (lệnh gán công việc). Đạt được hiệu suất tốt hơn bằng cách có những nút chủ yếu truy xuất các thiết bị cục bộ tạo cho cache vùng đệm tốt hơn đạt được tỷ lệ đối với mọi nút và giảm phần điều hành chung mạng và thời gian chờ I/O.

Đối với những cụm liên kết đĩa dùng chung SMP, Oracle sử dụng cơ chế xoay tua để gán những thiết bị vào các nút. Tương tự với mục 1, quan hệ cấu trúc thiết bị-tới-nút này được sử dụng để xác định phân bổ tiến trình song song và lệnh gán công việc.

Đối với SMP, cụm liên kết và các kiến trúc MPP, quan hệ cấu trúc tiến trình-tới-thiết bị được sử dụng để đạt được sự tách lập thiết bị. Điều này làm giảm các cơ hội có nhiều tiến trình server song song truy xuất cùng thiết bị đồng thời. Thông tin quan hệ cấu trúc tiến trình-tới-thiết bị này còn được sử dụng để bổ sung hoạt động giành được giữa các tiến trình.

Đối với các bảng được phân hoạch và chỉ mục, thông tin quan hệ cấu trúc phân hoạch-tới-nút xác định phân bổ tiến trình và gán công việc. Đối với những hệ thống MPP không dùng chung, Oracle Parallel Server cố gán những phân hoạch vào các môi trường làm việc lưu ý tới các quan hệ cấu trúc đĩa của các phân hoạch. Đối với MPP đĩa dùng chung và các hệ thống cụm liên kết, các phân hoạch được gán vào các môi trường làm việc theo phương thức xoay tua.

Quan hệ cấu trúc chỉ có thể sử dụng đối với DML khi tiến hành cấu hình Oracle Parallel Server. Thông tin quan hệ cấu trúc duy trì qua các câu lệnh sẽ tăng cache vùng đệm đạt tỷ lệ và giảm các chuyển giao khối dữ liệu giữa các môi trường làm việc.

Các kiểu trạng thái song song khác

Hơn nữa tiến hành SQL song song, Oracle có thể sử dụng trạng thái song song cho các kiểu điều hoạt sau:

Phục hồi song song
Truyền song song (sao chép)
Tải song song (tiện ích SQL*Loader)

Giống như SQL song song, phục hồi song song và truyền song song được tiến hành bằng một tiến trình bộ điều phối song song và nhiều tiến trình server song song. Tuy nhiên, tải song song sử dụng một cơ chế khác.

Thông tin bổ sung:
xem các tiện ích Oracle8 về tải song song và thông tin tổng quát về SQL*Loader. Xem thêm Oracle8 Tuning về tải song song.

Xử lý của bộ điều phối song song và các tiến trình server song song có thể khác nhau tùy theo loại điều hoạt chúng thực hiện (SQL, phục hồi hoặc truyền). Thí dụ, nếu tất cả các tiến trình server song song trong vùng chia được chiếm giữ và số những tiến trình server song song tối đa đã được khởi động:

- ở vai trò SQL song song, bộ điều phối song song chuyển sang xử lý nối tiếp
- ở vai trò truyền song song, bộ điều phối song song cho trở lại một lỗi.

Đối với một phiên cho trước, bộ điều phối chỉ phối hợp một loại điều hoạt. Thí dụ, một bộ điều phối song song không thể phối

hợp SQL song song và truyền song song hoặc phục hồi song song cùng một lúc.

Xem thêm "Thực hiện phục hồi song song".

Phần VII

BẢO VỆ DỮ LIỆU

Chương 23

Tính đồng thời và nhất quán dữ liệu

Chương này giải thích Oracle duy trì dữ liệu nhất quán trong môi trường CSDL nhiều người dùng như thế nào, gồm các chủ đề:

- **Tính đồng thời và nhất quán dữ liệu trong môi trường đa người dùng**
- **Oracle quan hệ tính đồng thời và nhất quán dữ liệu như thế nào**
- **Oracle khóa dữ liệu như thế nào**

Tính đồng thời và nhất quán dữ liệu trong môi trường đa người dùng

Trong cơ sở dữ liệu một người dùng, người dùng ấy có thể sửa đổi dữ liệu trong CSDL mà không cần quan tâm về những người

dùng khác sửa đổi cùng dữ liệu đồng thời. Tuy nhiên, một CSDL đa người dùng, các câu lệnh trong nhiều giao dịch đồng thời có thể cập nhật cùng dữ liệu đó. Các giao dịch tiến hành cùng một lúc phải tạo các kết quả có ý nghĩa và nhất quán. Do đó, kiểm soát tính đồng thời và nhất quán dữ liệu quan trọng trong cơ sở dữ liệu đa người dùng.
Tính đồng thời dữ liệu có nghĩa là nhiều người dùng có thể truy xuất dữ liệu cùng một lúc.
Tính nhất quán dữ liệu có nghĩa là mỗi người dùng thấy khung nhìn của dữ liệu nhất quán kể cả thấy được những thay đổi xảy ra do các giao dịch riêng của người dùng và những giao dịch của những người dùng khác.

Tính nguyên vẹn dữ liệu tăng cường những nguyên tắc hoạt động được kết hợp với CSDL được đề cập ở chương 24 "Tính nguyên vẹn dữ liệu".

Để mô tả hành vi giao dịch nhất quán khi những giao dịch tiến hành cùng một lúc, những nhà nghiên cứu CSDL đã xác định một môi trường tách lập giao dịch được gọi là tính năng nối tiếp. Mô hình nối tiếp của hành vi giao dịch cố bảo đảm các giao dịch tiến hành một cách sao cho chúng dường như được tiến hành mỗi lúc một lần hoặc nối tiếp thay vì đồng thời.

Trong khi mức tách lập giữa các giao dịch này thường được yêu cầu, tiến hành nhiều ứng dụng trong chế độ này có thể thỏa hiệp lưu lượng truyền ứng dụng nối tiếp. Tách lập hoàn toàn của những giao dịch tiến hành đồng thời có nghĩa là một giao dịch không thể thực hiện chèn vào một bảng đang được truy vấn bằng một giao dịch khác. Nói tóm lại, các nhận xét về thế giới thực thường đòi hỏi sự thỏa hiệp giữa sự tách lập giao dịch hoàn hảo và hiệu suất.

Oracle cung ứng hai mức tách lập để cung cấp cho các nhà phát triển ứng dụng các chế độ tùy chọn duy trì tính nhất quán và cung cấp hiệu suất cao.

Hiện tượng có thể ngăn chặn và các mức tách lập giao dịch

Tiêu chuẩn ANSI/ISO SQL (SQL92) xác định bốn mức tách lập giao dịch với các mức độ khác nhau tác động đến thống suất xử lý giao dịch. Những mức tách lập này được sử dụng với điều kiện ba hiện tượng phải được ngăn chặn giữa các giao dịch tiến hành đồng thời.

Ba hiện tượng có thể ngăn chặn là:

Đọc những thay đổi dữ liệu	Một giao dịch đọc dữ liệu đã được ghi bằng một giao dịch khác chưa được cam kết.
Đọc không lặp lại được (mờ)	Một giao dịch đọc lại dữ liệu nó đã đọc trước đây và tìm một giao dịch được cam kết đã sửa đổi hoặc xóa dữ liệu.
Đọc bóng ma	Một giao dịch tiến hành lại truy vấn để cho trở lại một tập hợp các hàng thỏa đáng điều kiện tìm kiếm và tìm một giao dịch được cam kết khác đã chèn các hàng có điều kiện thỏa đáng điều kiện ấy.

SQL92 xác định bốn mức tach lập với điều kiện hiện tượng một giao dịch tiến hành ở mức tách lập cá biệt được cho phép trải qua.

Mức tách lập	Đọc dữ liệu bị thay đổi	Đọc không lặp lại được	Đọc bóng ma
Đọc không được cam kết	Có thể xảy ra	Có thể xảy ra	Có thể xảy ra

Đọc được cam kết	Có thể xảy ra	Có thể xảy ra	Có thể xảy ra
Đọc lặp lại được	Không thể xảy ra	Không thể xảy ra	Có thể xảy ra
Nối tiếp được	Không thể xảy ra	Không thể xảy ra	Không thể xảy ra

Oracle cung ứng đọc được cam kết và các mức tách lập nối tiếp được, cùng như chế độ chỉ đọc không phải là thành phần của SQL92. Đọc được cam kết là mặc định và là mức tách lập tự động được cung cấp chỉ có trước Oracle Release 7.3. Đọc được cam kết và các mức tách lập nối tiếp được để cập chi tiết ở "Oracle quản lý tính đồng thời và nhất quán dữ liệu như thế nào".

Các cơ chế khóa

Nói chung, các CSDL đa người dùng sử dụng một số dạng khóa dữ liệu để giải quyết những vấn đề được kết hợp với tính đồng thời, nhất quán và tính nguyên vẹn dữ liệu. Các khóa là các cơ chế ngăn chặn sự tương tác hủy giữa các giao dịch truy xuất cùng tài nguyên.

Những tài nguyên có hai kiểu đối tượng thông thường:

- các đối tượng người dùng, chẳng hạn như các bảng và các hàng (các cấu trúc và dữ liệu)
- các đối tượng hệ thống người dùng không thấy, chẳng hạn như các cấu trúc dữ liệu dùng chung trong bộ nhớ và các hàng từ điển dữ liệu

Các kiểu khóa khác nhau - các khóa dữ liệu, các khóa DDL và các khóa bên trong được để cập ở "Oracle khóa dữ liệu như thế nào".

Oracle quản lý tính đồng thời và nhất quán dữ liệu như thế nào

Oracle duy trì tính đồng thời và nhất quán dữ liệu trong môi trường đa người dùng bằng cách sử dụng mô hình nhất quán đa phiên bản và các kiểu khóa và giao dịch khác nhau.

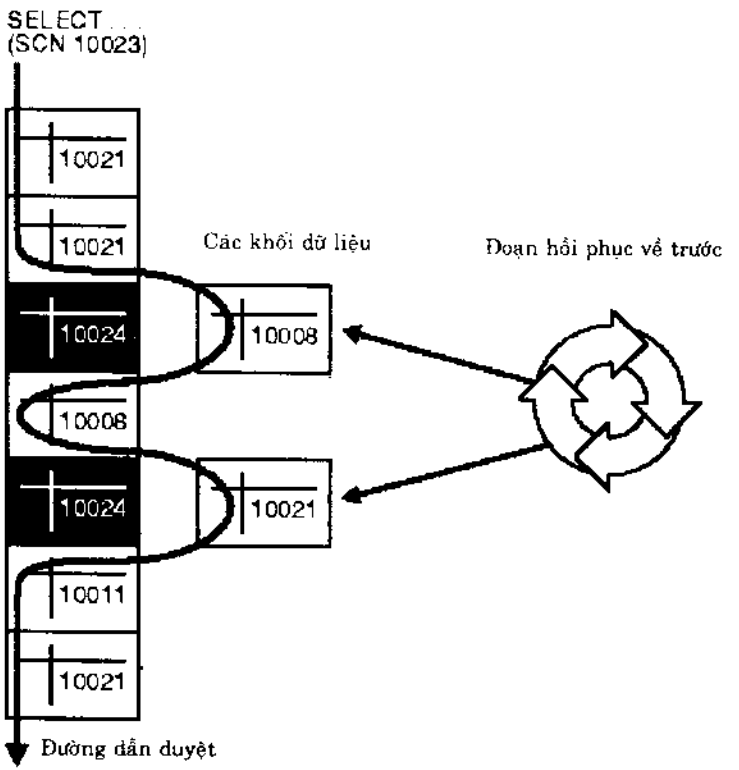
Kiểm soát tính đồng thời đa phiên bản

Oracle tự động cung cấp tính đọc nhất quán một truy vấn sao cho tất cả dữ liệu truy vấn thấy chúng phát xuất từ một thời điểm đơn (tính đọc nhất quán mức câu lệnh). Oracle còn có thể cung cấp tính đọc nhất quán đối với tất cả những truy vấn trong một giao dịch (tính đọc nhất quán mức giao dịch).

Oracle sử dụng thông tin được duy trì trong các đoạn hồi phục về trước của nó để cung cấp những khung nhìn nhất quán này. Các đoạn hồi phục về trước có những giá trị dữ liệu cũ đã bị thay đổi do các giao dịch được cam kết hoặc mới được cam kết. Hình 23-1 trình bày Oracle cung cấp tính đọc nhất quán mức câu lệnh sử dụng dữ liệu trong các đoạn hồi phục về trước như thế nào.

Khi một truy vấn nhập giai đoạn tiến hành, hệ thống hiện hành thay đổi số (SCN) được xác định; ở Hình 23-1, hệ thống này thay đổi số là 10023. Khi các khối dữ liệu được đọc thay cho truy vấn, chỉ các khối được ghi với SCN được quan sát được sử dụng. Các khối có dữ liệu thay đổi (các SCN gần đây hơn) được cấu trúc lại từ dữ liệu trong các đoạn hồi phục về trước và dữ liệu có cấu trúc lại được cho trở lại truy vấn ấy. Do đó, mỗi truy vấn cho trở lại tất cả dữ liệu cam kết liên quan đến SCN được ghi vào lúc tiến hành truy vấn đó bắt đầu. Những thay đổi của các giao dịch khác xảy ra khi tiến hành của một truy vấn không được quan sát, bảo đảm là dữ liệu nhất quán được cho trở lại đối với mỗi truy vấn.

Hình 23-1 Các giao dịch và tính đọc nhất quán



Thông điệp "Snapshot Too Old"

Trong rất ít trường hợp, Oracle không thể cho trở lại tập hợp những kết quả nhất quán (thường được gọi là ảnh chụp nhanh) cho truy vấn tiến hành lâu. Điều này xảy ra vì không đủ thông tin còn lại trong các đoạn hồi phục về trước để tái cấu trúc dữ liệu cũ hơn. Thường lỗi này được tạo khi nhiều hoạt động cập nhật làm cho đoạn hồi phục về trước để gói và ghi đè những thay đổi cần được cấu trúc lại dữ liệu mà truy vấn cần tiến hành lâu. Trong trường hợp này, lỗi 1555 sẽ cho kết quả:

ORA-1555: snapshot too old - anh chụp nhanh quá cũ (đoạn hồi phục về trước quá nhỏ)

Bạn có thể tránh lỗi này bằng cach tạo nhiều đoạn hồi phục về trước hoặc lớn hơn. Hoặc những truy vấn tiến hành lâu có thể được phát hành khi có một ít giao dịch đồng thời hoặc bạn có thể có được khóa dùng chung trên bảng bạn đang truy vấn, vì thế ngăn chặn bất kỳ khóa độc quyền khác trong khi giao dịch.

Tính đọc nhất quán mức câu lệnh

Oracle luôn luôn tăng cường tính nhất quán đọc mức câu lệnh. Điều này bảo đảm tất cả dữ liệu được cho trở lại bằng một truy vấn đơn xuất phát từ một thời điểm - thời điểm truy vấn bắt đầu. Do đó, một truy vấn không bao giờ thấy dữ liệu bị thay đổi hoặc bất kỳ thay đổi nào xảy ra do các giao dịch chỉ mục trong khi tiến hành truy vấn. Khi tiến hành truy vấn tiếp tục, chỉ dữ liệu được cam kết trước khi truy vấn đã bắt đầu có thể thấy truy vấn ấy. Truy vấn không thấy những thay đổi được cam kết sau khi tiến hành câu lệnh bắt đầu.

Tập hợp kết quả nhất quán được cung cấp cho mọi truy vấn để bảo đảm tính nhất quán dữ liệu, không có hoạt động nào về phía người dùng. Các câu lệnh SELECT, INSERT của SQL có một truy vấn con UPDATE và DELETE tất cả dữ liệu truy vấn, rõ ràng hoặc ngầm định và đều cho lại dữ liệu nhất quán. Mỗi câu lệnh này sử dụng một truy vấn để xác định dữ liệu nào nó sẽ ảnh hưởng (SELECT, INSERT, UPDATE hoặc DELETE tương ứng).

Câu lệnh SELECT là một truy vấn rõ ràng và có thể có những truy vấn lồng nhau hoặc một điều hoạt liên kết. Câu lệnh INSERT có thể sử dụng những truy vấn lồng nhau. Các câu lệnh UPDATE và DELETE có thể sử dụng mệnh đề WHERE hoặc các truy vấn con để chỉ ảnh hưởng đến một số hàng trong bảng thay vì tất cả các hàng.

Các truy vấn được sử dụng trong những câu lệnh INSERT, UPDATE và DELETE được bảo đảm một tập hợp kết quả nhất quán. Tuy nhiên, chúng không thấy những thay đổi được thực hiện do chính câu lệnh DML. Nói cách khác, truy vấn trong các điều hoạt này thấy dữ liệu khi nó đã tồn tại trước khi điều hoạt bắt đầu thực hiện các thay đổi.

Tính đọc nhất quán mức giao dịch

Oracle còn cung ứng tùy chọn tăng cường tính đọc nhất quán mức giao dịch. Khi một giao dịch tiến hành ở chế độ nối tiếp (xem bên dưới), tất cả truy xuất dữ liệu phản ánh trạng thái của CSDL như vào thời gian giao dịch bắt đầu. Điều này có nghĩa là dữ liệu được tất cả truy vấn thấy trong cùng giao dịch là nhất quán liên quan đến một thời điểm, ngoại trừ những truy vấn được thực hiện bởi một giao dịch nối tiếp thấy những thay đổi xảy ra bởi chính giao dịch ấy. Tính đọc nhất quán mức giao dịch tạo ra những lần đọc có thể lập lại và không trình bày truy vấn thành các bóng ma.

Các mức tách lập Oracle

Oracle cung cấp ba mức tách lập giao dịch:

Đọc được cam kết

Đây là mức tách lập giao dịch mặc định. Mỗi truy vấn được tiến hành bằng một giao dịch chỉ thấy dữ liệu đã được cam kết trước khi truy vấn (không phải giao dịch) đã bắt đầu. Một truy vấn Oracle sẽ không bao giờ đọc dữ liệu bị thay đổi (chưa cam kết).

Bởi vì Oracle không ngăn chặn các giao dịch khác không để sửa đổi dữ liệu được đọc bởi một truy vấn, dữ liệu đó có thể bị thay đổi bởi các giao dịch khác giữa hai lần tiến hành truy vấn ấy. Vì thế,

	một giao dịch tiến hành một truy vấn cho trước hai lần có thể trải qua cả đọc không lặp lại được và các bóng ma.
Các giao dịch nối tiếp hóa	Các giao dịch nối tiếp hóa chỉ thấy những thay đổi đã được cam kết đó vào thời điểm giao dịch bắt đầu, cùng với những thay đổi đó được thực hiện bằng chính giao dịch qua các câu lệnh INSERT, UPDATE và DELETE. Các giao dịch nối tiếp hóa không trải qua các lần đọc không lặp lại hoặc các bóng ma.
Chỉ đọc	Các giao dịch chỉ đọc chỉ thấy những thay đổi đã được cam kết vào thời điểm giao dịch đã bắt đầu và không cho phép các câu lệnh INSERT, UPDATE và DELETE.

Xác lập mức tách lập

Các nhà thiết kế ứng dụng, các nhà phát triển ứng dụng và các người điều hành CSDL có thể chọn các mức tách lập thích hợp cho các giao dịch khác nhau tùy theo ứng dụng và tải công việc. Bạn có thể xác lập mức tách lập giao dịch bằng cách sử dụng một trong những lệnh sau vào lúc bắt đầu giao dịch:

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;

SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;

SET TRANSACTION ISOLATION LEVEL READ ONLY;
```

Để tiết kiệm chi phí tiến trình và nối mạng vào lúc bắt đầu mỗi giao dịch bằng lệnh SET TRANSACTION, bạn có thể sử dụng lệnh ALTER SESSION để xác lập mức tách lập giao dịch cho tất cả những giao dịch kế tiếp:

