

Broadview®  
www.broadview.com.cn

IT PUB NET® 图书推荐

畅销书升级版

## 毫无保留

- 从数据库的基础知识入手，全面介绍Oracle 11g数据库的所有特性
- 全书配以翔实的示例，严谨的论述，具有很强的可操作性
- 为初学者，本书提供一条有组织准确地掌握Oracle核心技术的途径
- 为从业者，本书深入探讨了Oracle 11g众多的新特性，可以快速应用
- 为资深者，本书还讲解了Oracle数据仓库与非结构数据库，常备案头很有帮助

# 谷长勇

谷长勇 吴逸云 单永红 陈杰 等编著

权威指南

# ORACLE 11g

(第2版)



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
http://www.phei.com.cn

# Oracle 11g 权威指南

## ( 第 2 版 )

谷长勇 吴逸云 单永红 陈 杰 等编著

电子工业出版社  
Publishing House of Electronics Industry  
北京 · BEIJING

## 内 容 简 介

本书从数据库的基础知识入手,全面系统地介绍了 Oracle 11g 数据库管理系统的所有特性,并配以翔实的实际用例,论述严谨,深入探讨了这些特性的细节内容,同时具有很强的可操作性和实用性。

全书内容共 37 章,分为 10 大部分。第一部分为 Oracle 11g 数据库基础;第二部分为 Oracle 之 SQL 与 PL/SQL;第三部分为 Oracle 客户端工具,主要介绍 SQL\*Plus 和 SQL Developer;第四部分为 Oracle 11g 数据库管理,主要介绍管理控制文件/重做日志文件/表空间/数据文件、模式对象管理、表分区和索引分区、用户管理与安全、数据完整性和数据约束等;第五部分为 Oracle 数据库优化,着重介绍 Oracle 数据库的系统调整、SQL 优化等知识;第六部分为 Oracle 11g 数据库的备份与恢复;第七部分为 Oracle 11g 数据库集群技术与高可用性,包括真正应用集群(RAC)、自动存储管理(ASM)、容灾与数据卫士、故障诊断、真正应用测试等内容;第八部分为商业智能与数据仓库,介绍了 Oracle 11g 数据库在构建企业级数据仓库时的强大功能和诸多新特性;第九部分为非结构化数据库,包括空间数据库和 XML 数据库内容;第十部分为 Oracle 11g 数据库的其他新特性。

本书是 Oracle 11g 数据库所有特性的集大成手册,可供 Oracle 数据库管理员、Oracle 数据库应用开发人员、Oracle 数据仓库工程师使用,还可以作为 Oracle 技术支持和培训结构、高等院校数据库课程的参考教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

Oracle 11g 权威指南 / 谷长勇等编著. —2 版. —北京: 电子工业出版社, 2011.9  
ISBN 978-7-121-14580-3

I. ①O… II. ①谷… III. ①关系数据库—数据库管理系统, Oracle 11g IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2011)第 185782 号

责任编辑: 葛 娜

印 刷: 北京京科印刷有限公司

装 订: 三河市皇庄路通装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 850×1168 1/16 印张: 62.75 字数: 1600 千字

印 次: 2011 年 9 月第 1 次印刷

印 数: 3500 册 定价: 128.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zltz@phei.com.cn](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。



# 前言

# Foreword

## 关于本书

在 IT 技术高速发展的今天，数据库技术的地位越来越重要。任何投入运行的信息系统，都需要有数据库管理系统（DBMS）作为支撑。其中，Oracle 以其卓越的性能获得了广泛的应用，已经成为当今世界上最流行的关系型数据库管理系统。因其在数据安全性与数据完整性方面的优越性能，以及跨越操作系统、多硬件平台的数据互操作等特点，越来越多的用户使用 Oracle 作为其信息系统管理、企业数据处理、Internet、电子商务网站等领域应用数据的后台处理系统，其使用已遍及军队、邮政、电信、海关、税务、保险、电力、化工和汽车等各行各业。如何开发、使用与管理 Oracle 数据库系统，特别是如何保证大型数据库的安全运行，一直是广大数据库用户最为关心的问题。

为了追求更高的技术水平和更好地满足用户的需求，Oracle 不断发展和完善自己，最新推出的 11g 就是原有版本的又一次飞跃。它不但改进了许多已有的功能，还增加了很多更加智能、可靠的新特性。

## 本书特点

### 1. 内容丰富

本书不仅包括了 Oracle 数据库的基础概念和开发，也包括高级数据库管理技术（如备份、恢复、优化等），而且还涵盖了 Oracle 数据库中的数据仓库特性（如 ETL、OLAP、数据挖掘等）和非结构数据管理技术（包括 XML、空间数据）。

### 2. 逐层深入

在每个章节组织上，首先介绍初级知识，然后介绍高级新特性，最后结合实际用例解释。初级知识有助于初学者入门；而高级新特性可作为有经验的 Oracle 数据库爱好者快速学习 11g 的资料；用例部分则作为开发者理解各种特性的参考。

## 本书结构

全书共分为十大部分，按照内容模块组织章节，各大部分及每一部分的内部章节之间既前后有序、循序渐进，同时又具备相对独立性，方便不同层次的读者阅读和使用。

第一部分为 Oracle 11g 数据库基础，包括数据库基础知识、Oracle 11g 简介、Oracle 11g 数据库安装/启动/关闭、Oracle 数据库组成结构等内容，让我们在对数据库和 Oracle 11g 形成较好的了解和整体认知的基础上，为后继内容的学习打下良好基础。

第二部分为 Oracle 之 SQL 与 PL/SQL，主要介绍数据库之 SQL 语言，使用者通过对 SQL 语言的使

用可实现与数据库的各种交互,对 SQL 语言的介绍从基础语法到复杂 SQL 语句编写;同时介绍了 PL/SQL 编程,从基础编程到高级应用,循序渐进。读者通过对本部分的学习,可以实现 SQL 及 PL/SQL 日常编写和应用,并在日常的工作中通过应用和经验总结,不断提高自己这方面的能力和技巧。

第三部分为 Oracle 客户端工具,主要介绍 Oracle 公司提供的客户端产品,包括 SQL\*Plus 和 SQL Developer,读者通过本部分介绍的客户端工具并利用从第二部分学到的知识,可以方便地连接到数据库并进行各种数据库操作。

第四部分为 Oracle 11g 数据库管理,主要介绍 Oracle 数据库的管理控制文件/重做日志文件/表空间/数据文件、模式对象管理、表分区和索引分区、用户管理与安全、数据完整性和数据约束,以及如何使用 Oracle 11g Enterprise Manager 来管理 Oracle 11g 数据库。面向 Oracle 高级用户详细介绍了对 Oracle 数据库各方面的管理和应用。

第五部分为 Oracle 数据库优化,着重介绍 Oracle 数据库的系统调整、SQL 优化等知识。读者通过对本部分的学习可以编写出高质量、高性能的 SQL 语句,同时也能进行数据库级的系统优化,对数据库本身进行优化和调整。

第六部分为 Oracle 11g 数据库的备份与恢复,本部分内容在对备份与恢复进行了总体介绍以后,详细讲解了 RMAN 工具的使用、Oracle 闪回技术、使用 OEM 向导备份和恢复数据库及数据导入/导出等内容。

第七部分为 Oracle 11g 数据库集群技术与高可用性,包括真正应用集群(RAC)、自动存储管理(ASM)、容灾与数据卫士、故障诊断、真正应用测试等内容。

第八部分为商业智能与数据仓库,本部分介绍了 Oracle 11g 数据库在构建企业级数据仓库时的强大功能和诸多新特性,包括 ETL 相关的功能、Oracle Warehouse Builder、多维数据库、数据挖掘等内容。

第九部分为非结构化数据库,包括空间数据库和 XML 数据库内容。

第十部分为 Oracle 11g 数据库的其他新特性,对于 Oracle 11g 应用增强、高级数据库管理和数据库升级等方面的新特性进行了详细讲解。

全书理论与实践紧密结合,图文并茂,示例翔实,有助于广大爱好者的学习和把握。

## 本书作者

本书的大部分章节由谷长勇和吴逸云执笔完成。另外,由陈杰完成了自动存储管理(ASM)、在线管理维护、生命周期管理、故障诊断等章节,单永红执笔完成了数据卫士、SQL 增强、数据库控制、整体数据库管理及本书的内容简介与前言部分,而书中的表分区和索引分区、Data Pump 等部分章节则由吴逸云和陈杰合作完成。参与本书编写的还有王彬、杨静、王文雅、杨雪、王大才、戴东东、安佰胜。

## 读者对象

本书是第一本将所有主要 Oracle 11g 数据库命令、函数、特性和产品综合在一起的核心参考书籍,

每位 Oracle 用户和开发人员都应常备一册。

**Oracle 数据库初学者和一般用户：**本书试图提供一种有组织、有效地掌握 Oracle 核心内容的方法和思路，帮助 Oracle 的初级开发人员和一般用户快速了解基本概念，快速掌握开发技术，克服常见困难，以及获得正确的 Oracle 数据库知识。