```
ALTER SESSION SET ISOLATION_LEVEL SERIALIZABLE;
```

ALTER SESSION SET ISOLATION_LEVEL READ COMMITTED;

Tách lập đọc có cam kết

Mức tách lập mặc định đối với Oracle được đọc có cam kết. Mức tách lập này thích hợp với các môi trường nơi ít giao dịch có thể xung đột. Oracle làm cho mỗi truy vấn tiến hành liên quan đến thời gian ảnh chụp nhanh của riêng nó, bằng cách đó cho phép các lần đọc không lặp lại được và các bóng ma cho nhiều tiến hành của một truy vấn, nhưng cung cấp lưu lượng truyền có tiềm năng cao hơn. Tách lập đọc có cam kết là mức thích hợp tách lập đối với các môi trường nơi ít giao dịch có thể xung đột.

Tách lập nối tiếp hóa

Tách lập nối tiếp hóa thích hợp đối với các môi trường

- với các CSDL lớn và các giao dịch ngắn chỉ cập nhật ít hàng
- với cơ hội hai giao dịch đồng thời sẽ sửa đổi những hàng tương tự tương đối thấp và
- với những giao dịch tiến hành lâu tương đối chủ yếu chỉ đọc.

Tách lập nối tiếp hóa cho phép những giao dịch đồng thời chỉ tạo ra các thay đổi CSDL mà chúng có thể thực hiện nếu các giao dịch đã có lịch trình tiến hành lần lượt. Nhất là, Oracle cho phép một giao dịch nối tiếp hóa sửa đổi hàng dữ liệu chỉ nếu nó có thể xác định những thay đổi trước hàng đã được thực hiện bằng các giao dịch đã cam kết khi giao dịch nối tiếp hóa bắt đầu.

Để xác định một cách hiệu quả, Oracle sử dụng thông tin điều khiển được lưu trữ trong khối dữ liệu chỉ rõ các hàng nào trong khối ấy có những thay đổi được cam kết và chưa cam kết. Có

nghĩa khối có những giao dịch lịch sử gần đây đã ảnh hưởng đến mỗi hàng trong khối ấy. Số lượng giao dịch lịch sử được duy trì, được tham số INITRANS của CREAT TABLE và ALTER TABLE điều khiển.

Trong một số trường hợp, Oracle có thể có thông tin lịch sử không đủ xác định một hàng có được cập nhật bằng một giao dịch "gần đây nhất". Điều này có thể xảy ra khi nhiều giao dịch sửa đổi cùng khối dữ liệu đồng thời hoặc thực hiện như vậy trong khoảng thời gian rất ngắn. Bạn có thể tránh trường hợp này bằng cách xác lập những giá trị của INITRANS cao hơn cho các bảng sẽ trải qua nhiều giao dịch cập nhật cùng các khối. Thực hiện như vậy sẽ cho phép phân bổ bộ nhớ đủ trong mỗi khối để ghi các giao dịch truy xuất khối ấy mới đây.

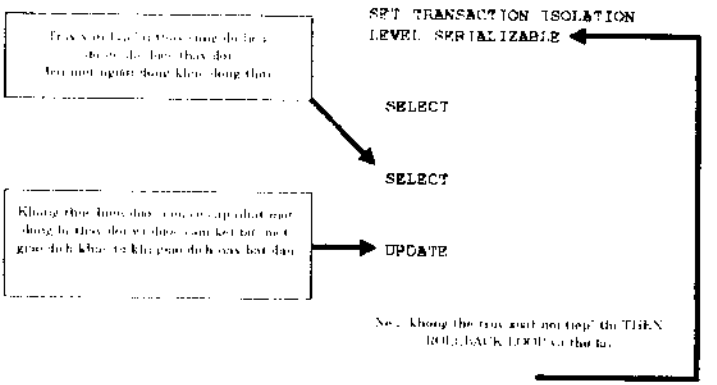
Oracle phát sinh một lỗi khi giao dịch nối tiếp hóa cố cập nhật hoặc xóa dữ liệu bị sửa đổi bởi một giao dịch cam kết sau khi giao dịch nối tiếp bắt đầu:
ORA-08177: Cannot serialize access for this transaction

Khi một giao dịch nối tiếp hóa không thực hiện được do lỗi "Cannot serialize access", thì ứng dụng ấy có thể sử dụng một trong nhiều hoạt động sau:

- cam kết công việc được tiến hành tới điểm đó
- tiến hành những câu lệnh bổ sung (nhưng khác nhau) (có thể sau khi hồi phục về trước tới điểm lưu được thiết lập ban đầu trong giao dịch ấy)
- hồi phục về trước toàn bộ giao diện

Hình 23-2 trình bày một thí dụ về một ứng dụng hồi phục về trước và thử lại giao dịch ấy sau khi nó không thực hiện được với lỗi "Cannot serialize access":

Hình 23-2 Sai sót giao dịch nối tiếp hóa



So sánh đọc có cam kết và tách lập nối tiếp hóa

Oracle cung cấp cho nhà phát triển ứng dụng một chọn lựa về hai mức tách lập giao dịch với cửa sổ đặc tính khác nhau. Cả đọc có cam kết và các mức tách lập nối tiếp hóa cung cấp mức độ nhất quán và đồng thời cao. Cả hai mức cung cấp các lợi ích làm giảm tranh chấp của mô hình điều khiển tính đồng thời đa phiên bản "đọc nhất quán" của Oracle và bổ sung khóa mức hàng độc quyền, được thiết kế để khai thác ứng dụng thế giới thực.

Giao dịch xác lập tính nhất quán

Một cách hữu dụng để xem các mức tách lập nối tiếp hóa và đọc có cam kết trong Oracle, xét các viễn tưởng sau: giả sử có một tập hợp các bảng CSDL (hoặc bất kỳ tập hợp dữ liệu), một chuỗi cá biệt những lần đọc các hàng trong các bảng đó và tập hợp những giao dịch được cam kết vào thời điểm riêng biệt bất kỳ. Một điều hoạt (một truy vấn hoặc giao dịch) là giao dịch xác lập nhất quán - *transaction set consistent* - nếu tất cả những lần đọc của nó cho trở lại dữ liệu được ghi bằng cùng tập hợp những giao dịch cam kết. Một điều hoạt không phải là giao dịch được xác lập nhất quán và các lần đọc khác phản ánh những thay đổi xảy ra bởi các giao dịch khác. Một điều hoạt không phải là giao dịch

được xác lập nhất quán thực sự, thấy CSDL ở trạng thái không phản ảnh tập hợp những giao dịch được cam kết nào.

Oracle cung cấp những giao dịch tiến hành ở chế độ đọc có cam kết với giao dịch được xác lập nhất quán trên phương thức mỗi câu lệnh. Chế độ nối tiếp hóa cung cấp giao dịch được xác lập nhất quán trên phương thức mỗi giao dịch.

Bảng 23-1 tóm tắt những khác biệt chính giữa các giao dịch đọc có cam kết và nối tiếp hóa trong Oracle.

Bảng 23-1 Các giao dịch đọc có cam kết và nối tiếp hóa

	Đọc có cam kết	**Nối tiếp hóa**
Ghi dữ liệu thay đổi	Không thể xảy ra	Không thể xảy ra
Đọc dữ liệu thay đổi	Không thể xảy ra	Không thể xảy ra
Đọc không lặp lại được	Có thể xảy ra	Không thể xảy ra
Các bóng ma	Có thể xảy ra	Không thể xảy ra
Theo ANSI/ISO SQL 92	Có	Có
Thời gian đọc ảnh chụp nhanh	Câu lệnh	Giao dịch
Giao dịch xác lập nhất quán	Mức câu lệnh	Mức giao dịch
Khóa mức hàng	Có	Có
Các bộ ghi khối các bộ đọc	Không	Không
Các bộ đọc khối các bộ ghi	Không	Không
Các bộ ghi khối bộ đọc hàng khác nhau	Không	Không

Các bộ ghi khối các bộ ghi hàng giống nhau	Có	Có
Chờ giao dịch lập khối	Có	Có
Phụ thuộc vào "cannot serialize access"	Không	Có
Lỗi sau khi lập khối giao dịch hủy ngang	Không	Không
Lỗi sau khi giao dịch khối chỉ mục	Không	Không

Khóa mức hàng

Cả các giao dịch đọc có cam kết và các giao dịch nối tiếp hóa sử dụng khóa mức hàng, và cả hai sẽ chờ nếu chúng cố thay đổi một hàng được cập nhật bởi một giao dịch đồng thời không có cam kết. Giao dịch thứ hai cố cập nhật một hàng cho trước chờ giao dịch khác để cam kết hoặc hồi phục về trước và phát hành khóa của nó. Nếu giao dịch khác hồi phục về trước, giao dịch chờ (bất kể chế độ tách lập của nó) có thể tiến hành thay đổi hàng có khóa trước, giống như giao dịch khác đã không tồn tại.

Tuy nhiên, nếu giao dịch (lập khối) khác chỉ mục và phát hành các khóa của nó, một giaọ dịch đọc có cam kết tiến hành với cập nhật có dụng ý của nó. Tuy nhiên, giao dịch nối tiếp hóa không thực hiện được với lỗi "Cannot serialize access", bởi vì giao dịch khác đã cam kết thay đổi đã xảy ra khi giao dịch nối tiếp hóa đã bắt đầu.

Tính nguyên vẹn tham chiếu

Bởi vì Oracle không sử dụng các khóa đọc trong các giao dịch đọc nhất quán hoặc nối tiếp hóa, đọc dữ liệu bằng một giao dịch có thể ghi đè bởi một giao dịch khác. Các giao dịch thực hiện kiểm tính đọc nhất quán ở mức ứng dụng sẽ không thừa nhận dữ liệu chúng đọc sẽ vẫn không thay đổi khi tiến hành giao dịch ấy (dù

những thay đổi như thế giao dịch không thấy). Tính không nhất quán CSDL có thể xảy ra nếu kiểm tính nhất quán mức ứng dụng không được mã hóa với CSDL này, ngay cả khi sử dụng những giao dịch nối tiếp hoa.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Application Developer về tính nguyên vẹn tham chiếu và các giao dịch nối tiếp hóa.

Oracle Parallel Server

Bạn có thể sử dụng cả mức tách lập giao dịch đọc có cam kết và giao dịch nối tiếp hóa trong Oracle Parallel Server (nhiều môi trường làm việc Oracle tiến hành dựa trên một cơ sở dữ liệu đơn).

Các giao dịch được phân phối

Trong môi trường CSDL được phân phối, một giao dịch cho trước cập nhật dữ liệu trong nhiều CSDL vật lý (được bảo vệ bởi cam kết hai pha để bảo đảm tất cả các nút hoặc không có cam kết) tham gia vào giao dịch nối tiếp hóa cần được hỗ trợ chế độ tách lập nối tiếp hóa.

Nếu giao dịch nối tiếp hóa có cập nhật dữ liệu trong CSDL được một server quản lý không hỗ trợ những giao dịch nối tiếp hóa, thì giao dịch ấy nhận một lỗi. Giao dịch có thể hồi phục về trước và thử lại chỉ khi server từ xa hỗ trợ các giao dịch nối tiếp hóa.

Trái lại, các giao dịch có cam kết đọc có thể thực hiện những giao dịch được phân phối với các server không hỗ trợ các giao dịch nối tiếp hoa.

Chọn mức tách lập

Các nhà thiết kế ứng dụng và các nhà phát triển ứng dụng nên chọn mức tách lập dựa trên hiệu suất ứng dụng và các nhu cầu tính nhất quán cũng như các đòi hỏi mã hóa ứng dụng.

Đối với các môi trường có nhiều người dùng đồng thời đệ trình các giao dịch nhanh chóng, nhà thiết kế phải truy xuất các đòi hỏi hiệu suất giao dịch với điều kiện đòi hỏi về tốc độ đi đến của giao dịch được yêu cầu và thời gian trả lời. Thường đối với các môi trường có hiệu suất cao, sự chọn lựa mức tách lập liên quan đến trao đổi giữa tính nhất quán và đồng thời (lưu lượng truyền giao dịch).

Logic ứng dụng kiểm tính nhất quán CSDL phải lưu ý tới sự kiện đọc khối ghi ở hai chế độ.

Cả hai chế độ tách lập của Oracle cung cấp các mức nhất quán và đồng thời cao (và hiệu suất) qua tổ hợp của khóa mức hàng và hệ thống điều khiển đồng thời đa phiên bản. Các bộ đọc và ghi không lập khối với nhau trong Oracle; do đó, khi các truy vấn vẫn thấy dữ liệu nhất quán, thì cả tách lập đọc có cam kết và tách lập nối tiếp hóa cung cấp tính đồng thời ở mức cao để có hiệu suất cao mà không cần đọc dữ liệu (thay đổi) chưa được cam kết.

Chọn tách lập đọc có cam kết

Đối với nhiều ứng dụng, đọc có cam kết là mức tách lập thích hợp nhất. Đây là mức tách lập được sử dụng bởi các ứng dụng tiến hành trên các bản phát hành của Oracle có trước Release 7.3.

Tách lập đọc có cam kết có thể cung cấp tính đồng thời nhiều hơn đáng kể, nhưng có phần gia tăng rủi ro về các kết quả không nhất quán (do các bóng ma và đọc không lặp lại được) đối với một số giao dịch.

Nhiều môi trường hiệu suất cao có những tốc độ giao dịch đến cao đòi hỏi lưu lượng truyền cao và thời gian trả lời nhanh hơn có thể đạt được bằng tách lập nối tiếp hóa. Các môi trường khác hỗ trợ một số người dùng có tốc độ giao dịch đến thấp cũng phải gặp một ít rủi ro về các kết quả không chính xác do các bóng ma và đọc không lặp lại được. Tách lập đọc có cam kết thích hợp với cả hai môi trường này.

Tách lập đọc có cam kết của Oracle cung cấp giao dịch xác lập tính nhất quán cho mọi truy vấn (nghĩa là mọi truy vấn thấy dữ liệu ở trạng thái nhất quán). Do đó, tách lập đọc có cam kết sẽ đủ đáp ứng cho nhiều ứng dụng có thể đòi hỏi mức tách lập cao hơn nếu được tiến hành trên những hệ thống quản lý CSDL khác không sử dụng điều khiển tính đồng thời đa phiên bản.

Chế độ tách lập đọc có cam kết không đòi hỏi logic ứng dụng để bẫy lỗi "Cannot serialize access" và quay trở lại khởi động giao dịch. Trong hầu hết những ứng dụng, một ít giao dịch có nhu cầu chức năng phát hành lại cùng truy vấn hai lần, vì vậy nhiều ứng dụng không cần bảo vệ tránh các bóng ma và đọc không lặp lại được. Do đó nhiều nhà phát triển chọn đọc có cam kết tránh nhu cầu đọc kiểm lỗi như thế và thử lại mã trong mỗi giao dịch.

Chọn tách lập nối tiếp hóa

Tách lập nối tiếp hóa của Oracle thích hợp với các môi trường nơi tương đối có ít cơ hội hai giao dịch đồng thời sẽ sửa đổi cùng các hàng và các giao dịch tiến hành tương đối lâu chủ yếu là chỉ đọc. Nó thích hợp nhất cho các môi trường có những giao dịch ngắn và các CSDL lớn chỉ cập nhật một ít hàng.

Chế độ tách lập nối tiếp hóa cung cấp tính nhất quán có phần hơn bằng cách tránh các bóng ma và đọc không lặp lại được và

có thể quan trọng đối với nơi giao dịch đọc/ghi tiến hành một truy vấn hơn một lần.

Không giống như các bổ sung khác về tách lập nối tiếp hóa, khóa lập khối để đọc cũng như ghi, Oracle cung cấp các truy vấn không lập khối và mức độ chi tiết nhất của khóa mức hàng, cả hai điều này làm giảm tranh chấp đọc/ghi. Đối với những ứng dụng trải qua phần lớn tranh chấp đọc/ghi, tách lập nối tiếp hóa của Oracle có thể cung cấp lưu lượng truyền nhiều hơn đáng kể so với các hệ thống khác. Do đó, một số ứng dụng có thể thích hợp với tách lập nối tiếp hóa với Oracle nhưng không thích hợp với các hệ thống khác.

Tất cả những truy vấn trong giao dịch nối tiếp hóa của Oracle thấy CSDL như vào một thời điểm đơn, vì vậy mức tách lập này thích hợp đối với nơi có nhiều truy vấn nhất quán phải được phát hành trong giao dịch đọc-ghi. Ứng dụng ghi báo biểu phát sinh dữ liệu tóm tắt và lưu trữ nó vào CSDL có thể sử dụng chế độ nối tiếp hóa vì nó cung cấp tính nhất quán mà một giao dịch READ ONLY cung cấp, nhưng cũng cho phép INSERT, UPDATE và DELETE.

Lưu ý:
Các giao dịch có các câu lệnh DML có những truy vấn con nên sử dụng tách lập nối tiếp hóa để bảo đảm đọc nhất quán.

Mã hóa các giao dịch nối tiếp hóa đòi hỏi nhà phát triển ứng dụng thêm việc (kiểm lỗi "Cannot serialize access" và hồi phục về trước và thử lại giao dịch). Tương tự như công việc mã hóa bổ sung cần thiết ở các hệ thống quản lý CSDL khác để quản lý các bế tắc. Để phối hợp chặt chẽ những tiêu chuẩn hoặc để các ứng dụng được tiến hành trên nhiều hệ thống quản lý CSDL, có thể cần thiết kế những giao dịch cho chế độ nối tiếp hóa. Những giao dịch kiểm các sai sót về khả năng nối tiếp và thử lại có thể phải

được sử dụng với chế độ đọc có cam kết của Oracle (để không phát sinh các lỗi về khả năng nối tiếp hóa).

Chế độ nối tiếp hóa có thể không phải là chọn lựa tốt nhất ở môi trường có các giao dịch tiến hành tương đối lâu phải cập nhật cùng các hàng được truy xuất bởi một dung lượng cao các giao dịch cập nhật ngắn. Vì giao dịch tiến hành lâu hơn không thể là giao dịch đầu tiên sửa đổi một hàng định trước, nó sẽ phải lập lại để hồi phục về trước, làm lãng phí công việc. (Lưu ý sự bổ sung "bi quan" khóa đọc quy ước sẽ không thích hợp với môi trường này, vì các giao dịch tiến hành lâu - thậm chí các giao dịch đọc - sẽ cản trở tiến trình cập nhật ngắn và ngược lại).

Các nhà phát triển ứng dụng sẽ quan tâm đến chi phí hồi phục về trước và thử lại các giao dịch khi sử dụng chế độ nối tiếp hóa. Giống như với các hệ thống khóa đọc, nơi các bế tắc thường xảy ra, sử dụng chế độ nối tiếp hóa đòi hỏi hồi phục về trước công việc được các giao dịch hủy ngang thực hiện và thử lại chúng. Trong môi trường có tranh chấp cao, hoạt động này có thể sử dụng nhiều tài nguyên đáng kể.

Trong hầu hết các môi trường, giao dịch khởi động lại sau khi nhận một lỗi "Cannot serialize access" không thể gặp phải xung đột thứ hai với một giao dịch khác. Vì lý do này, nó có thể giúp tiến hành các câu lệnh đó hoàn toàn không có thể tranh chấp với các giao dịch khác càng sớm càng tốt trong một giao dịch nối tiếp hóa. Tuy nhiên, không bảo đảm giao dịch sẽ hoàn tất thành công, vì vậy ứng dụng ấy phải được mã hóa để giới hạn số lần phải thử lại.

Dù chế độ nối tiếp hóa của Oracle tương thích với SQL92 và cung ứng nhiều lợi ích so với các bổ sung khóa đọc, nó không cung cấp ngữ nghĩa giống với các hệ thống ấy. Những nhà thiết kế ứng dụng phải quan tâm đến sự kiện đọc trong Oracle không lập khối ghi như chúng thực hiện trong các hệ thống khác. Các giao dịch

kiểm tính nhất quán CSDL ở mức ứng dụng có thể đòi hỏi các kỹ thuật mã hóa, chẳng hạn như sử dụng SELECT FOR UPDATE. Vấn đề này sẽ được xét khi những ứng dụng sử dụng chế độ nối tiếp hóa được cơ động sang Oracle từ các môi trường khác.

Oracle khóa dữ liệu như thế nào

Các khóa là các cơ chế ngăn chặn tương tác hủy giữa các giao dịch truy xuất cùng tài nguyên - đối tượng người dùng (chẳng hạn như các bảng và các hàng) hoặc các đối tượng hệ thống người dùng không thấy (chẳng hạn như các cấu trúc dữ liệu dùng chung trong bộ nhớ và các hàng từ điển dữ liệu).

Trong tất cả những trường hợp, Oracle tự động có được những khóa cần thiết khi tiến hành các câu lệnh SQL, vì vậy người dùng không cần quan tâm đến những chi tiết như thế. Oracle tự động sử dụng mức ứng dụng thấp nhất về sự giới hạn để cung cấp mức cao nhất tính đồng thời dữ liệu mà còn cung cấp tính nguyên vẹn dữ liệu an toàn sự cố. Oracle cũng cho phép người dùng khóa dữ liệu thủ công.

Xem thêm cửa sổ khóa bên trong được Oracle sử dụng "Các kiểu khóa"

Các giao dịch và tính đồng thời dữ liệu

Oracle cung cấp tính đồng thời và nguyên vẹn dữ liệu giữa các giao dịch sử dụng cơ chế khóa của nó. Vì các cơ chế khóa của Oracle liên kết chặt chẽ với điều khiển giao dịch, những nhà thiết kế ứng dụng chỉ cần xác định các giao dịch thích hợp và Oracle tự động quản lý khóa.

Nhớ là khóa Oracle tự động hoàn toàn và không đòi hỏi hoạt động của người dùng. Khóa ngầm định xảy ra đối với tất cả những câu lệnh SQL sao cho những người dùng CSDL không bao giờ phải bất kỳ các khóa tài nguyên nào. Các cơ chế khóa mặc

định của Oracle khóa dữ liệu ở mức thấp nhất về giới hạn để bảo đảm tính nguyên vẹn dữ liệu khi cho phép mức cao nhất về tính đồng thời dữ liệu.

Các phần sau còn đề cập những trường hợp nơi bạn có thể muốn có các khóa thủ công hoặc thay đổi xử lý khóa mặc định của Oracle và giải thích bạn thực hiện điều đó như thế nào, xem "Khóa dữ liệu rõ ràng (thủ công)".

Các chế độ khóa

Oracle sử dụng hai chế độ trong CSDL đa người dùng:

Chế độ khóa độc quyền	Ngăn chặn tiến trình được kết hợp với tài nguyên dùng chung. Chế độ khóa này có được để sửa đổi dữ liệu. Giao dịch thứ nhất để khóa tài nguyên độc quyền là giao dịch duy nhất có thể thay đổi tài nguyên ấy cho tới khi khóa độc quyền được giải phóng.
Chế độ khóa dùng chung	Cho phép tài nguyên kết hợp được dùng chung tùy theo các điều hoạt liên quan. Đọc dữ liệu đa người dùng có thể dùng chung dữ liệu, lưu giữ các khóa dùng chung để ngăn chặn bộ ghi truy xuất đồng thời (người gần khóa độc quyền). Nhiều giao dịch có thể đạt được các khóa dùng chung trên cùng tài nguyên.

Thời gian khóa

Tất cả các khóa các câu lệnh có được trong một giao dịch được lưu trữ cho thời gian của giao dịch ấy để ngăn chặn giao thoa hủy (kể cả đọc dữ liệu bị thay đổi, các cập nhật bị mất và điều hoạt DDL hủy) với cửa sổ giao dịch đồng thời. Các thay đổi do các câu lệnh SQL trở thành có thể thấy được chỉ đối với những giao dịch khác khởi động sau giao dịch thứ nhất được cam kết.

Oracle giải phóng tất cả những khóa các câu lệnh đạt được trong một giao dịch khi bạn cam kết hoặc hồi phục về trước giao dịch ấy. Oracle còn giải phóng các khóa đạt được sau một điểm lưu khi hồi phục về điểm lưu trước. Tuy nhiên, chỉ các giao dịch không chờ những tài nguyên có khóa trước đó có thể đạt được các khóa trên những tài nguyên có thể sử dụng bây giờ. Các giao dịch chờ sẽ tiếp tục chờ cho tới sau khi giao dịch ban đầu cam kết hoặc hồi phục về trước hoàn toàn.

Chuyển đổi khóa dữ liệu so với xếp chuỗi khóa

Một giao dịch lưu trữ các khóa hàng độc quyền đối với tất cả những hàng được chèn, cập nhật hoặc xóa trong giao dịch ấy. Vì các khóa hàng đạt được ở mức giới hạn cao nhất, không cần chuyển đổi khóa.

Oracle tự động chuyển đổi khóa bảng ở mức giới hạn thấp nhất thành khóa có giới hạn cao hơn thích hợp. Thí dụ, giả sử một giao dịch sử dụng câu lệnh SELECT với mệnh đề FOR UPDATE để khóa các hàng của một bảng. Do đó, nó đạt được các khóa hàng độc quyền và khóa bảng dùng chung hàng cho bảng ấy. Nếu sau đó giao dịch cập nhật một hoặc nhiều hàng có khóa, khóa bảng dùng chung hàng tự động được chuyển đổi thành khóa bảng độc quyền hàng. Xem thêm "Các khóa bảng".

Xếp chuỗi khóa xảy ra khi nhiều khóa được lưu trữ ở một mức độ chi tiết (thí dụ, các hàng) và một CSDL phát sinh các khóa thành mức độ chi tiết cao hơn (thí dụ, bảng). Thí dụ, nếu một người dùng khóa nhiều hàng trong một bảng, một số CSDL sẽ tự động xếp chuỗi các khóa hàng của người dùng ấy thành một bảng đơn. Số các khóa được giảm, nhưng sự giới hạn những gì đang được khóa tăng lên.

Oracle không bao giờ xếp chuỗi khóa. Xếp chuỗi khóa làm tăng khả năng bị bế tắc rất cao (được đề cập ở dưới). Hình dung

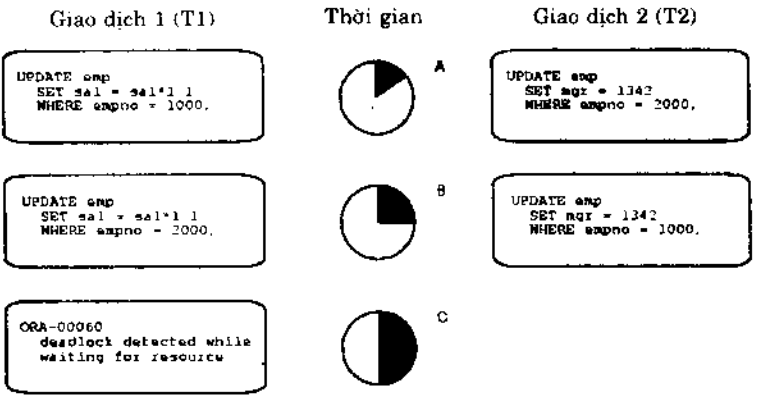
trường hợp nơi hệ thống đang cố xếp chuỗi các khóa thay cho một giao dịch T1 nhưng không thể được vì các khóa ấy được giao dịch T2 lưu giữ. Sự bế tắc được xảy ra nếu T2 còn đòi hỏi xếp chuỗi khóa của cùng dữ liệu trước khi nó có thể tiến hành.

Các bế tắc

Sự bế tắc có thể xảy ra khi hai hoặc nhiều người dùng đang chờ dữ liệu được khóa với nhau. Các bế tắc ngăn chặn một số giao dịch không cho tiếp tục hoạt động. Hình 23-3 minh họa hai giao dịch trong một bế tắc.

Trong Hình 23-3, không có vấn đề nào tồn tại ở thời điểm A, vì mỗi giao dịch có một khóa hàng mà nó cố cập nhật. Mỗi giao dịch tiến hành (không bị kết thúc). Tuy nhiên, mỗi cố gắng cập nhật tiếp hàng ấy hiện được một giao dịch khác lưu giữ. Do đó, một bế tắc xảy ra vào thời điểm B, vì không có giao dịch nào có thể có được tài nguyên nó cần để tiến hành hoặc kết thúc. Nó là một bế tắc vì bất kể giao dịch phải chờ bao lâu, các khóa xung đột được lưu trữ.

Hình 23-3 Hai giao dịch trong một bế tắc



Sự phát hiện bế tắc

Oracle tự động phát hiện những trường hợp bị bế tắc và giải quyết chúng bằng cách hồi phục về trước một trong những câu lệnh có liên quan đến bế tắc ấy, bằng cách đó để giải phóng một tập hợp những khóa hàng xung đột. Thông báo tương ứng cũng được cho trở lại giao dịch ấy phải qua hồi phục về trước mức câu lệnh. Câu lệnh được hồi phục về trước là một câu lệnh thuộc giao dịch phát hiện bế tắc ấy. Thường giao dịch có tín hiệu sẽ được hồi phục về trước rõ ràng, nhưng nó có thể thử lại câu lệnh hồi phục về trước sau khi chờ.

Lưu ý:
Trong các giao dịch được phân phối, các bế tắc cục bộ được phát hiện bằng cách phân tích một mạch "chờ", và các bế tắc toàn cục được phát hiện bằng hết thời gian. Một khi được phát hiện, các bế tắc không được phân phối và được phân phối được điều quản bằng CSDL và ứng dụng theo cách giống nhau.

Các bế tắc thường xảy ra nhất khi những giao dịch chạy đề khóa mặc định của Oracle. Bởi vì chính Oracle không xếp chuỗi khóa và không sử dụng các khóa đọc cho các truy vấn, nhưng sử dụng khóa mức hàng (thay vì khóa mức trang). Xem "Khóa dữ liệu rõ ràng (thủ công)" về đạt được các khóa thủ công và thí dụ về trường hợp bế tắc.

Tránh các bế tắc

Các bế tắc đa bảng thường có thể được tránh nếu các giao dịch truy xuất cùng bảng khóa các bảng đó theo cùng thứ tự, bằng các khóa ngầm định hoặc rõ ràng. Thí dụ, tất cả những nhà phát triển ứng dụng có thể theo nguyên tắc khi cả hai bảng gốc và bảng chi tiết đều được cập nhật, bảng gốc được khóa đầu tiên rồi sau đó đến bảng chi tiết. Nếu các nguyên tắc ấy được thiết kế thích hợp thì được thực hiện theo tất cả những ứng dụng, các bế tắc khó có thể xảy ra.
Khi bạn biết sẽ đòi hỏi một chuỗi các khóa cho một giao dịch, trước hết bạn nên xét cách để có được khóa độc quyền nhất (tương thích ít nhất).

Các kiểu khóa

Oracle tự động sử dụng những kiểu khóa khác nhau để điều khiển truy xuất đồng thời dữ liệu và ngăn chặn tương tác hủy giữa các người dùng. Oracle tự động khóa tài nguyên thay cho giao dịch để ngăn chặn những giao dịch khác cũng thực hiện điều gì đó đòi hỏi truy xuất độc quyền cùng tài nguyên ấy. Khóa được tự động giải phóng khi một sự kiện nào đó xảy ra sao cho giao dịch ấy không còn đòi hỏi tài nguyên ấy.

Trong khắp diều hoạt của nó, Oracle tự động đạt được những kiểu khóa khác nhau ở những mức giới hạn khác nhau tùy theo tài nguyên đang được khóa và điều hoạt đang được thực hiện.

Các khóa của Oracle rơi vào các biệt loại chung sau:

Các khóa DML (các khóa dữ liệu)	Các khóa DML bảo vệ dữ liệu. Thí dụ, các khóa bảng khóa toàn bộ các bảng, các khóa hàng khóa các hàng được lựa.
Các khóa DDL (các khóa từ điển)	Các khóa DDL bảo vệ cấu trúc của những đối tượng giản đồ, thí dụ, các định nghĩa của các bảng và các khung nhìn.
Các khóa bên trong và các mạch gài	Các khóa bên trong và các mạch gài bảo vệ những cấu trúc CSDL bên trong, chẳng hạn như các tập tin dữ liệu. Các khóa bên trong và các mạch gài hoàn toàn tự động.
Các khóa được phân phối	Các khóa được phân phối bảo đảm dữ liệu và những tài nguyên khác được phân phối trong các môi trường làm việc khác nhau của một Oracle Parallel Server vẫn duy trì nhất quán. Các khóa được phân phối được lưu trữ bởi các môi trường làm việc thay vì các giao dịch. Chúng truyền thông trạng thái hiện hành của tài nguyên trong các môi trường làm việc của một Oracle Parallel Server.
Các khóa quản lý cache song song (PCM)	Các khóa quản lý cache song song là các khóa được phân phối (bảng hoặc các khối chỉ mục) trong cache vùng đệm. Các khóa PCM không khóa bất kỳ hàng nào thay cho các giao dịch.

Chương này đề cập đến các khóa DML, các khóa DDL và các khóa bên trong tương ứng.

Các khóa DML (dữ liệu)

Mục đích của khóa DML (dữ liệu) để bảo đảm tính nguyên vẹn dữ liệu được nhiều người dùng truy xuất đồng thời. Các khóa DML ngăn chặn giao thoa hủy xung đột các điều hoạt DML và/hoặc DDL xảy ra cùng một lúc. Thí dụ, các khóa DML của Oracle xác định hàng trong bảng có thể được cập nhật chỉ bằng một giao dịch cùng một lúc và bảng không thể được bỏ nếu một giao dịch không được cam kết có một điều hoạt chèn vào bảng.

Các điều hoạt DML có thể đạt được có các khóa dữ liệu ở hai mức khác nhau: các hàng riêng biệt và đối với toàn bộ các bảng. Các phần sau giải thích các khóa hàng và bảng.

Lưu ý:
Tên tắt trong các dấu ngoặc sau mỗi kiểu khoa hoặc chế độ khóa trong các phần sau là chữ viết tắt được sử dụng trong Locks Monitor của Oracle Enterprise Manager. Oracle Enterprise Manager có thể hiển thị TM cho khóa hàng bất kỳ thay vì chỉ chế độ của khóa bảng (chẳng hạn như RS hoặc SRX).

Các khóa hàng (TX)

Chỉ các khóa DML tự động đạt được các khóa mức hàng. Không có giới hạn về số các khóa hàng được câu lệnh hoặc giao dịch lưu giữ và Oracle không xếp chuỗi khóa từ mức hàng tới mức độ không quá chi tiết. Khóa hàng có thể cung cấp khóa hạt mịn và vì vậy cung cấp được tính đồng thời và lưu lượng truyền tốt nhất.

Tổ hợp điều khiển đồng thời đa phiên bản và khóa mức hàng nghĩa là các người dùng tranh chấp dữ liệu chỉ khi truy xuất cùng các hàng, nhất là:
Các bộ đọc dữ liệu không chờ các bộ ghi của cùng những hàng dữ liệu.
Các bộ ghi dữ liệu không chờ các bộ đọc cùng các hàng dữ liệu (trừ khi SELECT... FOR UPDATE được sử dụng, nhất là yêu cầu một khóa cho bộ đọc ấy).
Các bộ ghi chỉ chờ các bộ ghi khác nếu chúng cố cập nhật cùng những hàng đồng thời.

Lưu ý:
Các bộ đọc dữ liệu có thể phải chờ các bộ ghi của cùng những khối dữ liệu trong một số trường hợp rất đặc-biệt về những giao dịch được phân phối chưa giải quyết.

Một giao dịch đạt được một khóa DML độc quyền cho mỗi hàng cá biệt được sửa đổi bằng một trong những câu lệnh sau: INSERT, UPDATE, DELETE và SELECT với mệnh đề FOR UPDATE.

Một hàng được sửa đổi luôn luôn được khóa độc quyền sao cho những người dùng khác không thể sửa đổi hàng ấy cho tới khi giao dịch ấy lưu giữ khóa được cam kết hoặc được hồi phục về trước. Các khóa hàng luôn luôn được Oracle tự động đạt được do các câu lệnh ở trên.

Nếu một giao dịch có một khóa hàng cho một hàng, thì giao dịch ấy cũng đạt được một khóa bảng cho bảng tương ứng. Khóa bảng cũng cần thiết để ngăn chặn xung đột các điều hoạt DDL chạy đè các thay đổi dữ liệu trong giao dịch hiện hành. Phần sau giải thích các khóa bảng và "Các khóa DDL (các khóa từ điển)" giải thích các khóa cần thiết cho những điều hoạt DDL.

Các khóa bảng (TM)

Một giao dịch đạt được một khóa bảng khi một bảng được sửa đổi trong các câu lệnh sau: INSERT, UPDATE, DELETE, SELECT với mệnh đề FOR UPDATE và LOCK TABLE. Các điều hoạt DML đạt được các khóa bảng cho hai mục đích: để đảo ngược truy xuất DML tới bảng thay cho một giao dịch và để ngăn chặn các điều hoạt DDL sẽ xung đột với giao dịch ấy. Khóa bảng bất kỳ ngăn chặn hoạt động để có được một khóa DDL độc quyền trên cùng bảng và bằng cách đó ngăn chặn các điều hoạt DDL đạt được các khóa như thế. Thí dụ, một bảng không thể được thay đổi hoặc bỏ nếu giao dịch không được cam kết lưu giữ một khóa bảng cho nó. (Xem thêm "Các khóa DDL độc quyền").

Một khóa bảng có thể được lưu giữ trong bất kỳ một trong nhiều chế độ: dùng chung hàng - row share (RS), hàng độc quyền - row exclusive (RX), dùng chung - share (S), hàng dùng chung độc quyền - share row exclusive (SRX), và độc quyền - exclusive (X). Giới hạn của chế độ của khóa bảng quyết định các chế độ trong đó các khóa bảng khác trên cùng bảng có thể đạt được và được lưu giữ.

Bảng 23-2 trình bày các chế độ khóa bảng mà các câu lệnh đạt được và các điều hoạt mà những khóa đó cho phép và ngăn chặn.

Bảng 23-2 Tóm tắt của các khóa bảng

Câu lệnh SQL	Chế độ khóa bảng	Các chế độ khóa có được cho phép không?				
		RS	RX	S	SRX	X
SELECT...FROM table...	Không	Y	Y	Y	Y	Y
INSERT INTO table ...	RX	Y	Y	N	N	N

UPDATE table ...	RX	Y*	Y*	N	N	N
DELETE FROM table ...	RX	Y*	Y*	N	N	N
SELECT ... FROM table FOR UPDATE OF ...	RS	Y*	Y*	Y*	Y*	N
LOCK TABLE table IN ROW SHARE MODE	RS	Y	Y	Y	Y	
LOCK TABLE table IN SHARE MODE	RX	Y	Y	N	N	N
LOCK TABLE table IN SHARE MODE	S	Y	N	Y	N	N
LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE	SRX	Y	N	N	N	N
LOCK TABLE table IN EXCLUSIVE MODE	X	N	N	N	N	N

RS: dùng chung hàng RX: độc quyền hàng S: dùng chung SRX: độc quyền hàng dùng chung X: độc quyền	*Yes, nếu không có khóa hàng xung đột nào được lưu trữ bằng một giao dịch khác; ngược lại phải chờ xảy ra.

Các phần sau giải thích từng chế độ của khóa bảng từ giới hạn ít nhất tới giới hạn nhiều nhất. Mỗi phần mô tả chế độ của khóa

bảng, các hoạt động làm cho giao dịch đạt được khóa bảng trong chế độ đó và những điều hoạt nào được cho phép và bị ngăn chặn trong các giao dịch khác bằng một khóa trong chế độ đó. Xem thêm khóa thủ công "Khóa dữ liệu rõ ràng (thủ công)".

Các khóa bảng dùng chung hàng (RS)*Error! Bookmark not defined.*

Một khóa bảng dùng chung hàng (đôi khi còn được gọi là khóa bảng dùng chung phụ, SỐ) chỉ giao dịch ấy lưu giữ khóa trên bảng có các hàng được khóa trong bảng ấy và có ý định cập nhật chúng. Một khóa bảng dùng chung hàng đạt được tự động cho một bảng khi một trong các câu lệnh SQL sau được tiến hành:
SELECT . . . FROM table . . . FOR UPDATE OF . . . ;

LOCK TABLE table IN ROW SHARE MODE;

Một khóa bảng dùng chung hàng ở chế độ giới hạn ít nhất của khóa bảng cung ứng mức đồng thời cao nhất cho một bảng.

Các điều hoạt được cho phép: khóa bảng dùng chung hàng được lưu giữ bằng một giao dịch cho phép các giao dịch khác truy vấn, chèn, cập nhật, xóa hoặc khóa các hàng đồng thời trong cùng bảng. Do đó, các giao dịch khác có thể có được các khóa dùng chung hàng đồng thời, độc quyền hàng, dùng chung và các khóa độc quyền hàng dùng chung cho cùng bảng.

Các điều hoạt bị ngăn chặn: khóa bảng dùng chung hàng được lưu giữ bằng một giao dịch ngăn chặn các giao dịch khác không truy xuất ghi độc quyền cùng bảng bằng cách chỉ sử dụng câu lệnh sau:
LOCK TABLE table IN EXCLUSIVE MODE;

Các khóa bảng độc quyền hàng (RX)*Error! Bookmark not defined.*

Khóa bảng độc quyền hàng (còn được gọi là khóa bảng độc quyền phụ, SẢN XUẤT) thường chỉ giao dịch lưu giữ khóa đã thực hiện một hoặc nhiều cập nhật những hàng trong bảng ấy. Khóa bảng độc quyền hàng bằng các kiểu câu lệnh sau:

INSERT INTO table . . . ;

UPDATE table . . . ;

DELETE FROM table . . . ;

LOCK TABLE table IN ROW EXCLUSIVE MODE;

Một khóa bảng độc quyền hàng giới hạn nhiều hơn khóa bảng dùng chung hàng một chút.

Các điều hoạt được cho phép: một khóa bảng độc quyền hàng được lưu giữ bằng một giao dịch cho phép các giao dịch khác truy vấn, chèn, cập nhật, xóa hoặc khóa các hàng đồng thời trong cùng bảng. Do đó, các khóa bảng độc quyền hàng cho phép nhiều giao dịch có được các khóa bảng dùng chung hàng và độc quyền hàng đồng thời cho cùng bảng.

Các điều hoạt bị ngăn chặn: một khóa bảng độc quyền hàng được lưu giữ bằng một giao dịch ngăn chặn các giao dịch khác không khóa thủ công được để đọc hoặc ghi độc quyền. Do đó, các giao dịch khác không thể khóa bảng ấy đồng thời bằng cách sử dụng các câu lệnh sau:

LOCK TABLE table IN SHARE MODE;

LOCK TABLE table IN SHARE EXCLUSIVE MODE;

LOCK TABLE table IN EXCLUSIVE MODE;

Các khóa bảng dùng chung (S)*Error! Bookmark not defined.*

Một khóa bảng dùng chung tự động đạt được cho *bảng* được ấn định trong câu lệnh sau:
LOCK TABLE table IN SHARE MODE;

Các điều hoạt được cho phép: một khoa bang dùng chung được lưu giữ bằng một giao dịch cho phep các giao dịch khác chi truy vấn bảng ấy, khoa các hàng riêng biệt với cac câu lệnh SELECT . . . FOR UPDATE hoặc tiến hanh các câu lệnh LOCK TABLE . . . IN SHARE MODE thành công; không có các cập nhật được các giao dịch khác cho phép. Nhiều giao dịch có thể lưu giữ các khóa bang dùng chung cho cùng bảng đồng thời. Trong trường hợp này, không giao dịch nào có thể cập nhật bảng (dù một giao dịch lưu giữ các khóa hàng bằng một câu lệnh SELECT với mệnh đề FOR UPDATE). Do đó, một giao dịch có một khóa bảng dùng chung có thể cập nhật chỉ nếu không có giao dịch nào khác cùng có khóa bảng dùng chung trên cùng bảng.

Các điều hoạt bị ngăn chặn: một khóa bảng dùng chung dược lưu giữ bằng một giao dịch ngăn chặn các giao dịch khác sửa đổi cùng bảng bằng cách tiến hành câu lệnh sau:
LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE;

LOCK TABLE table IN EXCLUSIVE MODE;

LOCK TABLE table IN ROW EXCLUSIVE MODE;

Các khóa bảng độc quyền hàng dùng chung (SRX)*Error! Bookmark not defined.*

Một khóa bảng độc quyền hàng (đôi khi còn được gọi là khóa bảng độc quyền phụ - *share-subexclusive table lock*, SSX) bị giới hạn nhiều hơn khóa bảng dùng chung. Khóa bảng độc quyền hàng dùng chung đạt được cho một bảng như sau:
LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE;

Các điều hoạt được cho phép: chỉ có một giao dịch mỗi lần có thể đạt được một khóa bảng độc quyền hàng dùng chung trên bảng cho trước. Một khóa bảng độc quyền hàng dùng chung được lưu giữ bằng một giao dịch cho phép các giao dịch khác truy vấn hoặc khóa các hàng riêng biệt bằng cách sử dụng SELECT với mệnh đề FOR UPDATE, một cập nhật bảng ấy.

Các điều hoạt bị ngăn chặn: một khóa bảng độc quyền hàng dùng chung được lưu giữ bằng một giao dịch ngăn chặn các giao dịch khác đạt được các khóa bảng độc quyền hàng và sửa đổi cùng bảng. Một khóa bảng độc quyền hang dùng chung còn ngăn chặn các giao dịch khác đạt được các khóa bảng dùng chung, độc quyền hàng dùng chung và các khóa bảng độc quyền để ngăn chặn các giao dịch khác tiến hành các câu lệnh sau:
LOCK TABLE table IN SHARE MODE;

LOCK TABLE table IN SHARE ROW EXCLUSIVE MODE;

LOCK TABLE table IN ROW EXCLUSIVE MODE;

LOCK TABLE table IN EXCLUSIVE MODE;

Các khóa bảng độc quyền (X)*Error! Bookmark not defined.*

Một khóa bảng độc quyền là chế độ khóa bảng giới hạn nhất, cho phép giao dịch ấy lưu giữ khóa độc quyền ghi truy xuất bảng ấy. Khóa bảng độc quyền đạt được cho một bảng như sau:
LOCK TABLE table IN EXCLUSIVE MODE;

Các điều hoạt cho phép: chỉ một giao dịch có thể đạt được khóa bảng độc quyền cho một bảng. Khoa bảng độc quyền chỉ cho phép các giao dịch khác truy vấn bảng ấy.

Các điều hoạt bị ngăn chặn: khóa bảng độc quyền được lưu giữ bằng một giao dịch ngăn chặn các giao dịch khác thực hiện bất kỳ kiểu câu lệnh DML nào hoặc đưa bất kỳ kiểu khóa nào vào bảng.

Các khóa DML đạt được tự động cho các câu lệnh DML Error! Bookmark not defined.

Các phần trên đã giải thích về những kiểu khóa dữ liệu khác nhau, các chế độ trong đó chúng có thể lưu giữ, khi chúng có thể đạt được; khi chúng đạt được, chúng ngăn chặn những gì. Các phần sau tóm tắt Oracle tự động khóa dữ liệu thay cho các điều hoạt DML khác như thế nào.

Bảng 23-3 tóm tắt thông tin trong các phần sau.

Bảng 23-3 Các khóa những câu lệnh DML đạt được

Các khóa hàng?	Câu lệnh DML	Chế độ khóa bảng
	SELECT ... FROM table	
X	INSERT INTO table ...	RX
X	UPDATE table ...	RX
X	DELETE FROM table ...	RX
X	SELECT ... FROM table ... FOR UPDATE OF ...	RS
	LOCK TABLE table IN ...	

	ROW SHARE MODE	RS
	ROW EXCLUSIVE MODE	RX
	SHARE MODE	S
	SHARE EXCLUSIVE MODE	SRX
	EXCLUSIVE MODE	X
X: độc quyền RX: độc quyền hàng		RS: dùng chung hàng S: dùng chung SRX: độc quyền hàng dùng chung

Tìm các truy vấn mặc định*Error! Bookmark not defined.*

Các truy vấn là các câu lệnh SQL ít có thể gây trở ngại cho các câu lệnh SQL khác vì chúng chỉ đọc dữ liệu. Các câu lệnh INSERT, UPDATE và DELETE có thể có những truy vấn ngầm định. Các truy vấn có các loại câu lệnh sau:
SELECT

INSERT . . . SELECT . . . ;

UPDATE . . . ;

DELETE . . . ;

Chúng không có câu lệnh sau:
SELECT . . . FOR UPDATE OF . . . ;

Các đặc tính sau đúng với tất cả những truy vấn không sử dụng mệnh đề FOR UPDATE:

Một truy vấn không đạt được các khóa dữ liệu. Do đó, các giao dịch khác có thể truy vấn và cập nhật bảng đang được truy vấn, kể cả những hàng riêng biệt đang được truy vấn. Bởi vì những truy vấn không có các mệnh đề FOR UPDATE đạt được các khóa dữ liệu nào, các truy vấn như thế thường được tham chiếu với Oracle bằng những truy vấn không lập khối.

Một truy vấn không phải chờ các khóa dữ liệu nào để giải phóng; nó luôn luôn có thể tiến hành. (Các truy vấn có thể phải chờ các khóa dữ liệu trong một số trường hợp rất cụ thể về các giao dịch được phân phối chưa giải quyết.)

Tìm mặc định INSERT, UPDATE, DELETE và SELECT ... FOR UPDATE*Error! Bookmark not defined.*

Các đặc tính khóa của các câu lệnh INSERT, UPDATE, DELETE và SELECT ... FOR UPDATE như sau:

Giao dịch có câu lệnh DML đạt được các khóa hàng độc quyền trên các hàng được sửa đổi bằng câu lệnh ấy. Các giao dịch khác không thể cập nhật hoặc xóa các hàng được khóa cho tới khi giao dịch khóa cam kết hoặc hồi phục về trước.

Giao dịch có câu lệnh DML không cần đạt được các khóa hàng trên bất kỳ những hàng được lựa bằng một truy vấn con hoặc truy vấn ngầm định, chẳng hạn như truy vấn trong mệnh đề WHERE. Một truy vấn con hoặc truy vấn ngầm định trong câu lệnh DML được bảo đảm là nhất quán như bắt đầu truy vấn ấy và không thấy các kết quả của câu lệnh DML mà nó là thành phần.

Một truy vấn trong một giao dịch có thể thấy những thay đổi được thực hiện bằng các câu lệnh DML trước trong cùng giao dịch, nhưng không thể thấy các thay đổi của các giao dịch khác được bắt đầu sau giao dịch của riêng nó.

Hơn nữa, các khóa hàng độc quyền cần thiết, một giao dịch có câu lệnh DML đạt được ít nhất một khóa bảng độc quyền hàng trên bảng có các hàng bị ảnh hưởng. Nếu giao dịch chứa lưu giữ khóa bảng dùng chung, độc quyền hàng dùng chung hoặc độc quyền, thì khóa bảng độc quyền hàng không đạt được. Nếu giao

dịch chứa lưu giữ một khóa bảng dùng chung hàng, thì Oracle tự động chuyển đổi khóa này thành khóa bảng độc quyền hàng.

Các khóa DDL (Các khóa từ điển)Error! Bookmark not defined.

Một khóa DDL bảo vệ định nghĩa về đối tượng giản đồ (thí dụ, một bảng) khi đối tượng đó được tác động hoặc được tham chiếu bằng một điều hoạt DDL đến. (Gọi lại một câu lệnh DDL cam kết ngầm định giao dịch của nó). Thí dụ, giả sử một người dùng tạo một thủ tục. Thay cho giao dịch câu lệnh đơn của người dùng, Oracle tự động đạt được các khóa DDL cho tất cả những đối tượng giản đồ được tham chiếu trong định nghĩa của thủ tục ấy. Các khóa DDL ngăn chặn các đối tượng được tham chiếu trong thủ tục ấy bị thay đổi hoặc bị bỏ trước khi tính toán thủ tục hoàn tất.

Oracle đạt được khóa từ điển tự động thay cho giao dịch DDL bất kỳ đạt được nó. Người dùng không thể yêu cầu rõ ràng các khóa DDL. Chỉ các đối tượng giản đồ cá biệt được sửa đổi hoặc được tham chiếu, được khóa trong các điều hoạt DDL; toàn bộ từ điển dữ liệu không bao giờ bị khóa.

Các khóa DDL rơi vào ba biệt loại: các khóa DDL độc quyền, các khóa DDL dùng chung và các khóa phân ngữ tách được.

Các khóa DDL độc quyềnError! Bookmark not defined.

Phần lớn các điều hoạt DDL (ngoại trừ đối với các điều hoạt được liệt kê ở phần tiếp theo, "Các khóa DDL dùng chung") đòi hỏi các khóa DDL độc quyền cho tài nguyên để ngăn chặn giao thoa hủy với các điều hoạt DDL khác có thể sửa đổi hoặc tham chiếu cùng đối tượng giản đồ. Thí dụ, điều hoạt DROP TABLE không được cho phép bỏ một bảng khi một điều hoạt ALTER TABLE đang bổ sung một cột vào nó và ngược lại.

Khi đạt được một khóa DDL độc quyền, nếu một khóa DDL khác được lưu giữ trên đối tượng giản đồ bằng một điều hoạt khác rồi, thì khóa đạt được chờ tới khi khóa DDL trước được giải phóng rồi tiến hành.

Các điều hoạt DDL còn đạt được các khóa DML (các khóa dữ liệu) trên đối tượng giản đồ để được sửa đổi.

Các khóa DDL dùng chung Error! Bookmark not defined.

Một số điều hoạt DDL đòi hỏi các khóa DDL dùng chung cho một tài nguyên để ngăn chặn giao thoa hủy với các điều hoạt DDL xung đột, nhưng cho phép tính đồng thời dữ liệu đối với các điều hoạt DDL tương tự. Thí dụ, khi một câu lệnh CREATE PROCEDURE được tiến hành, giao dịch chứa đạt được các khóa DDL dùng chung cho tất cả các bảng được tham chiếu. Các giao dịch khác có thể tạo các thủ tục đồng thời tham chiếu cùng các bảng và do đó đạt được các khóa DDL dùng chung đồng thời, nhưng không giao dịch nào có thể đạt được khóa DDL độc quyền trên bảng được tham chiếu bất kỳ. Không có giao dịch nào có thể thay đổi hoặc bỏ một bảng được tham chiếu. Do đó, một giao dịch lưu giữ một khóa DDL dùng chung được bảo đảm là định nghĩa của đối tượng giản đồ được tham chiếu sẽ vẫn không thay đổi trong thời gian giao dịch.

Một khóa DDL dùng chung đạt được trên đối tượng giản đồ cho các câu lệnh DDL có các lệnh sau: AUDIT, NOAUDIT, COMMENT, CREATE [OR REPLACE] VIEW/ PROCEDURE/PACKAGE/PACKAGE BODY/FUNCTION/ TRIGGER, CREATE SYNONYM và CREATE TABLE (khi không có tham số CLUSTER).

Các khóa phân ngữ tách được Error! Bookmark not defined.

Một câu lệnh SQL (hoặc đơn vị chương trình PL/SQL) trong vùng chia dùng chung lưu giữ một khóa phân ngữ cho mỗi đối tượng giản đồ mà nó tham chiếu. Các khóa phân ngữ đạt được sao cho vùng SQL dùng chung được kết hợp có thể vô hiệu hóa nếu đối tượng tham chiếu được thay đổi hoặc bỏ đi. Xem chương 19, "Quản lý tính phụ thuộc của Oracle". Một khóa phân ngữ cho phép điều hoạt DDL bất kỳ và có thể được tách ra để cho phép các điều hoạt DDL xung đột, vì thế có tên "Khóa phân ngữ tách được".

Một khóa phân ngữ đạt được khi pha phân ngữ tiến hành câu lệnh SQL và lưu giữ bao lâu vùng SQL dùng chung cho câu lệnh đó vẫn ở trong vùng chia dùng chung.

Thời gian của các khóa DDL Error! Bookmark not defined.

Thời gian của khóa DDL phụ thuộc vào kiểu của nó. Các khóa DDL độc quyền và dùng chung kéo dài trong thời gian tiến hành câu lệnh DDL và cam kết tự động. Một khóa phân ngữ duy trì bao lâu câu lệnh SQL được kết hợp còn trong vùng chia dùng chung.

Các khóa DDL và các cụm liên kết Error! Bookmark not defined.

Một điều hoạt DDL trên một cụm liên kết đạt được các khóa DDL độc quyền trên cụm liên kết và trên tất cả các bảng hoặc ảnh chụp nhanh trong một cụm liên kết. Một điều hoạt DDL trên một bảng hoặc ảnh chụp nhanh trong một cụm liên kết đạt được một khóa dùng chung trên cụm liên kết ấy bổ sung vào khóa DDL độc quyền hoặc dùng chung trên bảng hoặc ảnh chụp nhanh. Khóa DDL dùng chung trên cụm liên kết ngăn chặn một điều hoạt khác bỏ cụm liên kết ấy khi điều hoạt thứ nhất tiến hành.

Các mạch gài và các khóa bên trong Error! Bookmark not defined.

Các mạch gài và các khóa bên trong bảo vệ CƠ SỞ DỮ LIỆU bên trong và các cấu trúc bộ nhớ. Cả hai điều này người dùng đều không thể truy xuất được, bởi vì người dùng không cần kiểm soát sự kiện hoặc thời gian của chúng. Thông tin sau sẽ giúp bạn diễn dịch Oracle Enterprise Manager hoặc Server Manager LOCKS và các bộ giám sát LATCHES.

Các mạch gài Error! Bookmark not defined.

Các mạch gài là các cơ chế đơn giản, nối tiếp hóa ở mức thấp để bảo vệ những cấu trúc dữ liệu dùng chung trong vùng toàn cục hệ thống - system global area (SGA). Thí dụ, các mạch gài bảo vệ danh sách những người dùng truy xuất CƠ SỞ DỮ LIỆU và bảo vệ cấu trúc dữ liệu mô tả các khối trong cache vùng đệm. Một tiến trình server hoặc nền đạt được một mạch gài trong thời gian rất ngắn khi điều tác hoặc khảo sát một trong những cấu trúc này. Sự bổ sung các mạch gài phụ thuộc vào hệ điều hành, nhất là liên quan xem tiến trình sẽ chờ một mạch gài không và bao lâu.

Các khóa bên trong Error! Bookmark not defined.

Các khóa bên trong là các cơ chế ở mức cao hơn và phức tạp hơn các mạch gài và phục vụ các mục đích khác nhau.

Các khóa cache từ điển*Error! Bookmark not defined.*

Các khóa này có thời gian rất ngắn và được lưu giữ trên cửa sổ mục nhập vào các cache từ điển khi mục nhập được sửa đổi hoặc sử dụng. Chúng bảo đảm các câu lệnh được phân ngữ không hiểu những định nghĩa đối tượng không nhất quán.

Các khóa cache từ điển có thể được sử dụng chung hoặc độc quyền. Các khóa dùng chung được giải phóng khi phân ngữ ấy hoàn tất. Các khóa độc quyền được giải phóng khi điều hoạt DDL hoàn tất.

Các khóa quản lý sổ nhật ký và tập tin*Error! Bookmark not defined.*

Những khóa này bảo vệ các tập tin khác nhau. Thí dụ, một khóa bảo vệ tập tin điều khiển sao cho chỉ một tiến trình có thể thay đổi nó mỗi lần. Một khóa khác phối hợp sử dụng và tàng trữ các tập tin sổ nhật ký hoạt động. Các tập tin dữ liệu được khóa để bảo đảm nhiều môi trường làm việc kết gắn CƠ SỞ DỮ LIỆU vào chế độ dùng chung hoặc một môi trường làm việc kết gắn nó vào chế độ độc quyền. Bởi vì các khóa tập tin và sổ nhật ký chỉ rõ trạng thái của các tập tin, những khóa này cần được lưu giữ trong một thời gian dài.

Các khóa tập tin và sổ nhật ký đặc biệt quan trọng nếu bạn đang sử dụng Oracle Parallel Server.

Các khóa không gian bảng và đoạn hồi phục về trước*Error! Bookmark not defined.*

Những khóa này bảo vệ các không gian bảng và các đoạn hồi phục về trước. Tham số tất cả môi trường làm việc truy xuất một CƠ SỞ DỮ LIỆU phải phù hợp với một không gian bảng là trực tuyến hoặc ngoại tuyến. Các đoạn hồi phục về trước được khóa sao cho chỉ một môi trường làm việc có thể ghi vào một đoạn.

Khóa dữ liệu rõ ràng (thủ công) Error! Bookmark not defined.

Oracle luôn luôn thực hiện khóa tinh toán để bảo đảm tính đồng thời dữ liệu, tính nguyên vẹn dữ liệu và tính nhất quán đọc mức câu lệnh. Tuy nhiên, bạn có thể chạy đè các cơ chế khóa mặc

định của Oracle. Chạy đè khóa mặc định hữu dụng trong các trường hợp như sau:
Các ứng dụng đòi hỏi tính nhất quán đọc ở mức giao dịch hoặc "đọc lặp lại được". Nói cách khác, những truy vấn trong chúng phải tạo ra dữ liệu nhất quán trong thời gian giao dịch, không phản ảnh những thay đổi do các giao dịch khác. Bạn có thể đạt được tính nhất quán đọc mức giao dịch bằng cách sử dụng khóa rõ ràng, giao dịch chỉ đọc, các giao dịch nối tiếp hóa hoặc bằng cách chạy đè khóa mặc định.
Ứng dụng đòi hỏi giao dịch sử dụng truy xuất độc quyền một tài nguyên sao cho giao dịch ấy không phải chờ các giao dịch khác hoàn tất.

Khóa tự động của Oracle có thể chạy đè ở hai mức:

Giao dịch	Các giao dịch có những câu lệnh SQL sau chạy đè khóa mặc định của Oracle: Lệnh SET TRANSACTION ISOLATION LEVEL Lệnh LOCK TABLE (khóa một bảng hoặc khi được sử dụng với các khung nhìn, nằm dưới các bảng cơ sở) Lệnh SELECT... FOR UPDATE Các khóa đạt được bằng những câu lệnh này được giải phóng sau khi giao dịch ấy cam kết hoặc hồi phục về trước.
Phiên	Một phiên có thể xác lập mức tách lập giao dịch cần thiết với lệnh ALTER SESSION.

Lưu ý:
Nếu khóa mặc định của Oracle được chạy đè ở mức bất kỳ, thì người điều hành CƠ SỞ DỮ LIỆU sẽ bảo đảm các thủ tục chạy đè chạy đè điều hoạt chính xác. Các thủ tục khóa phải thỏa đáng

tiêu chuẩn sau: tính nguyên vẹn dữ liệu được bảo đảm, tính đồng thời dữ liệu được chấp nhận và các bế tắc không thể xảy ra hoặc được điều quản thích hợp.

Thông tin bổ sung:
Xem Oracle8 SQL Reference về các câu lệnh LOCK TABLE và SELECT ... FOR UPDATE của SQL.

Các thí dụ về tính đồng thời khi khóa thủ công Error! Bookmark not defined.

Hình minh họa sau trình bày Oracle duy trì tính đồng thời, tính nguyên vẹn dữ liệu và tính nhất quán như thế nào khi LOCK TABLE và SELECT với các câu lệnh của mệnh đề FOR UPDATE được sử dụng.

Lưu ý:
Để rút gọn, không có trong văn bản thông điệp ORA-00054 ("tài nguyên đang hoạt động và có NOWAIT được ấn định"). Văn bản người dùng nhập có chữ đậm.

Giao dịch 1	Thời điểm	Giao dịch 2
LOCK TABLE scott.dept IN ROW SHARE MODE; Câu lệnh được xử lý	1	
	2	**DROP TABLE scott.dept;** DROP TABLE scott.dept * ORA-00054 *(khóa DDL độc quyền không thể xảy ra vì khóa bảng của T1)*
	3	**LOCK TABLE scott.dept**

		IN EXCLUSIVE MODE NOWAIT; ORA-00054
	4	**SELECT LOC** **FROM scott.dept** **WHERE deptno = 20** **FOR UPDATE OF loc;** LOC - - - - - - DALLAS 1 hàng được lựa
UPDATE scott.dept **SET loc = 'NEW YORK'** **WHERE deptno = 20;** *(chờ bởi vì T2 đã khóa cùng hàng)*	5	
	6	**ROLLBACK;** *(giải phóng các khóa hàng)*
1 hàng được xử lý **ROLLBACK;**	7	
LOCK TABLE **scott.dept** **IN ROW EXCLUSIVE MODE;** Câu lệnh được xử lý.	8	
	9	**LOCK TABLE scott.dept** **IN EXCLUSIVE MODE** **NOWAIT;** ORA-00054
	10	**LOCK TABLE scott.dept** **IN SHARE ROW** **EXCLUSIVE** **MODE NOWAIT;** ORA-00054
	11	**LOCK TABLE scott.dept** **IN SHARE ROW**

		EXCLUSIVE **MODE NOWAIT;** ORA-00054
	12	**UPDATE scott.dept** **SET loc = 'NEW YORK'** **WHERE deptno = 20;** 1 hàng được xử lý.
	13	**ROLLBACK;**
SELECT loc **FROM scott.dept** **WHERE deptno = 20** **FOR UPDATE OF loc;** LOC - - - - - - DALLAS 1 hàng được lựa.	14	
	15	**UPDATE scott.dept** **SET loc = 'NEW YORK'** **WHERE deptno = 20;** *(chờ vì T1 đã khóa cùng hàng)*
ROLLBACK;	16	
	17	1 hàng được xử lý. *(các khóa xung đột được giải phóng)* **ROLLBACK;**
LOCK TABLE **scott.dept** **IN ROW SHARE MODE** Câu lệnh được xử lý	18	
	19	**LOCK TABLE scott.dept** **IN EXCLUSIVE MODE** **NOWAIT;** ORA-00054
	20	**LOCK TABLE scott.dept**

		IN SHARE ROW EXCLUSIVE MODE NOWAIT; ORA-00054
	21	**LOCK TABLE scott.dept IN SHARE MODE;** Câu lệnh được xử lý.
	22	**SELECT loc FROM scott.dept WHERE deptno = 20;** LOC - - - - - - DALLAS 1 hàng được lựa.
	23	**SELECT loc FROM scott.dept WHERE deptno = 20 FOR UPDATE OF loc;** LOC - - - - - - DALLAS 1 hàng được lựa.
	24	**UPDATE scott.dept SET loc = 'NEW YORK' WHERE deptno = 20;** *(chờ vì T1 lưu giữ khóa bảng xung đột)*
ROLLBACK;	25	
	26	1 hàng được xử lý. *(khóa bảng xung đột được giải phóng)* **ROLLBACK;**
LOCK TABLE scott.dept IN SHARE ROW EXCLUSIVE MODE; Câu lệnh được xử lý.	27	

	28	**LOCK TABLE scott.dept** **IN EXCLUSIVE MODE** **NOWAIT;** ORA-00054
	29	**LOCK TABLE scott.dept** **IN SHARE ROW** **EXCLUSIVE MODE** **NOWAIT;** ORA-00054
	30	**LOCK TABLE scott.dept** **IN SHARE MODE** **NOWAIT;** ORA-00054
	31	**LOCK TABLE scott.dept** **IN ROW EXCLUSIVE** **MODE NOWAIT;** ORA-00054
	32	**LOCK TABLE scott.dept** **IN SHARE MODE** **NOWAIT;** ORA-00054
	33	**SELECT loc** **FROM scott.dept** **WHERE deptno = 20;** LOC - - - - - - DALLAS 1 hàng được lựa.
	34	**SELECT loc** **FROM scott.dept** **WHERE deptno = 20** **FOR UPDATE OF loc;** LOC - - - - - - DALLAS 1 hàng được lựa.
	35	**UPDATE scott.dept**

		SET loc = 'NEW YORK' **WHERE deptno = 20;** *(chờ vì T1 lưu giữ khóa bảng xung đột)*
UPDATE scott.dept **SET loc = 'NEW YORK'** **WHERE deptno = 20;** *(chờ vì T2 đã khóa cùng các hàng)*	36	*(bế tắc)*
Hủy điều hoạt **ROLLBACK;**	37	
	38	1 hàng được xử lý.
LOCK TABLE **scott.dept** **IN EXCLUSIVE MODE;**	39	
	40	**LOCK TABLE scott.dept** **IN EXCLUSIVE MODE;** ORA-00054
	41	**LOCK TABLE scott.dept** **IN ROW EXCLUSIVE** **MODE** **NOWAIT;** ORA-00054
	42	**LOCK TABLE scott.dept** **IN SHARE MODE;** ORA-00054
	43	**LOCK TABLE scott.dept** **IN ROW EXCLUSIVE** **MODE NOWAIT;** ORA-00054
	44	**LOCK TABLE scott.dept** **IN ROW SHARE MODE** **NOWAIT;** ORA-00054
	45	**SELECT loc** **FROM scott.dept**

		WHERE deptno = 20; LOC - - - - - - DALLAS 1 hàng được lựa.
	46	**SELECT loc** **FROM scott.dept** **WHERE deptno = 20** **FOR UPDATE OF loc;** *(chờ vì T1 có khóa bảng xung đột)*
UPDATE scott.dept **SET deptno = 30** **WHERE deptno = 20;** 1 hàng được xử lý.	47	
COMMIT;	48	
	49	Không hàng nào được lựa. *(T1 giải phóng khóa xung đột)*
SET TRANSACTION **READ ONLY;**	50	
SELECT loc **FROM scott.dept** **WHERE deptno = 10;** LOC - - - - - - BOSTON	51	
	52	**UPDATE scott.dept** **SET loc = 'NEW YORK'** **WHERE deptno = 10;** 1 hàng được xử lý.
SELECT loc **FROM scott.dept** **WHERE deptno = 10;** LOC - - - - - - BOSTON	53	

(T1 không thấy dữ liệu chưa cam kết)		
	54	**COMMIT;**
SELECT loc **FROM scott.dept** **WHERE deptno = 10;** LOC - - - - - - *(cùng các kết quả được thấy ngay cả sau giao dịch T2 cam kết)*	55	
COMMIT;	56	
SELECT loc **FROM scott.dept** **WHERE deptno = 10;** LOC - - - - - - NEW YORK *(thấy được dữ liệu cam kết)*	57	

Các dịch vụ Oracle Lock Management

Với các dịch vụ Oracle Lock Management, một nhà phát triển ứng dụng có thể có những câu lệnh trong các khối PL/SQL để
yêu cầu khóa của một kiểu riêng biệt
cung cấp khóa một tên duy nhất có thể nhận biết trong một thủ tục khác ở cùng môi trường làm việc hoặc khác môi trường làm việc
thay đổi kiểu khóa
giải phóng khóa

Bởi vì khóa người dùng dành riêng giống như một khóa của Oracle, nó đều có tính năng khóa Oracle có phát hiện bế tắc. Các khóa người dùng xung đột với các khóa Oracle, vì chúng được nhận dạng với tiền tố "UL".

Các dịch vụ Oracle Lock Management có thể sử dụng bằng những thủ tục trong bộ phần mềm DBMS_LOCK.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Application Developer về các dịc vụ Oracle Lock Management.

Chương 24

Tính nguyên vẹn dữ liệu

Chương này giải thích cách sử dụng những hạn định tính nguyên vẹn dữ liệu để tăng cường những nguyên tắc hoạt động với CSDL và ngăn chặn nhập thông tin bất hợp lệ vào các bảng, gồm các chủ đề:

- **Định nghĩa của tính nguyên vẹn dữ liệu**
- **Giới thiệu về các hạn định tính nguyên vẹn dữ liệu**
- **Các kiểu của những hạn định tính nguyên vẹn**
- **Kiểm hạn định trì hoãn**
- **Các hạn định cho phép, không cho phép và vô hiệu hóa cho phép**

Thông tin bổ sung:
Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle về các hạn định tính nguyên vẹn trong môi trường đó.

Định nghĩa về tính nguyên vẹn

Quan trọng là dữ liệu gắn liền với những nguyên tắc được xác định trước, như được xác định bởi người điều hành CSDL hoặc

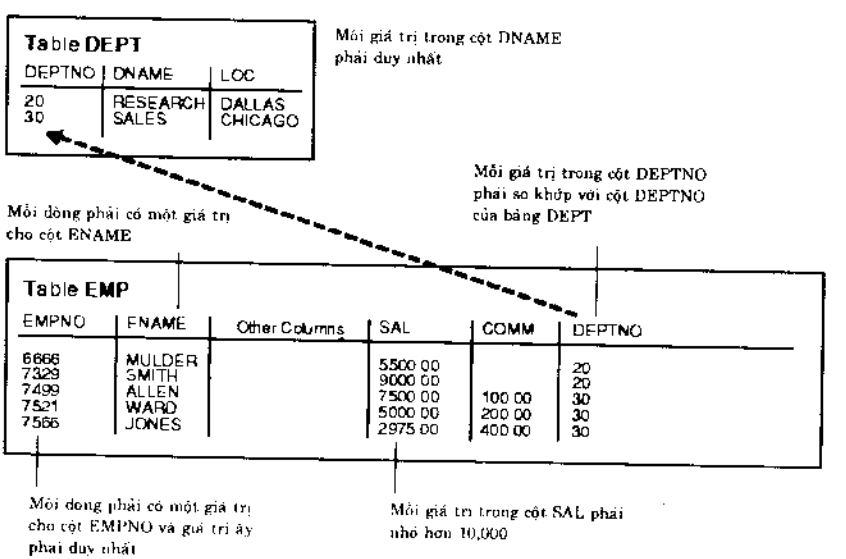
nhà phát triển ứng dụng. Thí dụ về tính nguyên vẹn dữ liệu, xét các bảng EMP và DEPT, những nguyên tắc hoạt động cung cấp thông tin trong mỗi bảng, như được minh họa ở Hình 24-1.

Lưu ý một số cột trong bảng có những nguyên tắc riêng hạn định dữ liệu chứa trong chúng.

Các kiểu tính nguyên vẹn dữ liệu

Phần này mô tả những nguyên tắc được áp dụng vào các cột của bảng để tăng cường những kiểu tính nguyên vẹn dữ liệu khác nhau.

Hình 24-1 Các thí dụ về tính nguyên vẹn dữ liệu



Một giá trị rỗng là một nguyên tắc được xác định trên một cột đơn cho phép hoặc vô hiệu hóa các lệnh chèn, cập nhật các hàng cho một giá trị rỗng (không có giá trị) trong cột đó.

Các giá trị cột duy nhất

Một giá trị duy nhất được xác định trên một cột (hoặc tập hợp các cột) cho phép chèn, cập nhật một hàng chỉ nếu nó có một giá trị duy nhất trong cột đó (hoặc tập hợp của các cột).

Các giá trị khóa chính

Giá trị một khóa chính được xác định trên một khóa (một cột hoặc tập hợp các cột) xác định mỗi hàng trong bảng có thể được nhận dạng duy nhất bằng các giá trị trong khóa ấy.

Tính nguyên vẹn tham chiếu

Một nguyên tắc được xác định trên một khóa (một cột hoặc tập hợp các cột) trong một bảng bảo đảm là những giá trị trong khóa đó so khớp với những giá trị trong một khóa ở bảng có liên quan (giá trị được tham chiếu).

Tính nguyên vẹn tham chiếu còn có những nguyên tắc chỉ những kiểu điều tác dữ liệu nào được cho phép trên các giá trị được tham chiếu và những hoạt động này tác động đến những giá trị phụ thuộc như thế nào. Những nguyên tắc được kết hợp với tính nguyên vẹn tham chiếu là:

Restrict	Không cho phép cập nhật hoặc xóa dữ liệu được tham chiếu.
Xác lập thành Null	Khi dữ liệu tham chiếu được cập nhật hoặc xóa, tất cả dữ liệu phụ thuộc kết hợp được xác lập thành NULL.
Xác lập thành mặc định Default	Khi dữ liệu tham chiếu được cập nhật hoặc xóa, tất cả dữ liệu phụ thuộc kết hợp được xác lập thành giá trị mặc định.
Cascade	Khi dữ liệu tham chiếu được cập nhật, tất cả dữ liệu phụ thuộc kết hợp được cập nhật tương ứng khi hàng tham chiếu được xóa, tất cả các hàng

	phụ thuộc kết hợp được xóa.
No Action	Không cho phép cập nhật hoặc xóa dữ liệu tham chiếu. Điều này khác với RESTRICT trong đó nó được kiểm ở cuối câu lệnh hoặc ở cuối giao dịch nếu hạn định được trì hoãn. (Oracle sử dụng No Action làm hoạt động mặc định của nó.)

Kiểm tính nguyên vẹn phức tạp

Kiểm tính nguyên vẹn phức tạp là nguyên tắc người dùng xác định cho một cột (hoặc tập hợp của các cột) cho phép hoặc không cho phép chèn, cập nhật hoặc xóa một hàng dựa trên giá trị nó chứa đối với cột ấy (hoặc tập hợp các cột).

Oracle tăng cường tính nguyên vẹn dữ liệu như thế nào

Oracle cho phép bạn xác định và tăng cường mỗi kiểu nguyên tắc tính nguyên vẹn dữ liệu được xác định trong phần trước. Phần lớn những nguyên tắc này sử dụng các hạn định tính nguyên vẹn hoặc các bộ kích khởi CSDL.

Những hạn định tính nguyên vẹn

Hạn định tính nguyên vẹn dữ liệu là phương pháp khai báo xác định một nguyên tắc cho một cột của bảng. Oracle hỗ trợ các hạn định sau:

NOT NULL hạn định cho những nguyên tắc được kết hợp với các giá trị rỗng trong một cột

Khóa UNIQUE hạn định cho nguyên tắc được kết hợp với những giá trị cột duy nhất

PRIMARY KEY hạn định cho nguyên tắc được kết hợp với những giá trị định danh chính

FOREIGN KEY hạn định cho những nguyên tắc được kết hợp với tính nguyên vẹn tham chiếu. Oracle hiện hành hỗ trợ sử dụng các hạn định tính nguyên vẹn dữ liệu FOREIGN KEY để xác định các hoạt động tính nguyên vẹn tham chiếu, gồm

Cập nhật hoặc xóa No Action
Xóa CASCADE
CHECK hạn định cho những nguyên tắc tính nguyên vẹn dữ liệu phức tạp

Lưu ý:
Bạn không thể tăng cường tính nguyên vẹn tham chiếu bằng cách sử dụng các hạn định tính nguyên vẹn khai báo nếu các bảng con và cha ở trên các nút khác nhau của CSDL được phân phối. Tuy nhiên, bạn có thể tăng cường tính nguyên vẹn tham chiếu trong CSDL được phân phối sử dụng các bộ kích khởi CSDL (xem phần tiếp theo).

Các bộ kích khởi CSDL

Oracle còn cho bạn tăng cường những nguyên tắc tính nguyên vẹn dữ liệu bằng cách giải quyết không khai báo sử dụng các bộ kích khởi CSDL (các thủ tục CSDL định sẵn tự động thu hồi các điều hoạt chèn, cập nhật hoặc xóa). Xem thêm các thí dụ về những bộ kích khởi CSDL được sử dụng để tăng cường tính nguyên vẹn dữ liệu ở chương 18, "Các bộ kích khởi CSDL".

Giới thiệu về những hạn định tính nguyên vẹn dữ liệu

Oracle sử dụng các hạn định tính nguyên vẹn dữ liệu để ngăn chặn nhập dữ liệu bất hợp lệ vào các bảng cơ sở của CSDL. Bảng có thể xác định các hạn định tính nguyên vẹn dữ liệu để tăng cường những nguyên tắc hoạt động bạn muốn kết hợp với thông tin trong CSDL. Nếu bất kỳ các kết quả của tiến hành câu lệnh DML xâm phạm hạn định tính nguyên vẹn dữ liệu, Oracle hồi phục về trước câu lệnh ấy và cho trở lại một lỗi.

Lưu ý:
Các điều hoạt trên những khung nhìn (và các từ đồng nghĩa cho các bảng) phụ thuộc vào các hạn định tính nguyên vẹn dữ liệu

được xác định trên các bảng cơ sở.

Thí dụ, giả sử bạn xác định một hạn định tính nguyên vẹn cho cột SAL của bảng EMP. Hạn định tính nguyên vẹn này tăng cường nguyên tắc không có hàng nào trong bảng này có thể có trị số lớn hơn 10.000 trong các cột này. Nếu câu lệnh INSERT hoặc UPDATE cố xâm phạm hạn định tính nguyên vẹn này, Oracle hồi phục về trước câu lệnh ấy và cho trở lại một thông báo lỗi.

Các hạn định tính nguyên vẹn được bổ sung trong Oracle tuân thủ hoàn toàn những tiêu chuẩn ANSI X3.135-1989 và ISO 9075-1989.

Các ưu điểm về những hạn định tính nguyên vẹn

Phần này mô tả một số ưu điểm tính nguyên vẹn có ưu điểm thay thế lẫn nhau, gồm:

- tăng cường những nguyên tắc hoạt động bằng mã của ứng dụng CSDL
- sử dụng những thủ tục định sẵn để kiểm soát truy xuất dữ liệu hoàn toàn
- tăng cường những nguyên tắc hoạt động với các thủ tục CSDL định sẵn có kích hoạt (xem chương 18, "Các bộ kích khởi CSDL")

Giảm bớt khai báo

Bạn xác định các hạn định tính nguyên vẹn bằng cách sử dụng các câu lệnh SQL. Khi bạn xác định hoặc sửa đổi một bảng, không cần lập trình thêm. Các câu lệnh SQL rất dễ viết, loại trừ các lỗi lập trình và Oracle điều khiển tính năng của chúng. Đối với các lý do này, các hạn định tính nguyên vẹn thích hợp với mã ứng dụng và các bộ kích khởi CSDL. Các giải quyết khai báo cũng tốt hơn sử dụng các thủ tục định sẵn, bởi vì giải pháp thủ

tục định sẵn đối với tính nguyên vẹn dữ liệu điều khiển truy xuất dữ liệu, nhưng các hạn định tính nguyên vẹn không loại trừ tính linh hoạt của truy xuất dữ liệu không định trước.

Các nguyên tắc trung tâm

Các hạn định tài nguyên được xác định cho các bảng (không cho ứng dụng) và được lưu trữ trong từ điển dữ liệu. Dữ liệu bất kỳ được nhập bằng một ứng dụng phải gắn liền với cùng các hạn định được kết hợp với bảng. Bằng cách di chuyển những nguyên tắc hoạt động từ mã ứng dụng tới các hạn định tính nguyên vẹn tập trung, các bảng của một CSDL được bảo đảm đối với dữ liệu hợp lệ nào đó, bất kể ứng dụng CSDL nào điều tác thông tin ấy. Các thủ tục định sẵn không thể cung cấp ưu điểm giống như các nguyên tắc tập trung với bảng. Các bộ kích khởi CSDL có thể cung cấp ưu điểm này, nhưng tính phức tạp bổ sung còn lớn hơn cách giải quyết khai báo được sử dụng cho các hạn định tính nguyên vẹn.

Năng suất phát triển ứng dụng tối đa

Nếu một nguyên tắc hoạt động được tăng cường bằng một hạn định tính nguyên vẹn thay đổi, người điều hành chỉ cần thay đổi hạn định tính nguyên vẹn và tất cả ứng dụng tự động gắn liền với hạn định được sửa đổi. Trái lại, nếu nguyên tắc hoạt động đã được tăng cường mã của mỗi ứng dụng CSDL, các nhà phát triển sẽ phải sửa đổi tất cả mã nguồn ứng dụng và biên dịch lại, gỡ rối và trắc nghiệm các ứng dụng được sửa đổi.

Phản hồi của người dùng trực tiếp

Oracle lưu trữ thông tin riêng biệt về mỗi hạn định tính nguyên vẹn trong từ điển dữ liệu. Bạn có thể thiết kế các ứng dụng CSDL sử dụng thông tin này để cung cấp phản hồi của người dùng trực tiếp về các xâm phạm hạn định tính nguyên vẹn, thậm chí trước khi Oracle tiến hành và kiểm câu lệnh SQL. Thí dụ,

ứng dụng SQL*Forms có thể sử dụng các định nghĩa hạn định tính nguyên vẹn được lưu trữ trong từ điển dữ liệu để kiểm các xâm phạm bằng những giá trị được nhập vào các trường của một biểu mẫu, ngay cả trước khi ứng dụng ấy phát hành một câu lệnh.

Hiệu suất cao

Ngữ nghĩa của các khai báo hạn định tính nguyên vẹn được xác định rõ ràng và các tối ưu hóa hiệu suất được bổ sung nguyên tắc khai báo cụ thể. Bộ tối ưu hóa truy vấn Oracle có thể sử dụng các khai báo để tìm hiểu về dữ liệu tăng hiệu suất truy vấn tổng quát. (Ngoài ra, đưa những nguyên tắc tính nguyên vẹn ra khỏi mã ứng dụng và các bộ kích khởi CSDL bảo đảm các kiểm tra chỉ được thực hiện khi cần.)

Tính linh hoạt đối với tải dữ liệu và nhận dạng các xâm phạm tính nguyên vẹn

Bạn có thể vô hiệu hóa những hạn định tính nguyên vẹn tạm thời sao cho những khối lượng dữ liệu lớn có thể được tải mà không có phần điều hành chung về kiểm hạn định. Khi tải dữ liệu hoàn tất, bạn dễ dàng có thể tự động báo cáo bất kỳ các hàng mới nào xâm phạm những hạn định tính nguyên vẹn đối với một bảng ngoại lệ tách riêng.

Phí tổn hiệu suất của các hạn định tính nguyên vẹn

Các ưu điểm tăng cường những nguyên tắc tính nguyên vẹn dữ liệu xảy ra không mất đi phần nào hiệu suất. Nói chung, "phí tổn" cao nhất để có hạn định tính nguyên vẹn bằng với tiến hành câu lệnh SQL để thẩm định hạn định.

Các kiểu hạn định tính nguyên vẹn

Bạn có thể sử dụng các hạn định tính nguyên vẹn sau để khai thác những giới hạn về nhập các giá trị cột:

Các hạn định tính nguyên vẹn NOT NULL
Các hạn định tính nguyên vẹn khóa UNIQUE
Các hạn định tính nguyên vẹn PRIMARY KEY
Các hạn định tính nguyên vẹn FOREIGN KEY (tham chiếu)
Các hạn định tính nguyên vẹn CHECK

Các hạn định tính nguyên vẹn NOT NULL

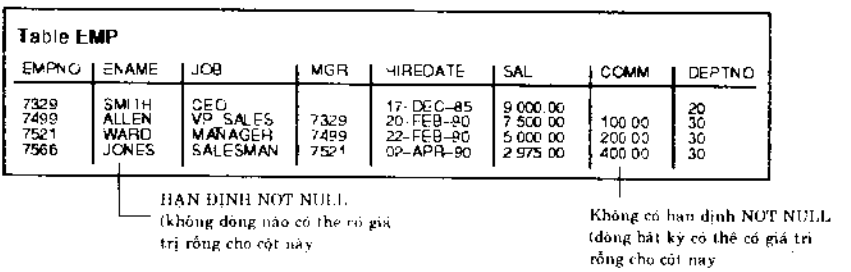
Theo mặc định, tất cả các cột trong một bảng cho phép các giá trị rỗng. Hạn định NOT NULL đòi hỏi một cột của bảng không chứa giá trị rỗng nào. Thí dụ, bạn có thể xác định hạn định NOT NULL để đòi hỏi một giá trị được nhập vào cột ENAME cho mọi hàng của bảng EMP.

Hình 23-2 minh họa hạn định tính nguyên vẹn NOT NULL.

Các hạn định tính nguyên vẹn khóa UNIQUE

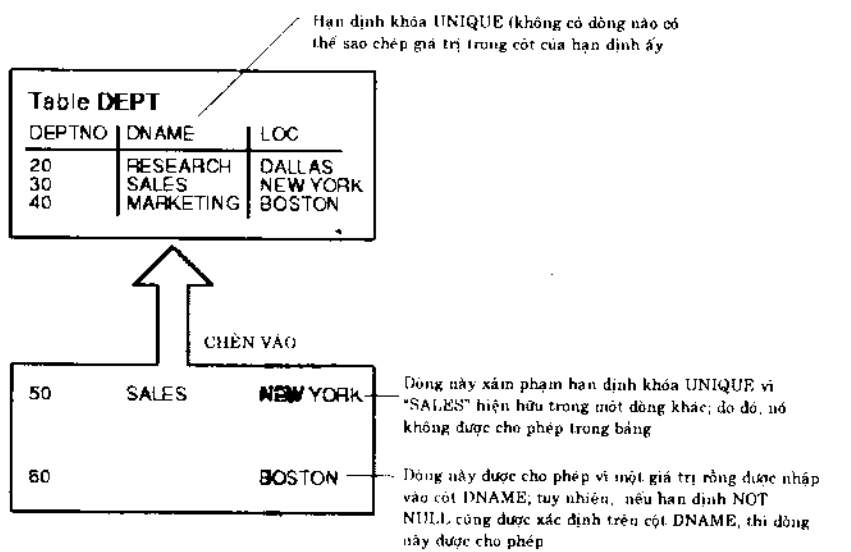
Một hạn định tính nguyên vẹn khóa UNIQUE đòi hỏi mọi giá trị trong một cột hoặc tập hợp các cột (khóa) phải duy nhất - nghĩa là không có hai hàng của một bảng có các giá trị sao chép trong một cột hoặc tập hợp các cột được xác định.

Hình 24-2 Hạn định tính nguyên vẹn NOT NULL



Thí dụ, ở Hình 24-3, một hạn định khóa UNIQUE được xác định trên cột DNAME của bảng DEPT không cho phép các hàng có các tên bộ phận (của một công ty) sao chép.

Hình 24-3 Một hạn định khóa UNIQUE

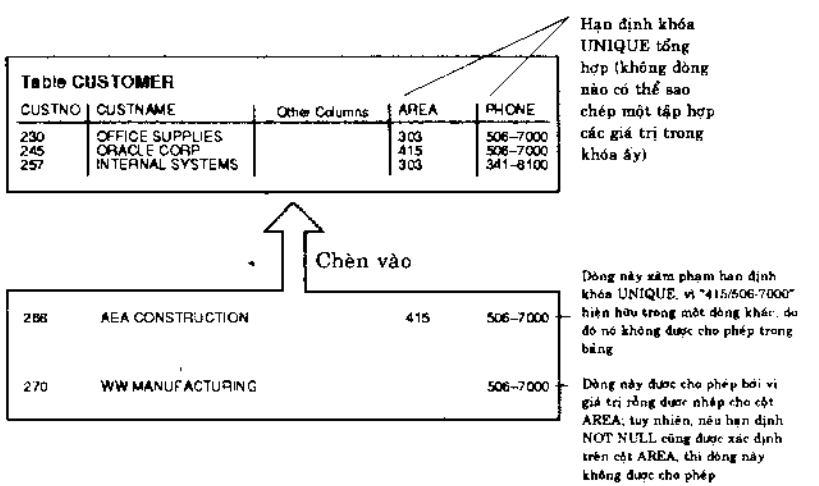


Các khóa Unique

Cột (hoặc tập hợp các cột) có trong định nghĩa của hạn định khóa UNIQUE được gọi là khóa duy nhất. Thuật ngữ "khóa duy nhất" thường được sử dụng không chính xác như một từ đồng nghĩa đối với các thuật ngữ "hạn định khóa UNIQUE " hoặc "chỉ mục UNIQUE "; tuy nhiên, lưu ý thuật ngữ "khóa" chỉ tham chiếu cột hoặc tập hợp các cột được sử dụng trong định nghĩa của hạn định tính nguyên vẹn.

Nếu khóa UNIQUE gồm hơn một cột, thì nhóm cột đó được coi là khóa duy nhất tổng hợp. Thí dụ, ở Hình 24-4, bảng CUSTOMER có hạn định khóa duy nhất được xác định trên khóa duy nhất tổng hợp: các cột AREA và PHONE.

Hình 24-4 Một khóa UNIQUE tổng hợp



Hạn định khóa UNIQUE này cho bạn nhập mã vùng và số điện thoại nhiều lần, nhưng tổ hợp của mã vùng cho trước và số điện thoại cho trước không thể được sao chép trong bảng. Điều này loại trừ sao chép không cố ý số điện thoại.

Các chỉ mục và các hạn định khóa UNIQUE

Oracle tăng cường các hạn định tính nguyên vẹn duy nhất với các chỉ mục. (ở Hình 24-4, Oracle tăng cường hạn định khóa UNIQUE bằng cách ngầm định tạo một chỉ mục duy nhất trên khóa duy nhất tổng hợp). Do đó, các hạn định khóa UNIQUE tổng hợp có cùng các giới hạn ảnh hưởng trên các chỉ mục tổng hợp: tối đa 32 cột có thể cấu tạo một khóa duy nhất tổng hợp và tổng kích thước (tính bằng byte) của một giá trị khóa không thể vượt nửa kích thước khối của CSDL được kết hợp. Nếu một chỉ mục có sẵn có thể sử dụng khi một hạn định khóa duy nhất được tạo, hạn định ấy sẽ sử dụng chỉ mục đó thay vì tạo một chỉ mục mới ngầm định.

Kết hợp khóa UNIQUE và các hạn định tính nguyên vẹn NOT NULL

Ở Hình 24-3 và Hình 24-4, các hạn định khóa UNIQUE cho phép nhập các giá trị rỗng trừ khi bạn cũng xác định các hạn định NOT NULL cho cùng các cột ấy. Thực ra, bất kỳ những số hàng có thể có các giá trị rỗng cho các cột không có hạn định NOT NULL bởi vì các giá trị rỗng không được xét bằng với bất kỳ cột nào. Một giá trị rỗng trong một cột (hoặc trong tất cả các cột của một khóa UNIQUE tổng hợp) luôn luôn thỏa mãn hạn định khóa UNIQUE.

Các cột với cả khóa duy nhất và các hạn định tính nguyên vẹn NOT NULL thường sử dụng. Tổ hợp này buộc người dùng nhập các giá trị trong khóa duy nhất và cũng loại trừ khả năng dữ liệu của hàng mới bất kỳ sẽ xung đột với một dữ liệu của hàng có sẵn.

Lưu ý:
Bởi vì cơ chế tìm các hạn định UNIQUE trên hơn một cột, bạn không thể có những giá trị giống nhau trong các cột không có giá trị rỗng của một hạn định khóa UNIQUE tổng hợp có giá

trị rỗng một phần.

Các hạn định tính nguyên vẹn PRIMARY KEY

Mỗi bảng trong CSDL có tối đa một hạn định PRIMARY KEY. Các giá trị trong nhóm của một hoặc nhiều cột phụ thuộc vào hạn định này cấu tạo dấu định danh duy nhất của hàng ấy. Thực ra, mỗi hàng được các giá trị khóa chính của nó đặt tên.

Bổ sung của Oracle về hạn định tính nguyên vẹn PRIMARY KEY bảo đảm cả hai điều sau đều đúng:
Không có hai hàng của một bảng có các giá trị sao chép trong một cột hoặc tập hợp các cột xác định.
Các cột khóa chính không cho phép các giá trị rỗng (nghĩa là một giá trị phải tồn tại cho các cột khóa chính trong mỗi hàng).

Các khóa chính

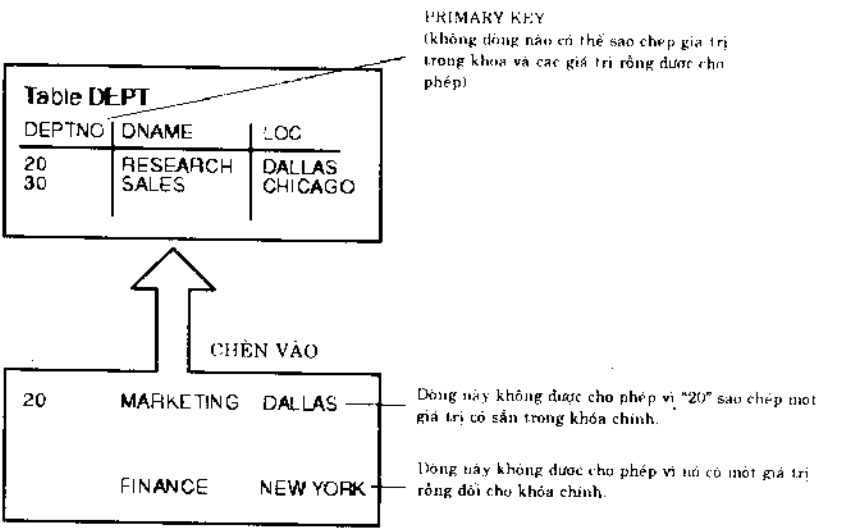
Cột (hoặc tập hợp các cột) có trong định nghĩa của hạn định tính nguyên vẹn PRIMARY KEY của một bảng được gọi là khóa chính. Dù nó không đòi hỏi, nhưng mọi bảng phải có một khóa chính sao cho
mỗi hàng trong bảng có thể được nhận dạng duy nhất
không có các hàng sao chép có sẵn trong bảng

Hình 24-5 minh họa hạn định PRIMARY KEY trong bảng DEPT và các thí dụ về các hàng xâm phạm hạn định ấy.

Các chỉ mục và các hạn định PRIMARY KEY

Oracle tăng cường tất cả các hạn định PRIMARY KEY sử dụng các chỉ mục. Ở Hình 24-5, hạn định khóa chính được tạo cho cột DEPTNO được tăng cường bằng
tạo ngầm định một chỉ mục duy nhất trên cột đó
tạo ngầm định một hạn định NOT NULL cho cột đó

Hình 24-5 Một hạn định khóa chính



Oracle tăng cường các hạn định khóa chính sử dụng các chỉ mục và các hạn định khóa chính tổng hợp được giới hạn là 32 cột, giới hạn giống như ảnh hưởng trên các chỉ mục tổng hợp. Tên của chỉ mục ấy giống như tên của hạn định đó. Hơn nữa, bạn có thể ấn định các tùy chọn lưu trữ cho chỉ mục ấy bằng mệnh đề ENABLE trong câu lệnh CREATE TABLE hoặc ALTER TABLE được sử dụng để tạo hạn định ấy. Nếu chỉ mục có sẵn có thể sử dụng khi một hạn định khóa chính được tạo, hạn định khóa chính sẽ sử dụng cập nhật đó thay vì tạo một chỉ mục mới ngầm định.

Các hạn định tính nguyên vẹn FOREIGN KEY (tham chiếu)

Các bảng khác nhau trong CSDL quan hệ có thể có liên quan bằng các cột chung và những nguyên tắc chi phối mối quan hệ

của các cột phải được duy trì. Những nguyên tắc tính nguyên vẹn tham chiếu bảo đảm những mối quan hệ này được lưu giữ.

Nhiều thuật ngữ được kết hợp với những hạn định tính nguyên vẹn tham chiếu:

Khóa ngoại	Cột hoặc tập hợp các cột có trong định nghĩa của hạn định tính nguyên vẹn tham chiếu, tham chiếu khóa được tham chiếu (xem ở dưới).
Khóa tham chiếu	Khóa duy nhất hoặc khóa chính của cùng bảng hoặc khác bảng được tham chiếu bằng một khóa ngoại.
Các bảng con và bảng phụ thuộc	Bảng có khóa ngoại. Do đó, nó là bảng phụ thuộc vào những giá trị hiệu hữu trong khóa chính hoặc duy nhất.
Bảng cha hoặc bảng được tham chiếu	Bảng được tham chiếu bằng khóa ngoại của bảng con. Nó là khóa được tham chiếu của bảng này để xác định xem các điều hoạt chèn hoặc cập nhật riêng biệt có được cho phép trong bảng con không.

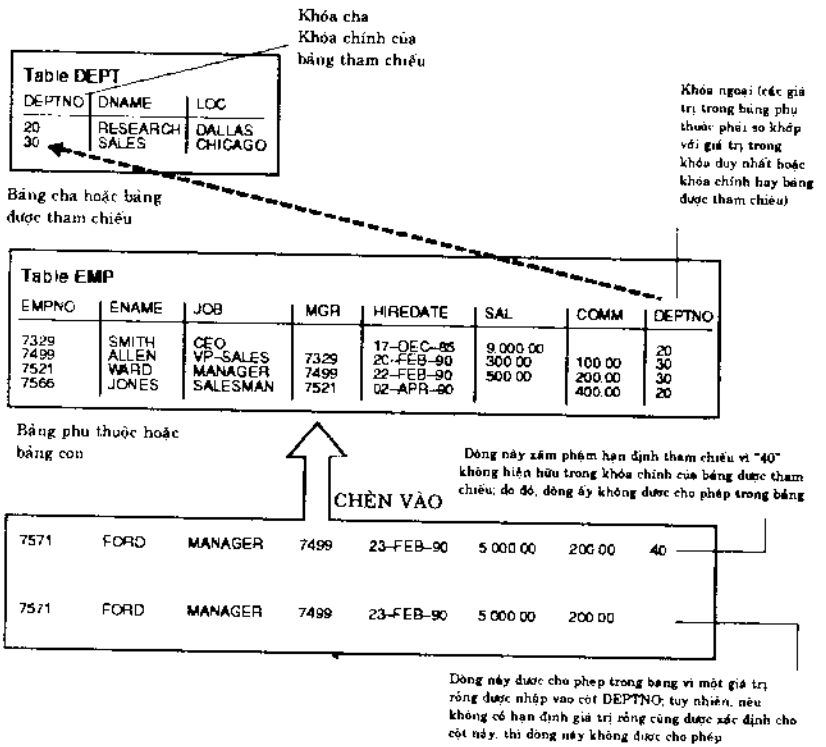
Một hạn định tính nguyên vẹn tham chiếu đòi hỏi cho mỗi hàng của một bảng, giá trị trong khóa ngoại so khớp với giá trị trong khóa cha.

Hình 24-6 trình bày một khóa ngoại được xác định trên cột DEPTNO của bảng EMP. Nó bảo đảm mọi giá trị của bảng DEPT (cả cột DEPTNO). Do đó, số bộ phận (của công ty) không bị sai sót có thể tồn tại trong cột DEPTNO của bảng EMP bảng.

Các khóa ngoại có thể gồm nhiều cột. Tuy nhiên, một khóa ngoại tổng hợp phải tham chiếu một khóa chính tổng hợp hoặc duy nhất với cùng số cột và cùng kiểu dữ liệu. Bởi vì các khóa chính

tổng hợp và duy nhất được giới hạn là 32 cột, một khóa ngoại tổng hợp cũng được giới hạn là 32 cột.

Hình 24-6 Các hạn định tính nguyên vẹn tham chiếu

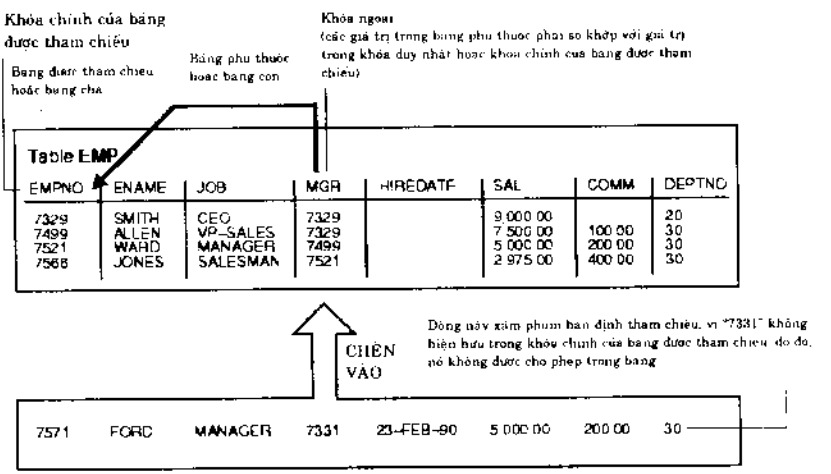


Các hạn định tính nguyên vẹn tự tham chiếu

Một kiểu hạn định tính nguyên vẹn tham chiếu khác được trình bày ở Hình 24-7 được gọi là hạn định tính nguyên vẹn tự tham chiếu. Kiểu khóa ngoại này tham chiếu một khóa cha trong cùng bảng.

Trong thí dụ ở Hình 24-7, hạn định tính nguyên vẹn tham chiếu bảo đảm mọi giá trị trong cột MGR của bảng EMP tương ứng với giá trị hiện có sẵn trong cột EMPNO của cùng bảng, nhưng không nhất thiết phải trong cùng hàng (nghĩa là, mọi trưởng phòng cũng là một nhân viên). Hạn định tính nguyên vẹn này loại trừ khả năng xảy ra số nhân viên bị sai sót trong cột MGR.

Hình 24-7 Các hạn định tham chiếu bảng đơn



Các khóa ngoại và các giá trị rỗng

Mô hình quan hệ cho phép giá trị của các khóa ngoại so khớp với giá trị khóa chính được tham chiếu và duy nhất hoặc rỗng. Nhiều diễn dịch nguyên tắc cơ bản này của mô hình quan hệ có thể xảy ra khi các khóa ngoại (đa cột) tổng hợp có liên quan.

Tiêu chuẩn ANSI/ISO SQL92 (mức mục nhập) cho phép một khóa ngoại tổng hợp có giá trị bất kỳ trong các cột không rỗng của nó nếu bất kỳ cột khác rỗng, dù các giá trị không rỗng đó không được thấy trong các khóa được tham chiếu. Bằng cách sử dụng các hạn định khác (thí dụ, các hạn định NOT NULL và CHECK),

bạn có thể thay đổi xử lý của các khóa ngoại rỗng từng phần từ cách xử lý mặc định này.

Một khóa ngoại tổng hợp có thể tất cả đều rỗng, tất cả không rỗng hoặc rỗng từng phần. Các thuật ngữ sau xác định ba nguyên tắc so khớp thay thế cho các khóa ngoại tổng hợp:

So khớp hoàn toàn	Các khóa ngoại rỗng từng phần không được cho phép. Tất cả các thành phần của khóa ngoại phải rỗng hoặc tổ hợp những giá trị phải xuất hiện bằng giá trị khóa chính hoặc duy nhất của một hàng đơn thuộc bảng được tham chiếu.
So khớp từng phần	Các khóa ngoại tổng hợp rỗng từng phần được cho phép. Tất cả các tập hợp của khóa ngoại phải rỗng hoặc tổ hợp những giá trị có trong khóa ngoại phải xuất hiện trong phần tương ứng của giá trị khóa chính hoặc duy nhất trong bảng được tham chiếu.
Không so khớp	Các khóa ngoại tổng hợp rỗng từng phần được cho phép. Nếu cột bất kỳ của một khóa ngoại tổng hợp rỗng, thì các phần không rỗng của khóa ấy không phải so khớp với bất kỳ phần tương ứng nào của khóa cha.

Các hoạt động được xác định bằng những hạn định tính nguyên vẹn tham chiếu

Hạn định tính nguyên vẹn tham chiếu có thể ấn định những hoạt động cá biệt để được thực hiện trên các hàng phụ thuộc trong bảng con nếu một giá trị khóa cha tham chiếu được sửa đổi. Các hoạt động tham chiếu được các hạn định tính nguyên vẹn FOREIGN KEY của Oracle hỗ trợ là UPDATE và DELETE No Action, và DELETE CASCADE.

Lưu ý:
Các hoạt động tham chiếu khác được các hạn định tính nguyên vẹn FOREIGN KEY của Oracle hỗ trợ có thể được tăng cường bằng cách sử dụng các bộ kích khởi CSDL. Xem chương 18, "Các bộ kích khởi CSDL".

Cập nhật và xóa No Action

Tùy chọn No Action (mặc định) ấn định những giá trị khóa được tham chiếu không thể được cập nhật hoặc xóa nếu dữ liệu kết quả xâm phạm hạn định tính nguyên vẹn tham chiếu. Thí dụ, nếu giá trị khóa chính được một giá trị trong khóa ngoại tham chiếu, giá trị khóa chính được tham chiếu không thể xóa bởi vì dữ liệu phụ thuộc.

Xóa xếp chuỗi

Hoạt động xóa xếp chuỗi ấn định khi các hàng có những giá trị khóa tham chiếu được xóa, tất cả các hàng trong các bảng con với những giá trị khóa ngoại phụ thuộc cũng được xóa - xóa "xếp chuỗi". Thí dụ, nếu hàng trong bảng cha được xóa và giá trị khóa chính của hàng này được tham chiếu bằng một hoặc nhiều giá trị khóa ngoại trong bảng con, thì các hàng trong bảng con tham chiếu giá trị khóa chính ấy cũng được xóa khỏi bảng con.

Các giới hạn DML liên quan đến các hoạt động tham chiếu

Bảng 24-1 trình bày các câu lệnh DML được các hoạt động tham chiếu khác cho phép trên những giá trị khóa chính/duy nhất trong bảng cha và các giá trị khóa ngoại trong bảng con.

Bảng 24-1 Các câu lệnh DML được cho phép bằng Update và Delete No Action

Câu lệnh DML	Được phát hành dựa trên bảng cha	Được phát hành dựa trên bảng con
INSERT	Luôn luôn OK nếu giá trị khóa cha là duy nhất.	OK chỉ nếu giá trị khóa ngoại có sẵn trong khóa cha hoặc từng phần hoặc tất cả rỗng.
UPDATE No Action	Được cho phép nếu câu lệnh không để bất kỳ hàng nào trong bảng con không có giá trị khóa cha được tham chiếu.	Được cho phép nếu giá trị khóa ngoại mới vẫn tham chiếu một giá trị khóa được tham chiếu.
DELETE No Action	Được cho phép nếu không có hàng nào trong bảng con tham chiếu giá trị khóa cha.	Luôn luôn OK.
DELETE Cascade	Luôn luôn OK.	Luôn luôn OK.

Các hạn định tính nguyên vẹn CHECK

Hạn định tính nguyên vẹn CHECK trên cột hoặc tập hợp các cột đòi hỏi điều kiện xác định là đúng hoặc không rõ cho mọi hàng của bảng. Nếu câu lệnh DML có kết quả trong điều kiện của hạn định CHECK thẩm định là sai, thì câu lệnh ấy được hồi phục về trước.

Điều kiện CHECK

Các hạn định CHECK cho bạn tăng cường mọi nguyên tắc tính nguyên vẹn tinh vi hoặc rất cụ thể. Điều kiện của hạn định CHECK có một số giao diện:

- nó phải là biểu thức Boolean được thẩm định bằng cách sử dụng những giá trị trong hàng được chèn hoặc cập nhật và
- nó không thể có các truy vấn con, các chuỗi, các hàm SQL, SYSDATE, UID, USER hoặc USERENV, hay các cột giả LEVEL hoặc ROWNUM.

Trong các hạn định CHECK thẩm định có những trực kiện chuỗi hoặc các hàm SQL với các tham chiếu NLS là các đối số (chẳng hạn như TO_CHAR, TO_DATE và TO_NUMBER), Oracle sử dụng các xác lập NLS của CSDL theo mặc định. Bạn có thể chạy đè các mặc định bằng cách ấn định những tham số NLS rõ ràng trong các hàm đó trong định nghĩa hạn định CHECK.

Thông tin bổ sung:
Xem Oracle8 Reference về các đặc trưng NLS.

Nhiều hạn định CHECK

Một cột đơn có thể có nhiều hạn định CHECK tham chiếu cột trong định nghĩa của nó. Không có giới hạn số hạn định CHECK mà bạn có thể xác định trên một cột.

Các cơ chế kiểm hạn định

Để biết được những kiểu hoạt động nào được cho phép khi các hạn định hiện hữu, cần biết khi Oracle thực sự thực hiện kiểm các hạn định. Để minh họa bằng một thí dụ hoặc hai. Giả sử như sau:
Bảng EMP đã xác định ở Hình 24-7.

Hạn định tự tham chiếu tạo cho các mục nhập trong cột MGR phụ thuộc vào những giá trị của cột EMPNO. Để dễ hiểu, phần còn lại chỉ đề cập đến các cột EMPNO và các cột MGR của bảng EMP.

Xét điều hoạt chèn hàng thứ nhất vào bảng EMP. Không có hàng nào hiện có sẵn, vì vậy một hàng có thể nhập vào như thế nào nếu giá trị trong cột MGR không thể tham chiếu bất kỳ giá trị có sẵn trong cột EMPNO?
Ba khả năng để thực hiện điều này là:

- Một giá trị rỗng có thể được nhập vào cột MGR của hàng thứ nhất, giả sử cột MGR không có hạn định NOT NULL được xác định trên nó. Bởi vì các giá trị rỗng được cho phép trong các khóa ngoại, hàng này được chèn thành công vào bảng ấy.
- Giá trị tương tự có thể được chèn vào cả cột EMPNO và MGR. Trường hợp này cho thấy Oracle thực hiện kiểm hạn định của nó sau khi câu lệnh đã tiến hành xong. Để cho phép một hàng được nhập vào với cùng những giá trị trong khóa cha và khóa ngoại, trước hết Oracle phải tiến hành câu lệnh ấy (nghĩa là, chèn hàng mới đó) và sau đó kiểm xem có bất kỳ hàng nào trong bảng có cột EMPNO tương ứng với MGR của hàng mới đó không.
- Một câu lệnh INSERT đa hàng, chức năng câu lệnh INSERT với câu lệnh SELECT lồng nhau có thể chèn các hàng tham chiếu một cột khác. Thí dụ, hàng thứ nhất có cột EMPNO là 200 và MGR là 300, trong khi hàng thứ hai có thể có cột EMPNO là 300 và MGR là 200. Trường hợp này cũng trình bày kiểm hạn định được trì hoãn cho tới khi tiến hành câu lệnh ấy hoàn tất; tất cả các hàng được chèn đầu tiên, sau đó tất cả các hàng được kiểm cho xem có xâm phạm hạn định không. (Bạn cũng có thể trì hoãn kiểm các hạn định cho tới khi kết thúc giao dịch; xem "Kiểm hạn định trì hoãn".)

Xét hạn định tính nguyên vẹn tự tham chiếu như viễn tưởng sau: Công ty đã được bán đi. Bởi vì việc này xảy ra, tất cả số nhân viên phải được cập nhật vào giá trị hiện hành cùng với 5000 nhân viên để phối hợp với số nhân viên của công ty mới. Vì số giám đốc thực sự cũng là nhân viên, những giá trị này cũng phải tăng lên theo số 5000 nhân viên.

Bảng hiện hành tồn tại được minh họa ở Hình 24-8.

```
UPDATE emp
  SET empno = empno + 5000,
     mgr = mgr + 5000;
```

Dù một hạn định được xác định để xác minh mỗi giá trị MGR so khớp với giá trị EMPNO, câu lệnh này hợp lệ vì Oracle thực hiện kiểm hạn định hiệu quả sau khi câu lệnh hoàn tất. Hình 24-9 trình bày Oracle thực hiện các hoạt động của toàn bộ câu lệnh SQL trước bất kỳ hạn định nào được kiểm.

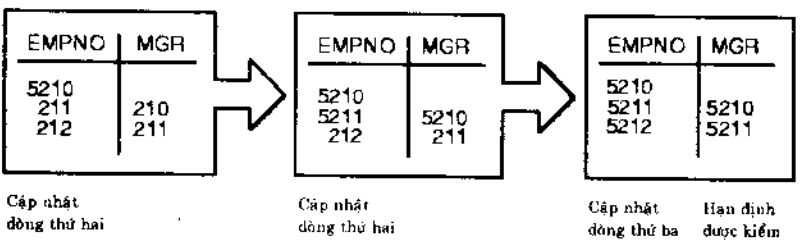
Hình 24-8 Bảng EMP trước khi cập nhật

EMPNO	MGR
210	
211	210
212	211

Các thí dụ trong phần này minh họa cơ chế kiểm khi có các câu lệnh INSERT và UPDATE. Cơ chế tương tự được sử dụng cho tất cả những kiểu câu lệnh DML, kể cả những câu lệnh UPDATE, INSERT và DELETE.
Các thí dụ cũng sử dụng các hạn định tính nguyên vẹn tự tham chiếu để minh họa cơ chế kiểm hạn định. Cơ chế tương tự được

sử dụng cho tất cả những kiểu hạn định, kể cả khóa NOT NULL, UNIQUE, PRIMARY KEY, tất cả các kiểu hạn định FOREIGN KEY và CHECK.

Hình 24-9 Kiểm hạn định



Các giá trị cột mặc định và kiểm hạn định tính nguyên vẹn

Các giá trị mặc định là thành phần của câu lệnh INSERT trước khi câu lệnh được phân ngữ, do đó những giá trị cột mặc định phụ thuộc vào tất cả kiểm hạn định tính nguyên vẹn.

Kiểm hạn định trì hoãn

Bạn có thể trì hoãn các hạn định tính hợp lệ cho tới khi kết thúc giao dịch.

Một hạn định được trì hoãn nếu hệ thống kiểm nó chỉ thỏa mãn cam kết. Nếu hạn định được trì hoãn bị xâm phạm, thì cam kết ấy làm cho giao dịch đó hồi phục về trước.

Nếu hạn định ngay (không trì hoãn), thì nó được kiểm vào cuối mỗi câu lệnh. Nếu nó bị xâm phạm, thì câu lệnh ấy được hồi phục về trước ngay.

Nếu hạn định làm cho một hoạt động (thí dụ xóa xếp chuỗi) thì hoạt động đó luôn luôn sử dụng làm thành phần của câu lệnh đã phát sinh nó, dù hạn định ấy được trì hoãn hoặc hạn định ngay.

Các thuộc tính hạn định

Bạn có thể xác định các hạn định trì hoãn được hoặc không trì hoãn được và được trì hoãn trước hoặc hạn định ngay trước. Những thuộc tính này có thể khác nhau đối với mỗi hạn định. Bạn ấn định chúng với các từ khóa trong mệnh đề CONSTRAINT:
DEFERRABLE hoặc NOT DEFERRABLE
INITIALLY DEFERRED hoặc INITIALLY IMMEDIATE

Thông tin bổ sung:
Xem Oracle8 SQL Reference về các thuộc tính hạn định và các giá trị mặc định của chúng.

Các hạn định có thể được bổ sung, bỏ, cho phép, vô hiệu hóa hoặc thẩm định, nhưng không được thay đổi. Nhất là, bạn không thể thay đổi hạn định không được trì hoãn để nó được trì hoãn.

Chế độ SET CONSTRAINTS

Câu lệnh SET CONSTRAINTS tạo các hạn định là DEFERRED hoặc IMMEDIATE cho một giao dịch cá biệt (theo các tiêu chuẩn ANSI SQL92 bằng cú pháp và ngữ nghĩa). Bạn có thể sử dụng câu lệnh này để xác lập chế độ cho một danh sách những tên hạn định hoặc cho cát hạn định ALL.

Chế độ SET CONSTRAINTS kéo dài trong thời gian giao dịch hoặc cho tới một câu lệnh SET CONSTRAINTS khác xác lập lại chế độ ấy.

SET CONSTRAINTS ... IMMEDIATE tạo các hạn định riêng biệt được kiểm ngay và lúc tiến hành mỗi câu lệnh được hạn định. Trước hết Oracle kiểm bất kỳ những hạn định nào đã được trì hoãn ban đầu trong giao dịch ấy và sau đó tiếp tục kiểm các hạn định ngay của bất kỳ câu lệnh bổ sung trong giao dịch đó (bao lâu tất cả các hạn định được kiểm là nhất quán và không có câu

lệnh SET CONSTRAINTS khác được phát hành). Nếu bất kỳ hạn định nào không kiểm được, thì có biểu hiện một lỗi; ở điểm đó, một COMMIT sẽ làm cho toàn bộ giao dịch hồi phục về trước.

Câu lệnh ALTER SESSION còn có các tùy chọn đối với SET CONSTRAINTS IMMEDIATE hoặc DEFERRED. Các tùy chọn này ám chỉ xác lập các hạn định trì hoãn ALL (nghĩa là, bạn không thể ấn định danh sách của những tên hạn định). Chúng tương đương với tạo một câu lệnh SET CONSTRAINTS vào lúc bắt đầu mỗi giao dịch trong phiên hiện hành.

Tạo các hạn định ngay vào cuối giao dịch là cách kiểm xem COMMIT có thể thành công hay không. Bạn có thể tránh các hồi phục về trước ngoài ý muốn bằng cách xác lập các hạn định là IMMEDIATE bằng câu lệnh cuối cùng trong một giao dịch. Nếu bất kỳ hạn định nào không thực hiện kiểm được, thì bạn có thể sửa lỗi trước khi cam kết giao dịch ấy.

Câu lệnh SET CONSTRAINTS không được cho phép bên trong các bộ kích khởi.

SET CONSTRAINTS có thể là một câu lệnh phân phối. Các CSDL có sẵn có các giao dịch trong tiến trình được khai báo khi câu lệnh SET CONSTRAINTS ALL xảy ra và các liên kết mới biết được nó đã xảy ra ngay khi chúng bắt đầu giao dịch.

Các hạn định duy nhất và các chỉ mục

Người dùng sẽ thấy các hạn định không nhất quán, kể cả các sao chép trong các chỉ mục duy nhất khi giao dịch của người dùng tạo ra những tính không nhất quán này.

Bạn có thể đưa các hạn định khóa ngoại và duy nhất được trì hoãn vào các ảnh chụp nhanh để cho phép tô bổ nhanh và hoàn toàn thực hiện thành công.

Các hạn định duy nhất trì hoãn được luôn luôn sử dụng các chỉ mục không duy nhất. Khi bạn xóa hạn định trì hoãn được, thì còn lại chỉ mục của nó. (Điều này thuận tiện vì những thông tin của bộ nhớ vẫn có thể còn lại sau khi bạn đã vô hiệu hóa một hạn định). Các hạn định duy nhất không trì hoãn được và các khóa chính còn sử dụng chỉ mục không duy nhất nếu chỉ mục không duy nhất ấy được đưa vào các cột khóa trước khi hạn định ấy được tăng cường.

Các hạn định cho phép, vô hiệu hóa và cho phép không hợp lệ

Bạn có thể cho phép hoặc vô hiệu hóa các hạn định tính nguyên vẹn ở mức bảng sử dụng câu lệnh CREATE TABLE hoặc ALTER TABLE. Tùy chọn ENABLE NOVALIDATE của câu lệnh ALTER TABLE bắt đầu lại kiểm hạn định trên các hạn định được vô hiệu hóa mà không thẩm định trước tất cả dữ liệu trong bảng.
ENABLE CONSTRAINT bảo đảm tất cả các hàng trong bảng hợp lệ, nghĩa là tất cả các hàng thích hợp với hạn định ấy.
DISABLE CONSTRAINT cho phép bảng chứa các hàng xâm phạm hạn định.
ENABLE NOVALIDATE CONSTRAINT cho phép các hàng có sẵn xâm phạm hạn định, nhưng bảo đảm tất cả các hàng mới hoặc các hàng được sửa đổi phải hợp lệ.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Administrator về cách sử dụng các tùy chọn ENABLE, DISABLE và ENABLE NOVALIDATE CONSTRAINT.

Chương 25

Điều khiển truy xuất cơ sở dữ liệu

Chương này giải thích cách điều khiển truy xuất một CSDL của Oracle, gồm các chủ đề:

- **Bảo mật CSDL**
- **Các giản đồ, các người dùng CSDL và các vùng bảo mật**
- **Xác minh người dùng**
- **Các chỉ tiêu và xác lập không gian bảng người dùng**
- **Nhóm người dùng PUBLIC**
- **Các giới hạn tài nguyên người dùng và các lược trình**
- **Cấp đăng ký**

Thông tin bổ sung:
Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle về truy xuất CSDL trong môi trường đó.

Bảo mật cơ sở dữ liệu

Bảo mật CSDL đòi hỏi phải được cho phép hoặc không được cho phép các hoạt động người dùng về CSDL và những đối tượng trong nó. Oracle sử dụng các giản đồ và các vùng bảo mật để điều khiển truy xuất dữ liệu và giới hạn sử dụng các tài nguyên CSDL khác nhau.

Oracle cung cấp điều khiển truy xuất thận trọng toàn diện. Điều khiển truy xuất tùy chọn quy định tất cả truy xuất người dùng đối với những đối tượng có tên qua các đặc quyền. Một đặc quyền là sự cho phép truy xuất một đối tượng có tên dựa vào phương thức được quy định; thí dụ, sự cho phép truy vấn một bảng. Các đặc quyền được cho người dùng một cách thận trọng - vì thuật ngữ "điều khiển truy xuất thận trọng". Xem thêm chương 26, "Các đặc quyền và các vai trò".

Các giản đồ, các người dùng cơ sở dữ liệu và các vùng bảo mật

Một người dùng (đôi khi còn gọi là tên người dùng – username) là tên được xác định trong CSDL có thể kết nối với những đối tượng truy xuất. Một giản đồ là một tập hợp các đối tượng có tên, chẳng hạn như các bảng, các cập nhật, các cụm liên kết, các thủ tục và các bộ phận mềm được kết hợp với một người dùng cá biệt. Các giản đồ và các người dùng giúp cho người điều hành CSDL quản lý bảo mật CSDL.

Để truy xuất một CSDL, người dùng phải chạy ứng dụng CSDL (chẳng hạn như Oracle Forms form, SQL*Plus hoặc chương trình biên dịch trước) và kết nối bằng cách sử dụng tên người dùng được xác định trong CSDL ấy.

Khi một người dùng CSDL được tạo, một giản đồ tương ứng có cùng tên được tạo cho người dùng đó. Theo mặc định, một khi người dùng kết nối với CSDL, thì người dùng ấy truy xuất được

tất cả những đối tượng có trong giản đồ tương ứng đó. Người dùng chỉ được kết hợp với cùng tên; do đó thuật ngữ tên người dùng và giản đồ thường được sử dụng thay thế lẫn nhau.

Quyền truy xuất của người dùng được điều khiển bằng các xác lập khác về vùng bảo mật của người dùng đó. Khi tạo một người dùng CSDL mới hoặc thay đổi một người dùng hiện có, người điều hành bảo mật phải đưa ra nhiều quyết định liên quan đến vùng bảo mật của người dùng. Chúng gồm:

- thông tin xác minh người dùng được duy trì bởi CSDL, hệ điều hành hoặc dịch vụ xác minh mạng
- các xác lập đối với những không gian bảng tạm thời và mặc định của người dùng
- danh sách, nếu có về các không gian bảng người dùng có thể truy xuất và các chỉ tiêu kết hợp cho mỗi không gian bảng có danh sách
- lược trình giới hạn tài nguyên của người dùng; nghĩa là giới hạn về số lượng tài nguyên hệ thống mà người dùng có thể sử dụng
- các đặc quyền và các vai trò cung cấp cho người dùng truy xuất thích hợp các đối tượng cần thiết để thực hiện các điều hoạt CSDL

Chương này mô tả bốn tùy chọn vùng bảo mật đầu tiên được liệt kê ở trên; các đặc quyền và các vai trò được đề cập ở chương 26, "Các Đặc Quyền và Các Vai Trò".

Lưu ý:
Thông tin trong chương này áp dụng cho tất cả những người dùng CSDL do người dùng xác định. Không áp dụng cho những người dùng CSDL SYS và SYSTEM đặc biệt. Các xác lập đối với những vùng bảo mật của người dùng này sẽ không bao giờ được thay đổi.

Thông tin bổ sung:
Xem hướng dẫn của Oracle8 Administrator về những người dùng SYS và SYSTEM đặc biệt.

Xác minh người dùng

Để ngăn chặn sử dụng không hợp lệ tên người dùng CSDL, Oracle cung cấp thẩm định người dùng bằng ba phương pháp khác nhau đối với những người dùng CSDL bình thường:

xác minh bằng hệ điều hành
xác minh bằng dịch vụ hệ thống
xác minh bằng CSDL Oracle kết hợp

Để đơn giản hóa, một phương pháp thường được sử dụng để xác minh tất cả những người dùng CSDL. Tuy nhiên, Oracle cho phép sử dụng tất cả những phương pháp trong cùng môi trường làm việc CSDL.
Oracle còn mã hóa các mật khẩu khi truyền để bảo mật xác minh của mạng.

Oracle đòi hỏi các thủ tục xác minh đặc biệt đối với các người điều hành CSDL, vì họ thực hiện những điều hoạt CSDL đặc biệt.

Xác minh bằng hệ điều hành

Một số hệ điều hành cho phép Oracle sử dụng thông tin được hệ điều hành duy trì để xác định các người dùng. Ưu điểm xác minh bằng hệ điều hành là:

Các người dùng có thể kết nối với Oracle thuận tiện hơn (không phải xác định tên người dùng hoặc mật khẩu). Thí dụ, một người dùng có thể viện dẫn SQL*Plus và bỏ qua tên người dùng và mật khẩu được nhắc nhập vào

```
SQLPLUS /
```

Kiểm soát quyền hạn của người dùng tập trung trong hệ điều hành; Oracle không cần lưu trữ hoặc quản lý các mật khẩu của

người dùng. Tuy nhiên, Oracle vẫn duy trì các tên người dùng trong CSDL.
Các mục nhập Username trong cơ sở dữ liệu và các vệt tìm kiếm soát hệ điều hành tương ứng.

Nếu hệ điều hành được sử dụng để xác minh các người dùng CSDL, thì có một số nhận xét đặc biệt phát sinh liên quan đến những môi trường CSDL được phân phối và những liên kết CSDL; xem chương 30, "Các CSDL được phân phối" về chủ đề này.

Xác định bằng mạng

Nếu các dịch vụ xác định mạng có thể sử dụng đối với bạn (chẳng hạn như DCE, Kerberos hoặc SESAME), Oracle có thể chấp nhận xác định từ dịch vụ mạng. Để sử dụng dịch vụ xác định mạng với Oracle, bạn cũng phải có sản phẩm Oracle Secure Network Services.

Thông tin bổ sung:
Nếu bạn sử dụng dịch vụ xác định mạng, một số nhận xét đặc biệt phát sinh cho các vai trò mạng và các liên kết CSDL. Xem "Các hệ thống CSDL Oracle8 phân phối" về xác định mạng.

Xác minh bằng CSDL Oracle

Oracle có thể xác minh những người dùng cố kết nối CSDL bằng cách sử dụng thông tin được lưu trữ trong CSDL đó. Bạn phải sử dụng phương pháp này khi hệ điều hành không thể được sử dụng để thẩm định người dùng.

Khi Oracle sử dụng xác định CSDL, bạn tạo mỗi người dùng với một mật khẩu kết hợp. Người dùng cung cấp mật khẩu chính xác khi thiết lập một kết nối để ngăn chặn sử dụng bất hợp lệ CSDL ấy. Oracle lưu trữ mật khẩu của người dùng trong từ điển dữ liệu

bằng định dạng mã hóa. Người dùng có thể thay đổi mật khẩu của họ bất kỳ lúc nào.

Mã hóa mật khẩu khi kết nối

Để bảo vệ mật khẩu một cách đáng tin cậy, Oracle cho phép bạn mã hóa các mật khẩu khi có các kết nối mạng (client/server và server/server). Nếu bạn cho phép tính năng này trên các máy khách và máy chủ, Oracle mã hóa các mật khẩu bằng cách sử dụng thuật toán DES (Data Encryption Standards - Các tiêu chuẩn mã hóa dữ liệu) sửa đổi trước khi truyền trên mạng.

Thông tin bổ sung:
Xem "Các hệ thống CSDL được phân phối của Oracle8" về mã hóa các mật khẩu trong các hệ thống mạng.

Khóa tài khoản

Oracle có thể khóa tài khoản của người dùng nếu người dùng không khởi nhập được hệ thống bằng một số cố gắng nhất định. Phụ thuộc vào tài khoản được cấu hình như thế nào, nó có thể được tự động mở khóa sau một khoảng thời gian nhất định hoặc nó phải được mở khóa bằng người điều hành CSDL.

Câu lệnh CREATE PROFILE cấu hình số lần khởi nhập không thành công mà một người dùng có thể cố thực hiện và thời lượng tài khoản duy trì khóa trước khi được tự động mở khóa. Xem "Các lược trình".

Người điều hành CSDL còn có thể khóa các tài khoản thủ công. Khi điều này xảy ra, tài khoản ấy không thể được mở khóa tự động, nhưng phải mở khóa bởi chính người điều hành CSDL.

Thời gian mật khẩu tồn tại và thời hạn

Các tùy chọn thời gian mật khẩu tồn tại và thời hạn cho phép người điều hành CSDL ấn định thời gian tồn tại đối với các mật khẩu, sau thời gian chúng hết hạn và phải được thay đổi trước khi khởi nhập vào tài khoản được hoàn tất. Lần khởi nhập đầu tiên vào tài khoản CSDL sau khi mật khẩu hết hạn, tài khoản của người dùng nhập khoảng thời gian gia hạn và một thông điệp cảnh báo được phát hành cho người dùng ấy mỗi lần họ khởi nhập cho tới khi thời gian gia hạn chấm dứt.

Người dùng muốn thay đổi mật khẩu trong thời gian gia hạn. Nếu mật khẩu không được thay đổi trong thời gian gia hạn, thì tài khoản ấy được khóa và không được phép đăng nhập vào tài khoản ấy nữa nếu không có sự trợ giúp của người điều hành CSDL.

Người điều hành CSDL còn có thể xác lập thời hạn của mật khẩu. Khi điều này xảy ra, tình trạng tài khoản của các người dùng được chuyển thành hết hạn và khi người dùng đăng nhập vào, thì tài khoản ấy nhập thêm thời gian gia hạn.

Lịch sử mật khẩu

Tùy chọn lịch sử mật khẩu kiểm từng mật khẩu mới được ấn định để bảo đảm mật khẩu không được sử dụng lại trong một thời gian nhất định hoặc trong số lần thay đổi mật khẩu. Người điều hành CSDL có thể cấu hình những nguyên tắc để sử dụng lại mật khẩu bằng các câu lệnh CREATE PROFILE.

Xác minh tính phức tạp mật khẩu

Kiểm xác minh tính phức tạp sao cho mỗi mật khẩu đủ phức tạp để cung cấp sự bảo vệ thích đáng đối với những tin tặc cố đột nhập vào hệ thống bằng cách đoán tìm mật khẩu.

Thủ tục xác minh tính phức tạp mật khẩu mặc định của Oracle đòi hỏi mỗi mật khẩu phải:
bằng ít nhất bốn ký tự
không bằng định danh người dùng
có ít nhất một ký tự chữ cái, một ký tự số và một dấu chấm câu
không giống bất kỳ từ nào trên danh sách bên trong của các từ đơn giản như welcome, account, database, user, vân vân.
khác với mật khẩu trước ít nhất ba ký tự.

Xác minh người điều hành CSDL

Những người điều hành CSDL thực hiện các hoạt động đặc biệt (chẳng hạn như đóng hoặc khởi động CSDL) sẽ không được những người dùng CSDL bình thường thực hiện. Oracle cung cấp phương pháp xác minh an toàn hơn cho những username người điều hành CSDL.

Bạn có thể chọn giữa xác định hệ điều hành hoặc tập tin mật khẩu để xác định những người điều hành CSDL.

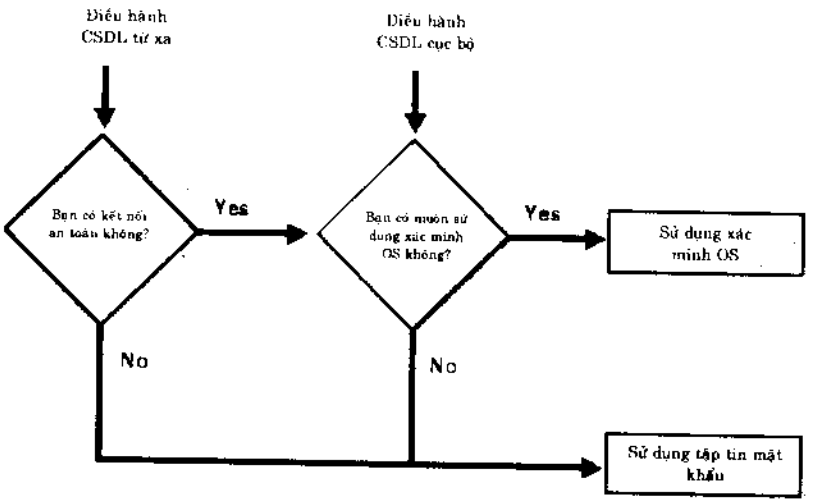
Hình 25-1 minh họa các chọn lựa bạn sử dụng cho các phương pháp xác định người điều hành CSDL, tùy theo người điều hành CSDL của bạn cục bộ (trên cùng máy mà CSDL cư trú) hoặc nếu bạn điều hành nhiều máy CSDL khác nhau từ một client đơn từ xa.

Trên hầu hết những hệ điều hành, xác minh OS đối với những người điều hành CSDL liên quan đến đưa tên người dùng OS của người điều hành CSDL vào một nhóm đặc biệt (trên các hệ thống UNIX, đây là nhóm dba) hoặc cung cấp tên người dùng OS bằng một tiến trình đặc biệt thích đáng.

Thông tin bổ sung:
Xem tài liệu Oracle xác định hệ điều hành về xác minh OS

của những người điều hành CSDL.

Hình 25-1 Các phương pháp xác minh người điều hành CSDL



CSDL sử dụng các tập tin mật khẩu để theo dõi các tên người dùng CSDL có được cho các đặc quyền SYSDBA và SYSOPER. Các đặc quyền này cho phép những người điều hành CSDL thực hiện các hoạt động sau:

SYSOPER	Cho phép bạn thực hiện các đặc quyền STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG và RECOVER, kể cả đặc quyền RESTRICTED SESSION.
SYSDBA	Có tất cả những đặc quyền với ADMIN OPTION và đặc quyền hệ thống SYSOPER; cho phép CREATE DATABASE và phục hồi dựa trên thời gian.

Các chỉ tiêu và các xác lập không gian bảng

Là thành phần của mọi vùng bảo mật của người dùng, người điều hành CSDL có thể xác lập nhiều tùy chọn liên quan đến sử dụng không gian bảng:

- không gian bảng mặc định của người dùng
- không gian bảng tạm thời của người dùng
- các chỉ tiêu sử dụng không gian trên các không gian bảng của CSDL đối với người dùng

Không gian bảng mặc định

Khi một người dùng tạo một đối tượng giản đồ không có ấn định không gian bảng chứa đối tượng đó, thì Oracle đưa đối tượng ấy vào không gian bảng mặc định của người dùng. Bạn xác lập không gian bảng mặc định của người dùng khi người dùng được tạo; bạn có thể thay đổi nó sau khi người dùng đã được tạo.

Không gian bảng tạm thời

Khi một người dùng tiến hành câu lệnh SQL đòi hỏi tạo đoạn tạm thời, thì Oracle phân bổ đoạn đó vào không gian bảng tạm thời của người dùng ấy.

Các chỉ tiêu và truy xuất không gian bảng

Bạn có thể gán cho mỗi người dùng một chỉ tiêu không gian bảng đối với bất kỳ một không gian bảng nào của CSDL. Thực hiện điều này có thể hoàn tất hai điều:

Bạn cho phép người dùng sử dụng không gian bảng ấn định để tạo các đối tượng, với điều kiện người dùng ấy có các đặc quyền thích hợp.

Bạn có thể giới hạn số lượng không gian được phân bổ để lưu trữ các đối tượng trong không gian bảng ấn định.

Theo mặc định, mỗi người dùng không có chỉ tiêu trên không gian bảng bất kỳ trong CSDL. Do đó, nếu người dùng có đặc quyền tạo một kiểu đối tượng giản đồ nào đó, thì người ấy cũng phải được gán một chỉ tiêu không gian bảng để tạo đối tượng hoặc được cung cấp đặc quyền để tạo đối tượng đó trong giản đồ của một người dùng khác đã được gán chỉ tiêu không gian bảng thích đáng.

Bạn có thể gán hai kiểu chỉ tiêu không gian bảng cho một người dùng: chỉ tiêu đối với số lượng không gian đĩa riêng biệt trong không gian bảng (được ấn định bằng byte, kilobyte hoặc megabyte) hoặc một chỉ tiêu đối với số lượng không gian đĩa không giới hạn số lượng không gian đĩa trong không gian bảng. Bạn nên gán các chỉ tiêu nhất định để ngăn chặn các đối tượng người dùng tiêu thụ quá nhiều không gian trong không gian bảng.

Các chỉ tiêu không gian bảng và các đoạn tạm thời không có ảnh hưởng với nhau:
Các đoạn tạm thời không tiêu thụ bất kỳ chỉ tiêu mà người dùng có thể xử lý.
Các đoạn tạm thời có thể được tạo trong không gian bảng cho người dùng không có chỉ tiêu.

Bạn có thể gán một chỉ tiêu không gian bảng cho người dùng khi bạn tạo người dùng đó và bạn có thể thay đổi chỉ tiêu đó hoặc bổ sung chỉ tiêu về sau.

Thu hồi truy xuất không gian bảng của người dùng bằng cách thay đổi chỉ tiêu hiện hành của người dùng thành zero. Với chỉ tiêu zero, các đối tượng người dùng trong không gian bảng bị hủy bỏ vẫn được duy trì, nhưng các đối tượng ấy không thể được phân bổ không gian mới.

Nhóm người dùng PUBLIC

Mỗi CSDL có một nhóm người dùng được gọi là PUBLIC. Nhóm người dùng PUBLIC cung cấp truy xuất chung các đối tượng giản đồ riêng biệt (các bảng, các khung nhìn, vân vân.) và cung cấp cho tất cả các người dùng những đặc quyền hệ thống ấn định. Mỗi người dùng tự động thuộc về nhóm người dùng PUBLIC.

Là các thành viên của PUBLIC, người dùng có thể thấy (lựa từ) tất cả các bảng từ điển dữ liệu có tiền tố USER và ALL. Ngoài ra, người dùng có thể cho PUBLIC một đặc quyền hoặc vai trò. Tất cả người dùng có thể sử dụng các đặc quyền được cho PUBLIC.

Bạn có thể cho (hoặc thu hồi) bất kỳ đặc quyền hệ thống, đặc quyền đối tượng hoặc vai trò đối với PUBLIC. Xem chương 26, "Các đặc quyền và các vai trò". Tuy nhiên, để duy trì tính bảo mật chặt chẽ về các quyền, chỉ cho các đặc quyền và các vai trò liên quan đến tất cả những người dùng đối với PUBLIC.

Cho và thu hồi các đặc quyền hệ thống và đối tượng tới và từ PUBLIC có thể làm cho mỗi khung nhìn, thủ tục, chức năng, bộ phần mềm và bộ kích khởi trong CSDL phải được biên dịch lại.

PUBLIC có những giới hạn sau:

Bạn không thể gán các chỉ tiêu không gian bảng vào PUBLIC, dù bạn có thể gán đặc quyền hệ thống UNLIMITED TABLESPACE vào PUBLIC.

Bạn có thể tạo các liên kết CSDL và các từ đồng nghĩa PUBLIC (sử dụng CREATE PUBLIC DATABASE LINK/SYNONYM), nhưng không có đối tượng giản đồ nào khác có thể được PUBLIC sở hữu. Thí dụ, câu lệnh sau không hợp lệ:

```
CREATE TABLE public.emp . . . ;
```

Lưu ý:
Các đoạn hồi phục về trước có thể được tạo bằng từ khóa PUBLIC, nhưng những đoạn này không được nhóm người dùng PUBLIC sở hữu. Tất cả những đoạn hồi phục về trước được SYS sở hữu. Xem chương 2 "Các khối dữ liệu, các khối lưu riêng và các đoạn" về các đoạn hồi phục về trước.

Các giới hạn tài nguyên người dùng và các lược trình

Bạn có thể xác lập những giao diện về số lượng tài nguyên hệ thống khác nhau mà mỗi người dùng thuộc vùng bảo mật người dùng có thể sử dụng. Bằng cách thực hiện này, bạn có thể ngăn chặn sự tiêu thụ không kiểm soát được về những tài nguyên hệ thống có giá trị, chẳng hạn như thời gian của CPU.

Đặc trưng giao diện tài nguyên này rất hữu dụng trong những hệ thống đa người dùng lớn, nơi những tài nguyên hệ thống rất tốn phí. Sự tiêu thụ quá mức những tài nguyên này do một hoặc nhiều người dùng có thể ảnh hưởng thiệt hại đến những người dùng khác của CSDL ấy. Trong những hệ thống CSDL đa người dùng quy mô nhỏ hoặc người dùng đơn, đặc trưng tiến trình hệ thống không quan trọng, vì sự tiêu thụ những tài nguyên hệ thống ít có thể gây ảnh hưởng thiệt hại.

Bạn quản lý các giới hạn tài nguyên của người dùng bằng tập hợp có tên lược trình của những giới hạn tài nguyên của người dùng ấy mà bạn có thể gán cho người dùng đó. Mỗi CSDL Oracle có thể có số lược trình không giới hạn. Oracle cho phép người điều hành bảo mật cho phép hoặc vô hiệu hóa sự tăng cường của tài nguyên lược trình giới hạn đối với mọi người.

Nếu bạn xác lập những giới hạn tài nguyên, hiệu suất hơi bị giảm đi khi những người dùng tạo các phiên. Điều này xảy ra vì

Oracle tải tất cả dữ liệu giới hạn tài nguyên đối với người dùng ấy khi một người dùng kết nối với CSDL.

Các kiểu tài nguyên hệ thống và các giới hạn

Oracle có thể giới hạn sử dụng nhiều kiểu tài nguyên hệ thống, kể cả thời gian CPU và các lần đọc logic. Nói chung, bạn có thể điều khiển từng tài nguyên này ở mức phiên, mức gọi hoặc cả hai:

Session Level (Mức phiên)	Mỗi lần một người dùng kết nối với CSDL, một phiên được tạo ra. Mỗi phiên tiêu thụ thời gian CPU và bộ nhớ trên máy tính tiến hành Oracle. Bạn có thể xác lập nhiều giới hạn tài nguyên ở mức phiên. Nếu một người dùng vượt giới hạn tài nguyên mức phiên, Oracle kết thúc (hồi phục về trước) câu lệnh hiện hành và cho trở lại thông điệp cho biết đã đạt tới giới hạn của phiên ấy. Ở điểm này, tất cả những câu lệnh trước trong giao dịch hiện hành không bị ảnh hưởng, và những điều hoạt người dùng chỉ có thể thực hiện là COMMIT, ROLLBACK hoặc thôi kết nối (trong trường hợp này, giao dịch hiện hành được cam kết); tất cả các điều hoạt khác tạo ra lỗi. Thậm chí sau khi giao dịch ấy được cam kết hoặc hồi phục về trước, người dùng không thể thực hiện thêm công việc nào khác trong phiên hiện hành.
Call Level (Mức gọi)	Mỗi lần câu lệnh SQL được tiến hành, nhiều bước được thực hiện để xử lý câu lệnh ấy. Trong khi xử lý, nhiều lệnh gọi được thực hiện đối với CSDL bằng thành phần của các pha tiến hành khác. Để ngăn chặn một lệnh gọi

bất kỳ sử dụng hệ thống độc quyền, Oracle cho phép bạn xác lập nhiều giới hạn tài nguyên ở mức gọi.

Nếu một người dùng vượt mức tài nguyên mức gọi, Oracle tạm dừng tiến trình của câu lệnh ấy, hồi phục câu lệnh ấy về trước và cho trở lại một lỗi. Tuy nhiên, tất cả những câu lệnh trước của giao dịch hiện hành vẫn không bị ảnh hưởng và phiên của người dùng vẫn được kết nối.

Thời gian CPU

Khi các câu lệnh SQL và các kiểu lệnh gọi khác được thực hiện đối với Oracle, một thời lượng CPU cần thiết để xử lý lệnh gọi ấy. Các lệnh gọi trung bình đòi hỏi một thời lượng CPU nhỏ. Tuy nhiên, câu lệnh SQL có liên quan đến số lượng dữ liệu lớn hoặc truy vấn nhanh có thể tiêu thụ thời lượng CPU lớn và làm giảm thời lượng CPU có thể sử dụng cho xử lý khác.

Để tránh sử dụng thời gian CPU không kiểm soát được, bạn có thể giới hạn thời gian CPU cho mỗi lệnh gọi và tổng thời gian CPU được sử dụng cho các lệnh gọi CPU trong một phiên. Các giới hạn được xác lập và tính toán bằng một phần trăm giây CPU (0,01 giây) được một lệnh gọi hoặc phiên sử dụng.

Đọc logic

Nhập/xuất (I/O) là một trong những điều hoạt tốn phí nhất trong hệ thống CSDL. Các câu lệnh SQL là cần nhiều I/O có thể chiếm độc quyền bộ nhớ và sử dụng đĩa làm cho các điều hoạt CSDL khác tranh chấp những tài nguyên này.

Để tránh các nguồn đơn vượt I/O, Oracle cho bạn giới hạn đọc khối dữ liệu logic trên mỗi lệnh gọi hoặc phiên. Đọc khối dữ liệu

logic kể cả đọc khối dữ liệu từ bộ nhớ và đĩa. Các giới hạn được xác lập và tính bằng số lần đọc khối dữ liệu được thực hiện bởi một lệnh gọi hoặc trong một phiên.

Các tài nguyên khác

Oracle còn cung cấp giới hạn của nhiều tài nguyên khác ở mức phiên:

Bạn có thể giới hạn số phiên đồng thời trên mỗi người dùng. Mỗi người dùng chỉ có thể tạo tối đa số phiên đồng thời được xác định trước.

Bạn có thể giới hạn thời gian không hoạt động đối với một phiên. Nếu thời gian giữa các lệnh gọi Oracle cho một phiên đạt tới giới hạn thời gian không hoạt động, thì giao dịch hiện hành được hồi phục về trước, phiên ấy bị hủy ngang và các tài nguyên của phiên được đưa trở về hệ thống. Lệnh gọi kế tiếp nhận một lỗi cho biết người dùng không còn được kết nối với môi trường làm việc đó nữa. Giới hạn này được xác lập bằng số phút trôi qua.

Lưu ý:

Ngay sau khi một phiên bị hủy ngang vì nó đã vượt giới hạn thời gian không hoạt động, xử lý nền của bộ giám sát tiến trình (PMON) dọn sạch sau khi phiên bị hủy ngang. Cho tới khi PMON hoàn tất tiến trình này, phiên bị hủy ngang vẫn được tính trong giới hạn tài nguyên phiên/người dùng bất kỳ.

Bạn có thể giới hạn thời gian kết nối đã trôi qua trên mỗi phiên. Nếu thời gian của phiên vượt giới hạn thời gian đã trôi qua, thì giao dịch hiện hành được hồi phục về trước và các tài nguyên của phiên được đưa trở lại hệ thống. Giới hạn này được xác lập bằng số phút đã trôi qua.

Lưu ý:
Oracle không giám sát liên tục thời gian không hoạt động trôi qua hoặc thời gian kết nối đã trôi qua. Thực hiện như vậy sẽ làm giảm hiệu suất hệ thống. Thay vào đó, cứ vài phút nó kiểm một lần. Do đó, một phiên có thể vượt giới hạn này một chút (thí dụ, khoảng 5 phút) trước khi Oracle tăng cường giới hạn và hủy ngang phiên ấy.

Bạn có thể giới hạn số không gian SGA riêng (được sử dụng cho các vùng SQL riêng) đối với một phiên. Giới hạn này chỉ quan trọng trong các hệ thống sử dụng cấu hình server đa xâu; trái lại, các vùng SQL riêng được định vị trong PGA. Giới hạn này được xác lập bằng số byte bộ nhớ trong SGA của một môi trường làm việc. Sử dụng các ký tự "K" hoặc "M" để ấn định số kilobyte hoặc megabyte.

Thông tin bổ sung:
Các hướng dẫn về cho phép và vô hiệu hóa các giới hạn tài nguyên ở phần hướng dẫn của Oracle8 Administrator.

Các lược trình

Một lược trình (profile) là một tập hợp có tên của những giới hạn tài nguyên được ấn định có thể được gán cho một username hợp lệ của CSDL Oracle. Các lược trình giúp quản lý dễ dàng các giới hạn tài nguyên.

Khi nào sử dụng các lược trình

Bạn cần tạo và quản lý các lược trình người dùng chỉ nếu các giới hạn tài nguyên cần thiết cho chính xác bảo mật CSDL của mình. Để sử dụng các lược trình, trước hết phân loại các kiểu người dùng có liên quan trong một CSDL. Giống như các vai trò được sử dụng để quản lý các đặc quyền của những người dùng có liên quan, các lược trình được sử dụng để quản lý các giới hạn tài

nguyên của những người dùng có liên quan. Xác định bao nhiêu lược trình cần thiết để bao gồm tất cả các kiểu người dùng trong một CSDL và sau đó xác định các giới hạn tài nguyên thích hợp cho mỗi lược trình.

Xác định những giá trị đối với các giới hạn tài nguyên của một lược trình

Trước khi tạo các lược trình và xác lập các giới hạn tài nguyên được kết hợp với chúng, bạn nên xác định những giá trị thích hợp cho mỗi giới hạn tài nguyên. Bạn có thể dựa vào các giá trị này về kiểu của các điều hoạt một người dùng tiêu biểu thực hiện. Thí dụ, nếu một lớp người dùng không thường thực hiện số lần đọc khối dữ liệu logic cao, thì các giới hạn LOGICAL_READS_PER_SESSION và LOGICAL_READS_PER_CALL sẽ xác lập vừa phải.

Thường, cách tốt nhất để xác định những giá trị tài nguyên thích hợp cho một lược trình người dùng định trước là thu thập những thông tin lịch sử về mỗi cách sử dụng tài nguyên. Thí dụ, người điều hành bảo mật hoặc CSDL có thể sử dụng tùy chọn AUDIT SESSION để thu thập thông tin về các giới hạn CONNECT_TIME, LOGICAL_READS_PER_SESSION và LOGICAL_READS_PER_CALL. Xem chương 27 "Kiểm soát".

Bạn có thể thu thập các thống kê về những giới hạn khác bằng cách sử dụng đặc trưng Monitor của Oracle Enterprise Manager (hoặc Server Manager), nhất là bộ giám sát Statistics.

Cấp đăng ký

Oracle thường được cho đăng ký sử dụng bằng số người dùng có tên tối đa hoặc bằng số người dùng kết nối đồng thời tối đa. Người điều hành CSDL (DBA) chịu trách nhiệm bảo đảm vùng đó tuân thủ hợp đồng đăng ký. Phương tiện đăng ký của Oracle giúp sử dụng hệ thống giám sát DBA bằng cách theo dõi và giới

hạn số phiên được kết nối đồng thời với một môi trường làm việc hoặc số người dùng được tạo trong một CSDL.

Nếu DBA phát hiện vượt số phiên đã đăng ký kết nối hoặc nhiều hơn số người dùng đã đăng ký cần được tạo, thì họ có thể nâng cấp cho đăng ký Oracle để phát sinh giới hạn thích hợp. (Nâng cấp cho đăng ký Oracle, bạn liên lạc với đại diện của Oracle).

Lưu ý:
Khi Oracle được nhúng trong một ứng dụng Oracle (chẳng hạn như Oracle Office), chạy trên một số hệ điều hành cũ hơn, hoặc được mua để sử dụng trong một số nước, nó không được cấp đăng ký xác lập số phiên hoặc nhóm người dùng. Chỉ trong các trường hợp đó, các cơ chế cấp đăng ký Oracle không áp dụng và sẽ không được cho phép.

Các phần sau giải thích hai kiểu đăng ký chính có thể sử dụng đối với Oracle.

Cấp đăng ký sử dụng đồng thời

Cấp đăng ký sử dụng đồng thời, sự cho phép đó ấn định số người dùng đồng thời là các phiên có thể được kết nối đồng thời với CSDL trên một máy tính riêng biệt vào bất kỳ lúc nào. Số này gồm tất cả các xử lý lô và các người dùng trực tiếp. Nếu một người dùng đơn có các phiên đồng thời, mỗi phiên tính riêng trong tổng số phiên. Nếu phần mềm đa công (chẳng hạn như TP monitor) được sử dụng để giảm số phiên được kết nối trực tiếp với CSDL, số người dùng đồng thời là số nhập liệu riêng vào máy tính tiền tiêu (front end) đa công.

Cơ chế cấp đăng ký sử dụng đồng thời cho phép một DBA:
Xác lập giới hạn về số phiên đồng thời có thể kết nối với một môi trường làm việc bằng cách xác lập tham số LICENSE_MAX_SESSIONS. Một khi đạt tới giới hạn này, chỉ

những người dùng có đặc quyền hệ thống RESTRICTED SESSION có thể kết nối với môi trường làm việc; điều này cho phép DBA kết thúc các phiên không cần thiết, cho phép các phiên khác kết nối.
Xác lập giới hạn cảnh báo về số phiên đồng thời có thể kết nối với một môi trường làm việc bằng cách xác lập tham số LICENSE_SESSIONS_WARNING. Một khi đạt tới giới hạn cảnh báo, Oracle cho phép các phiên bổ sung kết nối (tới giới hạn tối đa được đề cập ở trên), nhưng gửi các thông điệp cảnh báo tới người dùng bất kỳ kết nối có đặc quyền RESTRICTED SESSION và ghi thông điệp cảnh báo trong tập tin ALERT vào CSDL.

DBA có thể xác lập những giới hạn này trong tập tin tham số của CSDL sao cho chúng có ảnh hưởng khi môi trường làm việc khởi động và có thể thay đổi chúng khi môi trường làm việc đang tiến hành (sử dụng lệnh ALTER SYSTEM). Xác lập thứ hai hữu dụng đối với các CSDL không thể đưa vào trực tuyến.

Cơ chế cấp đăng ký phiên cho phép DBA kiểm số phiên được kết nối hiện hành và số phiên đồng thời tối đa kể từ khi môi trường làm việc được khởi động. Khung nhìn V$LICENSE trình bày những xác lập hiện hành đối với các giới hạn cấp đăng ký, số phiên hiện hành và số phiên đồng thời nhiều nhất kể từ khi môi trường làm việc được khởi động (phiên "đánh dấu mức nước cao"). DBA có thể sử dụng thông tin này để thẩm định các nhu cầu cấp đăng ký của hệ thống và kế hoạch để nâng cấp hệ thống.

Đối với các môi trường làm việc chạy bằng Oracle Parallel Server, mỗi môi trường làm việc có thể có giới hạn sử dụng đồng thời và giới hạn cảnh báo của riêng nó. Tổng số giới hạn môi trường làm việc không được vượt mức đăng ký sử dụng đồng thời.
Các giới hạn sử dụng đồng thời áp dụng cho tất cả các phiên người dùng, kể cả những phiên được tạo cho những liên kết CSDL đến. Chúng không áp dụng cho các phiên được Oracle tạo

hoặc các phiên tái quy. Các phiên kết nối bằng phần mềm đa công bên ngoài không được cơ chế cấp đăng ký Oracle tính riêng, dù mỗi phiên được tính riêng vào tổng số cấp đăng ký của Oracle. DBA chịu trách nhiệm tính các phiên này.

Cấp đăng ký người dùng có tên

Cấp đăng ký người dùng có tên, sự cho phép này ấn định số người dùng có tên, nơi người dùng có tên là một cá nhân được phép sử dụng Oracle trên máy tính cá biệt. Không có giới hạn được xác lập về số phiên mỗi người dùng có thể sử dụng đồng thời hoặc về số phiên đồng thời đối với CSDL đó.

Cấp đăng ký người dùng có tên cho phép DBA xác lập giới hạn về số người dùng được xác định trong CSDL, kể cả những người dùng được kết nối bằng các liên kết CSDL. Một khi đạt tới giới hạn này, không ai có thể tạo thêm người dùng mới. Cơ chế này thừa nhận mỗi người dùng truy xuất CSDL có một tên người dùng duy nhất trong CSDL ấy và không có hai người dùng chung một tên người dùng.

DBA có thể xác lập giới hạn này trong tập tin tham số của CSDL sao cho nó có tác động khi môi trường làm việc khởi động và có thể thay đổi nó khi môi trường làm việc tiến hành. (bằng cách sử dụng lệnh ALTER SYSTEM). Xác lập thứ hai hữu dụng cho các CSDL không thể đưa vào ngoại tuyến.

Nếu nhiều môi trường làm việc kết nối với cùng CSDL trong một Oracle Parallel Server, tất cả các môi trường làm việc được kết nối sẽ có cùng giới hạn người dùng có tên.

Chương 26

Các đặc quyền và các vai trò

Chương này giải thích bạn có thể điều khiển khả năng của người dùng để tiến hành các điều hoạt hệ thống và để truy xuất các đối tượng giản đồ bằng cách sử dụng các đặc quyền và các vai trò. Chương này gồm:

- **Các đặc quyền**
- **Các đặc quyền hệ thống**
- **Các đặc quyền đối tượng giản đồ**
- **Các vai trỏ**

Thông tin bổ sung:

> Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu của Trusted Oracle về các vai trò và các đặc quyền trong môi trường đó.

Các đặc quyền

Đặc quyền là quyền tiến hành một kiểu câu lệnh cá biệt hoặc để truy xuất một đối tượng người dùng khác. Một số thí dụ của các đặc quyền để có quyền

kết nối với cơ sở dữ liệu (tạo một phiên làm việc)

tạo một bảng
lựa các hàng từ bảng của một người dùng khác
tiến hành một thủ tục định sẵn của người dùng khác.

Bạn cho người dùng các đặc quyền vì vậy những người dùng này có thể thực hiện các tác vụ cần thiết cho tác vụ của họ. Bạn chỉ nên cho người dùng hết sức cần thiết đặc quyền ấy để thực hiện công việc cần thiết. Cho các đặc quyền không cần thiết có thể làm phương hại đến bảo mật. Người dùng có thể nhận được đặc quyền bằng hai cách khác nhau:
Bạn có thể cho người dùng đặc quyền rõ ràng. Thí dụ, bạn có thể cho đặc quyền chèn các bản ghi vào bảng EMP đối với người dùng SCOTT.
Bạn cũng có thể cho các đặc quyền về một vai trò (nhóm những đặc quyền có tên) và sau đó cho một người dùng hoặc nhiều người dùng vai trò đó. Thí dụ, bạn có thể cho các đặc quyền lựa, chèn, cập nhật và xóa các bản ghi từ bảng EMP đối với vai trò có tên CLERK, rồi tới lượt bạn có thể cho các người dùng SCOTT và BRIAN.

Bởi vì các vai trò cho phép quản lý các đặc quyền dễ dàng và tốt hơn, bạn nên cho các vai trò những đặc quyền và không nên cho riêng các người dùng.

Có hai biệt loại gồm các đặc quyền riêng biệt:
- Các đặc quyền hệ thống
- Các đặc quyền đối tượng giản đồ

Thông tin bổ sung:

Các danh sách hoàn chỉnh về những đặc quyền hệ thống và các đối tượng giản đồ cũng như các chỉ lệnh về quản lý đặc quyền xuất hiện ở "Oracle8 Administrator's Guide".

Các đặc quyền hệ thống

Một đặc quyền hệ thống là quyền thực hiện một hoạt động cá biệt hoặc thực hiện một hoạt động trên đối tượng giản đồ bất kỳ của một kiểu cá biệt. Thí dụ, các đặc quyền tạo các không gian bảng và xóa các hàng của một bảng bất kỳ trong cơ sở dữ liệu là các đặc quyền hệ thống. Có trên 60 đặc quyền hệ thống.

Cho và thu hồi các đặc quyền hệ thống

Bạn có thể cho hoặc thu hồi các đặc quyền hệ thống đối với các người dùng hoặc các vai trò. Nếu bạn cho các vai trò các đặc quyền hệ thống, bạn có thể sử dụng các vai trò để quản lý các đặc quyền hệ thống (thí dụ, các vai trò cho phép các đặc quyền có thể được sử dụng chọn lọc).

> Lưu ý:
> Thường bạn chỉ nên cho nhân viên điều hành hoặc các Developer hệ thống các đặc quyền, vì những người dùng cuối thường không cần các khả năng kết hợp.

Các đặc quyền hệ thống được cho hoặc được thu hồi từ những người dùng và các vai trò sử dụng như sau:
Hộp thoại Grant System Privileges/Roles và hộp thoại Revoke System Privileges/Roles của Oracle Enterprise Manager
Các câu lệnh GRANT và REVOKE của Oracle

Ai có thể cho hoặc thu hồi các đặc quyền?

Chỉ những người dùng đã được cho đặc quyền hệ thống riêng biệt có ADMIN OPTION hoặc những người dùng có đặc quyền hệ thống GRANT ANY PRIVILEGE (thường là những người dùng

điều hành bảo mật hoặc cơ sở dữ liệu) có thể cho hoặc thu hồi các đặc quyền đối với những người dùng khác.

Các đặc quyền đối tượng giản đồ

Đặc quyền đối tượng giản đồ ("object privilege") là đặc quyền hoặc quyền thực hiện một hoạt động cá biệt trên một bảng, khung nhìn, chuỗi, thủ tục, hàm hoặc bộ phần mềm riêng biệt. Các đặc quyền đối tượng khác nhau có thể sử dụng cho các kiểu đối tượng giản đồ khác nhau. Thí dụ, đặc quyền xóa các hàng khỏi bảng DEPT là một đặc quyền đối tượng.

Một số đối tượng giản đồ (chẳng hạn như các cụm liên kết, các chỉ mục, các bộ kích khởi và các liên kết cơ sở dữ liệu) không có các đặc quyền đối tượng kết hợp; cách sử dụng chúng được các đặc quyền hệ thống điều khiển. Thí dụ, thay thế một cụm liên kết, người dùng phải sở hữu cụm liên kết hoặc có đặc quyền hệ thống ALTER ANY CLUSTER.

Đối tượng giản đồ và từ đồng nghĩa của nó tương đương với các đặc quyền; nghĩa là các đặc quyền được cho sử dụng để tham chiếu đối tượng cơ sở bằng tên hoặc sử dụng từ đồng nghĩa.

Thí dụ, giả sử có một bảng JWARD.EMP với từ đồng nghĩa có tên JWARD.EMPLOYEE và người dùng JWARD phát hành câu lệnh sau:

```
GRANT SELECT ON emp TO swilliams;
```

Người dùng SWILLIAMS có thể truy vấn JWARD.EMP bằng cách tham chiếu bảng bằng tên hoặc sử dụng từ đồng nghĩa JWARD.EMPLOYEE:

```
SELECT * FROM jward.emp;
SELECT * FROM jward.employee;
```

Nếu bạn cho các đặc quyền đối tượng về một bảng, khung nhìn, chuỗi, thủ tục, hàm hoặc bộ phần mềm với *synonym* (từ đồng nghĩa) đối với đối tượng ấy, thì tác động tương tự như không có từ đồng nghĩa nào được sử dụng. Thí dụ, nếu JWARD muốn cho đặc quyền SELECT đối với bảng EMP tới SWILLIAMS, JWARD có thể phát hành các câu lệnh sau:

```
GRANT SELECT ON emp TO swilliams;
GRANT SELECT ON employee TO swilliams;
```

Nếu từ đồng nghĩa được bỏ, thì tất cả các đặc quyền được cho đối tượng giản đồ cơ sở vẫn có ảnh hưởng, dù các đặc quyền ấy đã được cho bằng cách ấn định từ đồng nghĩa được bỏ.

Cho và thu hồi các đặc quyền đối tượng giản đồ

Các đặc quyền đối tượng giản đồ có thể được cho và thu hồi từ những người dùng và các vai trò. Nếu bạn cho các vai trò các đặc quyền đối tượng, bạn có thể tạo cho các đặc quyền có thể chọn lọc. Các đặc quyền đối tượng đối với những người dùng và các vai trò có thể được cho hoặc được thu hồi bằng cách sử dụng các lệnh GRANT và REVOKE của SQL tương ứng hoặc hộp thoại Add Privilege tới Role/User và hộp thoại Revoke Privilege từ hộp thoại Role/User của Oracle Enterprise Manger.

Ai có thể cho các đặc quyền đối tượng giản đồ ?

Người dùng tự động có các đặc quyền về các đối tượng giản đồ trong giản đồ của họ. Người dùng có thể cho đặc quyền đối tượng bất kỳ trên đối tượng giản đồ bất kỳ mà người ấy sở hữu cho bất kỳ người dùng hoặc vai trò khác. Nếu đặc quyền được cho có GRANT OPTION (của lệnh GRANT), người được cho đặc quyền có thể cho người dùng khác đặc quyền đối tượng ấy; trái lại, người được cho đặc quyền có thể sử dụng đặc quyền ấy mà không thể cho người dùng khác.

Các chủ đề bảo mật bảng

Các đặc quyền đối tượng giản đồ đối với các bảng cho phép bảo mật bảng ở mức các điều hoạt DML và DDL.

Các điều hoạt ngôn ngữ điều tác dữ liệu - Data Manipulation Language (DML)

Các đặc quyền DELETE, INSERT, SELECT và UPDATE cho phép các điều hoạt DELETE, INSERT, SELECT và UPDATE DML tương ứng trên bảng hoặc khung nhìn. Bạn chỉ nên cho các người dùng và các vai trò các đặc quyền này cần truy vấn hoặc điều tác dữ liệu bảng.

Thông tin bổ sung:
Xem Oracle8 SQL ReferenceSee về các điều hoạt DML.

Bạn có thể hạn chế cắc đặc quyền INSERT và UPDATE đối với một bảng về các cột riêng biệt của bảng ấy. Với INSERT lựa chọn, người dùng có đặc quyền có thể chèn hàng với những giá trị đối với các cột lựa; tất cả các cột khác nhận NULL hoặc giá trị mặc định của cột ấy. Với UPDATE lựa chọn, người dùng chỉ có thể cập nhật những giá trị cột riêng biệt của một hàng. Các đặc quyền INSERT và UPDATE lựa chọn được sử dụng để hạn chế người dùng truy xuất dữ liệu nhạy cảm.

Thí dụ, nếu bạn không muốn những người dùng nhập dữ liệu để thay đổi cột SAL của bảng nhân viên, các đặc quyền INSERT và/hoặc UPDATE lựa chọn có thể được cho ngoại trừ cột SAL. (Hoặc một khung nhìn ngoại trừ cột SAL có thể thỏa mãn nhu cầu về tính bảo mật bổ sung.)

Điều hoạt ngôn ngữ định nghĩa dữ liệu - Data Definition Language (DDL)

Các đặc quyền ALTER, INDEX và REFERENCES cho phép các điều hoạt DDL được thực hiện trên bảng. Vì các đặc quyền này cho phép những người dùng khác thay thế hoặc tạo các phụ

thuộc trên bảng, bạn nên cho các đặc quyền một cách thận trọng. Người dùng cố thực hiện điều hoạt DDL trên bảng có thể cần các đặc quyền đối tượng hoặc hệ thống bổ sung (thí dụ, để tạo bộ kích khởi trên bảng, người dùng cần cả đặc quyền đối tượng ALTER TABLE đối với bảng và đặc quyền hệ thống CREATE TRIGGER).

Cũng như có các đặc quyền INSERT và UPDATE, đặc quyền REFERENCES có thể được cho trên các cột riêng biệt của bảng. Đặc quyền cho phép người được cho đặc quyền sử dụng bảng mà đặc quyền được thực hiện bằng khóa cha đối với các khóa ngoại bất kỳ mà người được cho đặc quyền muốn tạo riêng các bảng của họ. Hoạt động này được điều khiển với đặc quyền đặc biệt vì sự hiện hữu các khóa ngoại hạn chế điều tác dữ liệu và các thay đổi bảng có thực hiện đối với khóa cha. Đặc quyền REFERENCE cột ấn định hạn chế người được cho đặc quyền sử dụng các cột có tên (đương nhiên, đặc quyền hạn chế người được đặc quyền phải có ít nhất một khóa chính hoặc khóa duy nhất của bảng cha). Xem "Tính Nguyên Vẹn Dữ Liệu" về các khóa chính, các khóa duy nhất và các hạn định của tính nguyên vẹn dữ liệu.

Các chủ đề bảo mật khung nhìn

Các đặc quyền đối tượng giản đồ đối với các khung nhìn cho phép các điều hoạt DML khác nhau, các đặc quyền đối tượng DML đối với các bảng có thể được áp dụng tương tự với các khung nhìn.

Các đặc quyền cần thiết để tạo các khung nhìn

Để tạo một khung nhìn, bạn phải đáp ứng các đòi hỏi sau:
Bạn phải được cho đặc quyền hệ thống CREATE VIEW (để tạo một khung nhìn trong giản đồ của mình) hoặc CREATE ANY VIEW (để tạo một khung nhìn trong một giản đồ của người dùng khác), cụ thể hoặc qua một vai trò.

Bạn phải được cho đặc quyền SELECT, INSERT, UPDATE hoặc DELETE cụ thể trên tất cả các đối tượng nằm dưới khung nhìn ấy hoặc các đặc quyền hệ thống SELECT ANY TABLE, INSERT ANY TABLE, UPDATE ANY TABLE hay DELETE ANY TABLE. Bạn không thể có các đặc quyền này qua các vai trò.
Hơn nữa, để cho những người dùng khác đặc quyền truy xuất khung nhìn của bạn, bạn phải nhận được (các) đặc quyền đối tượng đối với các đối tượng cơ sở có tùy chọn GRANT OPTION hoặc các đặc quyền hệ thống thích hợp với tùy chọn ADMIN OPTION. Nếu bạn không có chúng, những người được đặc quyền không thể truy xuất khung nhìn của bạn.

Tăng cường tính bảo mật bằng sử dụng các khung nhìn

Để sử dụng một khung nhìn, bạn cần phải có các đặc quyền thích hợp đối với chính khung nhìn ấy. Bạn không cần các đặc quyền dựa trên (các) đối tượng cơ sở nằm dưới khung nhìn ấy.

Các khung nhìn bổ sung thêm các mức bảo mật cho các bảng, tính bảo mật mức bảng và tính bảo mật dựa trên giá trị:
Một khung nhìn có thể cung cấp truy xuất đối với các cột lựa của (các) bảng cơ sở. Thí dụ, bạn có thể xác định một khung nhìn trên bảng EMP để chỉ trình bày các cột EMPNO, ENAME và MGR:

```
CREATE VIEW emp_mgr AS
   SELECT ename, empno, mgr FROM emp;
```

Một khung nhìn có thể cung cấp tính bảo mật dựa trên đối tượng đối với thông tin trong bảng. Mệnh đề WHERE trong định nghĩa của khung nhìn chỉ trình bày các hàng lựa của các bảng cơ sở. Xét hai thí dụ sau:

```
CREATE VIEW lowsal AS
   SELECT * FROM emp
   WHERE sal < 10000;
```

Khung nhìn LOWSAL cho phép truy xuất tất cả cửa sổ hàng của bảng EMP có giá trị lương ít hơn 10000. Lưu ý là tất cả các cột của bảng EMP có thể truy xuất trong khung nhìn LOWSAL.

```
CREATE VIEW own_salary AS
   SELECT ename, sal
   FROM emp
   WHERE ename = USER;
```

Trong khung nhìn OWN_SALARY, chỉ các hàng có ENAME so khớp với người dùng hiện hành của khung nhìn có thể truy xuất. Khung nhìn OWN_SALARY sử dụng cột giả USER có những giá trị luôn luôn tham chiếu người dùng hiện hành. Khung nhìn này kết hợp tính bảo mật mức cột và tính bảo mật dựa trên giá trị.

Các chủ đề bảo mật thủ tục

Đặc quyền đối tượng giản đồ đối với các thủ tục (kể cả các thủ tục độc lập, các hàm và các bộ phần mềm) là EXECUTE. Bạn chỉ nên cho những người dùng cần tiến hành một thủ tục đặc quyền này.

Bạn có thể sử dụng các thủ tục để bổ sung mức bảo mật cơ sở dữ liệu. Người dùng chỉ cần có đặc quyền để tiến hành thủ tục và không cần các đặc quyền trên các đối tượng cơ sở mà thủ tục truy xuất. Bằng cách viết một thủ tục và chỉ cho người dùng đặc quyền EXECUTE, người dùng ấy có thể buộc phải truy xuất các đối tượng tham chiếu bằng thủ tục ấy (nghĩa là, người dùng đó không thể đệ trình các câu lệnh SQL không định trước đối với cơ sở dữ liệu ấy).

Tiến hành thủ tục và các vùng bảo mật

Người dùng có đặc quyền đối tượng EXECUTE đối với một thủ tục riêng biệt có thể tiến hành thủ tục ấy. Người dùng có đặc quyền hệ thống EXECUTE ANY PROCEDURE có thể tiến hành

thủ tục bất kỳ trong cơ sở dữ liệu ấy. Người dùng có thể được cho các đặc quyền qua các vai trò để tiến hành thủ tục.

Khi bạn tiến hành một thủ tục, nó điều hoạt vùng bảo mật của người dùng sở hữu thủ tục ấy, bất kể ai đang tiến hành nó. Do đó, người dùng không cần có các đặc quyền trên các đối tượng tham chiếu, những người dùng thủ tục ít cần được cho các đặc quyền do đó điều khiển truy xuất cơ sở dữ liệu được chặt chẽ hơn.

Các đặc quyền hiện hành của một chủ nhân về một thủ tục định sẵn luôn luôn được điều khiển trước khi thủ tục ấy được tiến hành. Nếu đặc quyền cần thiết về một đối tượng tham chiếu được thu hồi từ một chủ nhân của thủ tục, thì thủ tục ấy không thể được tiến hành bởi chủ nhân hoặc bất kỳ người dùng khác.

Lưu ý:
Tiến hành bộ kích khởi thực hiện theo cùng các mẫu này. Người dùng có thể tiến hành một câu lệnh SQL mà người dùng ấy được cho đặc quyền tiến hành. Bộ kích khởi được kích hoạt. Các câu lệnh trong hoạt động được kích hoạt tạm thời tiến hành dưới vùng bảo mật của người dùng sở hữu bộ kích khởi ấy.

Các đặc quyền hệ thống cần thiết để tạo hoặc thay đổi thủ tục

Để tạo một thủ tục, người dùng phải có các đặc quyền hệ thống CREATE PROCEDURE hoặc CREATE ANY PROCEDURE. Để thay thế một thủ tục, nghĩa là để biên dịch thủ công lại một thủ tục, người dùng phải sở hữu thủ tục ấy hoặc có đặc quyền hệ thống ALTER ANY PROCEDURE.

Người dùng sở hữu thủ tục ấy cũng phải có các đặc quyền đối với các đối tượng giản đồ được tham chiếu trong thân thủ tục. Để tạo một thủ tục, bạn phải được cho các đặc quyền cần thiết rõ ràng (hệ thống hoặc đối tượng) về tất cả các đối tượng được thủ tục ấy tham chiếu; bạn không thể có được các đặc quyền cần thiết qua

các vai trò. Điều này gồm đặc quyền EXECUTE đối với các đặc quyền được gọi bên trong thủ tục được tạo.

Các bộ kích khởi còn cần các đặc quyền đối với các đối tượng được tham chiếu mà chủ nhân của chúng phải được cho rõ ràng. Các khối PL/SQL nặc danh có thể sử dụng đặc quyền bất kỳ, dù đặc quyền được rõ ràng hoặc qua một vai trò.

Các bộ phần mềm và các đối tượng phần mềm

Một người dùng có đặc quyền đối tượng EXECUTE đối với bộ phần mềm có thể tiến hành một thủ tục (chung) bất kỳ hoặc ham trong bộ phần mềm và truy xuất hay sửa đổi giá trị của một bộ phần mềm (chung) khả biến bất kỳ. Các đặc quyền EXECUTE riêng biệt này không thể được cho đối với các cấu trúc của bộ phần mềm. Do đó, bạn có thể thấy nó để xét hai thay thế thiết lập tính bảo mật khi phát triển các thủ tục, các hàm và các bộ phần mềm rất hữu dụng đối với ứng dụng cơ sở dữ liệu. Những thay thế này được mô tả ở các thí dụ sau.

Thí dụ 1: Thí dụ này trình bày bốn thủ tục được tạo ở các thân của hai bộ phần mềm.

```
CREATE PACKAGE BODY hire_fire AS
  PROCEDURE hire(...) IS
    BEGIN
      INSERT INTO emp . . .
    END hire;
  PROCEDURE fire(...) IS
    BEGIN
      DELETE FROM emp . . .
    END fire;
END hire_fire;

CREATE PACKAGE BODY raise_bonus AS
  PROCEDURE give_raise(...) IS
    BEGIN
```

```
    UPDATE EMP SET sal = . . .
  END give_raise;
 PROCEDURE give_bonus(...) IS
  BEGIN
    UPDATE EMP SET bonus = . . .
  END give_bonus;
END raise_bonus;
```

Để truy xuất tiến hành các thủ tục được cho trước bằng cách cho đặc quyền EXECUTE đối với phần mềm ấy, sử dụng các câu lệnh sau:

```
GRANT EXECUTE ON hire_fire TO big_bosses;
GRANT EXECUTE ON raise_bonus TO little_bosses;
```

Cho đặc quyền EXECUTE đối với bộ phần mềm cung cấp truy xuất đồng nhất đối với tất cả các đối tượng bộ phần mềm.

Thí dụ 2: Thí dụ này trình bày bốn định nghĩa trong thân của một bộ phần mềm đơn. Hai thủ tục độc lập bổ sung và bộ phần mềm được tạo đặc biệt để truy xuất các thủ tục được xác định trong bộ phần mềm chính.

```
CREATE PACKAGE BODY employee_changes AS
  PROCEDURE change_salary(...) IS BEGIN ... END;
  PROCEDURE change_bonus(...) IS BEGIN ... END;
  PROCEDURE insert_employee(...) IS BEGIN ... END;
  PROCEDURE delete_employee(...) IS BEGIN ... END;
END employee_changes;

CREATE PROCEDURE hire
  BEGIN
    employee_changes.insert_employee(...)
  END hire;

CREATE PROCEDURE fire
  BEGIN
```

```
    employee_changes.delete_employee(...)
  END fire;

PACKAGE raise_bonus IS
  PROCEDURE give_raise(...) AS
    BEGIN
      employee_changes.change_salary(...)
    END give_raise;

  PROCEDURE give_bonus(...)
    BEGIN
      employee_changes.change_bonus(...)
    END give_bonus;
```

Sử dụng phương pháp này, các thủ tục thực sự thực hiện công việc ấy (các thủ tục trong bộ phần mềm EMPLOYEE_CHANGES) được xác định ở bộ phần mềm đơn và có thể dùng chung các biến toàn cục được khai báo, các con trỏ, vân vân. Bằng cách khai báo các thủ tục HIRE và FIRE ở cấp cao nhất và bộ phần mềm bổ sung RAISE_BONUS, bạn có thể cho các đặc quyền EXECUTE lựa chọn về các thủ tục ở bộ phần mềm chính:

```
GRANT EXECUTE ON hire, fire TO big_bosses;
GRANT EXECUTE ON raise_bonus TO little_bosses;
```

Các vai trò

Oracle cung cấp quản lý đặc quyền dễ dàng và có điều khiển bằng các vai trò. Các vai trò là các nhóm có tên về các đặc quyền liên quan mà bạn cho những người dùng hoặc các vai trò khác các đặc quyền. Các vai trò được thiết kế để tạo thuận lợi để điều hành những người dùng cuối và các đặc quyền đối tượng giản đồ. Tuy nhiên, các vai trò không có nghĩa là được sử dụng cho các Developer ứng dụng, vì các đặc quyền truy xuất những đối tượng giản đồ trong các cấu trúc lập trình được lưu trữ cần được cho

đặc quyền trực tiếp. Xem "Các câu lệnh ngôn ngữ định nghĩa dữ liệu và các vai trò" về các hạn chế đối với các thủ tục.

Các tính chất của những vai trò này cho phép quản lý đặc quyền trong cơ sở dữ liệu dễ hơn:

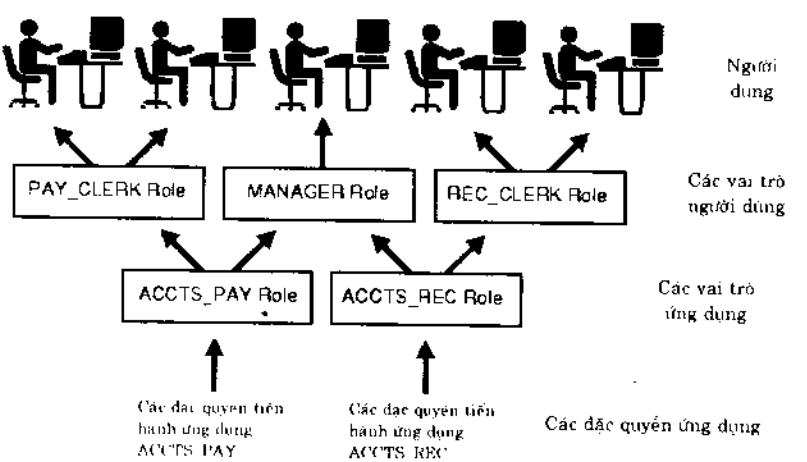
Giảm bớt điều hành đặc quyền	Thay vì cho đặc quyền cùng tập hợp những đặc quyền Rather cụ thể cho nhiều người dùng, bạn có thể cung cấp các đặc quyền cho một nhóm người dùng liên quan thành một vai trò và sau đó chỉ vai trò đó phải cho mỗi thành viên của nhóm đặc quyền.
Quản lý đặc quyền linh hoạt	Nếu các đặc quyền của nhóm phải thay đổi, chỉ những đặc quyền của vai trò ấy cần được sửa đổi. Các vùng bảo mật của tất cả người dùng được cho đặc quyền, vai trò của nhóm tự động phản ảnh các thay đổi xảy ra đối với vai trò ấy.
Khả năng lựa các đặc quyền	Bạn có thể cho phép lựa hoặc vô hiệu hóa các vai trò được cho đối với một người dùng. Điều này cho phép điều khiển riêng những đặc quyền của người dùng theo trường hợp cho trước.
Nhận biết ứng dụng	Các bản ghi từ điển dữ liệu mà các vai trò tồn tại, vì vậy bạn có thể thiết kế các ứng dụng để truy vấn từ điển và tự động cho phép (hoặc vô hiệu hóa) các vai trò được lựa khi một người dùng cố tiến hành ứng dụng bằng một tên người dùng cho trước.
Bảo mật xác định ứng dụng	Bạn có thể bảo vệ sử dụng vai trò bằng mật khẩu. Các ứng dụng có thể được tạo riêng biệt để cho phép một vai trò khi được cung cấp mật khẩu chính xác. Những người dùng không thể cho phép vai trò ấy nếu họ không biết mật khẩu đó.

Thông tin bổ sung:
Các chỉ lệnh để cho phép các vai trò từ một ứng dụng được trình bày ở "Oracle8 Application Developer's Guide".

Các sử dụng thông thường đối với các vai trò

Nói chung, bạn tạo một vai trò để đáp ứng một trong hai mục đích: để quản lý các đặc quyền đối với một ứng dụng cơ sở dữ liệu hoặc để quản lý các đặc quyền đối với một nhóm người dùng. Hình 26 -1 và các phần theo sau mô tả hai cách sử dụng các vai trò.

Hình 26-1 Các sử dụng thông thường đối với các vai trò



Các vai trò ứng dụng

Bạn cho một vai trò ứng dụng tất cả những đặc quyền cần thiết để tiến hành một ứng dụng cơ sở dữ liệu cho trước. Sau đó bạn cho những vai trò khác hoặc các người dùng riêng biệt vai trò ứng dụng đó. Một ứng dụng có thể có nhiều vai trò khác nhau,

với mỗi vai trò được gán một tập hợp các đặc quyền khác nhau để cho phép truy xuất dữ liệu ít nhiều khi sử dụng ứng dụng ấy.

Các vai trò người dùng

Bạn tạo một vai trò người dùng cho một nhóm những người dùng cơ sở dữ liệu với các đòi hỏi đặc quyền thông thường. Bạn quản lý các đặc quyền người dùng bằng cách cho vai trò người dùng các vai trò ứng dụng và các đặc quyền và sau đó cho các người dùng các vai trò người dùng thích hợp.

Cơ chế các vai trò

Các vai trò cơ sở dữ liệu có các tính năng sau:

Một vai trò có thể được cho các đặc quyền đối tượng giản đồ hoặc hệ thống.
Một vai trò có thể được cho các vai trò khác. Tuy nhiên, một vai trò không thể tự cho chính nó và không thể được cho luân chuyển (thí dụ, vai trò A không thể được cho vai trò B nếu vai trò B đã được cho vai trò A trước đó).
Người dùng cơ sở dữ liệu bất kỳ có thể được cho một vai trò bất kỳ.
Vào thời điểm định trước, người dùng được cho mỗi vai trò được cho phép hoặc bị vô hiệu hóa. Vùng bảo mật của người dùng có các đặc quyền về tất cả các vai trò hiện được cho phép đối với người dùng ấy và ngoại trừ các đặc quyền hiện được vô hiệu hóa đối với người dùng ấy. Oracle cho phép các ứng dụng cơ sở dữ liệu và người dùng cho phép và vô hiệu hóa các vai trò để cung cấp tính khả dụng của các đặc quyền.
Một vai trò cho gián tiếp (vai trò được cho một vai trò khác) có thể được cho phép rõ ràng hoặc vô hiệu hóa đối với một người dùng. Tuy nhiên, bằng cách cho phép một vai trò có các vai trò khác, bạn cho phép ngầm định tất cả các vai trò được cho gián tiếp của vai trò được cho trực tiếp.

Cho và thu hồi các vai trò

Bạn cho hoặc thu hồi các vai trò từ những người dùng hoặc các vai trò khác bằng cách sử dụng các tùy chọn sau:
Hộp thoại Grant System Privileges/Roles và hộp thoại Revoke System Privileges/Roles của Oracle Enterprise Manager
Các câu lệnh GRANT và REVOKE của SQL

Các đặc quyền được cho và được thu hồi từ các vai trò sử dụng cùng các tùy chọn. Các vai trò cũng có thể được cho và được thu hồi từ những người dùng bằng cách sử dụng hệ điều hành tiến hành Oracle hoặc bằng các dịch vụ mạng.

Thông tin bổ sung:
Các chỉ lệnh chi tiết về quản lý vai trò ở Oracle8 Administration's Guide.

Ai có thể cho hoặc thu hồi các vai trò?

Người dùng bất kỳ có đặc quyền hệ thống GRANT ANY ROLE có thể cho hoặc thu hồi vai trò bất kỳ (trừ vai trò toàn cục) tới hoặc từ những người dùng khác và các vai trò của cơ sở dữ liệu. Bạn nên thận trọng khi cho đặc quyền hệ thống này vì nó đầy năng lực.

Thông tin bổ sung:
Xem "Các Hệ Thống Cơ Sở Dữ Liệu được Phân Phối của Oracle8".

Người dùng bất kỳ được cho vai trò có ADMIN OPTION có thể cho hoặc thu hồi vai trò đó tới hoặc từ các người dùng khác hoặc các vai trò của cơ sở dữ liệu. Tùy chọn này cho phép các năng lực điều hành đối với các vai trò theo phương thức lựa chọn.

Đặt tên các vai trò

Trong một cơ sở dữ liệu, mỗi tên vai trò phải duy nhất và tên người dùng và tên vai trò không được giống nhau. Không giống như các đối tượng giản đồ, các vai trò không "nằm trong" trong bất kỳ giản đồ nào. Do đó, người dùng tạo một vai trò, có thể được bỏ đi mà không có ảnh hưởng gì đến vai trò ấy.

Các vùng bảo mật của các vai trò và các người dùng

Mỗi vai trò và người dùng có vùng bảo mật riêng duy nhất. Vùng bảo mật của một vai trò gồm các đặc quyền được cho vai trò ấy cùng với các đặc quyền được cho bất kỳ vai trò khác.

Vùng bảo mật của người dùng gồm các đặc quyền về tất cả đối tượng giản đồ trong giản đồ tương ứng, các đặc quyền được cho người dùng và các đặc quyền của các vai trò được cho người dùng hiện được cho phép. (Một vai trò có thể được cho phép đồng thời đối với một người dùng và vô hiệu hóa đối với người dùng khác). Vùng bảo mật của người dùng còn có các đặc quyền và các vai trò được cho nhóm người dùng PUBLIC.

Các khối PL/SQL có tên và các vai trò

Tất cả các vai trò được vô hiệu hóa trong khối PL/SQL có tên bất kỳ (thủ tục định sẵn, hàm hoặc bộ kích khởi)
được tạo trong giản đồ người dùng không có đối tượng đang được tham chiếu trong khối PL/SQL
có thể được tiến hành bằng một người dùng thay vì chủ nhân của khối PL/SQL
Tuy nhiên, các khối PL/SQL nặc danh được tiến hành dựa trên các đặc quyền được cho bằng các vai trò được cho phép.

Khung nhìn SESSION_ROLES trình bày tất cả các vai trò hiện cho phép. Nếu khối PL/SQL có tên truy vấn SESSION_ROLES, thì truy vấn ấy không cho trở lại bất kỳ hàng nào.

Các câu lệnh định nghĩa dữ liệu và các vai trò

Một người dùng cần có một hoặc nhiều đặc quyền để tiến hành câu lệnh ngôn ngữ định nghĩa dữ liệu (DDL) thành công tùy thuộc vào câu lệnh ấy. Thí dụ, để tạo một bảng, người dùng đó phải có đặc quyền hệ thống CREATE TABLE hoặc CREATE ANY TABLE. Để tạo một khung nhìn của bảng của một người dùng khác, bộ tạo đòi hỏi đặc quyền hệ thống CREATE VIEW hoặc CREATE ANY VIEW và đặc quyền đối tượng SELECT đối với bảng ấy hoặc đặc quyền hệ thống SELECT ANY TABLE.

Oracle tránh các phụ thuộc vào các đặc quyền được các vai trò nhận bằng cách hạn chế sử dụng các đặc quyền ấn định trong các câu lệnh DDL nào đó. Các nguyên tắc sau trình bày các hạn chế liên quan đến các câu lệnh DDL:

tất cả những đặc quyền hệ thống và các đặc quyền đối tượng giản đồ cho phép người dùng thực hiện điều hoạt DDL sử dụng được khi nhận được từ một vai trò.

Thí dụ:

System Privileges: các đặc quyền CREATE TABLE, CREATE VIEW và CREATE PROCEDURE.

Schema Object Privileges: các đặc quyền ALTER và INDEX đối với một bảng.

Ngoại trừ: Đặc quyền đối tượng REFERENCES đối với một bảng không thể được sử dụng để xác định một khóa ngoại của một bảng nếu đặc quyền ấy nhận được từ một vai trò.

Tất cả các đặc quyền hệ thống và các đặc quyền đối tượng cho phép người dùng thực hiện điều hoạt DML cần thiết để phát hành câu lệnh DDL không sử dụng được khi nhận được từ một vai trò.

Thí dụ: Nếu một người dùng nhận đặc quyền hệ thống SELECT ANY TABLE hoặc đặc quyền đối tượng SELECT đối với một bảng qua một vai trò, người ấy không thể sử

dụng được đặc quyền nào để tạo khung nhìn trên bảng của một người dùng khác.

Câu lệnh sau làm rõ các sử dụng được cho phép hoặc hạn chế về các đặc quyền nhận được qua các vai trò:

Thí dụ: Giả sử người dùng

- được cho một vai trò có đặc quyền hệ thống CREATE VIEW
- được cho một vai trò có đặc quyền đối tượng SELECT đối với một bảng EMP, nhưng người dùng ấy được cho gián tiếp đặc quyền đối tượng SELECT dịch vụ bảng EMP ấy
- được cho trực tiếp đặc quyền đối tượng SELECT đối với bảng DEPT

Các đặc quyền được cho trực tiếp và gián tiếp sau:

Người dùng có thể phát hành các câu lệnh SELECT trên cả bảng EMP và DEPT.

Dù người dùng có cả đặc quyền CREATE VIEW và SELECT đối với bảng EMP (cả hai qua một vai trò), nhưng người dùng ấy không thể tạo một khung nhìn có thể sử dụng trên bảng EMP, vì đặc quyền đối tượng SELECT đối với bảng EMP đã cho qua một vai trò. Bất kỳ khung nhìn được tạo sẽ phát sinh lỗi khi được truy xuất.

Người dùng có thể tạo một khung nhìn trên bảng DEPT vì có đặc quyền CREATE VIEW (qua một vai trò) và đặc quyền SELECT đối với bảng DEPT (trực tiếp).

Các vai trò được xác định trước

Các vai trò CONNECT, RESOURCE, DBA, EXP_FULL_DATABASE và IMP_FULL_DATABASE được xác định tự động đối với các cơ sở dữ liệu của Oracle. Các vai trò này được cung cấp để tương thích về trước đối với các phiên bản có

trước của Oracle và có thể được sửa đổi bằng cách tương tự như bất kỳ vai trò khác trong một cơ sở dữ liệu của Oracle.

Hệ điều hành và các vai trò

Trong một số môi trường, bạn có thể điều hành bảo mật cơ sở dữ liệu bằng cách sử dụng hệ điều hành. Hệ điều hành ấy có thể được sử dụng để quản lý cho (và thu hồi) các vai trò cơ sở dữ liệu và để quản lý các định danh mật khẩu của chúng.
Khả năng này không thể sử dụng trên tất cả các hệ điều hành.

Thông tin bổ sung:
Xem tài liệu Oracle hệ điều hành xác định những chi tiết về các vai trò quản lý bằng hệ điều hành.

Các vai trò trong môi trường được phân phối

Khi bạn sử dụng các vai trò trong môi trường cơ sở dữ liệu được phân phối, bạn phải chắc chắn tất cả vai trò cần thiết được xác lập là các vai trò mặc định đối với phiên (từ xa) được phân phối. Bạn không thể cho phép các vai trò khi kết nối với một cơ sở dữ liệu từ xa trong một phiên cơ sở dữ liệu cục bộ. Thí dụ, bạn không thể tiến hành thủ tục từ xa để cố cho phép một vai trò ở vùng địa chỉ từ xa.

Thông tin bổ sung:
Để có thêm thông tin về các môi trường cơ sở dữ liệu được phân phối, xem Các Hệ Thống Cơ Sở Dữ Liệu Được Phân Phối Của Oracle.

Chương 27

Kiểm soát

Chương này đề cập về đặc trưng kiểm soát (auditing) của Oracle, gồm:

- **Giới thiệu kiểm soát**
- **Kiểm soát câu lệnh**
- **Kiểm soát đặc quyền**
- **Kiểm soát đối tượng giản đồ**
- **Kiểm soát câu lệnh tập trung, đặc quyền và đối tượng giản đồ**

Thông tin bổ sung:

Nếu bạn đang sử dụng Trusted Oracle, xem tài liệu Trusted Oracle về kiểm soát trong môi trường đó.

Giới thiệu kiểm soát

Kiểm soát giám sát và ghi lại các hoạt động cơ sở dữ liệu người dùng lựa. Kiểm soát thường được sử dụng để

- khảo sát hoạt động bị nghi ngờ. Thí dụ, nếu người dùng không hợp lệ xóa dữ liệu khỏi các bảng, thì người điều

hành bảo mật có thể quyết định kiểm soát tất cả những kết nối đối với cơ sở dữ liệu ấy và tất cả điều tác xóa các hàng thành công và không thực hiện được khỏi tất cả các bảng trong cơ sở dữ liệu đó.

- giám sát và tập hợp dữ liệu về các hoạt động của cơ sở dữ liệu riêng biệt. Thí dụ, người điều hành có thể thu thập các thống kê về các bảng nào được cập nhật, bao nhiêu I/O logic được thực hiện hoặc bao nhiêu người dùng đồng thời vào thời gian cao điểm.

Các đặc trưng kiểm soát

Phần này trình bày các đặc trưng về cơ chế kiểm soát của Oracle.

Các kiểu kiểm soát

Oracle hỗ trợ ba kiểu kiểm soát chung:

Kiểm soát câu lệnh	Kiểm soát lựa chọn các câu lệnh SQL chỉ liên quan đến kiểu câu lệnh, không phải cửa sổ đối tượng giản đồ ấn định mà nó điều hoạt. Các tùy chọn kiểm soát câu lệnh thường rộng, kiểm soát sử dụng nhiều kiểu hoạt động liên quan trên mỗi tùy chọn. Thí dụ, AUDIT TABLE truy tìm nhiều câu lệnh DDL bất kể bảng mà chúng được phát hành. Bạn có thể thiết lập kiểm soát câu lệnh để kiểm soát các người dùng được lựa hoặc mọi người dùng trong cơ sở dữ liệu ấy.
Kiểm soát đặc quyền	Kiểm soát lựa chọn sử dụng các đặc quyền hệ thống có năng lực thực hiện các hoạt động tương ứng, chẳng hạn như AUDIT CREATE TABLE. Kiểm soát đặc quyền được tập trung hơn kiểm soát câu lệnh vì nó chỉ

	kiểm soát sử dụng đặc quyền đích. Bạn có thể thiết lập kiểm soát đặc quyền để kiểm soát một người dùng được lựa hoặc mọi người dùng trong cơ sở dữ liệu đó.
Kiểm soát đối tượng giản đồ	Kiểm soát lựa chọn những câu lệnh ấn định trên một đối tượng giản đồ cá biệt, chẳng hạn như AUDIT SELECT ON EMP. Kiểm soát đối tượng giản đồ rất tập trung, chỉ kiểm soát câu lệnh ấn định trên một đối tượng giản đồ riêng biệt. Kiểm soát đối tượng giản đồ luôn luôn áp dụng cho tất cả người dùng của cơ sở dữ liệu ấy.

Tập trung kiểm soát

Oracle cho phép các tùy chọn kiểm soát để được tập trung hoặc mở rộng. Bạn có thể:

- kiểm soát các câu lệnh được tiến hành thành công hoặc không thành công hay cả hai
- kiểm soát các câu lệnh tiến hành một lần trên mỗi phiên người dùng hoặc một lần cứ mỗi lần câu lệnh ấy được tiến hành
- kiểm soát các hoạt động của tất cả người dùng hoặc của một người dùng riêng biệt.

Các bản ghi kiểm soát và các vệt kiểm soát

Các bản ghi gồm thông tin, chẳng hạn như điều hoạt đã được kiểm soát, người dùng thực hiện điều hoạt ấy và ngày tháng và giờ điều hoạt. Các bản ghi kiểm soát có thể được lưu trữ trong bảng từ điển dữ liệu, được gọi là vệt kiểm soát hoặc vệt kiểm soát hệ điều hành.

Vệt kiểm soát cơ sở dữ liệu là một bảng đơn có tên AUD$ trong giản đồ SYS của mỗi từ điển dữ liệu của cơ sở dữ liệu Oracle. Nhiều khung nhìn được xác định trước được cung cấp để giúp người dùng sử dụng thông tin trong bảng này.

Thông tin bổ sung:
Các chỉ lệnh về tạo và sử dụng các khung nhìn này được tìm thấy ở Oracle8 Administrator's Guide.

Các bản ghi vệt kiểm soát có thể có các kiểu thông tin khác nhau, tùy theo các sự kiện được kiểm soát và các tùy chọn kiểm soát được xác lập. Thông tin sau luôn luôn có trong mỗi bản ghi kiểm soát, với điều kiện thông tin ấy có ý nghĩa đối với hoạt động kiểm soát cá biệt:

- tên người dùng
- dấu định danh người dùng
- dấu định danh đầu cuối
- tên của đối tượng giản đồ được truy xuất
- điều hoạt được thực hiện hoặc bị xâm phạm
- mã hoàn tất của điều hoạt ấy
- tem ngày tháng và giờ
- các đặc quyền hệ thống được sử dụng (kể cả các đặc quyền MAC đối với Trusted Oracle)
- nhãn của phiên người dùng (chỉ đối với Trusted Oracle)
- tên của đối tượng giản đồ được truy xuất (chỉ đối với Trusted Oracle only)

Vệt kiểm soát hệ điều hành được mã hóa và không đọc được, nhưng được giải mã trong các tập tin từ điển dữ liệu và các thông báo lỗi như sau:

Action Code	Thông báo này mô tả điều hoạt được thực hiện hoặc bị xâm phạm. Bảng từ điển dữ liệu AUDIT_ACTIONS có danh sách của những mã này và các mô tả của chúng.
Privileges Used	Thông báo này mô tả bất kỳ các đặc quyền hệ thống được sử dụng để thực hiện điều hoạt. Bảng SYSTEM_PRIVILEGE_MAP liệt kê tất cả những mã này và các mô tả của chúng.
Completion Code	Thông báo này mô tả kết quả của điều hoạt bị xâm phạm. Các điều hoạt thành công cho trở lại giá trị zêrô; các điều hoạt không thành công cho trở lại mã lỗi của Oracle mô tả lý do tại sao điều hoạt ấy không thành công. Những mã này được lưu trữ ở Các Oracle Error Messages.

Các cơ chế kiểm soát

Phần này giải thích các cơ chế được các đặc trưng kiểm soát Oracle sử dụng.

Khi nào các bản ghi kiểm soát được phát sinh?

Ghi những thông tin kiểm soát có thể được cho phép hoặc vô hiệu hóa. Tính năng này cho phép bất kỳ người dùng cơ sở dữ liệu hợp lệ xác lập các tùy chọn kiểm soát vào bất kỳ lúc nào nhưng duy trì điều khiển ghi thông tin kiểm soát cho người điều hành kiểm soát.

Thông tin bổ sung:
Các hướng dẫn cho phép và vô hiệu hóa kiểm soát được tìm thấy ở Oracle Administrator's Guide.

Khi kiểm soát được cho phép trong cơ sở dữ liệu, bản ghi kiểm soát được phát sinh khi pha tiến hành câu lệnh.

> Lưu ý:
> Nếu bạn chưa quen với các pha khác nhau về xử lý câu lệnh SQL và SQL dùng chung, xem "SQL và PL/SQL" về thông tin nền liên quan đến đề cập sau.

Các câu lệnh SQL bên trong các đơn vị chương trình PL/SQL được kiểm soát riêng nếu cần, khi đơn vị chương trình được tiến hành.

Sự phát sinh và chèn bản ghi vệt kiểm soát không phụ thuộc vào giao dịch của người dùng; do đó nếu giao dịch của người dùng được hồi phục về trạng thái cũ, thì bản ghi vệt kiểm soát vẫn được cam kết.

Lưu ý:
Các bản ghi kiểm soát không bao giờ được phát sinh bằng các phiên được thiết lập bởi SYS người dùng hoặc các kết nối với những đặc quyền người điều hành. Các kết nối của những người dùng này bỏ qua những đặc trưng nào đó ở bên trong của Oracle để cho phép ấn định các hoạt động điều hành xảy ra (thí dụ, khởi động, đóng, phục hồi cơ sở dữ liệu, vân vân.)

Các sự kiện luôn luôn được kiểm soát với vệt kiểm soát hệ điều hành

Bất kể kiểm soát cơ sở dữ liệu có được cho phép hay không, Oracle luôn luôn ghi một số hoạt động liên quan đến cơ sở dữ liệu vào vệt kiểm soát hệ điều hành:

Khởi động môi trường làm việc	Bản ghi kiểm soát được phát sinh mô tả chi tiết người dùng OS khởi động môi trường làm việc, dấu định danh đầu cuối của người dùng, tem ngày giờ và kiểm soát cơ sở dữ liệu có được cho phép hay không. Những thông tin này được ghi vào vệt kiểm soát OS vì vệt kiểm soát cơ sở dữ liệu không thể sử dụng cho tới sau khi khởi động đã hoàn tất thành công. Ghi trạng thái kiểm soát cơ sở dữ liệu vào lúc khởi động ngăn cản thêm người điều hành khởi động lại một cơ sở dữ liệu có kiểm soát cơ sở dữ liệu được vô hiệu hóa để chúng có thể thực hiện các hoạt động chưa được kiểm soát.
Đóng môi trường làm việc	Bản ghi kiểm soát được phát sinh mô tả chi tiết đóng môi trường làm việc của người dùng OS, dấu định danh đầu cuối của người dùng, tem về giờ và ngày tháng.
Các kết nối với cơ sở dữ liệu có người điều hành privileges	Bản ghi kiểm soát được phát sinh mô tả chi tiết người dùng OS kết nối Oracle với các đặc quyền người điều hành. Điều này cung cấp khả năng kế toán về những người dùng được kết nối với các đặc quyền người điều hành.

Trên các hệ điều hành không truy xuất được vệt kiểm soát Oracle, những bản ghi vệt kiểm soát này được đưa vào tập tin vệt kiểm soát của Oracle trong cùng thư mục bằng các tập tin ghi vệt xử lý nền.

Thông tin bổ sung:
Xem tài liệu Oracle xác định hệ điều hành về vệt kiểm soát hệ điều hành.

Khi nào các tùy chọn kiểm soát có tác động?

Những tham chiếu câu lệnh và kiểm soát đặc quyền có tác động vào lúc người dùng cơ sở dữ liệu kết nối với cơ sở dữ liệu ấy vẫn còn tác động trong thời gian của phiên. Một phiên không biết được các tác động của các tùy chọn câu lệnh hoặc kiểm soát đặc quyền được xác lập hoặc bị thay đổi. Câu lệnh được sửa đổi hoặc các tùy chọn kiểm soát đặc quyền chỉ ảnh hưởng khi phiên hiện hành được kết thúc và một phiên mới được tạo. Trái lại, các thay đổi đối với các tùy chọn đối tượng giản đồ trở thành có tác động đối với các phiên hiện hành ngay.

Kiểm soát trong cơ sở dữ liệu được phân phối

Kiểm soát là khu vực tự trị; một môi trường làm việc chỉ kiểm soát các câu lệnh được người dùng kết nối phát hành trực tiếp. Nút Oracle cục bộ không thể kiểm soát các hoạt động xảy ra trong cơ sở dữ liệu từ xa. Bởi vì các kết nối từ xa được thiết lập bằng tài khoản người dùng của liên kết cơ sở dữ liệu, nút Oracle từ xa kiểm soát các câu lệnh được phát hành bằng kết nối của liên kết cơ sở dữ liệu ấy. Xem "Các Cơ Sở Dữ Liệu Được Phân Phối" về các cơ sở dữ liệu được phân phối và các liên kết cơ sở dữ liệu.

Kiểm soát đối với vệt kiểm soát OS

Cả Oracle và Trusted Oracle đều cho phép các bản ghi vệt kiểm soát được định hướng vào một vệt kiểm soát hệ điều hành nếu hệ điều hành ấy tạo cho vệt kiểm soát như thế để Oracle có thể sử dụng. Trên một số hệ điều hành, các bản ghi kiểm soát này được viết vào tập tin bên ngoài cơ sở dữ liệu với định dạng tương tự như các tập tin vệt Oracle khác.

Thông tin bổ sung:
Xem tài liệu Oracle xác định nền để xem đặc trưng này có được bổ sung vào hệ điều hành của bạn không.

Cả Oracle và Trusted Oracle đều cho phép các hoạt động nào đó luôn luôn được kiểm soát tiếp tục, ngay cả khi vệt kiểm soát hệ điều hành (hoặc tập tin hệ điều hành có các bản ghi kiểm soát) không thể ghi bản ghi kiểm soát ấy. Nguyên nhân thông thường của điều này là vệt kiểm soát hệ điều hành hoặc hệ thống tập tin đầy và không thể nhận các bản ghi mới.

Các người điều hành hệ thống cấu hình kiểm soát OS sẽ bảo đảm vệt kiểm soát hoặc hệ thống tập tin không hoàn toàn đầy. Phần lớn các hệ điều hành cung cấp cho những người điều hành với đầy đủ thông tin và cảnh báo để điều này không xảy ra. Tuy nhiên, lưu ý cấu hình kiểm soát sử dụng vệt kiểm soát cơ sở dữ liệu xóa bỏ tình trạng có thể bị xâm phạm này, vì Oracle server ngăn cản các sự kiện được kiểm soát xảy ra nếu vệt kiểm soát không thể nhận bản ghi kiểm soát cơ sở dữ liệu ấy cho câu lệnh.

Kiểm soát câu lệnh

Kiểm soát câu lệnh là kiểm soát lựa chọn của các nhóm câu lệnh có liên quan rơi vào hai biệt loại sau:

- các câu lệnh DDL, đối với kiểu cấu trúc cơ sở dữ liệu cá biệt hoặc đối tượng giản đồ, mà không phải cấu trúc có tên hoặc đối tượng giản đồ cá biệt (thí dụ, AUDIT TABLE kiểm soát tất cả các câu lệnh CREATE và DROP TABLE)
- các câu lệnh DML, đối với kiểu cấu trúc cơ sở dữ liệu cá biệt hoặc đối tượng giản đồ, mà không phải cấu trúc có tên hoặc đối tượng giản đồ cá biệt (thí dụ, AUDIT SELECT TABLE kiểm soát tất cả các câu lệnh SELECT . . . FROM TABLE/VIEW/SNAPSHOT, bất kể bảng, khung nhìn hoặc ảnh chụp nhanh)

Kiểm soát câu lệnh có thể rộng hoặc tập trung, kiểm soát các hoạt động của tất cả người dùng cơ sở dữ liệu hoặc những hoạt

động của chỉ một danh sách lựa chọn của những người dùng cơ sở dữ liệu.

Kiểm soát đặc quyền

Kiểm soát đặc quyền là kiểm soát lựa chọn về các câu lệnh cho phép sử dụng đặc quyền hệ thống. Thí dụ, kiểm soát của đặc quyền hệ thống SELECT ANY TABLE kiểm soát các câu lệnh của những người dùng được tiến hành bằng cách sử dụng đặc quyền hệ thống SELECT ANY TABLE. Bạn có thể kiểm soát cách sử dụng đặc quyền hệ thống bất kỳ.

Trong mọi trường hợp, các đặc quyền chủ nhân và các đặc quyền đối tượng giản đồ được kiểm trước các đặc quyền hệ thống. Nếu chủ nhân và các đặc quyền đối tượng giản đồ đủ đáp ứng để cho phép hoạt động, thì hoạt động ấy không phải kiểm soát.

Nếu câu lệnh tương tự và các tùy chọn kiểm soát đặc quyền được xác lập, thì bản ghi kiểm soát đơn được phát sinh. Thí dụ, nếu tham chiếu câu lệnh và đặc quyền hệ thống CREATE TABLE cả hai đều được kiểm soát, thì chỉ bản ghi kiểm soát đơn được phát sinh mỗi lần bảng được tạo.

Kiểm soát đặc quyền tập trung hơn kiểm soát câu lệnh vì mỗi tùy chọn kiểm soát các kiểu câu lệnh riêng, không phải là danh sách các câu lệnh có liên quan. Thí dụ, tùy chọn kiểm soát câu lệnh TABLE kiểm soát các câu lệnh CREATE TABLE, ALTER TABLE và DROP TABLE, trong khi tùy chọn kiểm soát đặc quyền CREATE TABLE chỉ kiểm soát các câu lệnh CREATE TABLE, vì chỉ câu lệnh CREATE TABLE đòi hỏi đặc quyền CREATE TABLE.

Giống như kiểm soát câu lệnh, kiểm soát đặc quyền có thể kiểm soát các hoạt động của tất cả người dùng cơ sở dữ liệu hoặc các

hoạt động của chỉ một danh sách lựa chọn của những người dùng cơ sở dữ liệu.

Kiểm soát đối tượng giản đồ

Kiểm soát đối tượng giản đồ là kiểm soát chọn lựa những câu lệnh DML riêng biệt (kể cả những truy vấn) và các câu lệnh GRANT và REVOKE cho các đối tượng giản đồ ấn định. Kiểm soát đối tượng giản đồ kiểm soát các điều hoạt được các đặc quyền đối tượng giản đồ cho phép, chẳng hạn như câu lệnh SELECT hoặc DELETE trên bảng cho trước, cùng như các câu lệnh GRANT và REVOKE điều khiển các đặc quyền đó.

Bạn có thể kiểm soát các câu lệnh tham chiếu các bảng, khung nhìn và các bộ phần mềm (các thủ tục trong các bộ phần mềm không thể được kiểm soát riêng).

Các câu lệnh tham chiếu các cụm liên kết, các liên kết cơ sở dữ liệu, các chỉ mục hoặc các từ đồng nghĩa không được kiểm soát trực tiếp. Tuy nhiên, bạn có thể kiểm soát truy xuất các đối tượng giản đồ gián tiếp bằng cách kiểm soát các điều hoạt ảnh hưởng đến bảng cơ sở.

Các tùy chọn kiểm soát đối tượng giản đồ luôn luôn được xác lập cho tất cả những người dùng cơ sở dữ liệu; những tùy chọn này không thể được xác lập cho một danh sách riêng biệt những người dùng. Bạn có thể xác lập các tùy chọn kiểm soát đối tượng giản đồ mặc định cho tất cả những đối tượng giản đồ có thể kiểm soát.

Thông tin bổ sung:
Xem "AUDIT (Schema Objects)" ở Oracle8 SQL Reference.

Các tùy chọn kiểm soát đối tượng giản đồ đối với các khung nhìn và các thủ tục

Các khung nhìn và các thủ tục (kể cả các hàm được lưu trữ, các bộ phần mềm và bộ kích khởi) tham chiếu các đối tượng giản đồ cơ sở bằng định nghĩa của chúng. Do đó, kiểm soát liên quan đến các khung nhìn và các thủ tục có nhiều đặc điểm riêng biệt. Nhiều bản ghi kiểm soát có thể được phát sinh do sử dụng khung nhìn hoặc một thủ tục: sử dụng khung nhìn hoặc thủ tục tùy thuộc vào các tùy chọn kiểm soát được cho phép và các câu lệnh SQL được phát hành do sử dụng khung nhìn hoặc thủ tục tùy thuộc vào các tham chiếu kiểm soát được cho phép về những đối tượng giản đồ cơ sở (kể cả các tùy chọn kiểm soát mặc định).

Xét một loạt các câu lệnh SQL sau:

```
AUDIT SELECT ON emp;

CREATE VIEW emp_dept AS
  SELECT empno, ename, dname
    FROM emp, dept
    WHERE emp.deptno = dept.deptno;
AUDIT SELECT ON emp_dept;

SELECT * FROM emp_dept;
```

Do truy vấn trên khung nhìn EMP_DEPT, hai bản ghi kiểm soát: một đối với truy vấn trên khung nhìn EMP_DEPT và một đối với truy vấn trên bảng cơ sở EMP (gián tiếp qua khung nhìn EMP_DEPT view). Truy vấn trên bảng cơ sở DEPT không phát sinh bản ghi kiểm soát vì tùy chọn kiểm soát SELECT đối với bảng này không được cho phép. Tất cả những bản ghi kiểm soát liên quan đến người dùng đã truy vấn khung nhìn EMP_DEPT.

Các tùy chọn kiểm soát đối với một khung nhìn hoặc thủ tục được xác định khi khung nhìn hoặc thủ tục ấy lần đầu tiên được

sử dụng và được đưa vào vùng chia dùng chung. Các tham chiếu kiểm soát này còn được xác lập cho tới khi khung nhìn hoặc thủ tục ấy được xóa sạch khỏi vùng chia dùng chung và được thay thế tiếp tục vào vùng chia đó. Kiểm soát đối tượng giản đồ thẩm định đối tượng giản đồ đó trong cache và làm cho nó được nạp lại. Bất kỳ thay đổi đối với các tùy chọn của những đối tượng giản đồ cơ sở không được quan sát bằng các khung nhìn và các thủ tục trong vùng chia dùng chung.

Tiếp tục với thí dụ trên, nếu kiểm soát những câu lệnh SELECT được tắt đi đối với bảng EMP, sử dụng khung nhìn EMP_DEPT sẽ không còn phát sinh bản ghi kiểm soát cho bảng EMP ấy nữa.

Kiểm soát câu lệnh tập trung, đặc quyền và đối tượng giản đồ

Oracle cho phép bạn kiểm soát câu lệnh tập trung, đặc quyền và đối tượng giản đồ trong ba lãnh vực:

- các tiến hành thành công và không thành công của câu lệnh SQL được kiểm soát.
- kiểm soát BY SESSION và BY ACCESS

đối với những người dùng riêng biệt hoặc đối với tất cả người dùng trong cơ sở dữ liệu (chỉ kiểm soát câu lệnh và đặc quyền)

Các tiến hành kiểm soát câu lệnh thành công và không thành công

Đối với kiểm soát câu lệnh, đặc quyền và đối tượng giản đồ, Oracle cho phép kiểm soát lựa chọn về các tiến hành câu lệnh thành công, các câu lệnh cố tiến hành nhưng không thành công hoặc cả hai. Do đó, bạn có thể giám sát các hoạt động dù các câu lệnh được kiểm soát không thực hiện thành công.

Bạn có thể kiểm soát tiến hành câu lệnh không thành công chỉ nếu một câu lệnh SQL hợp lệ được phát hành nhưng không

thành công vì không có quyền hạn thích hợp hoặc vì nó tham chiếu đối tượng giản đồ không tồn tại. Các câu lệnh tiến hành không thành công chính vì chúng không hợp lệ, không thể được kiểm soát. Tham số tham chiếu kiểm soát đặc quyền được cho phép được xác lập để kiểm soát các tiến hành câu lệnh không thành công sử dụng đặc quyền hệ thống đích nhưng đã không thành công vì các lý do khác (chẳng hạn như khi CREATE TABLE được xử lý nhưng câu lệnh CREATE TABLE không thành công do thiếu chỉ tiêu đối với không gian bảng ấn định).

Sử dụng biểu mẫu lệnh AUDIT, bạn có thể có:

- tùy chọn WHENEVER SUCCESSFUL, chỉ để kiểm soát các tiến hành thành công của câu lệnh được kiểm soát
- tùy chọn WHENEVER NOT SUCCESSFUL, chỉ để kiểm soát các tiến hành không thành công câu lệnh được kiểm soát
- cả hai tham chiếu trên, để kiểm soát cả hai tiến hành thành công và không thành công câu lệnh được kiểm soát

Kiểm soát BY SESSION so với BY ACCESS

Phần lớn các tùy chọn kiểm soát có thể được xác lập để chỉ các bản ghi kiểm soát sẽ được phát sinh nếu câu lệnh được kiểm soát được phát hành nhiều lần trong một người dùng đơn. Phiên này mô tả sự phân biệt giữa các tùy chọn BY SESSION và BY ACCESS của lệnh AUDIT.

BY SESSION

Đối với kiểu kiểm soát bất kỳ (đối tượng giản đồ, câu lệnh hoặc đặc quyền), BY SESSION chỉ chèn một bản ghi kiểm soát trong vệt kiểm soát trong mỗi người dùng và đối tượng giản đồ khi phiên có một hoạt động kiểm soát.

Phiên là khoảng thời gian khi một người dùng kết nối và thôi kết nối cơ sở dữ liệu Oracle.

Thí dụ 1

Giả sử:

- tùy chọn kiểm soát câu lệnh SELECT TABLE được xác lập là BY SESSION.
- JWARD kết nối cơ sở dữ liệu và phát hành 5 câu lệnh SELECT dựa trên bảng có tên DEPT và sau đó thôi kết nối cơ sở dữ liệu.
- SWILLIAMS kết nối với cơ sở dữ liệu và phát hành 3 câu lệnh SELECT dựa trên bảng EMP và sau đó thôi kết nối cơ sở dữ liệu.

Trong trường hợp này, vệt kiểm soát sẽ có hai bản ghi kiểm soát cho 8 câu lệnh SELECT (mỗi câu lệnh cho một phiên được phát hành một câu lệnh SELECT).

Thí dụ 2

Hoặc giả sử:

Tùy chọn kiểm soát câu lệnh SELECT TABLE được xác lập là BY SESSION.

JWARD kết nối với cơ sở dữ liệu và phát hành 5 câu lệnh SELECT dựa trên bảng có tên DEPT và 3 câu lệnh SELECT dựa trên bảng EMP và sau đó thôi kết nối với cơ sở dữ liệu.

Trong trường hợp này, vệt kiểm soát sẽ có hai bản ghi (mỗi bảng cho một đối tượng giản đồ dựa trên người dùng đã phát hành câu lệnh SELECT trong một phiên).

Lưu ý:

Nếu bạn sử dụng tùy chọn BY SESSION khi định hướng các bản ghi kiểm soát tới vệt kiểm soát hệ điều hành, Oracle phát sinh và lưu trữ một bản ghi kiểm soát mỗi lần một truy xuất được thực hiện. Do đó, trong cấu hình kiểm soát, BY SESSION

tương đương với BY ACCESS.

BY ACCESS

Xác lập kiểm soát BY ACCESS chèn một bản ghi kiểm soát vào vệt kiểm soát cho mỗi tiến hành điều hoạt kiểm soát trong một con trỏ. Các sự kiện làm cho các con trỏ được sử dụng lại:

- một ứng dụng, chẳng hạn như Oracle Forms lưu trữ con trỏ mở để sử dụng lại
- tiến hành con trỏ kế tiếp sử dụng các biến kết gán mới
- các câu lệnh được tiến hành trong các vòng lặp PL/SQL nơi động cơ PL/SQL tối ưu hóa các câu lệnh để sử dụng con trỏ đơn

Lưu ý kiểm soát là NOT không bị ảnh hưởng bởi con trỏ dùng chung; mỗi người dùng tạo các bản ghi vệt kiểm soát riêng vào lúc tiến hành con trỏ lần đầu.

Thí dụ

Giả sử là

Tham chiếu kiểm soát câu lệnh SELECT TABLE được xác lập là BY ACCESS.

JWARD kết nối với cơ sở dữ liệu và phát hành 5 câu lệnh SELECT dựa trên bảng có tên DEPT và sau đó thôi kết nối cơ sở dữ liệu ấy.

SWILLIAMS kết nối cơ sở dữ liệu và phát hành 3 câu lệnh SELECT dựa trên bảng DEPT và sau đó thôi kết nối cơ sở dữ liệu.

Vệt kiểm soát đơn có 8 bản ghi cho 8 câu lệnh SELECT.

Các mặc định và các điều hoạt ngoại trừ

Lệnh AUDIT cho bạn ấn định BY SESSION hoặc BY ACCESS. Tuy nhiên, nhiều tùy chọn kiểm soát chỉ có thể được xác lập là BY ACCESS, gồm:

- tất cả tùy chọn kiểm soát câu lệnh kiểm soát các câu lệnh DDL
- tất cả tùy chọn kiểm soát đặc quyền kiểm soát các câu lệnh DDL
- đối với tất cả các tùy chọn kiểm soát khác, BY SESSION được sử dụng theo mặc định.

Kiểm soát bởi người dùng

Các tùy chọn kiểm soát câu lệnh và đặc quyền có thể kiểm soát các câu lệnh được người dùng bất kỳ phát hành hoặc các câu lệnh được một danh sách các người dùng riêng biệt phát hành, bằng cách tập trung vào các người dùng riêng biệt, bạn có thể giảm tối đa số bản ghi kiểm soát được phát hành.

Thông tin bổ sung:
Xem Oracle8 SQL Reference về kiểm soát bởi người dùng.

Thí dụ

Để kiểm soát các câu lệnh bởi người dùng SCOTT và BLAKE truy vấn hoặc cập nhật một bảng hoặc khung nhìn, phát hành các câu lệnh sau:

```
AUDIT SELECT TABLE, UPDATE TABLE
    BY scott, blake;
```

Chương 28

Phục hồi cơ sở dữ liệu

Chương này giới thiệu các cấu trúc được sử dụng khi phục hồi cơ sở dữ liệu và mô tả tiện ích Recovery Manager để đơn giản hóa sao lưu và các điều hoạt phục hồi. Các chủ đề trong chương này gồm:

- **Giới thiệu phục hồi cơ sở dữ liệu**
- **Các cấu trúc được sử dụng để phục hồi cơ sở dữ liệu**
- **Hồi phục về trạng thái cũ và Rolling Forward**
- **Trình quản lý phục hồi**
- **Thực hiện phục hồi song song**
- **Các tập tin điều khiển**
- **Các sao lưu cơ sở dữ liệu**
- **Khả năng tồn tại**

Thông tin bổ sung:

Các thủ tục cần thiết để tạo và duy trì sao lưu và các cấu trúc phục hồi được đề cập ở Hướng dẫn sao lưu và phục hồi của Oracle.

Giới thiệu phục hồi cơ sở dữ liệu

Trách nhiệm chính của người điều hành cơ sở dữ liệu là chuẩn bị khả năng của phần cứng, phần mềm, mạng, xử lý hoặc sai sót hệ thống. Nếu một sai sót như thế ảnh hưởng đến điều hoạt của hệ thống cơ sở dữ liệu, bạn thường phải phục hồi các cơ sở dữ liệu và đưa các điều hoạt trở lại bình thường càng sớm càng tốt. Phục hồi sẽ bảo vệ các cơ sở dữ liệu và các người dùng được kết hợp tránh các vấn đề không cần thiết và tránh hoặc ít phải sao chép công việc thủ công.

Các tiến trình phục hồi khác nhau tùy thep kiểu sai sót xảy ra, các cấu trúc bị ảnh hưởng và kiểu phục hồi bạn thực hiện. Nếu không có tập tin nào bị mất hoặc bị hư, phục hồi có thể không khác gì lúc khởi động một môi trường làm việc. Nếu dữ liệu đã bị mất, phục hồi đòi hỏi thêm một số bước bổ sung.

Lưu ý:
Recovery Manager là tiện ích để đơn giản hóa các hoạt động sao lưu và phục hồi. Xem "Trình Quản Lý Phục Hồi".

Thông tin bổ sung:
Xem Hướng Dẫn Phục Hồi và Sao Lưu Của Oracle về Recovery Manager và mô tả cách phục hồi dữ liệu bị mất.

Các lỗi và các sai sót

Nhiều vấn đề có thể làm tạm ngưng điều hoạt bình thường của cơ sở dữ liệu Oracle hoặc ảnh hưởng I/O cơ sở dữ liệu đến đĩa. Các phần sau mô tả các kiểu thông thường. Đối với một số vấn đề này, phục hồi tự động và đòi hỏi ít hoạt động hoặc không đòi hỏi hoạt động về phía người dùng cơ sở dữ liệu hoặc người điều hành cơ sở dữ liệu.

Lỗi người dùng

Người điều hành cơ sở dữ liệu có thể phải thực hiện một ít để tránh các lỗi người dùng (thí dụ, ngẫu nhiên bỏ một bảng). Thường, lỗi người dùng có thể được giảm xuống bằng nhờ đào tạo sử dụng cơ sở dữ liệu và các nguyên tắc ứng dụng tăng lên. Hơn nữa, nhờ đặt ra kế hoạch phục hồi có hiệu quả trước, người điều hành có thể tạo thuận lợi cho công việc cần thiết để phục hồi nhiều kiểu lỗi người dùng.

Sai sót câu lệnh

Sai sót câu lệnh xảy ra khi có một sai sót logic khi điều quản câu lệnh trong chương trình Oracle. Thí dụ, giả sử tất cả khối lưu riêng của một bảng (nói cách khác, số khối lưu riêng được ấn định trong tham số MAXEXTENTS của câu lệnh CREATE TABLE) được phân bổ và được điền dữ liệu hoàn toàn; bảng đó đầy hoàn toàn. Một câu lệnh INSERT hợp lệ không thể chèn một hàng vì không có không gian khả dụng. Do đó, nếu được phát hành, câu lệnh không thành công.

Nếu sai sót câu lệnh xảy ra, phần mềm Oracle hoặc hệ điều hành cho trở lại mã lỗi hoặc thông báo lỗi. Sai sót câu lệnh thường không đòi hỏi hoạt động hoặc các bước phục hồi nào; Oracle tự động điều chỉnh sai sót câu lệnh bằng cách hồi phục các ảnh hưởng về trạng thái cũ (nếu có) của câu lệnh và cho trở lại điều khiển ứng dụng đó. Người dùng có thể chỉ tiến hành lại câu lệnh sau khi điều chỉnh vấn đề được thông báo lỗi cho biết.

Sai sót tiến trình

Sai sót tiến trình là một sai sót xảy ra ở người dùng, server hoặc xử lý nền của môi trường làm việc cơ sở dữ liệu (thí dụ, thôi kết nối bất thường hoặc kết thúc tiến trình). Khi sai sót tiến trình xảy ra, tiến trình phụ thuộc không thể tiếp tục hoạt động, dù

những tiến trình khác của môi trường làm việc cơ sở dữ liệu có thể tiếp tục.

Tiến trình nền Oracle PMON phát hiện các tiến trình Oracle bị hủy ngang. Nếu tiến trình bị hủy ngang do xử lý của người dùng hoặc server, PMON giải quyết sự cố bằng cách hồi phục giao dịch hiện hành của tiến trình bị hủy ngang và giải phóng các tài nguyên bất kỳ mà tiến trình này đang sử dụng. Phục hồi tiến trình của server và người dùng bị sai sót tự động. Nếu tiến trình bị hủy ngang là tiến trình nền, thì môi trường làm việc thường không thể tiếp thực hiện chức năng chính xác. Do đó, bạn phải đóng môi trường làm việc và khởi động lại.

Sự cố mạng

Khi hệ thống của bạn sử dụng các mạng (thí dụ, các mạng vùng cục bộ, đường dây điện thoại, vân vân.) để kết nối các trạm làm việc của client với các server cơ sở dữ liệu hoặc để kết nối nhiều server cơ sở dữ liệu để thành lập một hệ thống cơ sở dữ liệu được phân phối, các sai sót mạng (chẳng hạn như các kết nối điện thoại bị hủy ngang hoặc sự cố phần mềm truyền thông mạng) có thể làm gián đoạn hoạt động bình thường của hệ thống cơ sở dữ liệu. Thí dụ:

Sự cố mạng có thể làm gián đoạn tiến trình bình thường của ứng dụng client và gây cho sự cố của tiến trình xảy ra. Trong trường hợp này, xử lý nền Oracle PMON phát hiện và giải quyết tiến trình server bị hủy ngang đối với tiến trình người dùng bị gián đoạn, như được mô tả ở phần trước.

Sự cố mạng có thể làm gián đoạn cam kết hai pha của giao dịch được phân phối. Một khi vấn đề mạng được điều chỉnh, xử lý nền Oracle RECO của mỗi server cơ sở dữ liệu liên quan tự động giải quyết bất kỳ giao dịch được phân phối nào chưa được giải quyết ở tất cả các nút của hệ thống cơ sở dữ liệu được phân phối. Các hệ thống cơ sở dữ liệu được phân phối được đề cập ở "Các Cơ Sở Dữ Liệu Được Phân Phối".

Sai sót môi trường làm việc cơ sở dữ liệu

Sai sót môi trường làm việc cơ sở dữ liệu xảy ra khi một vấn đề phát sinh ngăn cản môi trường làm việc Oracle (SGA và các tiến trình nền) tiếp tục hoạt động. Sai sót môi trường làm việc có thể xảy ra do vấn đề phần cứng, chẳng hạn như hệ điều hành bị sai sót. Sai sót môi trường làm việc còn xảy ra khi bạn phát hành câu lệnh SHUTDOWN ABORT hoặc STARTUP FORCE.

Phục hồi sai sót môi trường làm việc

Phục hồi môi trường làm việc để phục hồi cơ sở dữ liệu về trạng thái giao dịch nhất quán của nó ngay trước khi sai sót môi trường làm việc. Nếu bạn có kinh nghiệm về sai sót môi trường làm việc khi sao lưu trực tuyến, phục hồi môi trường có thể cần thiết. Trong tất cả những trường hợp khác, phục hồi sai sót môi trường làm việc tương đối tự động. Thí dụ, khi sử dụng Oracle Parallel Server, các môi trường làm việc khác thực hiện phục hồi môi trường làm việc. Trong các cấu hình môi trường làm việc đơn, Oracle thực hiện phục hồi môi trường làm việc đối với cơ sở dữ liệu khi cơ sở dữ liệu ấy được khởi động lại (được lắp ghép và được mở sang một môi trường làm việc mới). Chuyển từ trạng thái được lắp ghép sang trạng thái mở kích hoạt tự động phục hồi môi trường làm việc, nếu cần.

Phục hồi môi trường làm việc gồm các bước sau:

Phục hồi về sau (*Rolling forward*) để phục hồi dữ liệu chưa được ghi vào các tập tin dữ liệu, nhưng đã được ghi vào redo log (nhật ký redo log) trực tuyến, kể cả nội dung của các đoạn phục hồi về trạng thái cũ.

Mở cơ sở dữ liệu. Thay vì chờ tất cả những giao dịch được hồi phục về trạng thái cũ trước khi tạo cho cơ sở dữ liệu khả dụng, Oracle cho phép cơ sở dữ liệu được mở ngay khi phục hồi cache hoàn tất. Dữ liệu bất kỳ không bị các giao dịch chưa được phục hồi khóa có thể sử dụng ngay. Đặc trưng này được gọi là khởi động nóng nhanh.

Đánh dấu tất cả những giao dịch khắp hệ thống đã hoạt động vào lúc bị sự cố là DEAD và đánh dấu các đoạn hồi phục về trạng thái cũ có các giao dịch này là PARTLY AVAILABLE.

Phục hồi các giao dịch bị liệt là phần việc của phục hồi SMON.

Phục hồi bất kỳ giao dịch được phân phối chưa giải quyết phải cam kết qua hai pha vào thời gian sai sót môi trường làm việc.

Điểm kiểm gia tăng

Điểm kiểm gia tăng làm tăng hiệu suất phục hồi môi trường làm việc và phục hồi sự cố (một phục hồi môi trường đĩa). Một điểm kiểm gia tăng ghi vị trí trong xâu redo (log) mà phục hồi sự cố hoặc môi trường làm việc cần phải bắt đầu. Vị trí của nhật ký hoạt động này được xác định bởi vùng đệm sai sót cũ nhất trong cache vùng đệm ấy. Thông tin về điểm kiểm gia tăng được duy trì định kỳ ít có phần điều hành chung hoặc không có phần điều hành chung khi tiến trình hoạt động bình thường.

Hiệu suất phục hồi tỷ lệ gần bằng với số vùng đệm chưa được ghi vào cơ sở dữ liệu trước khi bị sự cố. Bạn có thể tác động đến hiệu suất phục hồi sự cố hoặc môi trường làm việc bằng cách xác lập tham số khởi hoạt DB_BLOCK_MAX_DIRTY_TARGET, để ấn định biên trên về số các vùng đệm bị sai sót có thể hiện hữu ở cache vùng đệm của một môi trường làm việc vào bất kỳ lúc nào. Do đó, nó có thể ảnh hưởng đến thời gian phục hồi đối với các trường hợp nơi cache vùng đệm quá lớn hoặc nơi có các giới hạn chặt chẽ về thời gian phục hồi sự cố/môi trường làm việc. Những giá trị này càng nhỏ tác động phần điều hành chung càng cao khi tiến trình bình thường vì nhiều vùng đệm phải được ghi. Mặt khác, giá trị của tham số này càng nhỏ thì hiệu suất phục hồi càng cao, vì ít khối phải được phục hồi.

Thông tin về điểm kiểm gia tăng được Oracle server duy trì tự động mà không ảnh hưởng đến các điểm kiểm khác (chẳng hạn như các điểm kiểm chuyển đổi nhật ký hoạt động và các điểm kiểm người dùng xác định). Nói cách khác, điểm kiểm gia tăng xảy ra độc lập với những điểm kiểm khác của các điểm kiểm khác xảy ra trong môi trường làm việc.

Các không gian bảng chỉ đọc và phục hồi môi trường làm việc

Không cần phục hồi đối với các tập tin dữ liệu chỉ đọc khi phục hồi môi trường làm việc. Phục hồi khi khởi động xác minh mỗi tập tin chỉ đọc không cần phải phục hồi môi trường. Nghĩa là cho tới khi bạn hoàn tất phục hồi môi trường.

Sai sót môi trường (đĩa)

Sai sót có thể phát sinh khi cố ghi hoặc đọc một tập tin cần thiết để điều hoạt một cơ sở dữ liệu Oracle. Sự kiện này được gọi là sai sót môi trường vì bị vấn đề vật lý về đọc hoặc ghi vào tập tin trên môi trường lưu trữ.

Một thí dụ thông thường về sai sót môi trường là đĩa bị chạm đầu làm mất tất cả những tập tin trên ổ đĩa. Tất cả những tập tin được kết hợp với cơ sở dữ liệu có thể bị hư hại do sự cố đĩa, kể cả các tập tin dữ liệu, các tập tin redo log và các tập tin điều khiển.

Phục hồi thích hợp sai sót môi trường phụ thuộc vào những tập tin bị ảnh hưởng.

Thông tin bổ sung:
Xem Hướng dẫn phục hồi và sao lưu của Oracle8.

Các sai sót môi trường ảnh hưởng đến điều hoạt cơ sở dữ liệu như thế nào

Sai sót môi trường có thể ảnh hưởng đến một hoặc nhiều kiểu tập tin cần thiết để điều hoạt cơ sở dữ liệu Oracle, kể cả những tập tin dữ liệu, các tập tin redo log và các tập tin điều khiển.

Điều hoạt cơ sở dữ liệu sau một sai sót môi trường của các tập tin redo log trực tuyến hoặc tập tin điều khiển theo đề nghị được truyền đa công. Redo log trực tuyến hoặc tập tin điều khiển đa công nghĩa là bản sao thứ hai của tập tin ấy được duy trì. Nếu môi trường sai sót làm hư một đĩa đơn và bạn có redo log trực tuyến đa công, thì cơ sở dữ liệu thường có thể tiếp tục hoạt động mà không bị gián đoạn quan trọng. Sự thiệt hại đối với một redo log trực tuyến không đa công làm cho điều hoạt cơ sở dữ liệu tạm ngưng và có thể làm mất dữ liệu vĩnh viễn. Sự thiệt hại đối với tập tin điều khiển bất kỳ tạm ngưng điều hoạt cơ sở dữ liệu dù có đa công hoặc không có đa công một khi Oracle cố đọc hoặc ghi tập tin kinh doanh bị hư.

Sai sót môi trường ảnh hưởng đến các tập tin dữ liệu có thể chia thành hai loại: các lỗi đọc và các lỗi ghi. Về lỗi đọc, Oracle phát hiện nó không thể đọc một tập tin dữ liệu và một lỗi hệ điều hành được cho trở lại ứng dụng ấy cùng mỗi lỗi Oracle chỉ tập tin không thể tìm thấy, không thể mở được hoặc không thể đọc được. Oracle tiếp tục chạy, nhưng có thông báo lỗi mỗi lần đọc không thành công. Ở điểm kiểm kế tiếp, lỗi ghi sẽ xảy ra khi Oracle cố ghi phần đầu tập tin ấy là thành phần của tiến trình điểm kiểm chuẩn.

Nếu Oracle phát hiện là nó không thể ghi vào một tập tin dữ liệu và Oracle đang lưu trữ những tập tin redo log trực tuyến, thì Oracle cho trở lại một lỗi ở tập tin vệt tìm DBWn và tự động đưa tập tin dữ liệu ngoại tuyến. Chỉ tập tin dữ liệu không thể được ghi được đưa ngoại tuyến; không gian bảng chứa tập tin đó vẫn duy trì trực tuyến.

Nếu tập tin dữ liệu không thể được ghi ở trong không gian bảng SYSTEM, thì tập tin ấy không thể được đưa ngoại tuyến. Thay vào đó, lỗi được cho trở lại và Oracle đóng cơ sở dữ liệu ấy. Lý do về sự ngoại lệ này là tất cả những tập tin trong không gian bảng SYSTEM phải được trực tuyến theo thứ tự để Oracle điều hoạt thích hợp. Tương tự với lý do đó, những tập tin dữ liệu của không gian bảng chứa các đoạn hồi phục về trạng thái cũ hoạt động phải duy trì trực tuyến.

Nếu Oracle phát hiện là nó không thể ghi vào cơ sở dữ liệu và Oracle hiện không lưu trữ các tập tin redo log trực tuyến, thì tiến trình nền DBWn không thực hiện thành công và môi trường làm việc hiện hành không thực hiện được. Nếu là vấn đề tạm thời (thí dụ, bộ điều kiện đĩa đã bị ngắt nguồn), phục hồi môi trường làm việc thường có thể được thực hiện bằng cách sử dụng các redo file trực tuyến, trong trường hợp môi trường làm việc có thể được khởi động lại. Tuy nhiên, nếu tập tin dữ liệu bị hư vĩnh viễn và dữ liệu lưu trữ không sử dụng được, thì toàn bộ cơ sở dữ liệu phải được phục hồi bằng cách sử dụng sao lưu mới nhất.

Các cấu trúc được sử dụng để phục hồi cơ sở dữ liệu

Nhiều cấu trúc của cợ sở dữ liệu Oracle bảo vệ dữ liệu tránh bị sự cố. Phần này giới thiệu về một trong những cấu trúc này và vai trò của nó trong phục hồi cơ sở dữ liệu.

Các sao lưu cơ sở dữ liệu

Sao lưu cơ sở dữ liệu gồm sao lưu các tập tin vật lý (tất cả những tập tin dữ liệu và tập tin điều khiển) cấu tạo nên một cơ sở dữ liệu. Để bắt đầu phục hồi cơ sở dữ liệu từ một sai sót môi trường, Oracle sử dụng các sao lưu tập tin để phục hồi các tập tin dữ liệu hoặc các tập tin điều khiển bị hư.

Oracle cung ứng nhiều tham chiếu để thực hiện các sao lưu cơ sở dữ liệu.

Thông tin bổ sung:
Xem Hướng dẫn phục hồi và sao lưu của Oracle.

Redo Log

Nhật ký redo log (nhật ký hoạt động) hiện hữu đối với mọi cơ sở dữ liệu của Oracle, nó ghi tất cả những thay đổi xảy ra trong cơ sở dữ liệu của Oracle. Redo log của một cơ sở dữ liệu gồm ít nhất hai tập tin redo log được tách ra từ những tập tin dữ liệu (thực ra nó lưu trữ dữ liệu của cơ sở dữ liệu). Là thành phần phục hồi cơ sở dữ liệu từ một sai sót môi trường làm việc hoặc môi trường đĩa, Oracle áp dụng những thay đổi thích hợp trong redo log của cơ sở dữ liệu vào các tập tin dữ liệu để cập nhật dữ liệu của cơ sở dữ liệu cho môi trường làm việc mà sự cố xảy ra.

Redo log của cơ sở dữ liệu có thể gồm hai phần: redo log trực tuyến và redo log tàng trữ.

Redo Log trực tuyến

Mọi cơ sở dữ liệu Oracle đều có redo log trực tuyến được kết hợp. Redo log trực tuyến hoạt động với tiến trình nền Oracle LGWR để ghi ngay tất cả những thay đổi xảy ra qua môi trường làm việc được kết hợp. Redo log trực tuyến gồm hai hoặc nhiều tập tin được phân bổ trước được sử dụng lại theo kiểu luân chuyển để ghi các thay đổi cơ sở dữ liệu khi tiến hành.

Redo Log tàng trữ (ngoại tuyến)

Bạn có thể tùy chọn cấu hình cơ sở dữ liệu Oracle để tàng trữ những tập tin của redo log trực tuyến một khi chúng được điền. Các tập tin redo log trực tuyến được tàng trữ, chúng được nhận dạng duy nhất và tạo thành redo log tàng trữ. Bằng cách tàng

trữ các tập tin redo log trực tuyến, thông tin redo log cũ hơn được lưu trữ cho nhiều điều hoạt phục hồi cơ sở dữ liệu mở rộng hơn, khi các tập tin redo log được phân bổ trước tiếp tục được sử dụng lại để lưu trữ những thay đổi của cơ sở dữ liệu hiện hành nhất.

Xem "Các Chế Độ Tàng Trữ Cơ Sở Dữ Liệu".

Các đoạn hồi phục về trạng thái cũ

Các đoạn hồi phục về trạng thái cũ được sử dụng cho một số chức năng trong điều hoạt của cơ sở dữ liệu Oracle. Nói chung, các đoạn hồi phục về trạng thái cũ của cơ sở dữ liệu lưu trữ những giá trị dữ liệu cũ bằng cách tiến hành các giao dịch (nghĩa là, các giao dịch chưa được cam kết).

Trong những vấn đề khác, thông tin trong đoạn hồi phục về trạng thái cũ được sử dụng khi phục hồi cơ sở dữ liệu "hủy bỏ" bất kỳ thay đổi "chưa được cam kết" được áp dụng từ redo log vào những tập tin dữ liệu. Do đó, nếu phục hồi cơ sở dữ liệu cần thiết, dữ liệu ở trạng thái nhất quán sau khi các đoạn hồi phục về trạng thái cũ được sử dụng để xóa tất cả dữ liệu chưa được cam kết khỏi những tập tin dữ liệu.

Các tập tin điều khiển

Nói chung, (các) tập tin điều khiển của cơ sở dữ liệu lưu trữ trạng thái cấu trúc vật lý của cơ sở dữ liệu. Thông tin trạng thái nào đó trong tập tin điều khiển (thí dụ, tập tin redo log trực tuyến, các tên của những tập tin dữ liệu, vân vân.) hướng dẫn Oracle khi phục hồi môi trường làm việc hoặc môi trường đĩa.

Xem "Các Tập Tin Điều khiển".

Hồi phục về sau và hồi phục về trước

Những vùng đệm cơ sở dữ liệu trong SGA được ghi vào đĩa chỉ khi cần thiết, bằng cách sử dụng thuật toán Least-Recently-Used. Vì cách mà tiến trình DBWn sử dụng thuật toán này để ghi các vùng đệm cơ sở dữ liệu vào những tập tin dữ liệu, những tập tin dữ liệu có thể có một số khối dữ liệu bị sửa đổi do các giao dịch chưa được cam kết và một số khối dữ liệu sai lạc những thay đổi từ giao dịch được cam kết.

Hai vấn đề có thể dẫn đến nếu sai sót của môi trường làm việc xảy ra:

Các khối dữ liệu bị sửa đổi do giao dịch không thể ghi vào các tập tin dữ liệu vào lúc cam kết và chỉ có thể xuất hiện ở redo log. Do đó, redo log có những thay đổi phải được áp dụng lại vào cơ sở dữ liệu khi phục hồi.

Vì redo log có thể còn có dữ liệu đã không được cam kết, những thay đổi của giao dịch chưa cam kết được redo log áp dụng (cũng như bất kỳ những thay đổi chưa cam kết được các redo log trước áp dụng) phải được xóa khỏi cơ sở dữ liệu.

Để giải quyết tình trạng khó xử này, hai bước tách riêng thường được Oracle sử dụng để phục hồi thành công sai sót của hệ thống: hồi phục về sau với redo log và hồi phục về trước với các đoạn hồi phục về trước.

Redo Log và hồi phục về sau

Redo log là một tập hợp những tập tin của hệ điều hành để ghi tất cả những thay đổi xảy ra đối với vùng đệm cơ sở dữ liệu bất kỳ, kể cả dữ liệu, chỉ mục và các đoạn hồi phục về trước, dù những thay đổi được cam kết hoặc chưa được cam kết. Redo log bảo vệ những thay đổi xảy ra đối với các vùng đệm cơ sở dữ liệu trong bộ nhớ chưa được ghi vào những tập tin dữ liệu.

Bước phục hồi đầu tiên từ môi trường làm việc hoặc từ sự cố đĩa để hồi phục về sau hoặc áp dụng lại tất cả những thay đổi được ghi trong redo vào những tập tin dữ liệu. Vì dữ liệu hồi phục về trước còn được ghi trong redo log, hồi phục về sau cũng phát sinh lại những đoạn hồi phục về trước tương ứng.

Hồi phục về sau tiến hành qua nhiều tập tin redo log theo nhu cầu để đưa cơ sở dữ liệu theo kịp thời gian. Hồi phục về sau thường có các tập tin redo log trực tuyến và có thể có các tập tin redo log tàng trữ.

Sau khi hồi phục về sau, các khối dữ liệu có tất cả những thay đổi được cam kết cũng như bất kỳ những thay đổi chưa được cam kết được ghi trong redo log.

Các đoạn hồi phục về sau và hồi phục về trước

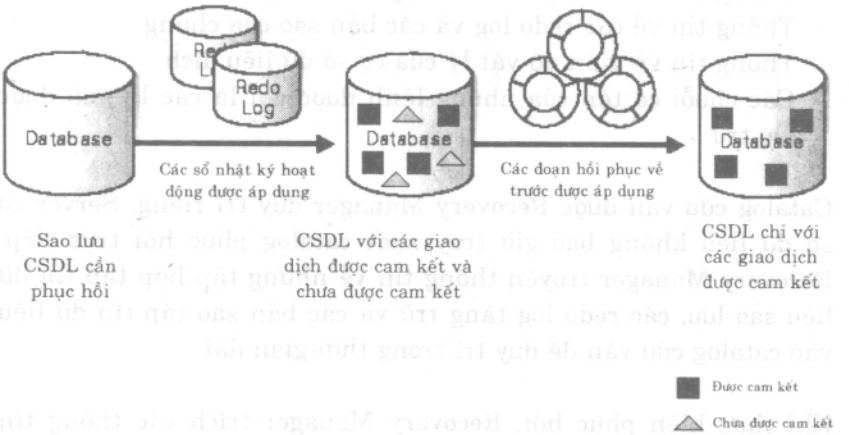
Các đoạn hồi phục về trước ghi những hoạt động của cơ sở dữ liệu sẽ được xóa bỏ khi những điều hoạt cơ sở dữ liệu nào đó. Trong phục hồi cơ sở dữ liệu, các đoạn hồi phục về trước xóa bỏ những ảnh hưởng của những giao dịch chưa được cam kết trước đó pha hồi phục về sau áp dụng.

Sau khi hồi phục về sau, bất kỳ những thay đổi đã chưa được cam kết phải được xóa bỏ. Sau khi những tập tin redo log đã áp dụng lại tất cả những thay đổi xảy ra đối với cơ sở dữ liệu, thì các đoạn hồi phục về sau tương ứng được sử dụng. Các đoạn hồi phục về trước được sử dụng để nhận dạng và xóa bỏ những giao dịch chưa bao giờ được cam kết, nhưng đã được ghi vào redo log và được áp dụng vào cơ sở dữ liệu khi hồi phục về sau.

Hình 28-1 minh họa hồi phục về sau và hồi phục về trước, hai bước cần thiết để phục hồi từ bất kỳ kiểu sai sót hệ thống.

Oracle có thể hồi phục về sau nhiều giao dịch đồng thời theo nhu cầu. Tất cả những giao dịch khắp hệ thống đã hoạt động vào lúc sai sót được đánh dấu là DEAD. Thay vì phải chờ SMON hồi phục về trước những giao dịch bị liệt, những giao dịch mới có thể phục hồi chính những giao dịch khối của chúng để có được những khóa hàng chúng cần. Đặc trưng này được gọi là hồi phục về trước giao dịch nhanh.

Hình 28-1 Các bước phục hồi căn bản: Hồi phục về sau và hồi phục về trước



Trình quản lý phục hồi

Trình quản lý phục hồi - Recovery Manager - là một tiện ích quản lý những tiến trình tạo các bản sao lưu những tập tin cơ sở dữ liệu và phục hồi hoặc cứu vãn những tập tin từ các bản sao chép.

Thông tin bổ sung:
Xem Hướng dẫn phục hồi và sao lưu của Oracle8 mô tả đầy đủ về Recovery Manager.

Catalog phục hồi

Recovery Manager duy trì một tàng thư được gọi là catalog phục hồi, có những thông tin về những tập tin sao lưu và những tập tin nhật ký tàng trữ. Recovery Manager sử dụng những catalog phục hồi để tự động hóa cả các hoạt động phục hồi và cứu vãn môi trường đĩa.

Catalog phục hồi có:

Thông tin về các sao lưu những tập tin dữ liệu và các nhật ký tàng trữ
Thông tin về những bản sao tập tin dữ liệu
Thông tin về các redo log và các bản sao của chúng
Thông tin về giản đồ vật lý của cơ sở dữ liệu đích
Các chuỗi có tên của những lệnh được gọi là các ký mã được lưu trữ

Catalog cứu vãn được Recovery Manager duy trì riêng. Server cơ sở dữ liệu không bao giờ truy xuất catalog phục hồi trực tiếp. Recovery Manager truyền thông tin về những tập hợp tập tin dữ liệu sao lưu, các redo log tàng trữ và các bản sao tập tin dữ liệu vào catalog cứu vãn để duy trì trong thời gian dài.

Khi thực hiện phục hồi, Recovery Manager trích các thông tin thích hợp từ catalog cứu vãn và chuyển nó tới server cơ sở dữ liệu. Server thực hiện những kiểm tra tính nguyên vẹn khác nhau trên những tập tin nhập liệu được ấn định để phục hồi. Recovery Manager xử lý không chính xác không làm hư cơ sở dữ liệu.

Cơ sở dữ liệu catalog phục hồi

Catalog phục hồi được lưu trữ trong cơ sở dữ liệu Oracle là trách nhiệm của người điều hành cơ sở dữ liệu tạo ra một cơ sở dữ liệu như thế để Recovery Manager có thể sử dụng. Chọn các bản sao lưu của catalog phục hồi cũng là trách nhiệm của người điều

hành. Vì catalog phục hồi được lưu trữ trong cơ sở dữ liệu Oracle, bạn có thể sử dụng Recovery Manager để sao lưu nó.

Nếu catalog phục hồi bị hủy và không có bản sao lưu nào có thể sử dụng, thì nó có thể được cấu trúc lại một phần từ tập tin điều khiển hiện hành hoặc các bản sao lưu tập tin điều khiển.

Điều hoạt không có catalog phục hồi

Sử dụng catalog phục hồi không cần thiết. Vì phần lớn thông tin trong catalog phục hồi còn có thể sử dụng từ tập tin điều khiển, Recovery Manager hỗ trợ một chế độ điều hoạt nơi nó chỉ sử dụng tập tin điều khiển.

Chế độ điều hoạt thích hợp đối với những cơ sở dữ liệu nhỏ nơi cài đặt và điều hành một cơ sở dữ liệu khác để đáp ứng khi catalog phục hồi sẽ thành nặng nề.

Phục hồi giờ và ngày tháng của không gian bảng không được hỗ trợ trong chế độ điều hoạt này.

Sự song song hóa

Recovery Manager có thể song song hóa các điều hoạt của nó, thiết lập nhiều phiên đăng nhập và quản lý nhiều hoạt động song song bằng cách sử dụng UPI không lập khối. Các hoạt động đồng thời phải điều hoạt trên những tập hợp tập tin cơ sở dữ liệu tách riêng.

Lưu ý:
Oracle8 chỉ có thể phân bổ một kênh Recovery Manager một lần, vì thế giới hạn trạng thái song song với một luồng. Oracle8 Enterprise Edition cho phép trạng thái song song. Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về những đặc trưng khả dụng với Oracle8 và Oracle8

Enterprise.

Tình trạng song song hóa sao lưu, bản sao và các lệnh phục hồi được Recovery Manager xử lý bên trong. Bạn chỉ cần ấn định:

- danh sách của một hoặc nhiều thiết bị I/O chuỗi
- các đối tượng được sao lưu, sao chép hoặc phục hồi.

Recovery Manager tiến hành các lệnh nối tiếp, nghĩa là nó hoàn tất lệnh trước đó trước khi bắt đầu lệnh kế tiếp. Tình trạng song song chỉ được khai thác trong ngữ cảnh của một lệnh đơn. Do đó, nếu 10 bản sao tập tin dữ liệu được yêu cầu, thì phát hành một lệnh sao chép đơn tốt hơn ấn định tất cả các bản sao thay vì các lệnh sao chép tách riêng.

Sự phát sinh báo biểu

Các lệnh *report* (báo biểu) và *list* (danh sách) cung cấp thông tin về các bản sao lưu và các bản sao ảnh. Kết xuất từ các lệnh này được ghi vào tập tin nhật ký thông điệp.

Lệnh báo biểu tạo ra các báo biểu có thể trả lời những câu hỏi chẳng hạn như:

Những tập tin nào cần sao lưu?
Những tập tin nào chưa được sao lưu?
Những tập tin sao lưu nào bị xóa?

Bạn có thể sử dụng *report need backup* (báo biểu cần sao lưu) và *report unrecoverable* (báo biểu không phục hồi được) trên phương thức thường xuyên để bảo đảm những bản sao lưu cần thiết có thể sử dụng để thực hiện phục hồi và phục hồi có thể được thực hiện trong thời gian hợp lý. Lệnh báo biểu xóa được liệt kê các tập hợp và các bản sao tập tin dữ liệu có thể được xóa vì chúng không cần thiết hoặc vì chúng không bao giờ được một lệnh phục hồi sử dụng.

Tập tin dữ liệu được xét không phục hồi được nếu:
Chỉ các bản sao lưu tập tin dữ liệu có sẵn không sử dụng được vì các nhật ký tàng trữ cần thiết để phục hồi chúng không ở trong catalog phục hồi hoặc catalog phục hồi ấy không biết cần những sổ nhật ký nào
Điều hoạt không được ghi sổ nhật ký được thực hiện dựa trên đối tượng giản đồ cư trú trong tập tin dữ liệu.

(Tập tin dữ liệu không có sao lưu không được xét phục hồi được. Những tập tin dữ liệu như thế có thể được phục hồi bằng cách sử dụng tạo lệnh tập tin dữ liệu với điều kiện các sổ nhật ký bắt đầu từ khi tập tin ấy đã được tạo vẫn tồn tại.)

Lệnh danh sách truy vấn catalog phục hồi và tạo danh sách những nội dung của nó. Bạn có thể sử dụng nó để phát hiện những bản sao lưu hoặc các bản sao nào có thể sử dụng:
Các bản sao lưu hoặc các bản sao của một danh sách những tập tin dữ liệu được ấn định
Các bản sao lưu hoặc các bản sao của bất kỳ tập tin dữ liệu nào là thành viên của danh sách các không gian bảng được ấn định
Các bản sao lưu hoặc các bản sao của bất kỳ nhật ký tàng trữ với tên ấn định và/hoặc trong phạm vi ấn định
Các bản hiện thân của cơ sở dữ liệu ấn định.

Thực hiện phục hồi song song

Phục hồi áp dụng lại những thay đổi được nhiều tiến trình đồng thời phát sinh, do đó môi trường làm việc hoặc phục hồi môi trường đìa có thể lâu hơn thời gian phát sinh ban đầu những thay đổi đối với một cơ sở dữ liệu. Với phục hồi nối tiếp, tiến trình đơn áp dụng những thay đổi trong những tập tin redo log liên tiếp. Sử dụng phục hồi song song, nhiều tiến trình áp dụng những thay đổi đồng thời từ những tập tin redo log.

Lưu ý:
Oracle8 cung cấp tình trạng song song với Recovery Manager; Oracle8 Enterprise Edition cho phép tình trạng song song không bị giới hạn. Xem Gettingto Know Oracle8 và Oracle8 Enterprise Edition về những đặc trưng có thể sử dụng trong Oracle8 và Oracle8 Enterprise Edition.

Phục hồi song song có thể được thực hiện thủ công bằng cách phát sinh nhiều phiên Oracle Enterprise Manager và phát hành lệnh RECOVER DATAFILE trên tập hợp khác nhau của những tập tin dữ liệu trong mỗi phiên. Tuy nhiên, phương pháp này tạo ra mỗi phiên Oracle Enterprise Manager để đọc tập tin redo log.

Phục hồi môi trường làm việc và môi trường đĩa có thể được thực hiện tự động song song bằng cách ấn định tham số khởi hoạt hoặc các tùy chọn đối với lệnh RECOVER. Oracle sử dụng một tiến trình đọc các tập tin nhật ký liên tiếp và phân công thông tin redo đối với nhiều tiến trình phục hồi áp dụng những thay đổi từ các tập tin nhật ký vào những tập tin dữ liệu. Các tiến trình phục hồi được Oracle khởi động tự động, vì vậy không cần phải sử dụng nhiều phiên để thực hiện phục hồi.

Các lợi ích của phục hồi song song

Nói chung, phục hồi song song có hiệu quả nhất vì thời gian phục hồi giảm xuống khi nhiều tập tin dữ liệu trên nhiều đĩa khác nhau đang được phục hồi đồng thời. Phục hồi sự cố (phục hồi sau khi sai sót môi trường làm việc) và phục hồi môi trường của nhiều tập tin dữ liệu trên nhiều ổ đĩa khác nhau là các ứng viên tốt đối với phục hồi song song.

Hiệu suất tăng lên nhờ phục hồi song song còn phụ thuộc vào hệ điều hành có hỗ trợ I/O không đồng bộ. Nếu I/O không đồng bộ không được hỗ trợ, thì phục hồi song song có thể giảm thời gian phục hồi rất nhiều. Nếu I/O không đồng bộ được hỗ trợ, thì thời

gian phục hồi chỉ có thể giảm xuống một ít nhờ sử dụng phục hồi song song.

Thông tin bổ sung:
Xem tài liệu hệ điều hành của bạn để xác định xem hệ thống có hỗ trợ I/O không đồng bộ không.

Các tiến trình phục hồi

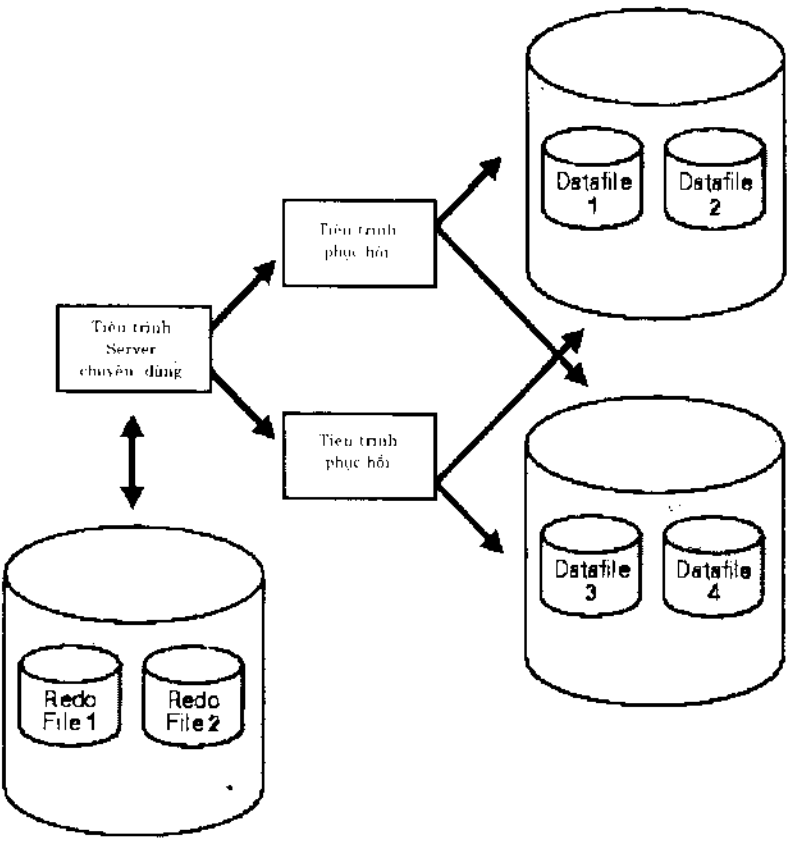
Trong trường hợp phục hồi song song thông thường, một tiến trình chịu trách nhiệm đọc và phân công các mục nhập redo của những tập tin redo log. Đây là tiến trình server chuyên dùng bắt đầu phiên phục hồi. Tiến trình server đọc các tập tin redo log đăng ký hai hoặc nhiều tiến trình phục hồi để áp dụng những thay đổi từ các mục nhập redo vào những tập tin dữ liệu.
Hình 28-2 minh họa phiên phục hồi song song tiêu biểu.

Trong hầu hết các trường hợp, một phiên phục hồi và một hoặc nhiều tiến trình phục hồi trên mỗi ổ đĩa có nhiều tập tin dữ liệu cần phục hồi có thể thực hiện được. Phục hồi là hoạt động mở rộng đĩa trái ngược với hoạt động mở rộng CPU và do đó một số tiến trình phục hồi cần thiết phụ thuộc hoàn toàn vào bao nhiêu ổ đĩa có liên quan đến hoạt động phục hồi. Nói chung, cần ít nhất có 8 tiến trình phục hồi trước khi phục hồi song song có thể cho thấy cải tiến được phục hồi nối tiếp.

Các chế độ tàng trữ CSDL

Cơ sở dữ liệu có thể điều hoạt trong hai chế độ riêng biệt: chế độ NOARCHIVELOG (phục hồi môi trường bị vô hiệu hóa) hoặc chế độ ARCHIVELOG (phục hồi môi trường được cho phép).

Hình 28-2 Phiên phục hồi song song tiêu biểu



Nếu một cơ sở dữ liệu được sử dụng ở chế độ NOARCHIVELOG, tàng trữ redo log trực tuyến bị vô hiệu hóa. Thông tin trong tập tin điều khiển của cơ sở dữ liệu chỉ những nhóm được điền không cần phải tàng trữ. Vì thế, khi một nhóm trở thành không hoạt động và điểm kiểm ở khóa chuyển hoàn tất, nhóm ấy có thể được tiến trình LGWR sử dụng lại.

Chế độ NOARCHIVELOG chỉ bảo vệ sai sót môi trường làm việc, không phải từ sai sót đĩa (môi trường). Chỉ những thay đổi mới

nhất được thực hiện đối với cơ sở dữ liệu được lưu trữ trong các nhóm của redo log trực tuyến có thể sử dụng để phục hồi môi trường làm việc.

Chế độ ARCHIVELOG (phục hồi môi trường được cho phép)

Nếu một cơ sở dữ liệu Oracle được điều hoạt ở chế độ ARCHIVELOG, tàng trữ redo log trực tuyến được cho phép. Thông tin trong tập tin điều khiển cơ sở dữ liệu chỉ nhóm của những tập tin redo log trực tuyến không thể được LGWR sử dụng lại cho tới khi nhóm ấy được tàng trữ. Một nhóm được điền có thể sử dụng ngay đối với tiến trình thực hiện tàng trữ một khi khóa chuyển nhật ký xảy ra (khi một nhóm trở thành không hoạt động); tiến trình thực hiện tàng trữ không phải chờ điểm kiểm của khóa chuyển nhật ký để hoàn tất trước khi nó có thể truy xuất nhóm không hoạt động để tàng trữ.

Hình 28-3 minh họa các tập tin redo log trực tuyến của cơ sở dữ liệu được sử dụng ở chế độ ARCHIVELOG và redo log tàng trữ được phát sinh bằng tiến trình tàng trữ các nhóm điền như thế nào (thí dụ, ARCH trong hình minh họa).

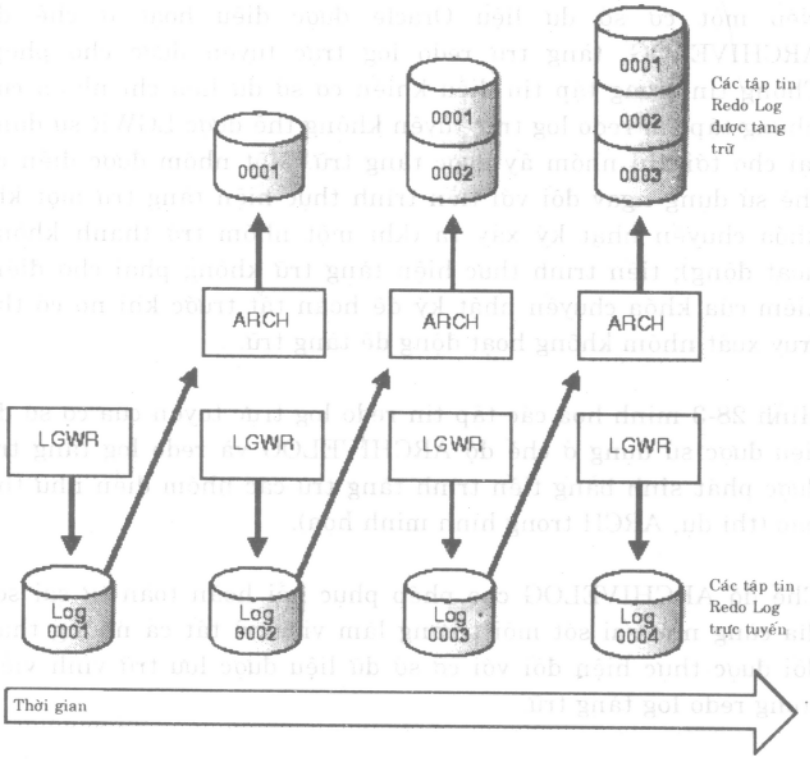
Chế độ ARCHIVELOG cho phép phục hồi hoàn toàn từ sai sót đĩa cùng như sai sót môi trường làm việc, vì tất cả những thay đổi được thực hiện đối với cơ sở dữ liệu được lưu trữ vĩnh viễn trong redo log tàng trữ.

Tàng trữ tự động và xử lý nền ARCH (Archiver)

Môi trường làm việc có thể được cấu hình có xử lý nền bổ sung, bộ tàng trữ (ARCH) tự động tàng trữ các nhóm tập tin redo log trực tuyến một khi chúng trở thành không hoạt động. Do đó, tàng trữ tự động để người điều hành không phải theo dõi và tàng trữ các nhóm điền thủ công. Đối với riêng sự thuận tiện này,

tàng trữ tự động là chọn lựa của phần lớn hệ thống cơ sở dữ liệu có redo log được tàng trữ.

Hình 28-3 Sử dụng tập tin Redo Log trực tuyến ở chế độ ARCHIVELOG



Nếu bạn yêu cầu tàng trữ tự động vào lúc khởi động môi trường làm việc bằng cách xác lập tham số khởi hoạt LOG_ARCHIVE_START, Oracle khởi động ARCH khi khởi động môi trường làm việc. Trái lại, ARCH không được khởi động khi khởi động môi trường làm việc.

Tuy nhiên, người điều hành cơ sở dữ liệu có thể khởi động hoặc dừng tàng trữ tự động bất kỳ lúc nào một cách tương tác. Nếu tàng trữ tự động đã không được ấn định để khởi động vào lúc khởi động môi trường làm việc và sau đó người điều hành khởi động tàng trữ tự động, thì xử lý nền ARCH được tạo. Sau đó ARCH vẫn duy trì thời gian của môi trường làm việc, dù tàng trữ tự động được tạm thời tắt đi và bật lại.

ARCH luôn luôn tàng trữ các nhóm theo thứ tự, bắt đầu với số hiệu chuỗi thấp nhất. ARCH tự động tàng trữ những nhóm được điền khi chúng trở thành không hoạt động. Một bản ghi của mọi tàng trữ tự động được ghi vào tập tin vệt tìm ARCH bằng tiến trình ARCH. Mỗi mục nhập trình bày thời gian tàng trữ bắt đầu và dừng.

Nếu ARCH gặp một lỗi khi cố tàng trữ một nhóm (thí dụ, do một đích đến không hợp lệ hoặc được điền), ARCH tiếp tục cố tàng trữ nhóm. Một lỗi cũng được ghi vào tập tin vệt tìm ARCH và tập tin ALERT. Nếu vấn đề ấy chưa được giải quyết, cuối cùng tất cả những nhóm redo log trực tuyến trở thành đầy, nhưng chưa được tàng trữ và tàng trữ các nhóm thủ công cho tới hiển thị vấn đề được giải quyết.

Tàng trữ thủ công

Nếu cơ sở dữ liệu đang điều hoạt ở chế độ ARCHIVELOG, thì người điều hành cơ sở dữ liệu có thể tàng trữ thủ công các nhóm điền của những tập tin redo log theo nhu cầu, dù tàng trữ tự động được cho phép hay không. Nếu tàng trữ tự động bị vô hiệu hóa, người điều hành cơ sở dữ liệu chịu trách nhiệm tàng trữ các nhóm được điền.

Đối với phần lớn các hệ thống, tàng trữ hệ thống được chọn vì người điều hành không phải theo dõi nhóm trở thành không hoạt động và có thể sử dụng để tàng trữ. Hơn nữa, nếu tàng trữ

tự động bị vô hiệu hóa và tàng trữ thủ công không được thực hiện đủ nhanh, điều hoạt cơ sở dữ liệu có thể bị treo tạm thời bất kỳ khi nào LGWR bị buộc phải chờ một nhóm không hoạt động trở thành có thể sử dụng lại.

Tùy chọn tàng trữ thủ công được cung cấp để người điều hành cơ sở dữ liệu có thể:
Tàng trữ một nhóm khi tàng trữ tự động đã bị dừngvì một vấn đề (thí dụ, thiết bị lưu trữ ngoại tuyến được ấn định khi được tàng trữ đích redo log đã trải qua sai sót hoặc trở thành đầy)
Tàng trữ một nhóm trong kiểu phi tiêu chuẩn (thí dụ, tàng trữ một nhóm vào thiết bị lưu trữ ngoại tuyến, nhóm kế tiếp vào thiết bị lưu trữ ngoại tuyến khác, vân vân)
Tàng trữ lại một nhóm nếu phiên bản được tàng trữ ban đầu bị mất hoặc bị hư

Khi một nhóm dược tàng trữ thủ công, thì tiến trình người dùng phát hành câu lệnh để tàng trữ một nhóm ấy. Dù xử lý nền ARCH hiện hữu đối với môi trường làm việc được kết hợp, nó là tiến trình người dùng tàng trữ nhóm những tập tin redo log trực tuyến.

Các tập tin điều khiển

Tập tin điều khiển của cơ sở dữ liệu là một tập tin thư viện nhỏ cần thiết để cơ sở dữ liệu khởi động và điều hoạt thành công. Một tập tin điều khiển được Oracle cập nhật liên tục khi cơ sở dữ liệu sử dụng, vì vậy nó phải có thể sử dụng để ghi bất kỳ lúc nào cơ sở dữ liệu được mở. Nếu đối với lý do nào đó, tập tin điều khiển không truy xuất được, thì cơ sở dữ liệu sẽ không thực hiện chức năng chính xác.
Mỗi tập tin điều khiển được kết hợp với chỉ một cơ sở dữ liệu của Oracle.

Nội dung tập tin điều khiển

Tập tin điều khiển có những thông tin về cơ sở dữ liệu được kết hợp cần thiết cho cơ sở dữ liệu để được một môi trường làm việc truy xuất vào lúc khởi động và điều hoạt bình thường. Thông tin của tập tin điều khiển chỉ có thể được Oracle sửa đổi; người điều hành cơ sở dữ liệu và người dùng cuối không thể hiệu chỉnh tập tin điều khiển của cơ sở dữ liệu.

Trong những vấn đề khác, tập tin điều khiển có những thông tin chẳng hạn như:

Tên cơ sở dữ liệu
Dấu thời gian về tạo cơ sở dữ liệu
Các tên và các vị trí của những tập tin dữ liệu được kết hợp và các tập tin redo log trực tuyến
Thông tin không gian bảng
Các phạm vi ngoại tuyến tập tin cơ sở dữ liệu
Lịch sử sổ nhật ký
Thông tin sổ nhật ký tàng trữ
Thông tin tập hợp sao lưu và phân đoạn sao lưu
Thông tin redo log và tập tin dữ liệu sao lưu
Thông tin sao lưu tập tin dữ liệu
Thông tin bản sao tập tin dữ liệu
Số hiệu chuỗi sổ nhật ký hiện hành
Thông tin điểm kiểm

Tên cơ sở dữ liệu và dấu thời gian phát sinh vào lúc tạo cơ sở dữ liệu. Tên của cơ sở dữ liệu được chọn từ tên được tham số khởi hoạt DB_NAME ấn định hoặc tên được sử dụng trong câu lệnh CREATE DATABASE.

Mỗi lần tập tin dữ liệu hoặc tập tin redo log được bổ sung vào, được đặt tên lại hoặc bỏ đi từ cơ sở dữ liệu, tập tin điều khiển được cập nhật để phản ánh sự thay đổi cấu trúc vật lý này. Những thay đổi này được ghi để Oracle có thể nhận dạng những

tập tin dữ liệu và các tập tin redo log trực tuyến để mở khi khởi động cơ sở dữ liệu.
Oracle có thể nhận dạng những tập tin cần thiết hoặc có thể sử dụng trong trường hợp cần phục hồi cơ sở dữ liệu.

Do đó, nếu bạn thay đổi cấu trúc vật lý của cơ sở dữ liệu, bạn nên sao lưu tập tin điều khiển ngay.

Thông tin bổ sung:
Xem Hướng dẫn phục hồi và sao lưu của Oracle8 về sao lưu một tập tin điều khiển của cơ sở dữ liệu.

Các tập tin điều khiển còn ghi thông tin về các điểm kiểm. Khi một điểm kiểm bắt đầu, tập tin điều khiển ấy ghi thông tin về mục nhập kế tiếp phải được nhập vào redo log trực tuyến. Thông tin này được sử dụng khi phục hồi cơ sở dữ liệu để cho Oracle biết tất cả những mục nhập redo được ghi trước điểm này trong nhóm redo log trực tuyến không cần thiết cho phục hồi cơ sở dữ liệu vì chúng đã được ghi vào những tập tin dữ liệu rồi.

Các tập tin điều khiển đa công

Giống như các tập tin redo log trực tuyến, Oracle cho phép nhiều tập tin điều khiển giống nhau được mở đồng thời và được ghi đối với cơ sở dữ liệu tương tự.
Bằng cách lưu trữ nhiều tập tin điều khiển cho một cơ sở dữ liệu đơn trên các đĩa khác nhau, bạn có thể bảo vệ dựa vào một điểm sai sót đơn giản liên quan đến những tập tin điều khiển. Nếu một đĩa đơn có những sai sót tập tin điều khiển, thì môi trường làm việc hiện hành không thực hiện được khi Oracle cố truy xuất tập tin điều khiển bị hư. Tuy nhiên, các bản sao của tập tin điều khiển hiện hành có thể sử dụng cho những đĩa khác, vì vậy một môi trường làm việc có thể được phục hồi lại dễ dàng mà không cần phục hồi cơ sở dữ liệu.

Mất vĩnh viễn tất cả những bản sao tập tin điều khiển của cơ sở dữ liệu là một vấn đề nghiêm trọng phải cần bảo vệ. Nếu tất cả những tập tin điều khiển của cơ sở dữ liệu bị mất vĩnh viễn khi điều hoạt (nhiều đĩa không thực hiện được), môi trường làm việc bị hủy ngang và cần phải phục hồi môi trường. Dù thậm chí như vậy, phục hồi môi trường không phải dễ dàng thực hiện nếu sao lưu cũ hơn của tập tin điều khiển phải sử dụng vì bản sao hiện hành không thể sử dụng. Do đó, hết sức cần những tập tin điều khiển đa công được sử dụng với mỗi cơ sở dữ liệu và mỗi bản sao được lưu trữ trên đĩa vật lý khác nhau.

Các sao lưu cơ sở dữ liệu

Dù kế hoạch sao lưu và phục hồi bạn đặt ra đối với một cơ sở dữ liệu của Oracle như thế nào đi nữa, nhưng các sao lưu các tập tin dữ liệu của cơ sở dữ liệu và các tập tin điều khiển hết sức cần thiết vì nó là thành phần của chiến lược để bảo vệ những sự cố môi trường có thể xảy ra và làm hư những tập tin.

Các phần sau cung cấp tổng quát khái niệm về những kiểu sao lưu khác nhau có thể được thực hiện và tính hữu dụng của chúng trong các kế hoạch phục hồi khác nhau.

Thông tin bổ sung:
Hướng dẫn Phục Hồi Và Sao Lưu Của Oracle8 về những nguyên tắc chủ đạo để thực hiện sao lưu cơ sở dữ liệu.

Sao lưu tổng thể cơ sở dữ liệu

Sao lưu tổng thể cơ sở dữ liệu là sao lưu hệ điều hành của tất cả những tập tin dữ liệu và tập tin điều khiển cấu tạo một cơ sở dữ liệu Oracle. Bạn có thể sao lưu tổng thể cơ sở dữ liệu khi cơ sở dữ liệu được đóng hoặc khi cơ sở dữ liệu được mở. Bạn không nên sau lưu sau khi môi trường làm việc bị sự cố hoặc những tình trạng bất thường khác.

Các sao lưu tổng thể nhất quán so với các sao lưu tổng thể không nhất quán

Thực hiện sau khi đóng không bị sai sót, tất cả những tập tin cấu tạo cơ sở dữ liệu được đóng và nhất quán liên quan đến thời gian hiện hành. Do đó, sao lưu tổng thể được thực hiện sau khi đóng có thể được sử dụng để phục hồi vào thời gian sao lưu tổng thể cuối cùng. Sao lưu tổng thể được thực hiện khi cơ sở dữ liệu mở không nhất quán vào thời gian ấn định, phải được phục hồi (với những tập tin redo log tàng trữ hoặc trực tuyến) trước khi cơ sở dữ liệu ấy có thể sử dụng được.

Các sao lưu và chế độ tàng trữ

Các tập tin có được từ sao lưu tổng thể được hữu dụng trong bất kỳ kiểu kế hoạch phục hồi môi trường:

Nếu cơ sở dữ liệu đang điều hoạt ở chế độ NOARCHIVELOG và sự cố đĩa làm hư một số hoặc tất cả những tập tin cấu tạo cơ sở dữ liệu ấy, sao lưu tổng thể nhất quán mới nhất có thể được sử dụng để phục hồi cơ sở dữ liệu.

Vì redo log tàng trữ không thể sử dụng để đưa cơ sở dữ liệu vào thời gian hiện hành, tất cả hoạt động cơ sở dữ liệu được thực hiện vì sao lưu phải được lặp lại. Dưới những trường hợp đặc biệt, sự cố đĩa ở chế độ NOARCHIVELOG có thể được phục hồi hoàn toàn, nhưng bạn không nên tin cậy vào điều này.

Nếu cơ sở dữ liệu đang điều hoạt ở chế độ ARCHIVELOG và sự cố đĩa làm hư một số hoặc tất cả những tập tin cấu tạo cơ sở dữ liệu ấy, những tập tin dữ liệu được tập hợp bằng sao lưu tổng thể gần nhất có thể được sử dụng làm thành phần của phục hồi cơ sở dữ liệu.

Sau khi phục hồi những tập tin dữ liệu cần thiết từ sao lưu tổng thể, phục hồi cơ sở dữ liệu có thể tiếp tục bằng cách áp dụng các tập tin redo log hiện hành và tàng trữ để đưa các tập tin dữ liệu được phục hồi vào thời gian hiện hành.

Nói tóm lại, nếu cơ sở dữ liệu được điều hoạt ở chế độ NOARCHIVELOG, sao lưu tổng thể cơ sở dữ liệu nhất quán là phương pháp duy nhất bảo vệ một phần cơ sở dữ liệu để tránh sự cố đĩa; nếu cơ sở dữ liệu đang điều hoạt ở chế độ ARCHIVELOG, sao lưu tổng thể cơ sở dữ liệu nhất quán hoặc không nhất quán có thể được sử dụng để phục hồi những tập tin bị hư là thành phần của phục hồi cơ sở dữ liệu từ một đĩa bị sự cố.

Các sao lưu cơ sở dữ liệu từng phần

Sao lưu cơ sở dữ liệu từng phần là sao lưu một phần của sao lưu tổng thể, được thực hiện khi cơ sở dữ liệu mở hoặc đóng. Sau đây là tất cả những thí dụ về các sao lưu cơ sở dữ liệu từng phần:

- sao lưu tất cả những tập tin dữ liệu đối với một không gian bảng cá biệt
- sao lưu một tập tin dữ liệu đơn
- sao lưu một tập tin điều khiển

Các sao lưu từng phần chỉ hữu dụng đối với cơ sở dữ liệu điều hoạt ở chế độ ARCHIVELOG. Bởi vì redo log tàng trữ hiện hữu, các tập tin dữ liệu được phục hồi từ sao lưu từng phần được thực hiện nhất quán với phần còn lại của cơ sở dữ liệu trong các thủ tục phục hồi.

Các sao lưu tập tin dữ liệu

Sao lưu từng phần chỉ có một số những tập tin dữ liệu của một cơ sở dữ liệu. Một phần hoặc các tập hợp của những cơ sở dữ liệu riêng biệt có thể được sao lưu độc lập với các tập tin dữ liệu khác và các tập tin điều khiển của một cơ sở dữ liệu. Bạn có thể sao lưu tập tin dữ liệu khi nó ngoại tuyến hoặc trực tuyến.

Chọn sao lưu tập tin dữ liệu trực tuyến hoặc ngoại tuyến chỉ phụ thuộc vào những đòi hỏi khả dụng về dữ liệu ấy - các sao lưu tập

tin dữ liệu trực tuyến là chọn lựa duy nhất nếu dữ liệu được sao lưu luôn luôn phải sử dụng được.

Các sao lưu tập tin điều khiển

Một hình thức sao lưu từng phần khác là sao lưu tập tin điều khiển. Vì tập tin điều khiển theo dõi cấu trúc tập tin vật lý của cơ sở dữ liệu được kết hợp, sao lưu một tập tin điều khiển của cơ sở dữ liệu phải được thực hiện mỗi lần thay đổi cấu trúc xảy ra đối với cơ sở dữ liệu ấy.

Lưu ý :
Recovery Manager tự động sao lưu tập tin điều khiển ở bất kỳ sao lưu nào, kể cả datafile 1 có từ điển dữ liệu.

Các tập tin điều khiển đa công bảo vệ sự mất mát tập tin điều khiển đơn. Tuy nhiên, nếu sự cố đĩa làm hư những tập tin dữ liệu và phục hồi được yêu cầu không hoàn toàn hoặc phục hồi về đúng thời gian được yêu cầu, một sao lưu tập tin điều khiển tương đương với cấu trúc cơ sở dữ liệu có dụng ý phải được sử dụng, không nhất thiết phải là tập tin điều khiển hiện hành. Do đó, sử dụng những tập tin điều khiển đa công không thay thế các sao lưu tập tin điều khiển được thực hiện mỗi lần cấu trúc của cơ sở dữ liệu ấy bị thay đổi.

Nếu bạn sử dụng Recovery Manager để phục hồi tập tin điều khiển trước khi phục hồi không hoàn toàn hoặc phục hồi vào đúng thời điểm, Recovery Manager tự động phục hồi tập tin điều khiển sao lưu thích hợp nhất.

Các tiện ích Export và Import

Các tiện ích Export và Import là những tiện ích được sử dụng để di chuyển dữ liệu của Oracle vào hoặc ra khỏi cơ sở dữ liệu Oracle. Export là một tiện ích để ghi dữ liệu từ một cơ sở dữ liệu

Oracle vào những tập tin hệ điều hành bằng định dạng cơ sở dữ liệu của Oracle. Các tập tin Export lưu trữ thông tin về các đối tượng giản đồ được tạo cho cơ sở dữ liệu. Import là một tiện ích đọc các tập tin Export và phục hồi thông tin tương ứng vào cơ sở dữ liệu có sẵn. Dù Export và Import được thiết kế để di chuyển dữ liệu Oracle, chúng cũng có thể được sử dụng làm phương pháp bổ sung để bảo vệ dữ liệu trong cơ sở dữ liệu Oracle.

Thông tin bổ sung :
Xem Các tiện ích Oracle8.

Các không gian bảng chỉ đọc và sao lưu

Bạn có thể tạo các sao lưu không gian bảng chỉ đọc khi cơ sở dữ liệu mở. Ngay sau khi tạo không gian bảng chỉ đọc, bạn nên sao lưu không gian ấy. Bao lâu không gian bảng vẫn duy trì chỉ đọc, không cần thực hiện thêm bất kỳ sao lưu nó.

Sau khi bạn thay đổi không gian bảng chỉ đọc đối với không gian bảng đọc-ghi, bạn cần hồi phục lại các sao lưu bình thường của không gian bảng ấy, giống như khi bạn đưa không gian bảng đọc-ghi trở lại trực tuyến.

Đưa các tập tin dữ liệu của không gian bảng chỉ đọc trực tuyến không tạo cho các tập tin này có thể ghi được và cũng không tạo cho tập tin ấy phần đầu được cập nhật. Do đó không nhất thiết phải thực hiện sao lưu những tập tin này như bạn phải thực hiện khi đưa tập tin dữ liệu đọc được trực tuyến.

Khả năng tồn tại

Trong trường hợp bị mất điện, sự cố phần cứng hoặc thiệt hại do ngắt hệ thống, Oracle cung ứng đặc trưng cơ sở dữ liệu dự phòng. Cơ sở dữ liệu dự phòng có mục đích đối với những vùng địa chỉ nơi khả năng tồn tại và phục hồi sự cố có tầm quan trọng tột bực.

Kế hoạch phục hồi hư hại

Cách duy nhất để bảo đảm được sự cố hệ thống hoặc thiệt hại khác là phải có kế hoạch kỹ lưỡng. Bạn phải có kế hoạch được xác lập với những thủ tục chi tiết. Dù bạn đang thực hiện một cơ sở dữ liệu dự phòng hoặc có một hệ thống cơ sở dữ liệu đơn, nhưng bạn phải có một kế hoạch về những gì phải thực hiện trong trường hợp bị sự cố thiệt hại nặng.

Cơ sở dữ liệu dự phòng

Oracle cung cấp một cơ chế hỗ trợ đáng tin cậy để thực hiện hệ thống cơ sở dữ liệu dự phòng, tạo thuận tiện phục hồi sự cố thảm khốc nhanh chóng. Kế hoạch sử dụng hệ thống thứ cấp vào phần cứng sao chép được duy trì ở trạng thái phục hồi môi trường không thay đổi bằng ứng dụng của những tập tin nhật ký được tàng trữ ở vùng sơ cấp. Trong trường hợp sự cố hệ thống sơ cấp, cơ sở dữ liệu dự phòng có thể được kích hoạt phục hồi nhanh nhất để cung cấp hệ thống có thể sử dụng ngay. Oracle cung cấp những lệnh có liên quan đến việc tạo và duy trì hệ thống dự phòng, làm tăng độ tin cậy của kế hoạch phục hồi sự cố bị thiệt hại nặng.

Cơ sở dữ liệu dự phòng sử dụng thông tin nhật ký tàng trữ từ cơ sở dữ liệu sơ cấp, vì vậy nó sẵn sàng thực hiện phục hồi và vào trực tuyến bất kỳ lúc nào. Khi cơ sở dữ liệu sơ cấp tàng trữ các redo log của nó, các sổ nhật ký phải được truyền tới vùng địa chỉ từ xa và được áp dụng vào cơ sở dữ liệu dự phòng. Do đó cơ sở dữ liệu dự phòng luôn luôn ở sau cơ sở dữ liệu sơ cấp lịch sử giao dịch và thời gian.

Phần cứng vật lý mà cơ sở dữ liệu dự phòng cư trú phải được sử dụng làm hệ thống phục hồi thiệt hại nặng; các ứng dụng khác không tiến hành nó. Bởi vì cơ sở dữ liệu dự phòng được thiết kế để phục hồi thiệt hại nặng, nó cư trú ở vị trí vật lý tách riêng với cơ sở dữ liệu sơ cấp rất tốt.

Cơ sở dữ liệu dự phòng tồn tại không những để bảo vệ tránh các sự cố nguồn (mất điện) và các sự cố phần cứng, mà còn bảo vệ dữ liệu của bạn trong trường hợp thiệt hại vật lý nặng, chẳng hạn như hỏa hoạn, bão táp hoặc động đất.

Thông tin bổ sung:
Xem Hướng dẫn Phục hồi và sao lưu Oracle8.

Phần VIII

XỬ LÝ ĐƯỢC PHÂN PHỐI VÀ CÁC CƠ SỞ DỮ LIỆU ĐƯỢC PHÂN PHỐI

Phần VIII mô tả các môi trường xử lý được phân phối cho Oracle server và các ứng dụng cơ sở dữ liệu, giải thích kiến trúc cơ sở dữ liệu được phân phối và sao chép dữ liệu qua các bảng.

Phần VIII có các chương sau:

Chương 29 Xử Lý Được Phân Phối
Chương 30 Cơ Sở Dữ Liệu Được Phân Phối
Chương 31 Sao Chép Cơ Sở Dữ Liệu

Chương 29

Xử lý được phân phối

Chương này xác định xử lý được phân phối và mô tả Oracle server và các ứng dụng cơ sở dữ liệu hoạt động trong môi trường xử lý được phân phối như thế nào. Tài liệu này áp dụng đối với hầu hết mọi kiểu môi trường hệ thống cơ sở dữ liệu Oracle.

Chương này gồm:

- **Kiến trúc Client/Server của Oracle**
- **Xử lý được phân phối**

Kiến trúc Oracle Client/Server

Trong môi trường hệ thống cơ sở dữ liệu Oracle, ứng dụng cơ sở dữ liệu và cơ sở dữ liệu được tách ra thành hai phần:
phần front-end (phía client) hoặc phần client và back-end (phía server) hoặc phần server - vì thế có thuật ngữ kiến trúc client/server. Client tiến hành ứng dụng cơ sở dữ liệu để truy xuất thông tin cơ sở dữ liệu và tương tác với người dùng bằng bàn phím, màn hình và thiết bị trỏ, chẳng hạn như chuột. Server tiến hành phần mềm Oracle và điều quản các chức năng cần

thiết đồng thời, truy xuất dữ liệu dùng chung đối với cơ sở dữ liệu Oracle.

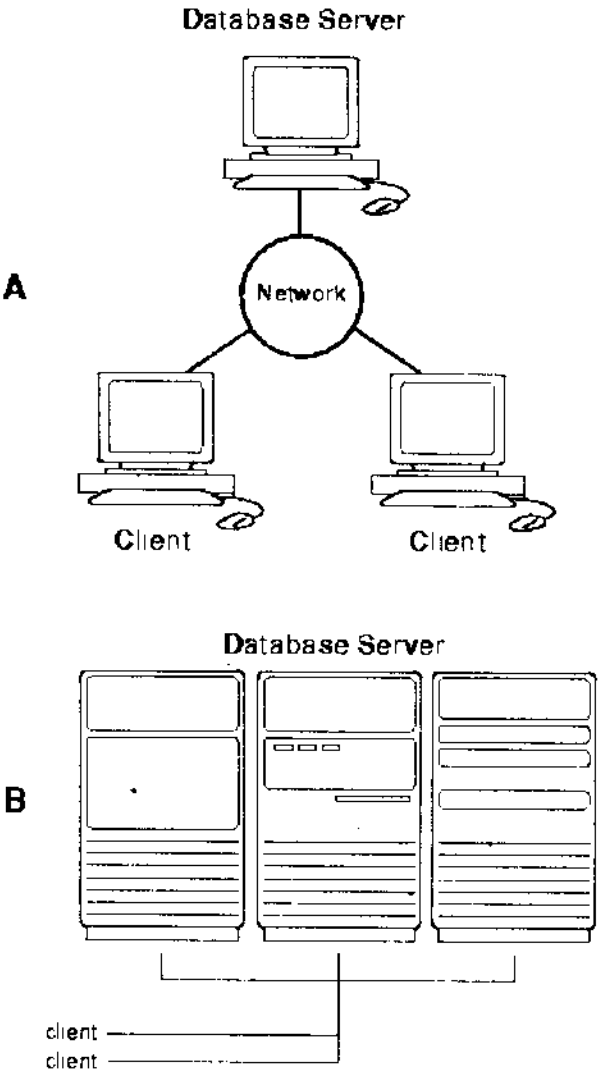
Dù ứng dụng client và Oracle có thể được tiến hành trên cùng máy tính, nhưng thường đạt được hiệu quả hơn khi (các) phần client và server được các máy tính khác nhau kết nối qua mạng tiến hành. Các phần sau đề cập đến những khác biệt có thể xảy ra trong kiến trúc client/Oracle server.

Xử lý được phân phối

Xử lý được phân phối là cách sử dụng hơn một bộ xử lý để thực hiện xử lý cho một tác vụ khác biệt. Các thí dụ về xử lý được phân phối trong các hệ thống cơ sở dữ liệu Oracle được trình bày ở Hình 29-1.
Ở Part A của hình này, client và server được định vị trên nhiều máy tính; những máy tính này được kích khởi bằng một mạng.
Ở Part B của hình này, một máy tính có hơn một bộ xử lý và các bộ xử lý khác tách sử dụng tiến hành của ứng dụng client từ Oracle.

Lưu ý:
Chương này áp dụng vào những môi trường có một cơ sở dữ liệu trên một server. Trong một cơ sở dữ liệu được phân phối, một server (Oracle) có thể cần truy xuất cơ sở dữ liệu trên một server khác. Xem "Các Cơ Sở Dữ Liệu Được Phân Phối" về các client và các server trong các cơ sở dữ liệu được phân phối.

Hình 29-1 Kiến trúc Client/Server và xử lý được phân phối



Trong thí dụ về nối mạng (Part A), server và các client truyền thông bằng Net8, giao diện mạng của Oracle. Xem "Net8".

Kiến trúc Oracle client/server trong môi trường xử lý được phân phối cung cấp những ưu điểm sau:

- Các ứng dụng Client không chịu trách nhiệm thực hiện bất kỳ xử lý dữ liệu nào. Đúng hơn, chúng yêu cầu nhập liệu từ các người dùng, dữ liệu yêu cầu từ server và sau đó phân tích và trình bày dữ liệu này bằng cách sử dụng những khả năng hiển thị của trạm làm việc client hoặc đầu cuối (thí dụ, sử dụng đồ họa hoặc các bảng tính).
- Các ứng dụng Client không phụ thuộc vào vị trí vật lý của dữ liệu. Nếu dữ liệu được di chuyển hoặc được phân phối tới các server cơ sở dữ liệu, ứng dụng ấy tiếp tục thực hiện chức năng với một ít sửa đổi hoặc không phải sửa đổi.
- Oracle khai thác những phương tiện bộ nhớ dùng chung và đa tác vụ của hệ điều hành cơ sở của nó. Do đó, nó cam kết ở mức độ đồng thời, tính nguyên vẹn dữ liệu và hiệu suất cao nhất tới các ứng dụng client của nó.
- Các trạm làm việc Client hoặc các trạm có thể được tối ưu hóa để biểu thị dữ liệu (thí dụ, cung cấp hỗ trợ đồ họa và chuột) và server có thể được tối ưu hóa để xử lý và lưu trữ dữ liệu (thí dụ, bằng cách có được số lượng bộ nhớ và không gian đĩa lớn).
- Trong các môi trường nối mạng, bạn có thể sử dụng các trạm làm việc client ít tiền để truy xuất dữ liệu từ xa của server một cách hiệu quả.
- Nếu cần, Oracle có thể được thích ứng khi hệ thống của bạn phát triển. Bạn có thể bổ sung nhiều server để phân phối nạp xử lý cơ sở dữ liệu trên khắp mạng (được thích ứng theo chiều ngang) hoặc bạn có thể di chuyển Oracle tới một máy vi tính hoặc máy tính lớn để khai thác hiệu suất của hệ thống lớn hơn (được thích ứng theo chiều dọc). Trong hai trường hợp, tất cả dữ liệu và các ứng dụng được duy trì có ít hoặc không có sửa đổi, vì Oracle khả chuyển giữa các hệ thống.

- Trong những môi trường nối mạng, dữ liệu dùng chung được lưu trữ trên các server, thay vì trên tất cả các máy tính trong hệ thống ấy. Điều này tạo quản lý truy xuất đồng thời dễ dàng và hiệu quả hơn.
- Trong các môi trường được nối mạng, những ứng dụng client đệ trình những yêu cầu cơ sở dữ liệu bằng cách sử dụng những câu lệnh SQL. Một khi nhận được, câu lệnh SQL được server xử lý và các kết quả được cho trở lại ứng dụng client. Lưu thông mạng phải lưu ở mức tối thiểu vì chỉ những yêu cầu và những kết quả được truyền trên mạng.

Net8

Net8 là giá trị mạng Oracle cho phép các công cụ Oracle tiến hành trên các trạm làm việc mạng và các server truy xuất, sửa đổi và lưu trữ dữ liệu trên các server khác. Net8 được coi là thành phần của giao diện có thể trong các truyền thông mạng. Xem "Cấu Trúc Xử Lý" về giao diện chương trình.

Net8 sử dụng các giao thức truyền thông hoặc các giao diện lập trình ứng dụng (APIs) được hỗ trợ bởi các mạng có phạm vi rộng để cung cấp cơ sở dữ liệu được phân phối và xử lý được phân phối cho Oracle.

Giao thức truyền thông là tập hợp những tiêu chuẩn được bổ sung vào phần mềm để điều phối truyền dữ liệu trên mạng.

Một API là tập hợp những trình con, trong trường hợp của các mạng để cung cấp phương tiện thiết lập truyền thông liên xử lý từ xa bằng một giao thức truyền thông.

Các giao thức truyền thông xác định cách dữ liệu được chuyển giao và nhận trên mạng. Trong môi trường được nối mạng, một

Oracle server truyền thông với các trạm client và các Oracle server khác bằng cách sử dụng Net8. Net8 hỗ trợ các truyền thông trên tất cả các giao thức mạng chính có phạm vi từ các giao thức được các PC LAN hỗ trợ tới các giao thức sử dụng các hệ thống máy tính lớn nhất.

Không sử dụng Net8, Developer ứng dụng phải viết mã thủ công tất cả các truyền thông trong một quyết định để điều hoạt trong môi trường xử lý phân phối được nối mạng. Nếu phần cứng mạng, tôpô hoặc giao thức thay đổi, thì ứng dụng ấy phải được sửa đổi thích ứng theo.

Tuy nhiên, bằng cách sử dụng Net8, Developer ứng dụng không phải quan tâm đến hỗ trợ các truyền thông mạng trong một ứng dụng cơ sở dữ liệu. Nếu giao thức cơ sở thay đổi, thì người điều hành cơ sở dữ liệu tạo ra một số thay đổi nhỏ, khi ứng dụng không đòi hỏi các sửa đổi, sẽ tiếp tục thực hiện chức năng.

Net8 hoạt động như thế nào

Các trình điều khiển Net8 cung cấp một giá trị giữa các xử lý Oracle tiến hành trên server cơ sở dữ liệu và các xử lý người dùng của những công cụ Oracle để tiến hành trên các máy tính khác của mạng.

Các trình điều khiển Net8 sử dụng các câu lệnh SQL từ giao diện của công cụ Oracle và đóng gói chúng để truyền tới Oracle bằng một trong những giao thức mức cao hơn tiêu chuẩn công nghệ hỗ trợ hoặc các giá trị lập trình. Các trình điều khiển còn nhận những trả lời từ Oracle và đóng gói chúng để truyền tới công cụ bằng cơ chế truyền thông mức cao hơn. Điều này đều được thực hiện độc lập với hệ điều hành mạng.

Thông tin bổ sung:
Tùy theo hệ điều hành tiến hành Oracle, phần mềm Net8 của

server cơ sở dữ liệu có thể có phần mềm trình điều khiển và khởi động xử lý nền Oracle bổ sung; xem thêm tài liệu hệ điều hành Oracle xác định.

Xem Hướng dẫn Oracle8 Net8 Administrator về Net8.

Chương 30

Các cơ sở dữ liệu được phân phối

Chương này mô tả các khái niệm căn bản và thuật ngữ về kiến trúc cơ sở dữ liệu được phân phối của Oracle, gồm:

- **Kiến trúc cơ sở dữ liệu được phân phối của Oracle**
- **Cơ sở dữ liệu được phân phối hỗn hợp**
- **Phát triển các ứng dụng cơ sở dữ liệu được phân phối**
- **Điều hành các ứng dụng cơ sở dữ liệu được phân phối**
- **Hỗ trợ ngôn ngữ quốc gia**

Kiến trúc cơ sở dữ liệu được phân phối của Oracle

Cơ sở dữ liệu được phân phối là tập hợp những cơ sở dữ liệu được lưu trữ trên nhiều máy tính thường xuất hiện trên các ứng dụng là một cơ sở dữ liệu đơn. Do đó, một ứng dụng có thể truy xuất và sửa đổi dữ liệu trong nhiều cơ sở dữ liệu của một mạng đồng thời. Mỗi cơ sở dữ liệu Oracle trong hệ thống được Oracle server

cục bộ của nó điều khiển nhưng nó phối hợp để duy trì tính nhất quán của cơ sở dữ liệu được phân phối toàn cục.

Hình 30-1 minh họa hệ thống cơ sở dữ liệu được phân phối của Oracle.

Các Client và các Server

Server cơ sở dữ liệu là phần mềm Oracle quản lý cơ sở dữ liệu và client là một ứng dụng yêu cầu thông tin từ một server. Mỗi máy tính trong một mạng là một nút. Một nút trong hệ thống cơ sở dữ liệu được phân phối là một client, server hoặc cả hai tùy theo từng trường hợp. Thí dụ, ở Hình 30-1, máy tính quản lý cơ sở dữ liệu HQ đóng vai trò là một server cơ sở dữ liệu khi một câu lệnh được phát hành dựa trên dữ liệu cục bộ của nó (thí dụ, câu lệnh thứ hai trong mỗi giao dịch phát hành một truy vấn dựa trên bảng DEPT cục bộ) và đóng vai trò là một client khi nó phát hành câu lệnh dựa trên dữ liệu từ xa (thí dụ, câu lệnh thứ nhất trong mỗi giao dịch được phát hành dựa trên bảng EMP từ xa trong cơ sở dữ liệu SALES).

Các kết nối trực tiếp và gián tiếp

Một client có thể kết nối trực tiếp hoặc gián tiếp với server cơ sở dữ liệu. Ở Hình 30-1, khi ứng dụng client phát hành các câu lệnh thứ nhất và thứ ba cho mỗi giao dịch, client được kết nối trực tiếp với cơ sở dữ liệu HQ trung gian và gián tiếp với cơ sở dữ liệu SALES có dữ liệu từ xa.

Mạng

Để liên kết các cơ sở dữ liệu cá biệt của hệ thống cơ sở dữ liệu được phân phối, cần phải có một mạng. Các phần sau giải thích thêm về các vấn đề mạng trong Hệ thống cơ sở dữ liệu được phân phối của Oracle.

Net8

Tất cả những cơ sở dữ liệu Oracle trong hệ thống cơ sở dữ liệu được phân phối sử dụng phần mềm mạng của Oracle, Net8 để tạo thuận tiện cho truyền thông liên cơ sở dữ liệu qua một mạng. Giống như Net8 kết nối các client và các server điều hoạt trên các máy tính khác nhau của mạng để hỗ trợ các giao diện được phân phối và các giao dịch từ xa trong một cơ sở dữ liệu được phân phối.

Net8 tạo kết nối thông suốt cần thiết để truyền các yêu cầu SQL và nhận những ứng dụng sử dụng hệ thống ấy. Net8 lấy các câu lệnh từ client và nén chúng để truyền tới Oracle server trên một giao thức hỗ trợ tiêu chuẩn công nghệ hoặc các giao diện lập trình. Net8 còn lấy các bản trả lời từ một server và đóng gói chúng để truyền trở lại client thích hợp. Net8 thực hiện tất cả xử lý độc lập với hệ điều hành mạng cơ sở.

Thông tin bổ sung:
Xem Hướng dẫn của Oracle8 Net8 Administrator về Net8 và các đặc trưng của nó.

Oracle Names

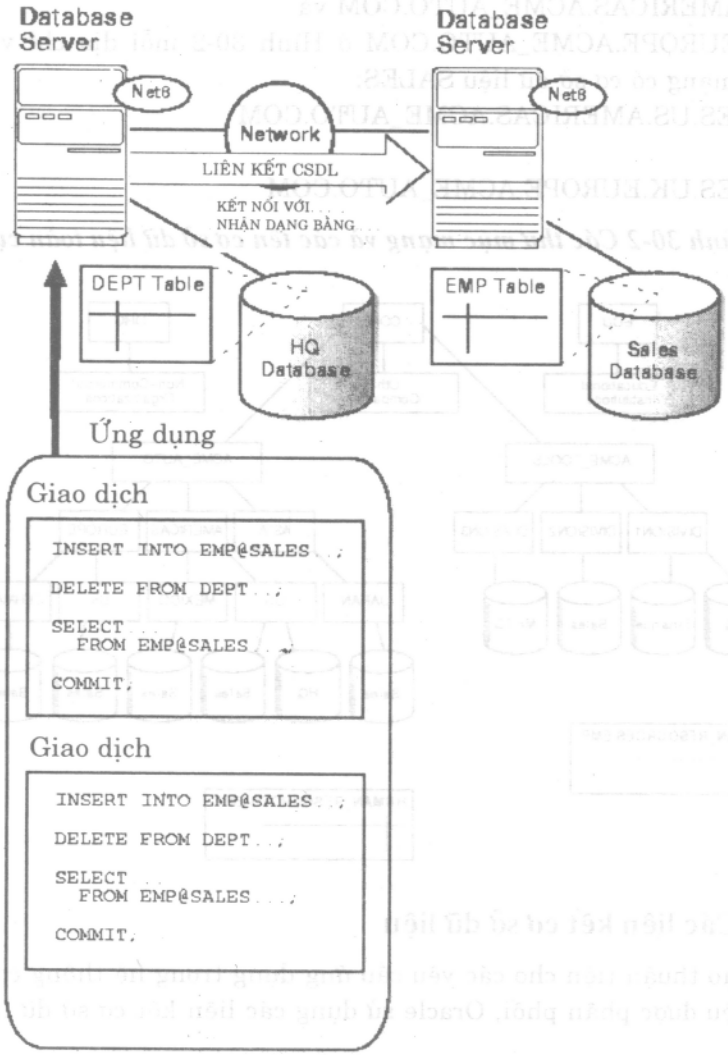
Mạng Oracle có thể sử dụng Oracle Names một cách tùy chọn để cung cấp cho hệ thống ấy dịch vụ thư mục toàn cục. Khi mạng Oracle hỗ trợ hệ thống cơ sở dữ liệu được phân phối, bạn có thể sử dụng các Oracle Names server làm kho lưu trữ thông tin trung tâm về cơ sở dữ liệu trong hệ thống ấy để dễ dàng cấu hình truy xuất cơ sở dữ liệu được phân phối.

Các cơ sở dữ liệu và các liên kết cơ sở dữ liệu

Mỗi cơ sở dữ liệu trong cơ sở dữ liệu được phân phối được phân biệt với tất cả những cơ sở dữ liệu khác trong hệ thống ấy và có

tên cơ sở dữ liệu riêng của nó. Oracle tạo tên cơ sở dữ liệu toàn cục của cơ sở dữ liệu bằng cách lập tiền tố vùng địa chỉ mạng của cơ sở dữ liệu với tên của cơ sở dữ liệu cá biệt.

Hình 30-1 Hệ thống cơ sở dữ liệu được phân phối

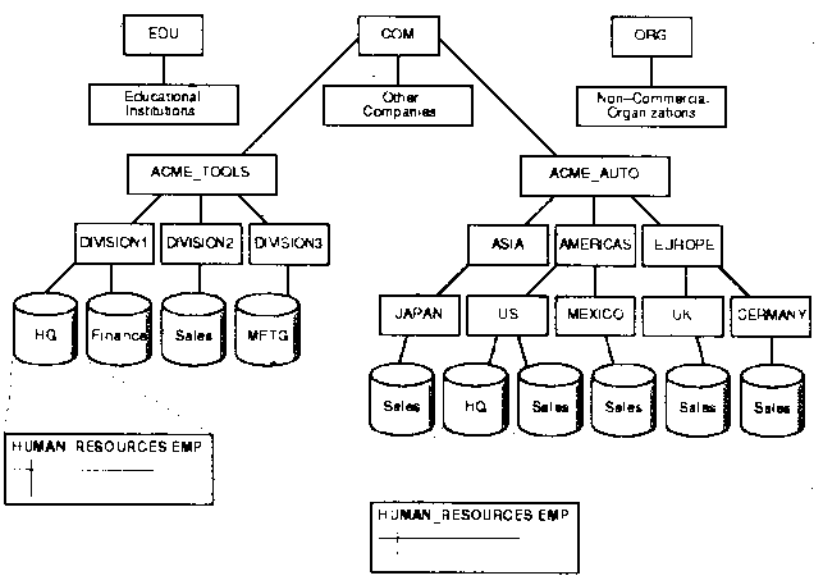


Thí dụ, Hình 30-2 minh họa sắp xếp phân cấp tiêu biểu của các cơ sở dữ liệu trên khắp mạng.

Dù nhiều cơ sở dữ liệu có thể có cùng tên cá biệt, nhưng mỗi cơ sở dữ liệu phải có tên cơ sở dữ liệu toàn cục.
Thí dụ, các địa chỉ vùng chủ mạng
US.AMERICAS.ACME_AUTO.COM và
UK.EUROPE.ACME_AUTO.COM ở Hình 30-2 mỗi địa chỉ vùng chủ mạng có cơ sở dữ liệu SALES:
SALES.US.AMERICAS.ACME_AUTO.COM

SALES.UK.EUROPE.ACME_AUTO.COM

Hình 30-2 Các thư mục mạng và các tên cơ sở dữ liệu toàn cục



Các liên kết cơ sở dữ liệu

Để tạo thuận tiện cho các yêu cầu ứng dụng trong hệ thống cơ sở dữ liệu được phân phối, Oracle sử dụng các liên kết cơ sở dữ liệu.

Một liên kết cơ sở dữ liệu xác định đường dẫn truyền thông một chiều từ cơ sở dữ liệu Oracle tới một cơ sở dữ liệu khác.

Các liên kết cơ sở dữ liệu đối với người dùng hệ thống cơ sở dữ liệu được phân phối của Oracle rất dễ hiểu, vì tên của liên kết cơ sở dữ liệu giống với tên toàn cục của cơ sở dữ liệu mà liên kết ấy chỉ. Thí dụ, câu lệnh SQL sau tạo liên kết cơ sở dữ liệu ở cơ sở dữ liệu cục bộ mô tả đường dẫn tới cơ sở dữ liệu SALES.US.AMERICAS.ACME_AUTO.COM từ xa.

```
CREATE DATABASE LINK sales.us.americas.acme_auto.com ... ;
```

Sau khi tạo một liên kết cơ sở dữ liệu, các ứng dụng được kết nối với cơ sở dữ liệu cục bộ có thể truy xuất dữ liệu ở cơ sở dữ liệu SALES.US.AMERICAS.ACME_AUTO.COM từ xa. Phần kế tiếp giải thích các ứng dụng có thể tham chiếu các đối tượng giản đồ từ xa trong cơ sở dữ liệu được phân phối như thế nào và có những thí dụ về các câu lệnh SQL sử dụng các liên kết cơ sở dữ liệu như thế nào.

Thông tin bổ sung:
Oracle hỗ trợ nhiều kiểu liên kết cơ sở dữ liệu khác nhau. Xem Các Hệ Thống Cơ Sở Dữ Liệu Được Phân Phối Của Oracle.

Giải pháp tên đối tượng giản đồ

Để giải quyết ứng dụng tham chiếu các đối tượng giản đồ (một tiến trình được gọi là giải pháp tên) Oracle hình thành các tên đối tượng sử dụng cách giải quyết phân cấp. Thí dụ, trong một cơ sở dữ liệu đơn, Oracle bảo đảm là mỗi giản đồ có một tên duy nhất và trong một giản đồ, mỗi đối tượng có một tên duy nhất. Do đó, tên của đối tượng giản đồ luôn luôn duy nhất trong cơ sở dữ liệu ấy. Hơn nữa, Oracle có thể dễ dàng giải quyết ứng dụng tham chiếu tên cục bộ của đối tượng giản đồ.

Trong một cơ sở dữ liệu được phân phối, đối tượng giản đồ chẳng hạn như một bảng truy xuất được tất cả những ứng dụng trong hệ thống. Oracle chỉ mở rộng mô hình tên phân cấp với các tên cơ sở dữ liệu toàn cục để tạo các tên đối tượng toàn cục một cách hiệu quả và giải quyết các tham chiếu những đối tượng giản đồ trong hệ thống cơ sở dữ liệu được phân phối. Thí dụ, một truy vấn có thể tham chiếu một bảng từ xa bằng cách ấn định tên hội chuẩn đầy đủ, kể cả cơ sở dữ liệu mà nó cư trú.