**具有一定基础的数据库从业人员：**Oracle 11g 数据库包括了众多的新特性，本书深入探讨了许多这样的问题（如真正应用测试、信息生命周期等），有助于有一定 Oracle 技术基础的技术人员快速掌握 11g 新技术。

**致力于 Oracle 数据库管理的人员：**本书内容丰富，不仅包括了常规的关系数据库开发和维护内容，也包括对 Oracle 数据仓库与非结构数据库的介绍，有助于全面掌握 Oracle 数据库管理技术。

本书也可以作为各大中专院校相关专业的教学辅导或参考用书，或者作为相关培训机构的培训教材。

## 致谢

本书在写作过程中，得到了甲骨文（中国）软件公司数据库咨询顾问团队的大力帮助，对此表示感谢。

由于作者水平有限，时间仓促，而且 Oracle 11g 的可借鉴资料比较少，错误和不妥之处在所难免，望各位专家和读者给予谅解和指正，不吝赐教。

“我们从您那里学到的与我们传授的一样多”。

编 者





# 目 录 Contents

## 第一部分 Oracle 数据库 11g 基础

### 第 1 章 数据库基础简介 2

本章主要介绍了数据库的基础知识，包括数据库、数据库管理系统及关系数据库等基本概念，此外还介绍了目前应用比较广泛的各种关系型数据库系统。

1.1 数据库起源	2
1.2 数据库主要特征	2
1.3 数据库构成	3
1.4 数据库管理系统	3
1.5 关系数据库	4
1.6 本章小结	8

### 第 2 章 Oracle 11g 概述 9

本章主要介绍了数据库的基础知识、Oracle 数据库的特点、发展历史及 Oracle 11g 的一些新特性。

2.1 Oracle 数据库发展演进史	9
2.2 Oracle 版本号的含义	10
2.3 Oracle 网格技术	11
2.3.1 网格概述	11
2.3.2 网格的基本属性	12
2.3.3 Oracle 网格基础架构	13
2.4 Oracle 11g 新特性	15
2.4.1 企业网格管理的高可用性	15
2.4.2 更加优化的性能	17
2.4.3 简化的信息管理	18
2.4.4 集成的信息	20
2.4.5 内置的业务智能	21
2.4.6 开发平台	22
2.5 本章小结	24

### 第 3 章 Oracle 11g 数据库的安装、启动及关闭 25

本章详细讲述了 Oracle 数据库的安装、启动、关闭的整个过程，以及如何使用 DBCA 创建一个数据库。对数据库管理员而言，这些操作都是最基本的。

3.1 在 Windows 环境下安装的软硬件要求	25
3.2 在 Windows 环境下安装 Oracle 11g	25
3.3 在 Linux 环境下安装前的准备	28
3.3.1 安装 Oracle 11g 对硬件的要求	28

3.3.2	安装 Oracle 11g 对操作系统的要求	29
3.3.3	安装 RedHat AS4 操作系统时的注意事项	31
3.3.4	检查安装 Oracle 11g 所必需的包	33
3.3.5	内核参数配置	33
3.3.6	创建安装数据库所需要的用户 (组)	34
3.3.7	添加 Oracle 用户的限制参数	34
3.3.8	创建安装所需要的目录	34
3.3.9	设置 Oracle 的环境变量	35
3.3.10	解压缩安装包	35
3.4	在 Linux 环境下安装 Oracle 11g	36
3.4.1	Oracle 用户图形终端方式登录	36
3.4.2	安装操作过程	36
3.4.3	登录 OEM 界面验证安装	40
3.5	Oracle 11g 数据库实例的创建	40
3.5.1	在 Windows 环境下创建数据库	40
3.5.2	在 Linux 环境下用 DBCA 创建数据库	41
3.6	Oracle 11g 实例的启动和关闭	46
3.6.1	在 Windows 操作系统下启动和关闭 Oracle 实例	46
3.6.2	在 Linux 环境下启动 Oracle 实例	47
3.6.3	在 Linux 环境下关闭 Oracle 实例	49
3.6.4	在 Linux 环境下启动和关闭 Oracle 监听进程	49
3.7	本章小结	50

## 第 4 章 Oracle 数据库结构

51

本章介绍了 Oracle 数据库的体系结构、服务器结构、Oracle 数据库文件及后台进程。

4.1	Oracle 体系结构	51
4.1.1	基本术语	51
4.1.2	体系结构图解	52
4.1.3	表空间与数据文件	52
4.1.4	临时表空间与临时文件	54
4.1.5	Oracle 存储结构介绍	55
4.1.6	控制文件	55
4.1.7	日志文件	56
4.1.8	服务器参数文件	58
4.1.9	密码文件/跟踪文件/警告日志	59
4.2	Oracle 服务器结构	60
4.2.1	Oracle 服务器与 Oracle 实例	60
4.2.2	物理结构与逻辑结构的关系	60
4.2.3	系统全局区 (SGA)	61
4.2.4	后台进程	62
4.2.5	程序全局区 PGA	64
4.3	数据字典	65



4.3.1 Oracle 数据字典的构成	65
4.3.2 Oracle 常用的数据字典	65
4.3.3 Oracle 常用的动态性能视图	67
4.4 本章小结	67

## 第二部分 Oracle 之 SQL&PL/SQL

### 第 5 章 SQL 语言介绍

70

本章主要介绍了SQL语言概述、如何使用简单和复杂的SELECT检索语句、如何使用基本函数、如何使用DDL语句、如何使用DML语句、如何使用描述语句等内容。

5.1 SQL 语言概述	70
5.1.1 SQL 语言特点	71
5.1.2 SQL 语言分类	71
5.1.3 SQL 语言的编写规则	71
5.2 使用 SELECT 检索数据	72
5.2.1 使用 FROM 子句指定表	73
5.2.2 使用 SELECT ( 必要元素 ) 指定列	73
5.2.3 使用 WHERE 子句指定行	74
5.2.4 使用 ORDER BY 子句	79
5.2.5 使用 DISTINCT 检索唯一的表列值	80
5.2.6 使用算术运算符	81
5.3 基本函数的使用	82
5.3.1 字符函数	82
5.3.2 数字函数	85
5.3.3 日期函数	87
5.3.4 转换函数	87
5.3.5 组函数	88
5.3.6 使用 GROUP BY 进行数据分组	89
5.3.7 使用 HAVING 子句限制分组数据	90
5.3.8 Oracle 递归函数的使用	91
5.4 复杂 SELECT 语句的使用	91
5.4.1 集合操作	91
5.4.2 子查询	96
5.4.3 表连接	96
5.4.4 CASE 语句的使用	99
5.4.5 强大的 DECODE 函数	101
5.5 DDL 语句的使用	102
5.5.1 常用的数据类型	102
5.5.2 表	104
5.5.3 主键	105
5.5.4 外键	107

5.5.5 约束	108
5.5.6 索引	110
5.5.7 视图	112
5.5.8 序列	113
5.5.9 同义词	114
5.6 DML 语句的使用	114
5.6.1 使用 INSERT 语句插入表数据	115
5.6.2 使用 UPDATE 语句更新表数据	116
5.6.3 使用 DELETE 语句删除表数据	117
5.6.4 使用 MERGE 语句修改表数据	117
5.6.5 TRUNCATE 语句的使用	121
5.7 使用描述语句	121
5.8 本章小结	122

## 第 6 章 PL/SQL 基础编程

123

本章介绍了PL/SQL概述、PL/SQL编程、过程和函数、错误处理等内容。

6.1 PL/SQL 概述	123
6.1.1 PL/SQL 语言	123
6.1.2 PL/SQL 主要特性	125
6.1.3 PL/SQL 的开发和运行环境	127
6.1.4 运行 PL/SQL 程序	128
6.2 PL/SQL 编程	129
6.2.1 基本语言块	130
6.2.2 字符集和语法注释	130
6.2.3 数据类型和类型转换	132
6.2.4 变量和常量	136
6.2.5 表达式和运算符	138
6.2.6 流程控制	140
6.3 过程和函数	147
6.3.1 过程	148
6.3.2 函数	149
6.4 错误处理	151
6.4.1 预定义异常	151
6.4.2 用户定义异常	153
6.5 本章小结	155

## 第 7 章 PL/SQL 高级应用

156

本章介绍了包的概念和使用、集合的使用及PL/SQL的新特性等内容。

7.1 包	156
7.1.1 包头	156
7.1.2 包体	157
7.1.3 重载	161

7.1.4	包的初始化	163
7.1.5	Oracle 内置包	165
7.2	集合	165
7.2.1	index-by 表	166
7.2.2	嵌套表	167
7.2.3	可变数组	169
7.2.4	集合的属性和方法	169
7.3	PL/SQL 游标	173
7.3.1	创建游标	174
7.3.2	使用游标的 FOR 循环	178
7.3.3	使用游标变量	180
7.3.4	游标变量实例	187
7.3.5	使用游标表达式	191
7.4	Oracle 11g 中 PL/SQL 的新特性	192
7.5	本章小结	194

## 第三部分 Oracle 客户端工具

### 第 8 章 SQL\*Plus 工具介绍

196

本章介绍了如何使用 SQL\*Plus 与数据库交互、如何使用 SQL\*Plus 的编辑功能、如何使用 SQL\*Plus 格式化查询结果等内容。

8.1	用 SQL*Plus 与数据库交互	196
8.1.1	SQL*Plus 的主要功能	197
8.1.2	启动 SQL*Plus 连接数据库	197
8.2	使用 SQL*Plus 的编辑功能	199
8.2.1	SQL 语句、PL/SQL 块与 SQL*Plus 命令的区别	199
8.2.2	编辑命令	200
8.2.3	保存命令	202
8.2.4	加入注释	203
8.2.5	运行命令	205
8.2.6	编写交互命令	206
8.2.7	使用绑定变量	213
8.2.8	跟踪语句	216
8.3	使用 SQL*Plus 格式化查询结果	221
8.3.1	格式化列	221
8.3.2	定义页与报告的标题和维	231
8.3.3	存储和打印结果	237
8.4	本章小结	238

### 第 9 章 Oracle SQL Developer 工具介绍

239

本章主要介绍了 Oracle SQL Developer 的使用，包括创建数据库连接，使用 SQL Worksheet 操作数据库，

开发调试PL/SQL语句，运行和创建报表。另外，通过实例演示了如何使用SQL Developer的迁移工作台完成数据库的迁移操作。

9.1	SQL Developer 概述	239
9.1.1	SQL Developer 版本演化	239
9.1.2	SQL Developer 产品特点	239
9.1.3	SQL Developer 下载与安装	239
9.2	创建数据库连接	240
9.2.1	连接到 Oracle 数据库	240
9.2.2	连接到 MySQL 数据库	242
9.2.3	连接到 SQL Server 数据库	242
9.3	使用 SQL Worksheet 操作数据库	242
9.3.1	检索数据	243
9.3.2	插入数据	245
9.3.3	修改数据	246
9.3.4	删除数据	248
9.4	使用 SQL Developer 开发和调试 PL/SQL	249
9.4.1	准备工作	249
9.4.2	创建并编译 PL/SQL 过程	249
9.4.3	运行 PL/SQL 过程	254
9.4.4	调试 PL/SQL 过程	256
9.5	使用 SQL Developer 运行或创建报表	261
9.5.1	运行预定义报表	261
9.5.2	创建自定义报表	262
9.6	使用 SQL Developer 导出操作	263
9.6.1	导出数据	264
9.6.2	导出对象定义	265
9.7	使用 SQL Developer 迁移工作台	266
9.7.1	迁移工作台概述	266
9.7.2	数据迁移实例演示	267
9.8	本章小结	273