```
SELECT * FROM scott.emp@sales.us.americas.acme_auto.com;
```

Để hoàn tất yêu cầu, server cơ sở dữ liệu cục bộ sử dụng ngầm định liên kết cơ sở dữ liệu kết nối với cơ sở dữ liệu SALES từ xa.

Kết nối giữa các phiên bản Oracle Server

Hệ thống cơ sở dữ liệu được phân phối của Oracle có thể phối hợp các cơ sở dữ liệu Oracle của các phiên bản khác nhau. Tất cả các bản phát hành Oracle được hỗ trợ có thể tham gia vào hệ thống cơ sở dữ liệu được phân phối. Tuy nhiên, các ứng dụng hoạt động với cơ sở dữ liệu được phân phối phải hiểu tính năng có thể sử dụng tại mỗi nút trong hệ thống. Thí dụ, ứng dụng cơ sở dữ liệu được phân phối không thể chờ một cơ sở dữ liệu Oracle7 hiểu các phần mở rộng SQL đối tượng có thể sử dụng với Oracle8.

Các cơ sở dữ liệu được phân phối và xử lý được phân phối

Các thuật ngữ "cơ sở dữ liệu được phân phối" và "xử lý được phân phối" có liên quan mật thiết, nhưng có ý nghĩa rất khác biệt.

Cơ sở dữ liệu được phân phối (Distributed Database)	Cơ sở dữ liệu được phân phối là tập hợp các cơ sở dữ liệu trên nhiều máy tính xuất hiện với các quyết định bằng một cơ sở dữ liệu đơn.

Xử lý được phân phối (Distributed Processing)	Xử lý được phân phối xảy ra khi một hệ thống ứng dụng phân phối các tác vụ của nó giữa những máy tính khác nhau trong một mạng. Thí dụ, ứng dụng cơ sở dữ liệu thường phân phối các cách trình bày front-end với các client PC hoặc NC và cho phép server cơ sở dữ liệu back-end (phía server) quản lý truy xuất một cơ sở dữ liệu dùng chung. Do đó, hệ thống xử lý ứng dụng cơ sở dữ liệu được phân phối thường tham chiếu khi là một hệ thống ứng dụng cơ sở dữ liệu "client-server" hơn.

Các hệ thống cơ sở dữ liệu được phần mềm của Oracle sử dụng kiến trúc xử lý được phân phối để thực hiện chức năng. Thí dụ, một Oracle server đóng vai trò một client khi nó yêu cầu dữ liệu của một Oracle server khác quản lý.

Các cơ sở dữ liệu được phân phối và sao chép cơ sở dữ liệu

Các thuật ngữ "cơ sở dữ liệu được phân phối - distributed database" và "sao chép cơ sở dữ liệu - database replication" cũng có liên quan mật thiết, nhưng khác nhau. Trong cơ sở dữ liệu được phân phối thuần túy, hệ thống quản lý một bản sao đơn của tất cả dữ liệu và những đối tượng cơ sở dữ liệu hỗ trợ. Các ứng dụng cơ sở dữ liệu được phân phối thường sử dụng các giao dịch được phân phối để truy xuất cả dữ liệu cục bộ và từ xa để sửa đổi cơ sở dữ liệu toàn cục vào thời gian thực.

Lưu ý:
Chương này đề cập các cơ sở dữ liệu được phân phối thuần túy. Xem "Sao Chép Cơ Sở Dữ Liệu" về sao chép.

Sao chép là tiến trình sao chép và duy trì những đối tượng cơ sở dữ liệu trong nhiều cơ sở dữ liệu để thiết lập hệ thống cơ sở dữ liệu được phân phối. Dù sao chép dựa vào công nghệ cơ sở dữ liệu được phân phối để thực hiện chức năng, nhưng sao chép cơ sở dữ liệu có thể cung ứng những ưu điểm ứng dụng không thể có trong môi trường cơ sở dữ liệu được phân phối. Phổ biến nhất, khi sao chép cần thiết để tăng hiệu suất và bảo vệ tính khả dụng của các ứng dụng vì các tùy chọn truy xuất dữ liệu tồn tại. Thí dụ, một ứng dụng thường có thể truy xuất cơ sở dữ liệu cục bộ thay vì server từ xa để giảm tối đa lưu thông mạng và đạt được hiệu suất cao tối đa. Hơn nữa, ứng dụng có thể tiếp tục thực hiện chức năng nếu server cục bộ gặp phải sai sót, nhưng các server có dữ liệu đã sao chép vẫn truy xuất được.

Thông tin bổ sung:
Xem Sao chép của Oracle8 về các đặc trưng sao chép của Oracle.

Các cơ sở dữ liệu được phân phối hỗn hợp

Cơ sở dữ liệu được phân phối hỗn hợp là cơ sở dữ liệu được phân phối trong đó ít nhất một trong các cơ sở dữ liệu ấy là hệ thống khác Oracle. Truy xuất các hệ thống khác Oracle từ Oracle server được Oracle Open Gateways cung cấp. Oracle server cùng với lối cổng ấy có thể cung cấp toàn bộ khả năng phân phối hỗn hợp thông suốt cho ứng dụng ấy. Nghĩa là ứng dụng đó không phải nhận biết hệ thống khác Oracle được truy xuất.

Các liên kết cơ sở dữ liệu được sử dụng giữa Oracle server và hệ thống khác Oracle để truy xuất dữ liệu trong hệ thống khác Oracle hoặc tiến trình thủ tục từ xa trong hệ thống khác Oracle. Bằng cách tích hợp hệ thống giao dịch của hệ thống khác Oracle với Oracle server, tính nguyên vẹn dữ liệu có thể được bảo đảm.

Truy xuất SQL thông suốt

Ứng dụng chỉ phát hành cửa sổ câu lệnh SQL của Oracle dựa trên Oracle server cục bộ. Câu lệnh SQL ấy có thể tham chiếu dữ liệu trong hệ thống khác Oracle từ xa, giống như nếu nó là hệ thống Oracle từ xa. Oracle server cùng với lối cổng sẽ thực hiện các phiên dịch cần thiết để truy xuất hệ thống khác Oracle bằng phương ngữ SQL của riêng nó.

Truy xuất thủ tục

Một số hệ thống khác Oracle cần được truy xuất theo thủ tục. Ứng dụng ấy chỉ phát hành một lệnh gọi thủ tục từ xa PL/SQL và Oracle server cùng với lối cổng, sẽ thực hiện các phiên dịch để tiến hành các thủ tục hoặc các hàm trong hệ thống khác Oracle từ xa. Thí dụ, thủ tục từ xa có thể giao diện với một hệ thống gửi thông điệp và đưa các thông điệp vào hệ thống gửi thông điệp khác Oracle. Nếu hệ thống gửi thông điệp ấy có hỗ trợ giao dịch, thì Oracle server cùng với lối cổng có thể thực hiện các điều hoạt trong hệ thống gửi thông điệp ở giao dịch được phân phối Oracle lớn hơn để bảo đảm tất cả các thay đổi (cả thông tin gửi thông điệp và trong Oracle server) được cam kết hoặc được hồi phục về trạng thái cũ.

Các đặc trưng Gateway

Nói tóm lại, có các đặc trưng Oracle Open Gateways, nhưng không bị giới hạn vào:

Distributed Transactions. Một giao dịch có thể nối cả các hệ thống Oracle và khác Oracle, khi vẫn bảo đảm, qua cơ chế cam kết hai pha của Oracle mà các thay đổi đều được cam kết hoặc được hồi phục về trạng thái cũ.

Truy xuất Transparent SQL. Tích hợp dữ liệu từ các hệ thống khác Oracle vào môi trường Oracle như dữ liệu được lưu trữ trong một cơ sở dữ liệu cục bộ đơn. Các câu lệnh SQL được phát hành

bằng ứng dụng được biến thể thông suốt thành câu lệnh SQL được hệ thống khác Oracle nhận biết.

Procedural Access. Các hệ thống thủ tục, như các hệ thống gửi thông điệp và sắp hàng chờ được truy xuất từ Oracle server bằng cách sử dụng các lệnh gọi thủ tục PL/SQL từ xa.

Các phiên dịch Data Dictionary. Để tạo cho hệ thống khác Oracle xuất hiện như một Oracle server khác, các câu lệnh SQL có các tham chiếu các bảng từ điển dữ liệu của Oracle được biến thể thành các câu lệnh SQL có các tham chiếu các bảng từ điển dữ liệu của hệ thống khác Oracle.

Pass-through SQL. Các lập trình viên ứng dụng có thể tùy chọn truy xuất trực tiếp hệ thống khác Oracle từ ứng dụng Oracle bằng cách sử dụng phương ngữ SQL của hệ thống khác Oracle.

Truy xuất các thủ tục định sẵn. Các thủ tục định sẵn trong các hệ thống khác Oracle dựa trên SQL được truy xuất như chúng là các thủ tục PL/SQL.

National Language Support. Gateways hỗ trợ các bộ ký tự đa byte và sẽ phiên dịch các bộ ký tự giữa hệ thống khác Oracle và Oracle server.

Tối ưu hóa truy vấn toàn cục. Số và các chỉ mục trên các bảng ở hệ thống khác Oracle bộ tối ưu hóa Oracle lưu ý và được phân tích để tạo các câu lệnh SQL có hiệu lực được tiến hành ở hệ thống khác Oracle.

Thông tin bổ sung:

> Không phải tất cả những đặc trưng được liệt kê ở trên có thể áp dụng cho các lối cổng cá biệt của bạn. Xem tài liệu về các lối cổng đối với các đặc trưng được hỗ trợ.

Version 8 Gateways

Version 8 Gateways được tích hợp chặt chẽ với Oracle server bằng cách sử dụng đặc trưng Heterogeneous Services của Oracle. Heterogeneous Services được tích hợp vào Oracle server và do đó,

các tác vụ điều hành đối với truy xuất hỗn hợp hiện nay chung cho các lối cổng khác nhau và được tích hợp vào Oracle server.

Thông tin bổ sung:
Xem Oracle8 Replication về Heterogeneous Services.

Version 4 Gateways

Các lối cổng của Version 4 được hỗ trợ dựa trên các Oracle server7 và Oracle8.

Thông tin bổ sung:
Xem Oracle Open Gateway Technology, Concepts và Administration Guide Version 4 về thông tin lối cổng.

Phát triển các ứng dụng cơ sở dữ liệu được phân phối

Khi bạn thiết lập các ứng dụng trên đầu của hệ thống cơ sở dữ liệu được phân phối, có nhiều vấn đề cần xét. Các phần sau giải thích các ứng dụng truy xuất dữ liệu trong một cơ sở dữ liệu được phân phối như thế nào.

Các câu lệnh SQL được phân phối và từ xa

Một truy vấn từ xa là truy vấn lựa thông tin từ một hoặc nhiều bảng từ xa, tất cả cư trú ở cùng nút từ xa. Thí dụ:

```
SELECT * FROM scott.dept@sales.us.americas.acme_auto.com;
```

Cập nhật từ xa là một cập nhật sửa đổi dữ liệu ở một hoặc nhiều bảng, tất cả được định vị ở cùng nút từ xa. Thí dụ:

```
UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK'
  WHERE deptno = 10;
```

Lưu ý :
Cập nhật từ xa có thể có truy vấn con để truy hồi dữ liệu từ một hoặc nhiều nút từ xa, nhưng vì cập nhật xảy ra chỉ ở một nút từ xa đơn, câu lệnh ấy được phân loại là một cập nhật từ xa.

Truy vấn được phân phối truy hồi thông tin từ hai hoặc nhiều nút. Thí dụ:

```
SELECT ename, dname
FROM scott.emp e, scott.dept@sales.us.americas.acme_auto.com d
  WHERE e.deptno = d.deptno;
```

Một cập nhật được phân phối sửa đổi dữ liệu trên hai hoặc nhiều nút. Một cập nhật được phân phối có thể sử dụng đơn vị chương trình con PL/SQL, chẳng hạn như một thủ tục hoặc bộ kích khởi có hai hoặc nhiều cập nhật từ xa để truy xuất dữ liệu trên các nút khác nhau. Thí dụ:

```
BEGIN
  UPDATE scott.dept@sales.us.americas.acme_auto.com
    SET loc = 'NEW YORK'
    WHERE deptno = 10;
  UPDATE scott.emp
    SET deptno = 11
    WHERE deptno = 10;
END;
```

Các câu lệnh trong chương trình ấy được gửi tới các nút từ xa và tiến hành của nó thành công hoặc không thành công bằng một đơn vị.

Các lệnh gọi thủ tục từ xa - Remote Procedure Calls (RPCs)

Các Developer có thể viết mà các bộ phần mềm PL/SQL và các thủ tục để hỗ trợ các ứng dụng hoạt động với cơ sở dữ liệu được phân phối. Các ứng dụng có thể lập các lệnh gọi thủ tục cục bộ

để thực hiện hoạt động ở cơ sở dữ liệu cục bộ và các lệnh gọi thủ tục từ xa (RPCs) để thực hiện hoạt động ở cơ sở dữ liệu từ xa. Khi một chương trình gọi một thủ tục từ xa, thì server cục bộ truyền tất cả các tham số thủ tục tới server từ xa trong lệnh gọi ấy. Thí dụ:

```
BEGIN
  emp_mgmt.del_emp@sales.us.americas.acme_auto.com(1257);
END;
```

Khi phát triển các bộ phần mềm và các thủ tục cho các hệ thống cơ sở dữ liệu được phân phối, các Developer phải viết mã để nhận ra những đơn vị chương trình nào thực hiện ở các vị trí từ xa và cách để các kết quả trở lại ứng dụng gọi.

Các giao dịch được phân phối và từ xa

Remote transaction (giao dịch từ xa) có một hoặc nhiều câu lệnh từ xa, tất cả những câu lệnh tham chiếu cùng nút từ xa. Thí dụ:

```
UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK'
  WHERE deptno = 10;
UPDATE scott.emp@sales.us.americas.acme_auto.com
  SET deptno = 11
  WHERE deptno = 10;
COMMIT;
```

Một giao dịch được phân phối có một hoặc nhiều câu lệnh cá biệt hoặc bằng một nhóm để cập nhật dữ liệu trên hai hoặc nhiều nút riêng biệt của cơ sở dữ liệu được phân phối. Thí dụ:

```
UPDATE scott.dept@sales.us.americas.acme_auto.com
  SET loc = 'NEW YORK'
  WHERE deptno = 10;
UPDATE scott.emp
  SET deptno = 11
  WHERE deptno = 10;
```

COMMIT;

Lưu ý:
Nếu tất cả các câu lệnh của một giao dịch chỉ tham chiếu một nút từ xa đơn, thì giao dịch từ xa ấy không được phân phối.

Cơ chế cam kết hai pha

DBMS phải bảo đảm tất cả các câu lệnh trong một giao dịch được phân phối hoặc không được phân phối cam kết hoặc hồi phục về trạng thái cũ bằng một đơn vị, để nếu giao dịch ấy được thiết kế chính xác, thì dữ liệu trong cơ sở dữ liệu logic luôn luôn nhất quán. Các kết quả của giao dịch trên đường đi phải không để cho toàn bộ các giao dịch khác ở tất cả các nút quan sát được; điều này chính xác đối với dịch vụ các giao dịch có điều hoạt bất kỳ, kể cả các truy vấn, cập nhật hoặc các lệnh gọi thủ tục từ xa.

Các cơ chế chung về điều khiển giao dịch trong cơ sở dữ liệu không được phân phối được đề cập ở "Quản Lý Giao Dịch". Trong cơ sở dữ liệu được phân phối, Oracle phải phối hợp điều khiển giao dịch với các đặc tính tương tự trên một mạng và duy trì tính nhất quán dữ liệu, dù sự cố mạng hoặc hệ thống xảy ra.

Cơ chế cam kết hai pha của Oracle bảo đảm tất cả các server cơ sở dữ liệu tham gia vào giao dịch được phân phối đều cam kết hoặc đều hồi phục về trạng thái cũ các câu lệnh trong giao dịch. Cơ chế cam kết hai pha còn bảo vệ các điều hoạt DML ngầm định được thực hiện bằng các hạn định tính nguyên vẹn dữ liệu, các lệnh gọi thủ tục từ xa và các bộ kích khởi.

Thông tin bổ sung:
Các hệ thống cơ sở dữ liệu được phân phối của Oracle có thêm nhiều thông tin về cơ chế cam kết hai pha của Oracle.

Tính thông suốt trong hệ thống cơ sở dữ liệu được phân phối

Với một ít nỗ lực, bạn có thể tạo được tính năng về hệ thống cơ sở dữ liệu được phân phối của Oracle thông suốt tới các người dùng hoạt động với hệ thống ấy. Mục đích của tính thông suốt là tạo hệ thống cơ sở dữ liệu được phân phối xuất hiện như nó là một cơ sở dữ liệu Oracle đơn giản. Do đó, hệ thống không gây phức tạp cho các Developer và người dùng hệ thống ấy với những phức tạp, trái lại sẽ làm cho việc phát triển ứng dụng cơ sở dữ liệu được phân phối gặp trở ngại và làm mất năng suất của người dùng.

Các phần sau giải thích về tính thông suốt trong hệ thống cơ sở dữ liệu được phân phối.

Tính thông suốt vị trí

Một cơ sở dữ liệu được Oracle phân phối có các đặc trưng cho phép các Developer ứng dụng và các người điều hành ẩn vị trí vật lý của các đối tượng cơ sở dữ liệu khỏi các ứng dụng và các người dùng. Tính thông suốt vị trí tồn tại khi một người dùng có thể truy xuất toàn bộ đối tượng cơ sở dữ liệu, chẳng hạn như, một bảng, bất kể nút mà ứng dụng kết nối. Tính thông suốt vị trí có nhiều ưu điểm, gồm:
Truy xuất dữ liệu từ xa đơn giản, vì những người dùng cơ sở dữ liệu không cần biết vị trí vật lý của các cơ sở dữ liệu.
Các người điều hành có thể di chuyển các đối tượng cơ sở dữ liệu mà không ảnh hưởng đến các người dùng cuối hoặc các ứng dụng cơ sở dữ liệu có sẵn.

Tiêu biểu nhất là các người điều hành và những Developer sử dụng các từ đồng nghĩa để thiết lập tính thông suốt vị trí đối với các bảng và các đối tượng hỗ trợ trong một giản đồ ứng dụng. Thí dụ, các câu lệnh sau tạo các từ đồng nghĩa trong một cơ sở dữ liệu cho các bảng trong một bảng khác, cơ sở dữ liệu từ xa.

```
CREATE PUBLIC SYNONYM emp
  FOR scott.emp@sales.us.americas.acme_auto.com
CREATE PUBLIC SYNONYM dept
  FOR scott.dept@sales.us.americas.acme_auto.com
```

Bây giờ, thay vì truy xuất các bảng từ xa với một truy vấn, chẳng hạn như:

```
SELECT ename, dname
  FROM scott.emp@sales.us.americas.acme_auto.com e,
       scott.dept@sales.us.americas.acme_auto.com d
  WHERE e.deptno = d.deptno;
```

Một ứng dụng có thể phát hành một truy vấn đơn giản hơn nhiều mà không phải tính toán đến vị trí của các bảng từ xa.

```
SELECT ename, dname
  FROM emp e, dept d
  WHERE e.deptno = d.deptno;
```

Hơn nữa với các từ đồng nghĩa, các Developer còn có thể sử dụng các khung nhìn và các thủ tục định sẵn để thiết lập tính thông suốt vị trí hoạt động trong hệ thống cơ sở dữ liệu được phân phối.

Tính thông suốt giao dịch và câu lệnh

Kiến trúc cơ sở dữ liệu được phân phối của Oracle còn cung cấp tính thông suốt truy vấn, cập nhật và giao dịch. Thí dụ, các lệnh SQL tiêu chuẩn, chẳng hạn như SELECT, INSERT, UPDATE và DELETE hoạt động giống như chúng thực hiện trong môi trường cơ sở dữ liệu được phân phối. Hơn nữa, các giao dịch điều khiển ứng dụng sử dụng các lệnh SQL tiêu chuẩn COMMIT, SAVEPOINT và ROLLBACK - không đòi hỏi lập trình phức tạp hoặc các hoạt động đặc biệt khác để cung cấp điều khiển giao dịch được phân phối.

Các câu lệnh trong một giao dịch đơn có thể tham chiếu bất kỳ bao nhiêu bảng cục bộ hoặc từ xa.
Oracle bảo đảm tất cả các nút liên quan trong giao dịch được phân phối thực hiện giống nhau: chúng đều cam kết hoặc hồi phục giao dịch ấy về trạng thái cũ.
Nếu sự cố mạng hoặc hệ thống xảy ra khi cam kết một giao dịch được phân phối, giao dịch ấy tự động giải quyết toàn bộ thông suốt; nghĩa là khi mạng hoặc hệ thống được phục hồi, các nút đều cam kết hoặc đều phục hồi giao dịch ấy về trạng thái cũ.

Các điều hoạt bên trong

Mỗi giao diện được cam kết có một số hiệu thay đổi hệ thống (SCN) để nhận dạng duy nhất các thay đổi xảy ra do các câu lệnh bên trong giao dịch đó. Trong một cơ sở dữ liệu được phân phối, các SCN của các nút truyền thông được phối hợp khi:
Một kết nối được thiết lập bằng cách sử dụng đường dẫn được mô tả bằng một hoặc nhiều liên kết cơ sở dữ liệu.
Một câu lệnh SQL được phân phối được tiến hành.
Một giao dịch được phân phối được cam kết.

Trong các ưu điểm khác, sự phối hợp của các SCN giữa các nút của hệ thống cơ sở dữ liệu được phân phối cho phép đọc nhất quán được phân phối toàn cục ở mức câu lệnh và mức giao dịch. Nếu, phục hồi dựa trên thời gian phân phối toàn cục còn có thể được thực hiện.

Tính thông suốt sao chép

Oracle còn cung cấp nhiều đặc trưng để sao chép dữ liệu thông suốt giữa các nút của hệ thống.

Thông tin bổ sung:
Xem Oracle8 Replication về những đặc trưng sao chép của Oracle.

Điều hành một hệ thống cơ sở dữ liệu được phân phối của Oracle

Cũng như có các vấn đề riêng biệt phải xét khi phát triển các ứng dụng cho Hệ thống cơ sở dữ liệu được phân phối của Oracle, nên cũng có các vấn đề đặc biệt cần hiểu về điều hành cơ sở dữ liệu được phân phối. Các phần sau giải thích một số chủ đề đặc biệt về quản lý những cơ sở dữ liệu trong Hệ thống cơ sở dữ liệu được phân phối của Oracle.

Vùng tự trị

Site autonomy (vùng tự trị) nghĩa là mỗi server tham gia trong cơ sở dữ liệu phân phối được điều hành độc lập với tất cả các cơ sở dữ liệu khác, dù mỗi cơ sở dữ liệu điều hoạt bằng một cơ sở dữ liệu không được phân phối. Tuy nhiên, nhiều cơ sở dữ liệu có thể hoạt động cùng với nhau, mỗi cơ sở dữ liệu là một kho lưu trữ dữ liệu riêng biệt mà bạn quản lý riêng biệt. Một số ưu điểm về vùng tự trị trong cơ sở dữ liệu được Oracle phân phối gồm:

Các nút của hệ thống có thể phản ảnh tổ chức logic của các công ty hoặc các tổ chức hợp tác cần phải duy trì quan hệ "hợp tác lâu dài".

Các người điều hành cơ sở dữ liệu cục bộ điều khiển dữ liệu cục bộ tương ứng. Do đó, mỗi địa chỉ vùng chủ của người điều hành cơ sở dữ liệu chịu trách nhiệm ít hơn và quản lý được nhiều hơn.

Các sự cố xảy ra độc lập ít có thể ảnh hưởng đến các nút khác của cơ sở dữ liệu được phân phối. Cơ sở dữ liệu Oracle toàn cục có thể sử dụng từng phần bao lâu cơ sở dữ liệu và mạng khả dụng; không có sự cố cơ sở dữ liệu đơn có thể tạm ngưng tất cả các hoạt động toàn cục hoặc gây trở ngại về hiệu suất (bottleneck).

Các người điều hành có thể phục hồi từ các sự cố hệ thống biệt lập độc lập với những nút khác trong hệ thống.

Từ điển dữ liệu tồn tại cho mỗi cơ sở dữ liệu cục bộ - catalog toàn cục không cần truy xuất dữ liệu cục bộ.
Các nút có thể nâng cấp phần mềm độc lập.

Dù Oracle cho phép bạn quản lý mỗi cơ sở dữ liệu trong hệ thống cơ sở dữ liệu được phân phối độc lập, không có nghĩa là bạn sẽ bỏ qua các đòi hỏi toàn cục của hệ thống. Thí dụ, các tài khoản người dùng bổ sung có thể cần thiết trong mỗi cơ sở dữ liệu để hỗ trợ những liên kết bạn tạo thuận tiện cho các kết nối liên server. Các phần sau giải thích thêm về những chủ đề cá biệt này và trình bày nhu cầu dịch vụ phối cảnh toàn cục của toàn bộ môi trường cơ sở dữ liệu được phân phối khi quản lý các nút cá biệt trong hệ thống.

Bảo mật cơ sở dữ liệu được phân phối

Oracle hỗ trợ tất cả những đặc trưng bảo mật có thể sử dụng với môi trường cơ sở dữ liệu không được phân phối cho các hệ thống cơ sở dữ liệu được phân phối, gồm:

- mật khẩu hoặc xác minh dịch vụ bên ngoài cho cửa sổ người dùng và các nút
- mật mã hóa gói tin khởi nhập cho các kết nối client-với-server và server-với-server

Các phần sau giải thích về một số chủ đề bổ sung để xét khi cấu hình một Hệ thống cơ sở dữ liệu được phân phối của Oracle.

Hỗ trợ các tài khoản người dùng và các vai trò

Trong hệ thống cơ sở dữ liệu được phân phối, bạn phải đặt kế hoạch kỹ lưỡng các tài khoản người dùng và các vai trò cần thiết để hỗ trợ các ứng dụng sử dụng hệ thống ấy.
Các tài khoản người dùng cần thiết để thiết lập các kết nối server-với-server phải có thể sử dụng trong tất cả các cơ sở dữ liệu của hệ thống cơ sở dữ liệu được phân phối.

Các vai trò cần thiết để thiết lập các đặc quyền ứng dụng đối với các người dùng ứng dụng cơ sở dữ liệu được phân phối phải được trình bày trong tất cả những cơ sở dữ liệu của hệ thống cơ sở dữ liệu được phân phối.

Giống như bạn tạo các liên kết cơ sở dữ liệu cho các nút trong hệ thống cơ sở dữ liệu được phân phối, xác định xem những tài khoản và các vai trò nào mà mỗi vùng địa chỉ cần hỗ trợ các kết nối server-với-server sử dụng các liên kết ấy.

Thông tin bổ sung:
Xem Các cơ sở dữ liệu được phân phối của Oracle8 về các tài khoản người dùng phải có thể sử dụng đối với các kiểu hỗ trợ các liên kết cơ sở dữ liệu khác nhau trong hệ thống.

Các người dùng toàn cục và các vai trò

Trong một môi trường được phân phối, các người dùng thường cần truy xuất nhiều dịch vụ mạng. Khi cần cấu hình các xác minh riêng biệt cho mỗi người dùng truy xuất từng dịch vụ mạng, điều hành bổ sung có thể trở thành khó điều khiển, nhất là đối với các hệ thống lớn. Sử dụng dịch vụ xác minh toàn cục là kỹ thuật thông thường để đơn giản hóa quản lý bảo mật cho các môi trường được phân phối.

Trong một Oracle client/server hoặc môi trường cơ sở dữ liệu được phân phối, bạn có hai tùy chọn để hỗ trợ xác minh toàn cục cho các người dùng và các vai trò:
Oracle Security Server là một sản phẩm hỗ trợ xác minh trung tâm và xác minh phân phối trên mạng Oracle. Oracle Security Server là tùy chọn tiêu chuẩn của Oracle.
Khi xác minh người dùng cơ sở dữ liệu toàn cục và vai trò phải hoạt động trong một khuôn khổ của dịch vụ xác minh khác Oracle (thí dụ, DCE), một môi trường cơ sở dữ liệu được Oracle phân phối sử dụng Advanced Networking Option của Net8. Net8

Advanced Networking Option là một sản phẩm tùy chọn gói chung một số đặc trưng ban có thể sử dụng để tăng cường Net8 và tính bảo mật của hệ thống cơ sở dữ liệu Oracle phân phối.

Lưu ý:
Advanced Networking Option chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition.

Mã hóa dữ liệu

Net8 Advanced Networking Option còn cho phép Net8 và các sản phẩm liên quan sử dụng mã hóa dữ liệu mạng và kiểm chuẩn để dữ liệu không đọc được hoặc bị thay đổi. Nó bảo vệ dữ liệu tránh được bị xem trái phép bằng cách sử dụng RSA Data Security RC4 hoặc thuật toán mã hóa Data Encryption Standard (DES). Để bảo đảm dữ liệu không bị sửa đổi, xóa hoặc sử dụng lại khi truyền, các dịch vụ bảo mật của Advanced Networking Option có thể phát sinh sắp xếp thông điệp bảo mật bằng mật mã và sử dụng nó trong mỗi gói tin được gửi trên mạng.

Thông tin bổ sung:
Xem Hướng dẫn của Oracle8 Net8 Administrator về những đặc trưng này và các đặc trưng có thể sử dụng với Oracle8 Enterprise Edition.

Các công cụ điều hành các cơ sở dữ liệu được Oracle phân phối

Người điều hành cơ sở dữ liệu có nhiều chọn lựa về các công cụ để sử dụng khi quản lý một Hệ thống cơ sở dữ liệu được phân phối của Oracle:
Oracle Enterprise Manager
Các công cụ điều hành bên thứ ba
Hỗ trợ SNMP

Oracle Enterprise Manager

Oracle Enterprise Manager là công cụ điều hành cơ sở dữ liệu của Oracle. Thành phần đồ họa của Oracle Enterprise Manager cho phép bạn thực hiện các tác vụ điều hành cơ sở dữ liệu thuận tiện bằng giao diện người dùng đồ họa (GUI). Thành phần chế độ tuyến của Oracle Enterprise Manager cung cấp giao diện chế độ tuyến.

Oracle Enterprise Manager cung cấp tính năng điều hành bằng một giao diện dễ dùng. Bạn có thể sử dụng Oracle Enterprise Manager để:

Thực hiện các tác vụ điều hành truyền thống, chẳng hạn như khởi động cơ sở dữ liệu, đóng, sao lưu và phục hồi. Thay vì nhập các lệnh SQL thủ công để thực hiện những tác vụ này, bạn có thể sử dụng giao diện đồ họa của Oracle Enterprise Manager để tiến hành các lệnh ấy nhanh chóng và thuận tiện bằng cách trỏ và nhắp chuột.

Thực hiện đồng thời nhiều tác vụ. Vì bạn có thể mở nhiều cửa sổ đồng thời trong Oracle Enterprise Manager, bạn có thể thực hiện nhiều tác vụ điều hành hoặc không điều hành đồng thời.

Điều hành nhiều cơ sở dữ liệu. Bạn có thể sử dụng Oracle Enterprise Manager để điều hành một cơ sở dữ liệu đơn hoặc nhiều cơ sở dữ liệu đồng thời.

Tập trung các tác vụ điều hành cơ sở dữ liệu. Bạn có thể điều hành ca các cơ sở dữ liệu cục bộ và từ xa bằng cách tiến hành các nền Oracle bất kỳ ở vị trí khắp thế giới. Hơn nữa, những nền Oracle này có thể được kết nối bằng (các) giao thức mạng bất kỳ được Net8 hỗ trợ.

Tiến hành các lệnh SQL, PL/SQL và Oracle Enterprise Manager linh hoạt. Bạn có thể sử dụng Oracle Enterprise Manager để nhập, sửa và tiến hành các câu lệnh. Oracle Enterprise Manager còn duy trì lịch sử các câu lệnh được tiến hành. Do đó, bạn có thể tiến hành lại các câu lệnh mà không phải gõ lại chúng, một đặc trưng đặc biệt hữu dụng nếu bạn cần phải tiến hành những

câu lệnh dài lặp đi lặp lại trong một hệ thống cơ sở dữ liệu được phân phối.
Thực hiện những tác vụ điều hành bằng cách sử dụng giao diện chế độ hàng của Oracle Enterprise Manager khi không thể hoặc không muốn sử dụng giao diện đồ họa người dùng.

Các công cụ điều hành bên thứ ba

Hiện nay có hơn 60 công ty sản xuất hơn 150 sản phẩm giúp quản lý các cơ sở dữ liệu Oracle và các mạng để cung cấp môi trường mở thực sự.

Hỗ trợ SNMP

Ngoài những khả năng điều hành mạng, hỗ trợ Oracle Simple Network Management Protocol (SNMP) cho phép Oracle server được định vị và truy vấn bằng hệ thống quản lý dựa trên SNMP bất kỳ. SNMP là tiêu chuẩn được chấp nhận nằm dưới nhiều hệ thống quản lý mạng phổ biến, chẳng hạn như:
HP's OpenView
Digital's POLYCENTER Manager on NetView
IBM's NetView/6000
Novell's NetWare Management System
SunSoft's SunNet Manager

Thông tin bổ sung:
Xem Hướng dẫn của Oracle SNMP Support Reference Guide.

Hỗ trợ ngôn ngữ quốc gia

Oracle hỗ trợ các môi trường client/server nơi các client và các server sử dụng các bộ ký tự khác nhau. Bộ ký tự được client sử dụng được xác định bằng giá trị của tham số NLS_LANG cho phiên client ấy. Bộ ký tự được server sử dụng là bộ ký tự cơ sở

dữ liệu của nó. Chuyển đổi dữ liệu được thực hiện tự động giữa các bộ ký tự này nếu chúng khác nhau.

Thông tin bổ sung:
Xem Oracle8 Reference về các đặc trưng National Language Support.

Chương 31

Sao chép cơ sở dữ liệu

Chương này giải thích các khái niệm căn bản và các thuật ngữ liên quan đến những đặc trưng sao chép của Oracle.

- **Sao chép là gì?**
- **Các khái niệm sao chép căn bản**
- **Các khái niệm sao chép cao cấp**

Lưu ý:
Các đặc trưng Advanced Replication mô tả ở chương này chỉ có thể sử dụng nếu bạn đã có Oracle8 Enterprise Edition.

Thông tin bổ sung:
Oracle Replication có những thông tin chi tiết về sao chép cơ sở dữ liệu.

Sao chép là gì?

Sao chép (Replication) là tiến hành sao lại và duy trì những đối tượng cơ sở dữ liệu trong nhiều cơ sở dữ liệu để thiết lập hệ thống cơ sở dữ liệu được phân phối. Sao chép có thể làm tăng hiệu suất và bảo vệ tính khả dụng của các ứng dụng vì có sẵn các tùy chọn thay thế truy xuất dữ liệu. Thí dụ, một ứng dụng thường

có thể truy xuất một cơ sở dữ liệu cục bộ thay vì một server từ xa có thể giảm tối đa lưu thông mạng và đạt được hiệu suất tối đa. Hơn nữa, ứng dụng ấy có thể tiếp tục thực hiện chức năng nếu server cục bộ gặp phải sự cố, nhưng các server khác có dữ liệu được sao chép vẫn có thể truy xuất.

Oracle hỗ trợ hai dạng sao chép khác nhau: sao chép căn bản và sao chép cao cấp.

Sao chép căn bản

Với sao chép căn bản, các bản sao chép dữ liệu cung cấp truy xuất chỉ đọc bảng dữ liệu từ một vùng chính (master). Các ứng dụng có thể truy vấn dữ liệu từ các bản sao chép dữ liệu cục bộ để tránh truy xuất mạng bất kể tính khả dụng của mạng. Tuy nhiên, các ứng dụng trên khắp hệ thống phải truy xuất dữ liệu ở vùng chính khi cần phải cập nhật.

Hình 31-1 minh họa sao chép căn bản.

Oracle có thể hỗ trợ những môi trường sao chép chỉ đọc, căn bản bằng cách sử dụng các ảnh chụp nhanh bảng chỉ đọc. Để tìm hiểu thêm về sao chép căn bản và các ảnh chụp nhanh chỉ đọc, xem "Các Khái Niệm Sao Chép Căn Bản".

Sao chép cao cấp (đối xứng)

Các đặc trưng sao chép cao cấp của Oracle mở rộng những khả năng sao chép chỉ đọc căn bản bằng cách cho phép các ứng dụng cập nhật các bản sao của bảng ở khắp hệ thống cơ sở dữ liệu được sao chép. Với sao chép cao cấp, các bản sao chép dữ liệu bất kỳ ở nơi nào trong hệ thống có thể cung cấp truy xuất đọc và cập nhật dữ liệu của bảng. Các server CSDL Oracle tham gia tự động hoạt động để quy tụ dữ liệu của tất cả các bản sao chép bảng và bảo đảm tính nhất quán của giao dịch toàn cục và tính nguyên vẹn dữ liệu.

Hình 31-2 minh họa sao chép cao cấp.

Oracle có thể hỗ trợ các đòi hỏi của những môi trường sao chép cao cấp bằng cách sử dụng nhiều cấu hình. Để tìm hiểu thêm về các hệ thống sao chép cao cấp, xem "Các Khái Niệm Sao Chép Cao Cấp".

Các khái niệm sao chép căn bản

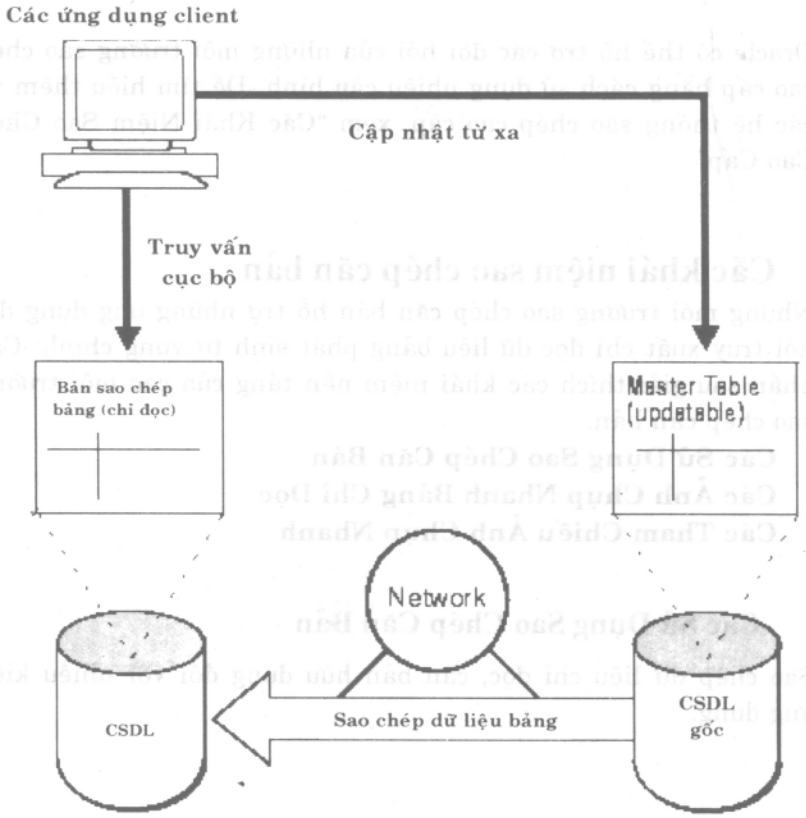
Những môi trường sao chép căn bản hỗ trợ những ứng dụng đòi hỏi truy xuất chỉ đọc dữ liệu bảng phát sinh từ vùng chính. Các phần sau giải thích các khái niệm nền tảng của các môi trường sao chép căn bản.

Các Sử Dụng Sao Chép Căn Bản
Các Ảnh Chụp Nhanh Bảng Chỉ Đọc
Các Tham Chiếu Ảnh Chụp Nhanh

Các Sử Dụng Sao Chép Căn Bản

Sao chép dữ liệu chỉ đọc, căn bản hữu dụng đối với nhiều kiểu ứng dụng.

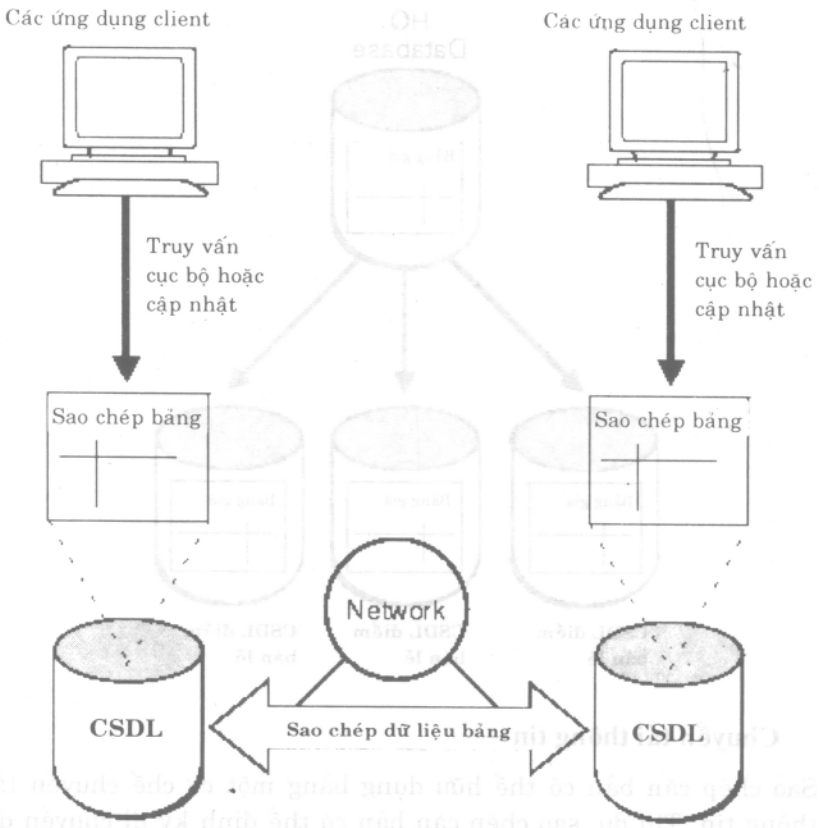
Hình 31-1 Sao chép chỉ đọc, căn bản



Phân phối thông tin

Sao chép căn bản hữu dụng để phân phối thông tin. Thí dụ, xét hoạt động của một dây chuyền cửa hàng tiêu thụ lớn. Với kiểu doanh nghiệp này, tối quan trọng phải bảo đảm thông tin giá cả sản phẩm luôn luôn có thể sử dụng, tương đối hiện hành và nhất quán ở tất cả các điểm bán lẻ. Để đạt được các mục đích này, mỗi cửa hàng bán lẻ có thể có dữ liệu giá sản phẩm riêng mà nó tô bổ vào ban đêm từ bảng giá chính ở các cửa hàng tổn hợp.

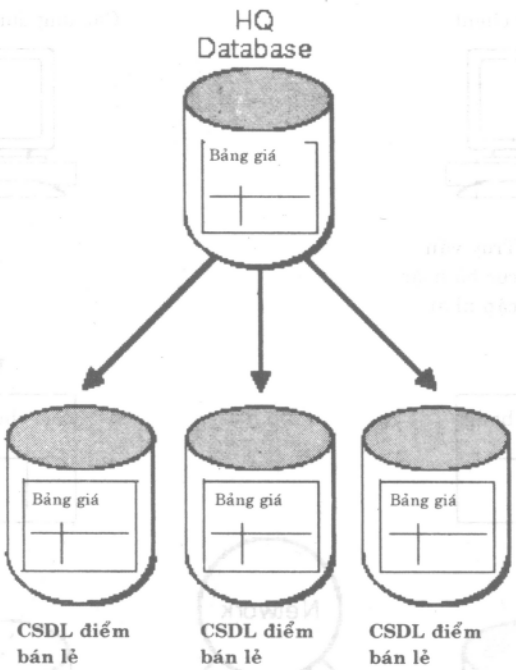
Hình 31-2 Sao chép cao cấp



Nạp rời thông tin

Sao chép căn bản là cách hữu dụng để sao chép toàn bộ những cơ sở dữ liệu hoặc thông tin nạp rời. Thí dụ, khi hiệu suất của hệ thống xử lý giao dịch dung lượng cao tới mức giới hạn, thì nó có lợi điểm duy trì một cơ sở dữ liệu sao chép để biệt lập các truy vấn đòi hỏi của các ứng dụng hỗ trợ quyết định.

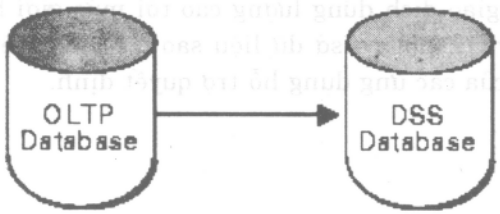
Hình 31-3 Phân phối thông tin



Chuyển tải thông tin

Sao chép căn bản có thể hữu dụng bằng một cơ chế chuyển tải thông tin. Thí dụ, sao chép căn bản có thể định kỳ di chuyển dữ liệu từ giao dịch sản xuất xử lý cơ sở dữ liệu tới một kho dữ liệu.

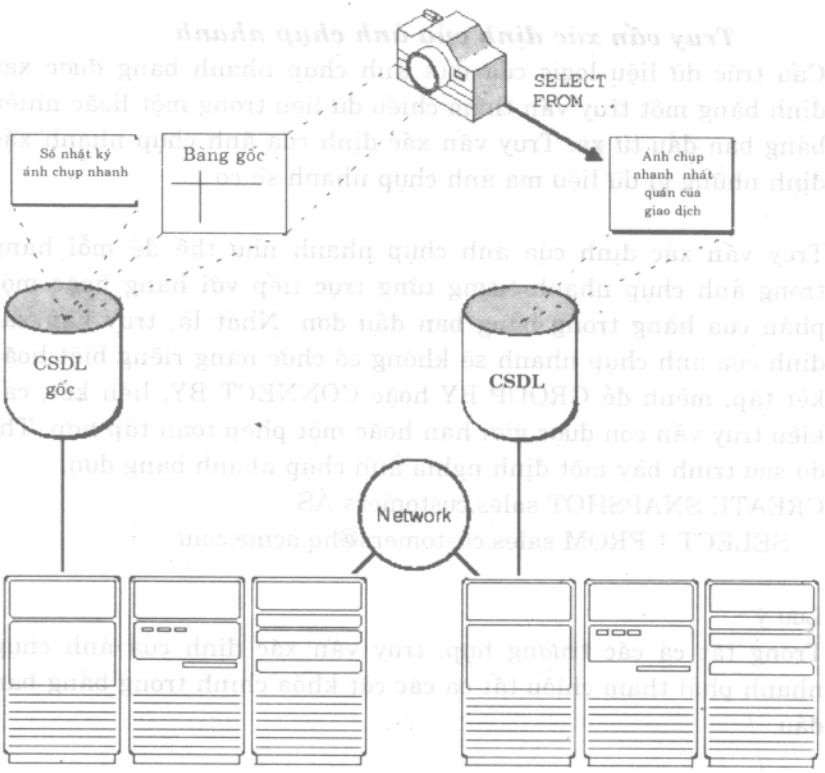
Hình 31-4 Thông tin Off-Loading



Các ảnh chụp nhanh bảng chỉ đọc

Ảnh chụp nhanh bảng chỉ đọc là bản sao cục bộ của dữ liệu bảng phát sinh từ một hoặc nhiều bảng ban đầu từ xa. Ứng dụng có thể truy vấn dữ liệu trong ảnh chụp bảng chỉ đọc, nhưng không thể chèn, cập nhật hoặc xóa các hàng trọng ảnh chụp nhanh. Hình 31-5 và các phần sau giải thích thêm về các ảnh chụp nhanh bảng chỉ đọc và sao chép căn bản.

Hình 31-5 Các ảnh chụp nhanh chỉ đọc, các bảng ban đầu và các sổ nhật ký ảnh chụp nhanh



Kiến trúc ảnh chụp nhanh chỉ đọc

Oracle hỗ trợ sao chép dữ liệu căn bản với cơ chế ảnh chụp nhanh bảng của nó. Các phần sau giải thích kiến trúc của các ảnh chụp nhanh bảng chỉ đọc đơn.

Lưu ý:
Oracle cung ứng các đặc trưng sao chép căn bản, chẳng hạn như các ảnh chụp nhanh phức tạp và các ảnh chụp nhanh ROWID cho các đòi hỏi sao chép duy nhất. Để tìm hiểu thêm về các cấu hình đặc biệt này, xem "Các Tùy Chọn Sao Chép Căn Bản Khác".

Truy vấn xác định của ảnh chụp nhanh

Cấu trúc dữ liệu logic của một ảnh chụp nhanh bảng được xác định bằng một truy vấn tham chiếu dữ liệu trong một hoặc nhiều bảng ban đầu từ xa. Truy vấn xác định của ảnh chụp nhanh xác định những gì dữ liệu mà ảnh chụp nhanh sẽ có.

Truy vấn xác định của ảnh chụp nhanh như thế để mỗi hàng trong ảnh chụp nhanh tương ứng trực tiếp với hàng hoặc một phần của hàng trong bảng ban đầu đơn. Nhất là, truy vấn xác định của ảnh chụp nhanh sẽ không có chức năng riêng biệt hoặc kết tập, mệnh đề GROUP BY hoặc CONNECT BY, liên kết, các kiểu truy vấn con được giới hạn hoặc một phép toán tập hợp. Thí dụ sau trình bày một định nghĩa ảnh chụp nhanh bảng đơn.

```
CREATE SNAPSHOT sales.customers AS
  SELECT * FROM sales.customers@hq.acme.com
```

Lưu ý :
Trong tất cả các trường hợp, truy vấn xác định của ảnh chụp nhanh phải tham chiếu tất cả các cột khóa chính trong bảng ban đầu.

Truy xuất xác định của ảnh chụp nhanh có thể có các kiểu giới hạn của những truy vấn con tham chiếu nhiều bảng để lọc các

hàng từ bảng ban đầu của ảnh chụp nhanh. Ảnh chụp nhanh truy vấn con có thể được sử dụng để tạo các ảnh chụp nhanh "đi lên" các tham chiếu nhiều so với một từ các bảng con tới cha có thể liên quan đến nhiều mức cấp. Thí dụ sau tạo một ảnh chụp nhanh truy vấn con.

```
CREATE SNAPSHOT sales.orders AS
   SELECT * FROM sales.orders@hq.acme.com o
      WHERE EXISTS
      ( SELECT c_id FROM sales.customers@hq.acme.com c
          WHERE o.c_id = c.c_id AND zip = 19555);
```

Các tô bổ ảnh chụp nhanh

Dữ liệu mà một ảnh chụp nhanh trình bày không nhất thiết phải so khớp với dữ liệu hiện hành của các bảng ban đầu của nó. Ảnh chụp nhanh bảng là một phản ảnh giao dịch nhất quán của dữ liệu ban đầu của nó như dữ liệu đã tồn tại ở một thời điểm cụ thể. Để duy trì dữ liệu của ảnh chụp nhanh tương đối hiện hành với dữ liệu của bảng ban đầu của nó, Oracle phải định kỳ tô bổ ảnh chụp nhanh ấy. Tô bổ ảnh chụp nhanh là điều hoạt lô có hiệu lực tạo ảnh chụp nhanh đó phản ảnh trạng thái hiện hành của bảng ban đầu của nó.

Bạn phải quyết định như thế nào và khi nào thích hợp để tô bổ mỗi ảnh chụp nhanh để tạo cho nó biểu diễn dữ liệu ban đầu của nó hiện hành hơn. Thí dụ, các ảnh chụp nhanh phát xuất từ các bảng ban đầu mà các ứng dụng thường cập nhật, chúng đòi hỏi phải tô bổ thường xuyên. Trái lại, các ảnh chụp nhanh phụ thuộc vào các bảng tương đối tĩnh thường đòi hỏi các tô bổ thường xuyên. Nói tóm lại, phân tích những đặc tính ứng dụng và các đòi hỏi để giúp xác định các thời khoảng tô bổ ảnh chụp nhanh thích hợp.

Tô bổ các ảnh chụp nhanh, Oracle hỗ trợ các kiểu tô bổ khác nhau (hoàn toàn và nhanh), các nhóm tô bổ ảnh chụp nhanh và các tô bổ tự động và thủ công.

Các tô bổ hoàn toàn và nhanh

Oracle có thể tô bổ một ảnh chụp nhanh cá biệt bằng cách sử dụng tô bổ hoàn toàn hoặc tô bổ nhanh.

Các tô bổ hoàn toàn

Để thực hiện một tô bổ hoàn toàn, server quản lý ảnh chụp nhanh để tiến hành truy vấn xác định của ảnh chụp nhanh. Tập hợp kết quả của truy vấn thay thế dữ liệu ảnh chụp nhanh có sẵn để tô bổ ảnh chụp nhanh ấy. Oracle có thể thực hiện tô bổ hoàn toàn đối với bất kỳ ảnh chụp nhanh.

Các tô bổ nhanh

Để thực hiện tô bổ nhanh, server quản lý ảnh chụp nhanh đó trước hết để nhận dạng các thay đổi xảy ra ở bản ban đầu vì tô bổ gần nhất của ảnh chụp nhanh và sau đó áp dụng chúng vào ảnh chụp nhanh ấy. Các tô bổ nhanh hiệu quả hơn các tô bổ toàn bộ khi một ít thay đổi đối với bảng ban đầu vì các server tham gia và các mạng ít phải sao chép dữ liệu. Các tô bổ nhanh có thể sử dụng cho các ảnh chụp nhanh chỉ khi bảng ban đầu có sổ nhật ký ảnh chụp nhanh.

Các sổ nhật ký ảnh chụp nhanh

Khi một bảng ban đầu tương ứng với một hoặc nhiều ảnh chụp nhanh, tạo một sổ nhật ký ảnh chụp nhanh cho bảng ấy để các tô bổ nhanh của các ảnh chụp nhanh là một tùy chọn. Sổ nhật ký ảnh chụp nhanh của bảng ban đầu theo dõi dữ liệu tô bổ nhanh cho tất cả sổ nhật ký ảnh chụp nhanh có thể trên mỗi bảng ban đầu. Khi một server thực hiện tô bổ nhanh cho một ảnh chụp nhanh, nó sử dụng dữ liệu trong sổ nhật ký cập nhật

Tô bổ các ảnh chụp nhanh, Oracle hỗ trợ các kiểu tô bổ khác nhau (hoàn toàn và nhanh), các nhóm tô bổ ảnh chụp nhanh và các tô bổ tự động và thủ công.

Các tô bổ hoàn toàn và nhanh

Oracle có thể tô bổ một ảnh chụp nhanh cá biệt bằng cách sử dụng tô bổ hoàn toàn hoặc tô bổ nhanh.

Các tô bổ hoàn toàn

Để thực hiện một tô bổ hoàn toàn, server quản lý ảnh chụp nhanh để tiến hành truy vấn xác định của ảnh chụp nhanh. Tập hợp kết quả của truy vấn thay thế dữ liệu ảnh chụp nhanh có sẵn để tô bổ ảnh chụp nhanh ấy. Oracle có thể thực hiện tô bổ hoàn toàn đối với bất kỳ ảnh chụp nhanh.

Các tô bổ nhanh

Để thực hiện tô bổ nhanh, server quản lý ảnh chụp nhanh đó trước hết để nhận dạng các thay đổi xảy ra ở bản ban đầu vì tô bổ gần nhất của ảnh chụp nhanh và sau đó áp dụng chúng vào ảnh chụp nhanh ấy. Các tô bổ nhanh hiệu quả hơn các tô bổ toàn bộ khi một ít thay đổi đối với bảng ban đầu vì các server tham gia và các mạng ít phải sao chép dữ liệu. Các tô bổ nhanh có thể sử dụng cho các ảnh chụp nhanh chỉ khi bảng ban đầu có sổ nhật ký ảnh chụp nhanh.

Các sổ nhật ký ảnh chụp nhanh

Khi một bảng ban đầu tương ứng với một hoặc nhiều ảnh chụp nhanh, tạo một sổ nhật ký ảnh chụp nhanh cho bảng ấy để các tô bổ nhanh của các ảnh chụp nhanh là một tùy chọn. Sổ nhật ký ảnh chụp nhanh của bảng ban đầu theo dõi dữ liệu tô bổ nhanh cho tất cả sổ nhật ký ảnh chụp nhanh có thể trên mỗi bảng ban đầu. Khi một server thực hiện tô bổ nhanh cho một ảnh chụp nhanh, nó sử dụng dữ liệu trong sổ nhật ký cập nhật

của bảng ban đầu để tô bổ ảnh chụp nhanh ấy một cách hiệu quả. Oracle tự động xóa bỏ dữ liệu tô bổ ấn định từ sổ nhật ký ảnh chụp nhanh sau khi tất cả những ảnh chụp nhanh thực hiện tô bổ sao cho dữ liệu của sổ nhật ký không còn cần thiết.

Các nhóm tô bổ ảnh chụp nhanh

Để duy trì tính nguyên vẹn tham chiếu và tính nhất quán của giao dịch trong các ảnh chụp nhanh bảng của nhiều bảng ban đầu liên quan, Oracle tổ chức và tô bổ mỗi ảnh chụp nhanh bằng một thành phần của nhóm tô bổ. Oracle tô bổ tất cả các ảnh chụp nhanh trong một nhóm tương ứng cùng với thời điểm giao dịch nhất quán.

Tô bổ ảnh chụp nhanh tự động

Khi tạo một nhóm tô bổ ảnh chụp nhanh, các người điều hành thường cấu hình nhóm ấy để Oracle tự động tô bổ các ảnh chụp nhanh của nó. Trái lại, các người điều hành sẽ phải tô bổ thủ công bất kỳ lúc nào cần.

Khi cấu hình một nhóm tô bổ cho các tô bổ tự động, nó cần để

- ấn định thời khoảng tô bổ cho nhóm ấy
- cấu hình server quản lý các ảnh chụp nhanh với một hoặc nhiều xử lý nền SNPn để đánh thức định kỳ và tô bổ bất kỳ các ảnh chụp nhanh đến lúc phải tô bổ

Các thời khoảng tô bổ tự động

Khi bạn tạo một nhóm tô bổ ảnh chụp nhanh, bạn có thể ấn định thời khoảng tô bổ tự động cho nhóm ấy. Khi xác lập thời khoảng của một nhóm, hiểu rõ các xử lý sau:
Ngày tháng hoặc các biểu thức ngày tháng ấn định thời khoảng tô bổ phải thẩm định thời điểm trong tương lai.

Thời khoảng tô bổ phải lớn hơn chiều dài thời gian cần thiết để thực hiện tô bổ.

Các biểu thức ngày tháng liên quan thẩm định thời điểm liên quan đến ngày tháng tô bổ gần nhất. Nếu sự cố mạng hoặc hệ thống xảy ra gây trở ngại tô bổ nhóm có lịch trình định kỳ, sự thẩm định của biểu thức ngày tháng liên quan có thể thay đổi tương ứng.

Các biểu thức ngày tháng rõ ràng thẩm định thời điểm ấn định, bất kể ngày tháng tô bổ gần nhất.

Các kiểu tô bổ

Theo mặc định, Oracle cố thực hiện mỗi ảnh chụp nhanh trong một nhóm tô bổ. Đối với lý do nào đó, nếu Oracle không thể thực hiện tô bổ nhanh cho một ảnh chụp nhanh cá biệt (thí dụ, khi bảng ban đầu không có sổ nhật ký ảnh chụp nhanh). Thì server thực hiện một tô bổ hoàn toàn cho ảnh chụp nhanh ấy.

Các xử lý nền SNPn Background

Phương tiện tô bổ ảnh chụp nhanh tự động của Oracle thực hiện chức năng bằng cách sử dụng các hàng đợi tác vụ để lập lịch trình các thủ tục hệ thống tiến hành định kỳ bên trong. Các hàng đợi tác vụ đòi hỏi ít nhất một xử lý nền SNPn sẽ được tiến hành. Xử lý nền SNPn đánh thức định kỳ, kiểm tra hàng đợi tác vụ và tiến hành bất kỳ tác vụ chưa giải quyết.

Các tô bổ ảnh chụp nhanh thủ công

Các tô bổ ảnh chụp nhanh tự động có lịch trình không thể luôn luôn thỏa đáng. Thí dụ, theo ngay sau lần nạp hàng loạt dữ liệu vào bảng ban đầu, các ảnh chụp nhanh phụ thuộc sẽ không còn biểu thị dữ liệu của bảng ban đầu. Thay vào đó phải chờ các tô bổ nhóm tự động có lịch trình kế tiếp, bạn có thể cần tô bổ các nhóm ảnh chụp nhanh thủ công để truyền ngay các hàng mới của bảng ban đầu tới các ảnh chụp nhanh được kết hợp.

Các tùy chọn sao chép căn bản khác

Oracle hỗ trợ một số đặc trưng sao chép căn bản bổ sung có thể hữu dụng trong những trường hợp nào đó:

Các Ảnh Chụp Nhanh Phức Tạp
Các Ảnh Chụp Nhanh ROWID

Các ảnh chụp nhanh phức tạp

Khi truy vấn xác định của một ảnh chụp nhanh có một chức năng riêng biệt hoặc kết tập, mệnh đề GROUP BY hoặc CONNECT BY, liên kết, các kiểu giới hạn của những truy vấn con hoặc phép toán tập hợp, ảnh chụp nhanh đó là ảnh chụp nhanh phức tạp.

Thí dụ sau là định nghĩa ảnh chụp nhanh bảng phức tạp.