## 第四部分 Oracle 11g 数据库管理

### 第 10 章 管理控制文件

276

本章主要讲述了Oracle数据库对控制文件的管理，包括控制文件的内容、更新、多路复用及创建和维护。控制文件对数据库的正常运行至关重要，通过本章的学习，读者可以对控制文件有一定的理解。

10.1	控制文件概述	276
10.1.1	控制文件的组成	276
10.1.2	控制文件的大小	277
10.1.3	控制文件更新	277
10.1.4	可复用区与不可复用区	277

10.2	控制文件的多路复用	277
10.2.1	使用 init.ora 多路复用控制文件	277
10.2.2	使用 SPFILE 多路复用控制文件	278
10.2.3	添加更多控制文件副本的步骤	278
10.3	查询控制文件信息	279
10.4	控制文件的创建步骤	279
10.5	维护控制文件	281
10.6	本章小结	283

## 第 11 章 管理日志文件 284

本章讲述了联机重做日志的作用和管理。联机重做日志是数据库正常运行不可或缺的文件，对于实例出现故障时的正常恢复是十分重要的。

11.1	管理重做日志文件	284
11.2	日志文件组、日志切换和日志归档	284
11.3	了解检查点	285
11.4	查询、新建、删除日志文件	285
11.5	本章小结	287

## 第 12 章 管理表空间和数据文件 288

本章主要介绍了 Oracle 数据库的逻辑结构、默认表空间，以及表空间的创建、设置、删除等管理操作。这些操作在数据库的日常管理中是十分常用的。

12.1	表空间与数据文件的关系	288
12.2	Oracle 11g 默认的表空间列表	289
12.2.1	SYSTEM 表空间	289
12.2.2	SYSAUX 表空间	290
12.3	建立表空间	291
12.3.1	创建表空间的语法	291
12.3.2	选择盘区管理方式建立表空间	292
12.3.3	选择段空间管理方式建立表空间	293
12.3.4	创建非标准块表空间	294
12.3.5	建立大文件表空间	294
12.4	维护表空间与数据文件	295
12.4.1	改变表空间的读写状态	295
12.4.2	改变表空间的名称	296
12.4.3	设置默认表空间	296
12.4.4	删除表空间	297
12.4.5	增建新的数据文件到表空间	297
12.4.6	删除表空间中无数据的数据文件	298
12.4.7	数据文件的自动扩展设置	298
12.5	管理撤销表空间 (UNDO)	299
12.5.1	撤销表空间的作用	299
12.5.2	与撤销表空间有关的参数	300

12.5.3	表空间的基本操作	301
12.6	管理临时表空间	303
12.6.1	临时表空间的概念	303
12.6.2	建立临时表空间	304
12.6.3	查询临时表空间的信息	304
12.6.4	临时表空间组	304
12.7	本章小结	306

## 第 13 章 模式对象管理

307

本章讲述了Oracle数据库中最重要几个模式对象的管理，包括表、索引、视图、同义词、序列等。

13.1	模式与模式对象	307
13.2	管理表	307
13.2.1	建表时的考量	307
13.2.2	列数据类型	308
13.2.3	创建表的语法	309
13.2.4	创建表的实例	310
13.2.5	查看表的信息	311
13.2.6	修改表结构	312
13.2.7	重命名表	313
13.2.8	监控表的存储空间	313
13.2.9	修改表空间的存储设置	314
13.2.10	删除表	314
13.2.11	外部表的管理	314
13.3	索引	316
13.3.1	概念	316
13.3.2	各种类型索引的比较和选择	317
13.3.3	索引的基本操作	317
13.4	视图	320
13.4.1	概念	320
13.4.2	视图的基本操作	320
13.5	同义词	322
13.6	序列	323
13.7	本章小结	323

## 第 14 章 表分区与索引分区

324

本章介绍了表分区和索引分区及其在数据库性能优化方面的作用，并通过实例介绍了Oracle 11g中对索引技术最新的增强功能。

14.1	分区技术概述	324
14.2	表分区的方法说明及实例	325
14.2.1	范围分区 ( Range )	325
14.2.2	散列 ( Hash ) 分区	327
14.2.3	列表分区 ( List )	328

14.2.4	组合分区	328
14.2.5	Interval 分区	329
14.2.6	外键分区	330
14.2.7	虚拟列分区	331
14.3	表的分区策略	332
14.4	管理表分区	332
14.4.1	表分区管理的操作列表	332
14.4.2	表分区管理的操作实例	333
14.5	分区表联机重定义	336
14.6	创建索引分区	338
14.6.1	索引分区概念	338
14.6.2	本地索引分区	338
14.6.3	全局索引分区	340
14.7	管理索引分区	340
14.7.1	索引分区管理的操作列表	341
14.7.2	索引分区管理的操作实例	341
14.8	查看分区信息	341
14.9	本章小结	342

## 第 15 章 用户管理与安全

343

本章介绍了Oracle系统中的用户管理、权限控制、角色、资源及数据库的审计等方面的内容。

15.1	创建与管理用户账户	343
15.1.1	配置身份验证	343
15.1.2	创建用户的语法	343
15.1.3	创建用户实例	344
15.1.4	修改用户语法与实例	345
15.1.5	删除用户	347
15.2	权限管理	347
15.2.1	权限概述	347
15.2.2	授予系统权限	347
15.2.3	系统权限收回	348
15.2.4	授予对象权限	349
15.2.5	对象权限回收	349
15.2.6	用户与权限查询	350
15.3	角色管理	350
15.3.1	角色概述	351
15.3.2	预定义的 Oracle 系统角色	351
15.3.3	角色创建与角色授权	353
15.3.4	角色生效/失效	355
15.3.5	角色与权限的查询	355
15.4	资源配置 PROFILE	356
15.4.1	PROFILE 概述	356



15.4.2	创建 PROFILE 语法	357
15.4.3	创建 PROFILE 实例	359
15.4.4	修改资源配置 PROFILE	361
15.4.5	显示资源配置 PROFILE 信息	362
15.5	数据库审计	362
15.5.1	审计概念	362
15.5.2	审计环境设置	362
15.5.3	登录审计	364
15.5.4	数据活动审计	364
15.5.5	对象审计	365
15.5.6	清除审计数据	365
15.5.7	查询审计信息	366
15.6	本章小结	366

## 第 16 章 数据完整性和数据约束 367

本章介绍了与数据库完整性和一致性有关的重要概念：事务、锁、约束等。

16.1	什么是事务	367
16.2	数据并行性和一致性	368
16.2.1	数据并行性和一致性概述	369
16.2.2	Oracle 如何管理数据并行和一致性	369
16.3	锁	372
16.3.1	锁的功能、分类及类型	373
16.3.2	表锁和事务锁	376
16.3.3	检测和解决锁冲突	376
16.3.4	实例分析	379
16.4	数据完整性	381
16.4.1	数据完整性概述	381
16.4.2	完整性约束	382
16.4.3	完整性约束类型	383
16.4.4	约束检验机制	387
16.4.5	约束状态	388
16.5	本章小结	390

## 第 17 章 Oracle 11g Enterprise Manager简介 391

Oracle Enterprise Manager是一个功能全面的Oracle图形界面管理工具，自从推出到现在已经有多个版本了，每个新版本都比前一个版本功能更强。目前Oracle 11g企业管理器（简称EM，业内常叫OEM）是初学者管理Oracle系统的最佳伴侣，本章简单介绍了企业管理器的安装配置和使用方法。

17.1	Oracle 11g EM 简单介绍	391
17.2	Oracle 11g EM 安装配置	392
17.2.1	安装选择说明	392
17.2.2	客户化安装	392
17.2.3	设置与安装数据库控制	393

17.2.4	启动 Oracle 11g EM .....	393
17.2.5	登录到 Oracle 11g EM .....	394
17.3	用 Oracle 11g EM 管理 Oracle 系统 .....	395
17.3.1	数据库配置管理 .....	396
17.3.2	方案管理 .....	396
17.3.3	数据文件管理 .....	397
17.3.4	数据库用户管理 .....	398
17.4	用 EM 维护 Oracle 系统 .....	400
17.4.1	执行导出操作 .....	400
17.4.2	管理 (本地) 表空间 .....	403
17.4.3	浏览预警文件信息 .....	404
17.5	用 Oracle 11g EM 进行性能调整 .....	405
17.5.1	查看顶级活动 .....	405
17.5.2	监视 Oracle 实例是否存在锁 .....	406
17.6	本章小结 .....	406

## 第五部分 Oracle 数据库优化

### 第 18 章 Oracle 系统调整

408

本章主要介绍了与 Oracle 性能优化有关的初始化参数的设置及内存区域的调整。Oracle 性能优化包括的内容和技巧很多，限于篇幅这里只做简单的介绍。

18.1	Oracle 初始化参数文件 .....	408
18.1.1	Oracle 的系统参数 .....	408
18.1.2	初始化参数文件 SPFILE .....	408
18.1.3	Oracle 11g 新增加和淘汰参数 .....	409
18.2	Oracle 动态视图 .....	409
18.3	Oracle SGA 调整 .....	412
18.3.1	理解内存分配 .....	412
18.3.2	调整日志缓冲区分配 .....	412
18.3.3	调整共享池分配 .....	414
18.3.4	调整数据库缓冲区 .....	417
18.3.5	SGA 调整建议 .....	417
18.4	排序区调整 .....	417
18.4.1	排序区与其他内存区的关系 .....	418
18.4.2	理解排序活动 .....	418
18.4.3	监视排序活动 .....	419
18.4.4	专用模式下排序区的调整 .....	419
18.4.5	共享模式下排序区的调整 .....	421
18.5	Oracle 11g 中的自动内存优化功能 .....	422
18.6	本章小结 .....	422

### 第 19 章 Oracle SQL 语句优化

423

SQL的优化主要与数据库开发人员及应用程序开发人员能否写出高效的SQL语句有关系。其实大多数性能问题往往都与SQL效率低下有很大的关系。本章介绍了编写SQL语句的一些最基本的技巧和需要注意的问题，以及SQL优化器及Oracle 11g中一些SQL优化工具。