```
CREATE SNAPSHOT scott.emp AS
   SELECT ename, dname
      FROM scott.emp@hq.acme.com a, scott.dept@hq.acme.com b
         WHERE a.deptno = b.deptno
         SORT BY dname
```

Bất lợi chính của ảnh chụp nhanh phức tạp là Oracle không thể thực hiện tô bổ nhanh ảnh chụp nhanh - Oracle chỉ có thể thực hiện các tô bổ hoàn toàn một ảnh chụp nhanh phức tạp. Do đó, sử dụng các ảnh chụp nhanh phức tạp có thể ảnh hưởng đến hiệu suất mạng khi tô bổ ảnh chụp nhanh hoàn toàn.

Các ảnh chụp nhanh ROWID

Các ảnh chụp nhanh khóa chính (được đề cập ở các phần trước của chương này) là mặc định đối với Oracle. Oracle dựa trên ảnh chụp nhanh khóa chính của bảng ban đầu của nó. Vì cấu trúc này, bạn có thể:

- tổ chức lại các bảng ban đầu của ảnh chụp nhanh mà không phải thực hiện tô bổ hoàn toàn ảnh chụp nhanh ấy
- tạo ảnh chụp nhanh với truy vấn xác định kể cả kiểu giới hạn của truy vấn con

Để chỉ tương thích về sau, Oracle còn hỗ trợ các ảnh chụp nhanh ROWID dựa trên các dấu định danh hàng vật lý (ROWIDs) của những hàng trong bảng ban đầu. Các ảnh chụp nhanh ROWID chỉ nên sử dụng cho các ảnh chụp nhanh của các bảng ban đầu trong cơ sở dữ liệu Oracle Release 7.3 và không nên sử dụng khi các ảnh chụp nhanh mới của bảng ban đầu ở cơ sở dữ liệu Oracle8.

Lưu ý:
Để hỗ trợ ảnh chụp nhanh ROWID, Oracle tạo một chỉ mục bổ sung vào bảng cơ sở của ảnh chụp nhanh với tên I_SNAP$_snapshotname.

Các khái niệm sao chép cao cấp

Trong những môi trường sao chép cao cấp, các bản sao dữ liệu bất kỳ nơi nào trong hệ thống có thể cung cấp cả truy xuất đọc và cập nhật dữ liệu của bảng.

Lưu ý:
Những thông tin trong phần này chỉ áp dụng cho đặc trưng sao chép cao cấp của Oracle8 Enterprise Edition. Xem Getting to Know Oracle8 và Oracle8 Enterprise Edition về các đặc trưng có thể sử dụng với Oracle8 Enterprise Edition.

Phần này giải thích các khái niệm của một hệ thống sao chép cao cấp, gồm các chủ đề sau.

Các Sử Dụng Sao Chép Cao Cấp
Các Cấu Hình Sao Chép Cao Cấp

Các Đối Tượng, Các Nhóm, Các Vùng Và Các Catalog Sao Chép
Các Người Điều Hành Sao Chép, Các Bộ Truyền Và Các Bộ Nhận
Các Xung Đột Sao Chép
Các Tùy Chọn Sao Chép Cao Cấp Duy Nhất

Các sử dụng sao chép cao cấp

Sao chép dữ liệu đối xứng, cao cấp hữu dụng cho nhiều loại hệ thống ứng dụng với các đòi hỏi đặc biệt.

Các môi trường không được kết nối

Sao chép cao cấp hữu dụng để khai thác các ứng dụng xử lý giao dịch điều hoạt bằng cách sử dụng các thành phần không được kết nối. Thí dụ, xét hệ thống tự động thúc đẩy kinh doanh đối với một công ty bảo hiểm nhân thọ. Mỗi nhân viên phải tới nơi các khách hàng thường xuyên với một máy tính xách tay và ghi phiếu mua bảo hiểm trong cơ sở dữ liệu cá nhân khi không được kết nối với mạng máy tính phối hợp và hệ thống cơ sở dữ liệu trung tâm. Khi trở về văn phòng, mỗi nhân viên phải gửi tất cả các phiếu bảo hiểm tới cơ sở dữ liệu phối hợp ở trung tâm.

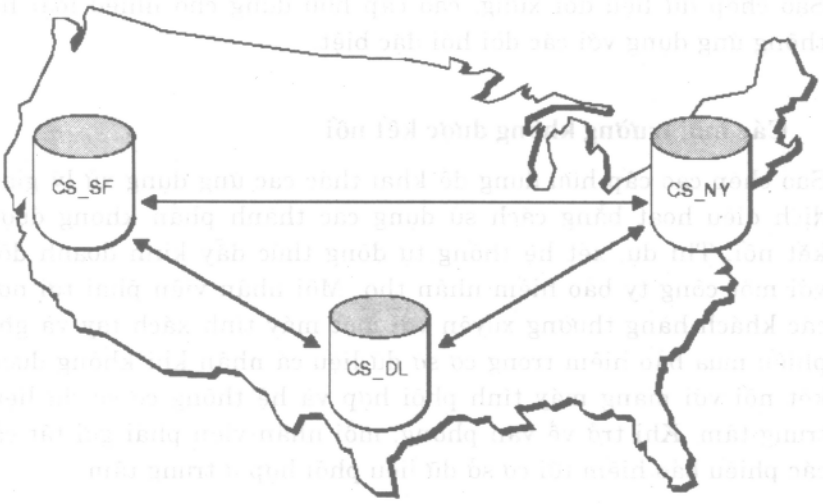
Vùng Failover

Sao chép cao cấp có thể hữu dụng để bảo vệ tính khả dụng của cơ sở dữ liệu nhiệm vụ quan trọng. Thí dụ, hệ thống sao chép đối xứng có thể sao chép toàn bộ cơ sở dữ liệu để thiết lập vùng failover nếu vùng chính trở thành không sử dụng được do sự hạn chế của hệ thống hoặc mạng. Trái lại, với đặc trưng cơ sở dữ liệu dự phòng của Oracle, một vùng failover như thế còn có thể đáp ứng là một cơ sở dữ liệu có chức năng đầy đủ để hỗ trợ truy xuất ứng dụng khi vùng chính hoạt động đồng thời.

Phân phối tải ứng dụng

Sao chép cao cấp hữu dụng đối với các ứng dụng xử lý giao dịch đòi hỏi nhiều điểm truy xuất thông tin cơ sở dữ liệu cho các mục đích phân phối tải ứng dụng nặng, bảo đảm tính khả dụng tiếp tục hoặc cung cấp truy xuất dữ liệu cục bộ hóa hơn.
Các ứng dụng có các đòi hỏi như thế thường có các ứng dụng hướng dịch vụ khách hàng, như được trình bày ở Hình 31-6.

Hình 31-6 Hệ thống sao chép cao cấp với truy xuất đa điểm



Chuyển tải thông tin

Sao chép cao cấp có thể hữu dụng là một cơ chế chuyển tải thông tin. Thí dụ, hệ thống sao chép cao cấp có thể định kỳ nạp rời dữ liệu từ cơ sở dữ liệu điều hoạt mở rộng cập nhật tới kho dữ liệu hoặc trung tâm dữ liệu.

Các cấu hình sao chép cao cấp

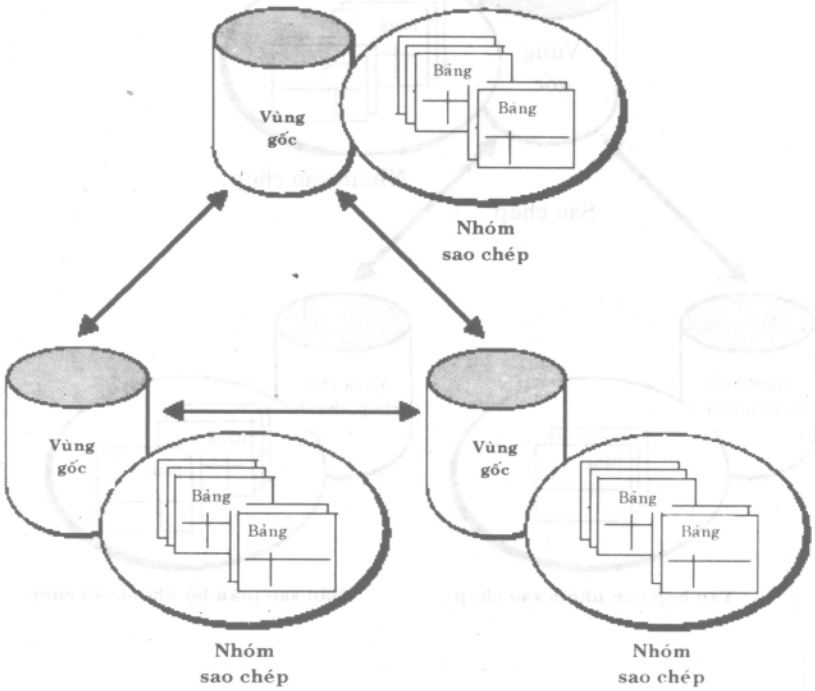
Oracle hỗ trợ các đòi hỏi về các môi trường sao chép cao cấp bằng cách sử dụng sao chép đa bản gốc cũng như các vùng ảnh chụp nhanh.

Sao chép đa bản gốc

Sao chép multimaster (đa bản gốc) của Oracle cho phép nhiều vùng, đóng vai trò đồng đẳng quản lý các nhóm đối tượng cơ sở dữ liệu được sao chép. Các ứng dụng có thể cập nhật bảng sao chép bất kỳ bằng cấu hình đa bản gốc.

Hình 31-7 minh họa hệ thống sao chép đối xứng đa bản gốc.

Hình 31-7 Hệ thống sao chép đa bản gốc

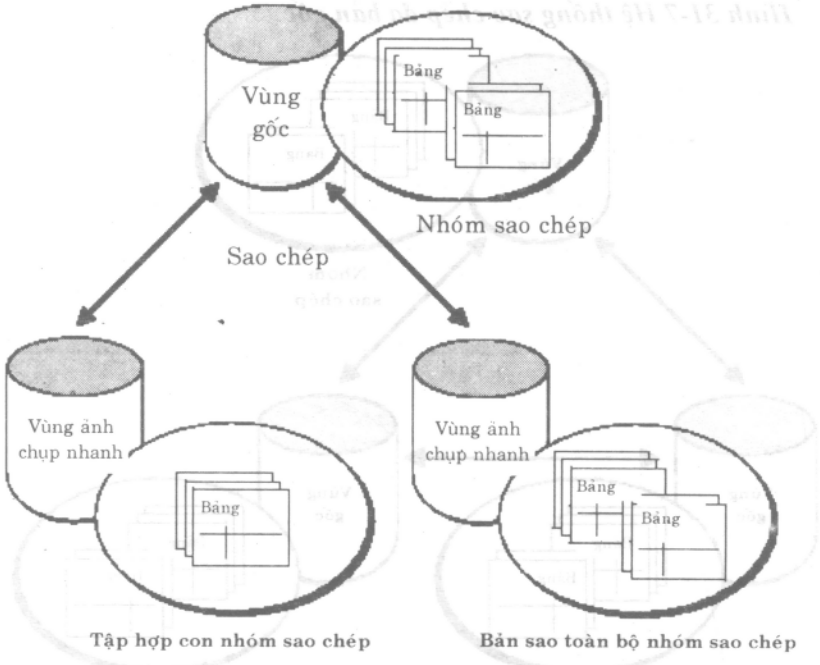


Các vùng ảnh chụp nhanh và các ảnh chụp nhanh cập nhật được

Các vùng ban đầu trong một hệ thống sao chép cao cấp có thể củng cố thông tin mà các ứng dụng cập nhật ở các ảnh chụp nhanh từ xa. Phương tiện sao chép đối xứng của Oracle cho phép các ứng dụng chèn, cập nhật và xóa các hàng bảng qua các ảnh chụp nhanh cập nhật được.

Hình 31-8 minh họa môi trường sao chép cao cấp với các ảnh chụp nhanh cập nhật được.

Hình 31-8 Hệ thống sao chép cao cấp với các ảnh chụp nhanh cập nhật được



Các ảnh chụp nhanh cập nhật được có các tính chất sau:

Các ảnh chụp nhanh cập nhật được luôn luôn là các ảnh chụp nhanh bảng tô bổ nhanh và đơn giản
Oracle truyền các thay đổi xảy ra bằng ảnh chụp nhanh cập nhật được với bảng ban đầu từ xa của ảnh chụp nhanh. Nếu cần, sau đó các cập nhật xếp tầng tới tất cả vùng ban đầu khác.
Oracle tô bổ ảnh chụp nhanh cập nhật được bằng thành phần của một nhóm tô bổ, giống với các ảnh chụp nhanh chỉ đọc.
Các ảnh chụp nhanh cập nhật được có cùng các đối tượng cơ sở. Hơn nữa, Oracle tạo bảng USLOG$_snapshotname để hỗ trợ các ảnh chụp nhanh cập nhật được.

Các cấu hình lai ghép

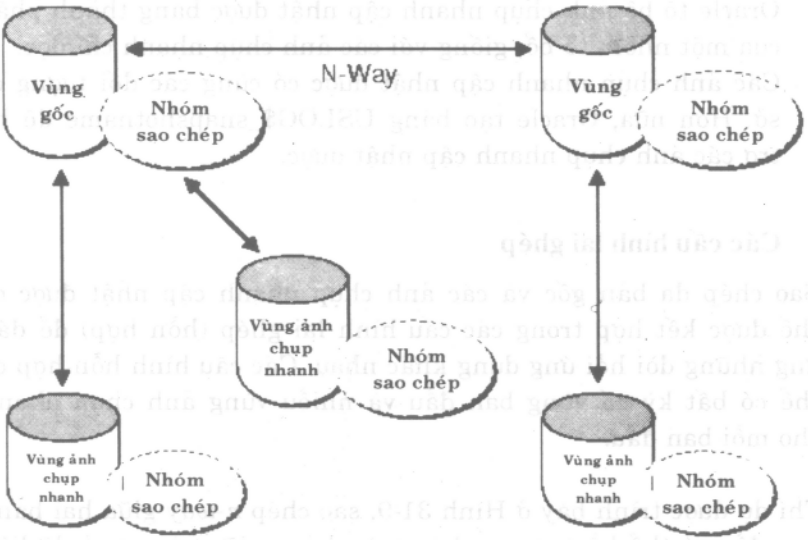
Sao chép đa bản gốc và các ảnh chụp nhanh cập nhật được có thể được kết hợp trong các cấu hình lai ghép (hỗn hợp) để đáp ứng những đòi hỏi ứng dụng khác nhau. Các cấu hình hỗn hợp có thể có bất kỳ số vùng ban đầu và nhiều vùng ảnh chụp nhanh cho mỗi ban đầu.

Thí dụ được trình bày ở Hình 31-9, sao chép *n-way* giữa hai bảng ban đầu có thể hỗ trợ sao chép toàn bảng giữa các cơ sở dữ liệu hỗ trợ hai vùng địa lý. Các ảnh chụp nhanh có thể được xác định trên các bảng ban đầu để sao chép toàn bộ các bảng hoặc các tập hợp con của bảng tới các vùng trong mỗi vùng.

Một số khác biệt chính giữa các ảnh chụp nhanh cập nhật được và các bảng ban đầu được sao chép như sau:
Các bảng ban đầu được sao chép phải có dữ liệu cho toàn bộ bảng được sao chép, trong khi các ảnh chụp nhanh có thể sao chép các tập hợp con của dữ liệu bảng ban đầu.
Sao chép đa bản gốc cho phép bạn sao chép các thay đổi đối với mỗi giao dịch khi các thay đổi xảy ra. Các tô bổ ảnh chụp nhanh

là các thay đổi truyền, được xác lập hướng từ nhiều giao dịch trong một điều hoạt hướng lô có hiệu quả hơn, nhưng thời khoảng thường xuyên ít hơn.
Nếu có bất kỳ xung đột xảy ra do những thay đổi xảy ra đối với nhiều bản sao của cùng dữ liệu, các vùng ban đầu phát hiện và giải quyết các xung đột ấy.

Hình 31-9 Cấu hình lai ghép



Sao chép cao cấp và trình quản lý sao chép Oracle

Các môi trường sao chép cao cấp hỗ trợ mô hình cập nhật dữ liệu bất kỳ nơi nào có thể gây khó khăn cho cấu hình và quản lý. Để giúp điều hành các môi trường sao chép cao cấp, Oracle cung cấp công cụ quản lý tinh vi, Oracle Replication Manager (Trình Quản Lý Sao Chép Oracle).

Thông tin bổ sung:
Oracle8 Replication có những thông tin và các thí dụ về sử dụng Replication Manager.

Các đối tượng, các nhóm, các vùng và các catalog sao chép

Các phần sau giải thích những thành phần của hệ thống sao chép cao cấp, gồm các đối tượng, các nhóm, các vùng và các catalog sao chép.

Các đối tượng sao chép

Đối tượng sao chép là một đối tượng cơ sở dữ liệu tồn tại trên nhiều server trong hệ thống cơ sở dữ liệu được phân phối. Phương tiện sao chép cao cấp của Oracle cho phép bạn sao chép các bảng và các đối tượng hỗ trợ, chẳng hạn như các khung nhìn, các bộ kích khởi cơ sở dữ liệu, các chỉ mục và các từ đồng nghĩa.

Các nhóm sao chép

Trong môi trường sao chép cao cấp, Oracle quản lý các đối tượng sao chép bằng cách sử dụng các nhóm sao chép. Bằng cách tổ chức các đối tượng cơ sở dữ liệu liên quan trong một nhóm sao chép, dễ dàng điều hành nhiều đối tượng với nhau. Bạn thường tạo và sử dụng một nhóm sao chép để tổ chức các đối tượng giản đồ cần thiết để hỗ trợ sao chép cơ sở dữ liệu cá biệt. Điều đó không có nghĩa là các nhóm sao chép và các giản đồ phải tương ứng với nhau - các đối tượng trong nhóm sao chép có thể phát sinh từ nhiều giản đồ cơ sở dữ liệu và giản đồ có thể có các đối tượng là các thành viên của các nhóm sao chép khác nhau. Giới hạn căn bản là đối tượng sao chép có thể là thành viên của chỉ một nhóm.

Các vùng sao chép

Một nhóm sao chép có thể tồn tại ở nhiều vùng sao chép. Các môi trường sao chép cao cấp hỗ trợ hai kiểu căn bản của các vùng: các vùng ban đầu và các vùng ảnh chụp nhanh.

Master site (vùng gốc) duy trì một bản sao toàn bộ tất cả đối tượng trong một nhóm sao chép. Tất cả các vùng gốc trong môi trường sao chép bản gốc, đối xứng truyền thông trực tiếp với một dữ liệu truyền và các thay đổi giản đồ trong nhóm sao chép. Nhóm sao chép ở vùng gốc đặc biệt được tham chiếu bằng *maste group* - một nhóm gốc.

Hơn nữa, mọi nhóm sao chép có một và chỉ một vùng định nghĩa ban đầu. Vùng định nghĩa gốc của nhóm sao chép là vùng gốc đáp ứng điểm điều khiển để quản lý nhóm sao chép ấy và các đối tượng ở nhóm đó.

Vùng ảnh chụp nhanh hỗ trợ các ảnh chụp nhanh cập nhật được và chỉ đọc đơn ở một vùng gốc được kết hợp. Các ảnh chụp nhanh bảng của vùng ảnh chụp nhanh có thể có tất cả hoặc chỉ một tập hợp con của dữ liệu bảng trong nhóm sao chép, nhưng phải là ảnh chụp nhanh đơn có các bảng tương ứng một-một ở vùng gốc ấy. Thí dụ, một vùng ảnh chụp nhanh có thể có các ảnh chụp nhanh chỉ cho các bảng được lựa trong nhóm sao chép và ảnh chụp nhanh cá biệt có thể chỉ là phần được chọn của bảng được sao chép nào đó. Nhóm sao chép ở vùng ảnh chụp nhanh được tham chiếu bằng một nhóm ảnh chụp nhanh nhiều hơn. Nhóm ảnh chụp nhanh còn có thể có các đối tượng sao chép khác.

Catalog sao chép

Mọi vùng ảnh chụp nhanh và ban đầu trong môi trường sao chép cao cấp có catalog sao chép. Một catalog sao chép của vùng là tập hợp riêng biệt của các bảng từ điển dữ liệu và các khung nhìn duy trì thông tin điều hành về những đối tượng sao chép và các nhóm sao chép ở vùng đó. Mỗi server tham gia vào môi trường sao chép cao cấp có thể tự động hóa sao chép các đối tượng trong các nhóm sao chép sử dụng thông tin trong catalog sao chép của nó.

Quản lý sao chép API và các yêu cầu điều hành

Để cấu hình và quản lý môi trường sao chép cao cấp, mỗi server tham gia sử dụng giao diện lập trình ứng dụng sao chép (API). Một API quản lý sao chép của server là tập hợp các phần mềm PL/SQL gói riêng các thủ tục và các hàm mà người điều hành có thể sử dụng để cấu hình các đặc trưng sao chép cao cấp của Oracle. Oracle Replication Manager còn sử dụng các thủ tục và các hàm của mỗi API quản lý sao chép của vùng để thực hiện công việc.

Yêu cầu điều hành là một lệnh gọi thủ tục hoặc hàm trong API quản lý sao chép của Oracle. Thí dụ, khi bạn sử dụng Replication Manager để tạo một nhóm ban đầu mới, Replication Manager hoàn tất tác vụ bằng cách đưa ra một lệnh gọi thủ tục DBMS_REPCAT.CREATE_MASTER_REPGROUP. Một số yêu cầu điều hành phát sinh các lệnh gọi API quản lý sao chép bổ sung để hoàn tất yêu cầu.

Kiến trúc sao chép cao cấp của Oracle

Oracle hội tụ dữ liệu của các cấu hình sao chép cao cấp tiêu biểu bằng cách sử dụng sao chép mức hàng với truyền dữ liệu đồng bộ. Các phần sau giải thích chức năng cơ chế này như thế nào.

> Lưu ý:
> Oracle cung ứng các đặc trưng sao chép cao cấp khác, chẳng hạn như sao chép thủ tục và truyền dữ liệu đồng bộ đối với các đòi hỏi ứng dụng riêng. Để tìm hiểu thêm về các cấu hình đặc biệt này, xem "Các Tùy Chọn Sao Chép Cao Cấp Riêng".

Sao chép mức hàng

Các ứng dụng xử lý giao dịch tiêu biểu sửa đổi một ít hàng trên mỗi giao dịch. Những ứng dụng như thế hoạt động ở môi trường

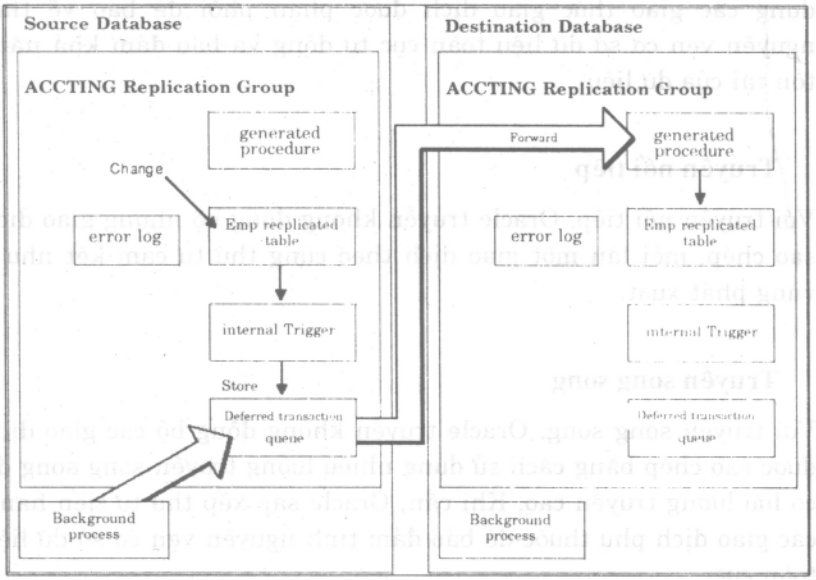
sao chép cao cấp thường phụ thuộc vào cơ chế sao chép mức hàng của Oracle, các ứng dụng sử dụng những câu lệnh DML tiêu chuẩn để sửa đổi dữ liệu của các bản sao dữ liệu cục bộ. Khi các giao dịch thay đổi dữ liệu cục bộ, thì server tự động đón bắt thông tin về các sửa đổi và các hàng đợi tương ứng những giao dịch bị trì hoãn để gửi các thay đổi cục bộ tới các vùng từ xa.

Truyền dữ liệu không đồng bộ (lưu trữ rồi gửi)

Những cấu hình sao chép cao cấp tiêu biểu dựa vào sao chép mức hàng để truyền các thay đổi mức dữ liệu bằng cách sử dụng sao chép dữ liệu không đồng bộ. Sao chép dữ liệu không đồng bộ xảy ra khi một ứng dụng cập nhật bản sao cục bộ của bảng, lưu trữ thông tin sao chép trong hàng đợi cục bộ và sau đó gửi thông tin sao chép tới các vùng sao chép khác vào thời gian sau. Do đó, sao chép dữ liệu không đồng bộ còn được gọi là sao chép dữ liệu lưu trữ rồi gửi.

Như Hình 31-10 trình bày Oracle sử dụng các bộ kích khởi của hệ thống bên trong của nó, các giao dịch bị trì hoãn, các hàng đợi giao dịch bị trì hoãn và các hàng đợi tác vụ để truyền những thay đổi mức dữ liệu không đồng bộ trong các vùng ban đầu trong hệ thống sao chép cao cấp cũng như từ ảnh chụp nhanh cập nhật tới bảng ban đầu của nó.
Khi các ứng dụng hoạt động trong môi trường sao chép cao cấp, Oracle sử dụng các bộ kích khởi bên trong để đón bắt và lưu trữ thông tin về các cập nhật dữ liệu được sao chép. Các bộ kích khởi bên trong thiết lập các lệnh gọi thủ tục từ xa (RPCs) sẽ tạo lại các thay đổi dữ liệu xảy ra ở vùng cục bộ tới các vùng sao chép từ xa. Các bộ kích khởi bên trong hỗ trợ sao chép dữ liệu là các thành phần chính yếu trong Oracle server khả thi; do đó, Oracle có thể đón bắt và lưu trữ các cập nhật tới dữ liệu sao chép rất nhanh ít sử dụng các tài nguyên hệ thống.

Hình 31-10 Các cơ chế sao chép dữ liệu không đồng bộ



Oracle lưu trữ các RPC được các bộ kích khởi bên trong tạo ra trong hàng đợi giao dịch bị trì hoãn của vùng để truyền sau. Oracle còn ghi thông tin về các giao dịch khởi đầu để tất cả các RPC thiết lập một giao dịch có thể được truyền và được áp dụng từ xa cũng như một giao dịch. Phương tiện sao chép cao cấp của Oracle bổ sung hàng đợi giao dịch bị trì hoãn bằng cách sử dụng cơ chế sắp hàng đợi cao cấp của Oracle.

Oracle quản lý tiến trình truyền bằng cách sử dụng cơ chế hàng đợi tác vụ của Oracle và các giao dịch bị trì hoãn. Mỗi server tham gia vào một hệ thống sao chép cao cấp, có hàng đợi tác vụ cục bộ. Một hàng đợi tác vụ của server là một bảng cơ sở dữ liệu lưu trữ thông tin về các tác vụ cục bộ, chẳng hạn như lệnh gọi PL/SQL để tiến hành tác vụ, vân vân. Các tác vụ tiêu biểu trong môi trường sao chép cao cấp có các tác vụ để đẩy các giao dịch bị trì hoãn tới các vùng ban đầu từ xa, các tác vụ xóa bỏ những giao dịch được áp dụng từ hàng đợi giao dịch bị trì hoãn và các tác vụ tô bổ các nhóm tô bổ ảnh chụp nhanh.

Oracle gửi thông tin sao chép dữ liệu bằng cách tiến hành các RPC là thành phần của những giao dịch bị trì hoãn. Oracle sử dụng các giao thức giao dịch được phân phối để bảo vệ tính nguyên vẹn cơ sở dữ liệu toàn cục tự động và bảo đảm khả năng tồn tại của dữ liệu.

Truyền nối tiếp

Với truyền nối tiếp, Oracle truyền không đồng bộ những giao dịch sao chép, mỗi lần một giao dịch theo cùng thứ tự cam kết như ở vùng phát xuất.

Truyền song song

Với truyền song song, Oracle truyền không đồng bộ các giao dịch được sao chép bằng cách sử dụng nhiều luồng truyền song song để có lưu lượng truyền cao. Khi cần, Oracle sắp xếp thứ tự tiến hành các giao dịch phụ thuộc để bảo đảm tính nguyên vẹn cơ sở dữ liệu toàn cục.

Truyền song song sử dụng cơ chế tiến hành tương tự Oracle sử dụng đối với truy vấn, tải, phục hồi song song và các điều hoạt song song khác. Mỗi tiến trình server truyền các giao dịch qua một luồng đơn. Bộ phối hợp song song điều khiển các tiến trình server này. Bộ phối hợp truy tìm các phụ thuộc giao dịch, phân bổ tác vụ cho các tiến trình server và theo dõi các tiến trình của chúng.

Xóa bỏ hàng đợi giao dịch bị trì hoãn

Sau khi một vùng đẩy một giao dịch bị trì hoãn tới đích của nó, giao dịch vẫn duy trì hàng đợi giao dịch bị trì hoãn cho tới khi một tác vụ khác xóa bỏ giao dịch được áp dụng từ hàng đợi ấy.

Các cơ chế truyền các ảnh chụp nhanh

Các ảnh chụp nhanh cập nhật trong môi trường sao chép cao cấp có thể "đẩy" và "kéo" dữ liệu tới và từ bảng ban đầu của nó tương ứng.

Các cập nhật bảng bản gốc

Cập nhật ảnh chụp nhanh có thể cập nhật được đẩy không đồng bộ tới bảng gốc của nó bằng cách sử dụng cơ chế truyền dữ liệu không đồng bộ, mức hàng của Oracle (các RPC, cửa sổ giao dịch bị trì hoãn và các hàng đợi tác vụ).

Tô bổ ảnh chụp nhanh

Giống như các môi trường sao chép căn bản, các hệ thống sao chép cao cấp sử dụng cơ chế tô bổ ảnh chụp nhanh của Oracle để kéo các thay đổi không đồng bộ từ bảng gốc tới các ảnh chụp nhanh cập nhật được kết hợp (và chỉ đọc).

Các nhận xét khác

Các tác vụ đẩy và kéo ảnh chụp nhanh cập nhật là các điều hoạt độc lập bạn có thể cấu hình kết hợp hoặc tách riêng.

Vùng ảnh chụp nhanh có thể cấu hình một nhóm tô bổ để kéo tất cả những thay đổi xảy ra đối với các ảnh chụp nhanh thành viên tới vùng gốc và sau đó tô bổ các ảnh chụp nhanh.
Vùng ảnh chụp nhanh có thể cấu hình các ảnh chụp nhanh cập nhật để đẩy những thay đổi tới vùng gốc và tô bổ các ảnh chụp nhanh ở những thời điểm và các thời khoảng khác nhau.

Thí dụ, môi trường sao chép cao cấp củng cố thông tin ở vùng gốc có thể cấu hình các ảnh chụp nhanh cập nhật cứ mỗi giờ đẩy những thay đổi tới vùng gốc nhưng ít khi tô bổ các ảnh chụp nhanh cập nhật, nếu có.

Các người điều hành sao chép, các bộ truyền và các bộ nhận

Một môi trường sao chép đối xứng Oracle đòi hỏi nhiều tài khoản người dùng cơ sở dữ liệu riêng để thực hiện chức năng thích hợp, kể cả những người điều hành sao chép, các bộ truyền và các bộ nhận.

Mỗi vùng trong hệ thống sao chép đối xứng Oracle đòi hỏi ít nhất một người điều hành sao chép, người dùng chịu trách nhiệm lập cache và duy trì các đối tượng cơ sở dữ liệu được sử dụng.

Mỗi vùng sao chép trong hệ thống sao chép đối xứng Oracle đòi hỏi các tài khoản người dùng để truyền và áp dụng các thay đổi vào dữ liệu được sử dụng.

Các tùy chọn cấu hình

Trong hầu hết các cấu hình sao chép cao cấp, chỉ một tài khoản được sử dụng cho tất cả các mục đích - như một người điều hành sao chép, bộ truyền sao chép và bộ nhận sao chép. Tuy nhiên, Oracle còn hỗ trợ các tài khoản riêng biệt cho các cấu hình cá biệt.

Các xung đột sao chép

Các hệ thống sao chép cao cấp hỗ trợ mô hình cập nhật bất kỳ nơi nào của các bản sao chép dữ liệu phải lập địa chỉ những xung đột có thể xảy ra. Các phần sau giải thích những kiểu xung đột sao chép khác nhau, khi chúng có thể xảy ra và Oracle có thể phát hiện và giải quyết các xung đột sao chép như thế nào.

Các kiểu xung đột sao chép

Có ba kiểu xung đột có thể xảy ra trong môi trường sao chép cao cấp: các xung đột cá biệt, các xung đột cập nhật và các xung đột xóa.

Các xung đột cá biệt

Xung đột cá biệt xảy ra khi sao chép một hàng cô xâm phạm đến tính toàn vẹn của thực thể (hạn định PRIMARY KEY hoặc UNIQUE). Thí dụ, xét những gì xảy ra khi hai giao dịch phát xuất từ hai vùng khác nhau, mỗi vùng chèn một hàng vào bản sao bảng tương ứng với cùng giá trị khóa chính - sao chép của những giao dịch sẽ gây ra xung đột cá biệt.

Các xung đột cập nhật

Xung đột cập nhật xảy ra khi sao chép một cập nhật các xung đột hàng với một cập nhật khác cùng hàng. Các xung đột cập nhật có thể xảy ra khi hai giao dịch phát xuất từ các vùng khác nhau, cập nhật cùng hàng ở thời gian gần như nhau.

Các xung đột xóa

Xung đột xóa xảy ra khi hai giao dịch phát xuất từ các vùng khác nhau, với một giao dịch xóa một hàng mà giao dịch khác cập nhật hoặc xóa.

Các mô hình dữ liệu sao chép và các xung đột

Khi thiết kế các ứng dụng hoạt động trên đầu hệ thống cơ sở dữ liệu sử dụng sao chép cao cấp, bạn phải xét khả năng các xung đột sao chép xảy ra. Thực hiện như vậy, các ứng dụng phải sử dụng một trong nhiều mô hình quan hệ sở hữu dữ liệu được sao chép khác nhau để bảo đảm tính nguyên vẹn cơ sở dữ liệu toàn cục hoặc giải quyết các xung đột sao chép.

Vùng chính, quan hệ sở hữu tĩnh

Quan hệ sở hữu chính còn được gọi là quan hệ sở hữu tĩnh là mô hình dữ liệu mà các môi trường sao chép chỉ đọc căn bản hỗ trợ. Quan hệ sở hữu chính tránh tất cả các xung đột sao chép, vì chỉ server đơn cho phép cập nhật truy xuất tập hợp dữ liệu được sao chép.

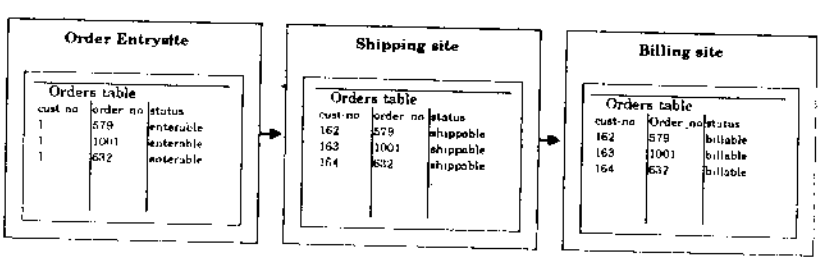
Thay vì điều khiển quan hệ sở hữu dữ liệu ở mức bảng, các ứng dụng có thể sử dụng phân hoạch ngang và dọc để thiết lập quan hệ sở hữu dữ liệu mức độ chi tiết tĩnh. Thí dụ, các ứng dụng có thể truy xuất cập nhật các cột hoặc các hàng riêng biệt trong bảng được sao chép trên phương thức theo từng vùng.

Quan hệ sở hữu động

Mô hình dữ liệu sao chép quan hệ sở hữu động ít giới hạn hơn quan hệ sở hữu vùng chính. Với quan hệ sở hữu động, khả năng cập nhật bản sao dữ liệu di chuyển từ vùng này tới vùng kia, vẫn bảo đảm chỉ một vùng cập nhật truy xuất dữ liệu xác định ở bất kỳ thời điểm ấn định nào. Hệ thống hoạt động theo luồng minh họa rõ ràng khái niệm về quan hệ sở hữu động. Thí dụ, các ứng dụng cửa hàng có liên quan có thể đọc mã trạng thái của đơn đặt hàng để quyết định khi nào chúng có thể và không thể cập nhật đơn đặt hàng ấy.

Hình 31-11 minh họa ứng dụng sử dụng mô hình sử dụng quan hệ sở hữu động.

Hình 31-11 Quan hệ sở hữu động trong hệ thống xử lý đơn đặt hàng



Quan hệ sở hữu dùng chung

Các mô hình dữ liệu sao chép quan hệ sở hữu động và quan hệ sở hữu vùng chính tăng khả năng tránh xung đột thường quá hạn chế hoặc không thể bổ sung một số ứng dụng cơ sở dữ liệu. Một số ứng dụng phải điều hoạt bằng cách sử dụng mô hình dữ liệu

quan hệ sở hữu dùng chung trong đó các ứng dụng có thể cập nhật dữ liệu của bản sao bảng bất kỳ lúc nào.

Khi hệ thống quan hệ sở hữu dữ liệu dùng chung sao chép các thay đổi không đồng bộ (sao chép lưu trữ để gửi), các ứng dụng tương ứng phải chắc chắn tránh được hoặc phát hiện và giải quyết các xung đột sao chép nếu và khi chúng xảy ra. Thí dụ:
Các xung đột xóa khó giải quyết trong mô hình quan hệ sở hữu dữ liệu dùng chung không đồng bộ trừ khi hệ thống sao chép ghi các thông tin lịch sử bằng các giao dịch xóa dữ liệu. Do đó, các ứng dụng điều hoạt trong mô hình dữ liệu quan hệ sở hữu dữ liệu dùng chung không đồng bộ sẽ tránh các xung đột xóa bằng cách không sử dụng các câu lệnh DELETE để xóa các hàng. Thay vào đó, các ứng dụng có thể đánh dấu các hàng và cấu hình hệ thống định kỳ để xóa các hàng sử dụng sao chép thủ tục.
Sự phát sinh chuỗi được phối hợp là kỹ thuật các ứng dụng có thể sử dụng để tránh các xung đột cá biệt trong hệ thống quan hệ sở hữu dữ liệu dùng chung. Thí dụ, các ứng dụng tiêu biểu sử dụng các chuỗi Oracle để phát sinh các khóa chính bằng số. Ở mỗi vùng trong hệ thống quan hệ sở hữu dữ liệu dùng chung, tạo các chuỗi sao chép để mỗi chuỗi phát sinh một tập hợp các số hiệu chuỗi riêng biệt hỗ tương.

Phát hiện xung đột

Khi một ứng dụng sử dụng mô hình dữ liệu quan hệ sở hữu dùng chung với sao chép mức hàng không đồng bộ, Oracle tự động phát hiện các xung đột cá biệt, cập nhật và xóa. Để phát hiện các xung đột khi sao chép, Oracle so sánh số nhỏ nhất của dữ liệu hàng từ vùng phát sinh với thông tin hàng tương ứng ở vùng nhận. Khi có sự khác biệt, Oracle phát hiện xung đột đó.

Để phát hiện các xung đột sao chép chính xác, Oracle phải có thể nhận dạng và so khớp các hàng tương ứng ở mỗi vùng khác nhau khi sao chép dữ liệu. Phương tiện sao chép cao cấp của

Oracle thường sử dụng khóa chính của bảng để nhận dạng riêng biệt các hàng trong bảng. Khi bảng không có khóa chính, bạn phải thiết kế một khóa thay thế - cột hoặc các cột mà Oracle có thể sử dụng để nhận dạng các hàng trong bảng ấy khi sao chép dữ liệu. Trong hai trường hợp, các ứng dụng sẽ không cho phép cập nhật các cột nhận dạng của bảng để bảo đảm Oracle có thể nhận dạng các hàng và duy trì tính nguyên vẹn của dữ liệu được sao chép.

Giải quyết xung đột

Khi một vùng nhận trong hệ thống sao chép cao cấp sử dụng sao chép mức hàng không đồng bộ và nó phát hiện một xung đột ở một giao dịch, thì hoạt động mặc định được ghi sổ xung đột ấy và toàn bộ giao dịch, để phiên cục bộ của dữ liệu không bị thay đổi. Trong hầu hết các trường hợp, bạn sử dụng phương tiện sao chép cao cấp của Oracle để tự động giải quyết các xung đột sao chép. Bạn cũng có thể kiểm từng khung nhìn từ điển dữ liệu DEFERROR của mỗi server cho các giao dịch gây ra các xung đột và giải quyết chúng thủ công, nếu cần.

Các nhóm cột

Oracle sử dụng các nhóm cột để phát hiện và giải quyết cửa sổ xung đột khi sao chép đối xứng mức hàng không đồng bộ. Nhóm cột là lập nhóm logic của một hoặc nhiều cột trong bảng. Mỗi cột trong bảng được sao chép là thành phần của nhóm cột đơn. Khi cấu hình các bảng được sao chép, bạn có thể tạo các nhóm cột và sau đó gán các cột và các phương pháp giải quyết xung đột đối với mỗi nhóm.

Mỗi nhóm cột trong bảng được sao chép có thể có một danh sách về một hoặc nhiều phương pháp giải quyết xung đột. Chỉ các phương pháp giải quyết nhiều xung đột cho một nhóm cho phép Oracle giải quyết xung đột bằng nhiều cách khác nhau nếu các cách khác không giải quyết được xung đột đó. Khi cố giải quyết

một xung đột cho một nhóm, Oracle tiến hành các phương pháp giải quyết của nhóm ấy theo thứ tự bạn liệt kê cho nhóm ấy.

Bằng mặc định, mỗi bảng được sao chép có một nhóm cột bóng. Một nhóm cột bóng có tất cả các cột không có trong nhóm cột ấn định. Bạn không thể gán các phương pháp giải quyết xung đột vào nhóm bóng của bảng.

Các phương pháp giải quyết xung đột

Khi thiết kế các nhóm cột bạn có thể chọn trong nhiều phương pháp giải quyết xung đột được lập sẵn. Thí dụ, để giải quyết các xung đột cập nhật, bạn có thể chọn để được Oracle ghi đè các giá trị cột ở vùng đích với những giá trị cột từ vùng phát sinh. Oracle cung ứng nhiều phương pháp giải quyết xung đột khác.

Các tùy chọn sao chép cao cấp cá biệt

Một số ứng dụng có các đòi hỏi đặc biệt về hệ thống sao chép cao cấp. Các phần sau giải thích các tùy chọn sao chép cao cấp cá biệt của Oracle, gồm

Sao chép thủ tục

Truyền dữ liệu đồng bộ (Real-Time)

Sao chép thủ tục

Các ứng dụng xử lý lô có thể thay đổi số lượng dữ liệu lớn trong một giao dịch đơn. Trong các trường hợp như thế, sao chép mức hàng tiêu biểu có thể bão hòa một mạng với số lượng thay đổi dữ liệu khổng lồ. Để tránh những vấn đề như thế, một ứng dụng xử lý lô điều hoạt trong môi trường sao chép cao cấp có thể sử dụng sao chép thủ tục của Oracle để sao chép các lệnh gọi thủ tục định sẵn đơn để quy tụ các sao chép dữ liệu. Sao chép thủ tục chỉ sao chép lệnh gọi ấy tới các thủ tục định sẵn mà ứng dụng sử dụng để cập nhật bảng. Sao chép thủ tục không sao chép các sửa đổi dữ liệu.

Để sử dụng sao chép thủ tục, ở tất cả các vùng bạn phải sao chép đóng gói dữ liệu sửa đổi trong hệ thống ấy. Sau khi sao chép một vỏ bọc, bạn phải phát sinh một bộ gói (wrapper) cho vỏ bọc này ở mỗi vùng. Khi ứng dụng gọi một thủ tục được đóng gói ở vùng cục bộ để sửa đổi dữ liệu, bộ gói bảo đảm lệnh gọi ấy cuối cùng được thực hiện với cùng thủ tục được đóng gói ở tất cả các vùng khác trong môi trường được sao chép. Sao chép thủ tục có thể xảy ra không đồng bộ hoặc đồng bộ.

Phát hiện xung đột và sao chép thủ tục

Khi một hệ thống sao chép cao cấp sao chép dữ liệu bằng cách sử dụng sao chép thủ tục, thì các thủ tục ấy sao chép dữ liệu chịu trách nhiệm bảo đảm tính nguyên vẹn dữ liệu được sao chép. Nghĩa là bạn phải thiết kế các thủ tục như thế để tránh hoặc để phát hiện các xung đột sao chép và giải quyết chúng thích hợp. Do đó, sao chép thủ tục thường được sử dụng nhất khi các cơ sở dữ liệu chỉ có thể xử lý các điều hoạt lô lớn. Trong các trường hợp như thế, những xung đột sao chép không thể xảy ra vì nhiều giao dịch cạnh tranh cùng dữ liệu.

Thông tin bổ sung:
Xem Sao chép của Oracle8.

Truyền dữ liệu đồng bộ (Real-Time)

Dữ liệu không đồng bộ là cấu hình bình thường cho những môi trường sao chép cao cấp. Tuy nhiên, Oracle còn hỗ trợ truyền dữ liệu đồng bộ cho các ứng dụng với các đòi hỏi đặc biệt. Truyền dữ liệu đồng bộ xảy ra khi một ứng dụng cập nhật một bản sao cục bộ của cùng bảng. Do đó sao chép dữ liệu đồng bộ còn được gọi là sao chép dữ liệu thời gian thực. Chỉ sử dụng sao chép đồng bộ khi ứng dụng đòi hỏi các vùng được sao chép vẫn được đồng bộ hóa tiếp tục.

Lưu ý:
Hộp thoại sao chép sử dụng truyền thời gian thực của dữ liệu sao chép rất phụ thuộc vào tính khả dụng hệ thống và mạng vì nó chỉ có thể thực hiện chức năng khi tất cả các vùng trong hệ thống có thể sử dụng đồng thời.

Như được trình bày ở Hình 31-12, Oracle sử dụng cùng hệ thống của các bộ kích khởi cơ sở dữ liệu bên trong để phát sinh các RPC sao chép các thay đổi mức dữ liệu đối với các vùng sao chép khác để hỗ trợ sao chép dữ liệu mức hàng không đồng bộ. Tuy nhiên, Oracle không trì hoãn tiến hành các RPC như thế. Thay vào đó, các RPC sao chép dữ liệu tiến hành trong biên của cùng giao dịch sửa đổi bản sao cục bộ. Do đó, thay đổi mức dữ liệu phải có thể xảy ra ở tất cả các vùng quản lý bảng được sao chép nếu không hồi phục về trạng thái cũ của giao dịch xảy ra.

Bạn có thể chọn để tạo một môi trường được sao chép trong đó một số vùng truyền các thay đổi đồng bộ khi các vùng khác sử dụng truyền không đồng bộ (các giao dịch bị trì hoãn).

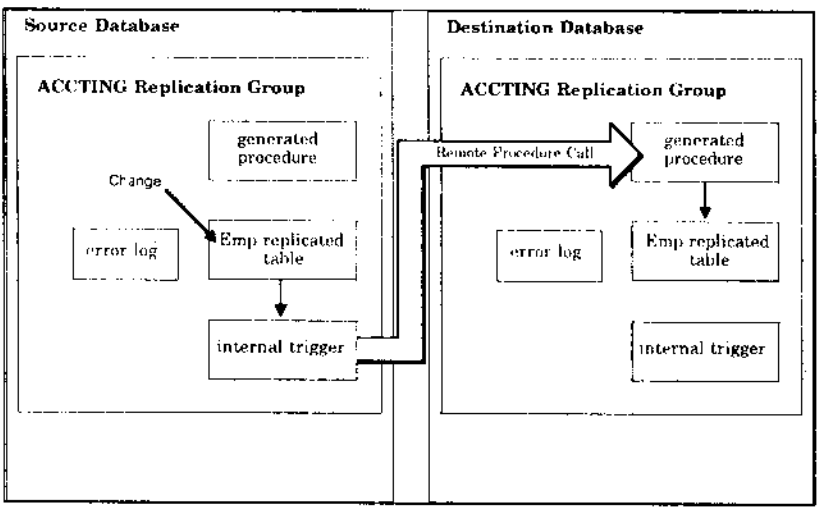
Các xung đột sao chép
và các sao chép dữ liệu đồng bộ

Khi hệ thống quan hệ sở hữu dùng chung sao chép tất cả những thay đổi đồng bộ (sao chép thời gian thực), các xung đột sao chép không thể xảy ra. Với sao chép thời gian thực, các ứng dụng sử dụng các giao dịch được phân phối để cập nhật tất cả các bản sao của bảng vào cùng thời gian. Như trường hợp ở các môi trường cơ sở dữ liệu không được phân phối, Oracle tự động khóa các hàng thay cho mỗi giao diện được phân phối để tránh tất cả kiểu giao thoa hủy trong các giao dịch. Trong khi hệ thống sao chép thời gian thực có thể tránh được các xung đột sao chép, kiểu hệ thống này rất phụ thuộc vào tính khả dụng của hệ thống và mạng vì nó chỉ có thể thực hiện chức năng khi tất cả những vùng trong hệ thống ấy có thể sử dụng.

Thông tin bổ sung:

Xem Sao chép của Oracle8 về sao chép cơ sở dữ liệu cao cấp và căn bản.

Hình 31-12 Các cơ chế sao chép dữ liệu đồng bộ



Các quy ước kiểu chữ

Tài liệu này có các quy ước sau:

Kiểu Chữ	Ý Nghĩa
Chữ hoa	Các chữ hoa chỉ các lệnh Oracle, các đối tượng cơ sở dữ liệu tiêu chuẩn và các hằng và các thủ tục PL/SQL của Oracle tiêu chuẩn.
Chữ nghiêng thường	Các chữ nghiêng chỉ tên các biến, các tên của những đối tượng người dùng (các bảng, các khung nhìn, các tham chiếu,

	các phương thức, vân vân.) các tên tham số/đối số của PL/SQL và các tên của thủ tục PL/SQL điển hình. Chữ nghiêng cũng chỉ sự nhấn mạnh.
Đơn cách	Kiểu đơn cách chỉ các lệnh SQL*Plus điển hình và mã PL/SQL điển hình. Kiểu chữ trong lệnh hoặc mã chính xác như nó biểu thị.

Oracle 8 Database

Lê Minh Trí

Chịu trách nhiệm xuất bản:
LÊ HOÀNG
Biên tập: **PHẠM SỸ SÁU**
Bìa và trình bày: **AZDesign**
Sửa bản in: **NGUYỄN LỤC**
Đơn vị liên doanh: **Cty Văn hóa Phương Nam**

Tổng phát hành

Công ty Văn hóa Phương Nam
160/12-14 Quân Sự, Phường 9, Quận 11, TP.HCM
ĐT: 8558504 - 8589592 * Fax: 8588908

Nhà sách Tiền Phong 175 Nguyễn Thái Học, Hà Nội * ĐT: 7.336.235 * Fax: 7.336236
Nhà sách Phương Nam 19 Phan Chu Trinh, Đà Nẵng * ĐT, Fax: (05-11) 821470
Nhà sách Phương Nam 06 Hòa Bình, TP Cần Thơ * ĐT: (071) 813436 * Fax: (071) 813437

In 800 cuốn, khổ 14.5 x 20cm tại Xí nghiệp in Công ty Văn hóa Phương Nam. Số đăng ký kế hoạch xuất bản 402/53 do Cục Xuất bản ký ngày 21-4-2000 và giấy trích ngang KHXB số 620/2000. In và nộp lưu chiểu tháng 08 năm 2000.

NEW !

Oracle 8 Database

TỦ SÁCH TIN HỌC

PHƯƠNG NAM

Đã phát hành

THUẬT NGỮ TIN HỌC ANH - VIỆT

Sắp phát hành

SQL* PLUS for Windows NT & Windows 95-98

8 932000 101024

Giá : 85.000 Đ