19.1	常用的 SQL 技巧	423
19.1.1	不要用“*”代替所有列名	423
19.1.2	用 TRUNCATE 代替 DELETE	423
19.1.3	在确保完整性的情况下多用 COMMIT 语句	424
19.1.4	尽量减少表的查询次数	424
19.1.5	用 NOT EXISTS 替代 NOT IN	425
19.1.6	用 EXISTS 替代 IN	425
19.1.7	用 EXISTS 替代 DISTINCT	425
19.1.8	有效利用共享游标	425
19.1.9	以合理的方式使用函数	426
19.2	表的连接方法	427
19.2.1	选择 FROM 表的顺序	427
19.2.2	驱动表的选择	428
19.2.3	WHERE 子句的连接顺序	428
19.3	有效使用索引	428
19.3.1	何时使用索引	429
19.3.2	索引列和表达式的选择	429
19.3.3	选择复合索引主列	429
19.3.4	避免对大表的全表扫描	430
19.3.5	监视索引是否被使用	430
19.3.6	影响索引有效使用的因素	431
19.4	Oracle 的优化器与执行计划	431
19.4.1	优化器概念	431
19.4.2	运行 EXPLAIN PLAN	432
19.4.3	Oracle 11g 中 SQL 执行计划的管理	432
19.5	Oracle 11g 中的执行计划管理	435
19.5.1	执行计划管理的工作原理	435
19.5.2	执行计划管理的实例测试	437
19.6	Oracle 11g 中的数据库重演和 SQL 重演	439
19.6.1	数据库重演	439
19.6.2	SQL 重演	439
19.7	Oracle 的性能顾问	440
19.7.1	Advisor Central	440
19.7.2	SQL 调优顾问	440
19.7.3	SQL 访问顾问	440
19.8	本章小结	441

## 第六部分 Oracle 11g 数据库的备份与恢复

本章介绍了数据库备份与恢复的概念、分类、方法等基本常识。读者通过本章的学习，能够对数据库备份和恢复有一个总体的了解，同时还介绍了 Oracle 11g 中备份和恢复的最新特性。

20.1	备份与恢复概述	444
20.2	备份分类与恢复分类	445
20.2.1	备份分类	445
20.2.2	恢复分类	446
20.3	备份与恢复方法	447
20.4	Oracle 11g 的备份和恢复特性	448
20.5	本章小结	449

本章介绍了 Oracle 推荐的备份工具 RMAN，包括 RMAN 的基本配置、恢复目录的介绍、通道分配，以及相关的备份和恢复命令、各种不同的备份方法等。RMAN 涉及的知识点很多，希望大家多做一些备份和恢复的试验，加深对 RMAN 的理解。

21.1	RMAN 简介	450
21.2	RMAN 资料档案库/恢复目录/控制文件	451
21.3	创建恢复目录	451
21.4	RMAN 与数据库的连接	453
21.4.1	与目标数据库的连接	453
21.4.2	与恢复目录的连接	453
21.5	注册数据库	454
21.5.1	注册目标数据库	454
21.5.2	实例演示	454
21.6	通道分配	455
21.6.1	通道概述	455
21.6.2	RUN 命令介绍	455
21.6.3	自动通道配置	456
21.6.4	手动通道配置	457
21.6.5	显示通道配置参数	458
21.6.6	设置通道控制参数	458
21.7	使用 BACKUP 命令生成备份集	460
21.7.1	备份集与备份片	460
21.7.2	BACKUP 命令语法	460
21.7.3	备份文件的存储格式	461
21.7.4	BACKUP 备份实例	462
21.7.5	BACKUP 的冷备份与热备份	464
21.8	使用 COPY 与 BACK AS COPY 命令	464
21.8.1	COPY 命令语法	464
21.8.2	COPY 备份实例	465
21.8.3	BACKUP AS COPY 命令	466
21.9	备份	467

21.9.1	压缩备份	467
21.9.2	完全备份与增量备份	467
21.9.3	查看备份信息	469
21.10	使用 RMAN 恢复数据库	470
21.10.1	RESTORE 命令	470
21.10.2	RECOVER 命令	470
21.10.3	实例解析	470
21.11	本章小结	472

## 第 22 章 Oracle 闪回技术 ( Flashback ) 473

本章主要讲述了数据库的闪回技术，闪回是数据库进行逻辑恢复的一个快捷工具。本章对 Oracle 11g 在闪回方面的最新技术，闪回数据归档也进行了详细介绍。

22.1	闪回技术概述	473
22.2	闪回恢复区	474
22.2.1	闪回恢复区的作用	474
22.2.2	配置闪回恢复区	474
22.2.3	闪回恢复区的文件保留策略	476
22.2.4	使用闪回恢复区	476
22.2.5	与闪回恢复区有关的视图	477
22.3	闪回数据库	477
22.3.1	闪回数据库概述	477
22.3.2	配置闪回数据库	478
22.3.3	使用闪回数据库	480
22.3.4	与闪回数据库有关的视图	481
22.4	闪回表	481
22.4.1	闪回表概念	481
22.4.2	使用闪回表	483
22.5	闪回丢弃	487
22.5.1	回收站概念	487
22.5.2	使用回收站	487
22.5.3	回收站与空间利用	489
22.6	闪回版本查询	491
22.6.1	闪回版本查询概念	491
22.6.2	使用闪回版本查询	491
22.7	闪回事务查询	493
22.7.1	闪回事务查询概念	493
22.7.2	使用闪回事务查询	494
22.8	闪回数据归档	495
22.8.1	闪回数据归档概念	495
22.8.2	闪回数据归档区	496
22.8.3	使用闪回数据归档	497
22.8.4	清除闪回数据归档区数据	500

22.8.5 与闪回数据归档有关的视图.....	500
22.9 本章小结.....	501

## 第 23 章 使用OEM向导备份和恢复数据库 502

本章介绍了如何使用OEM备份和恢复向导来备份和恢复数据库。对于数据库初学者而言，使用OEM备份和恢复向导会使操作更加直观。

23.1 设置首选身份证明.....	502
23.2 配置备份设置.....	505
23.3 调度备份.....	507
23.4 管理当前备份.....	509
23.5 配置恢复设置.....	511
23.6 执行恢复.....	512
23.7 本章小结.....	513

## 第 24 章 逻辑备份及数据导入/导出 514

本章着重介绍了在数据导入/导出时常用的数据泵技术EXPDP/IMPDP，以及传统的EXP/IMP导入/导出工具。

24.1 数据泵技术概述.....	514
24.2 创建目录对象.....	515
24.3 使用 Data Pump 导出数据.....	516
24.3.1 Data Pump 导出说明.....	516
24.3.2 基于命令行数据泵导出实例.....	516
24.3.3 基于 DBMS_DATAPUMP 数据泵导出实例.....	518
24.4 使用 Data Pump 导入数据.....	519
24.4.1 Data Pump 导入说明.....	520
24.4.2 基于命令行数据泵导入实例.....	521
24.4.3 基于 DBMS_DATAPUMP 数据泵导入实例.....	521
24.5 监控 Data Pump 作业进度.....	522
24.6 EXP/IMP 导出/导入数据.....	523
24.6.1 EXP/IMP 概要说明.....	523
24.6.2 EXP 导出实例.....	523
24.6.3 IMP 导入实例.....	524
24.6.4 字符集冲突问题.....	525
24.7 本章小结.....	526

# 第七部分 Oracle 11g 数据库集群技术与高可用性

## 第 25 章 Oracle RAC应用 528

本章主要讲述了RAC的基本概念、系统架构，以及安装、运行、管理RAC所需要的工具，并且通过具体的实例演示了RAC集群数据库的安装。

25.1 RAC 概述.....	528
25.2 Oracle 集群件.....	528

25.3	RAC 硬件架构	529
25.4	文件系统和卷管理	529
25.5	OCR 与投票磁盘	530
25.6	虚拟 IP 地址 (VIP)	530
25.7	集群校验工具	530
25.8	扩展的远距离集群上的 RAC	530
25.9	系统扩展与并行	531
25.10	管理集群数据库	531
25.10.1	企业管理器	531
25.10.2	以滚动的方式给应用打补丁	532
25.11	集群 11g R2 增强	532
25.12	RAC 的安装与配置	534
25.12.1	共享存储方案说明	534
25.12.2	软件包检查	535
25.12.3	创建用户和组及相关配置	536
25.12.4	设置主机名称和网关	538
25.12.5	Linux 系统参数配置	539
25.12.6	安装并配置 ASM 驱动	541
25.12.7	安装 cvuqdisk 软件包	543
25.12.8	安装 grid infrastructure	543
25.12.9	创建 ASM 磁盘组	555
25.13	安装 Oracle 数据库软件	558
25.14	使用 DBCA 创建数据库	563
25.15	TNS 配置	569
25.16	检查 RAC 集群数据库的状态	571
25.17	使用 Direct NFS Client	574
25.18	Oracle RAC 日常维护	574
25.19	本章小结	575

## 第 26 章 自动存储管理 ( ASM )

576

本章主要介绍了 Oracle 11g 数据库在 ASM 方面的新功能。

26.1	ASM 概述	576
26.2	Oracle 11g 中 ASM 都有哪些新特征	576
26.2.1	快速重新同步 ( ASM Fast Mirror Resync )	576
26.2.2	ASM 滚动升级	577
26.2.3	为 ASM 管理员新增了 SYSASM 权限和 OSASM 操作系统用户组	579
26.2.4	ASM 可扩展性和性能的增强	579
26.2.5	新的 ASM 命令行 ( ASMCMD ) 命令和选项	580
26.3	磁盘组兼容性方面新的属性	580
26.3.1	磁盘组兼容性概要	580
26.3.2	COMPATIBLE.ASM 和 COMPATIBLE.RDBMS	581
26.3.3	设置磁盘组兼容性的方法	581
26.4	ASM 优先镜像读取	582



26.4.1	优先镜像读取新特征概要	582
26.4.2	配置和管理优先读取失败组	582
26.5	ASM 快速再平衡	583
26.6	Oracle 自动存储管理集群文件系统 (ACFS)	583
26.6.1	Oracle ASM 动态卷管理器	584
26.6.2	Oracle ACFS 快照	585
26.6.3	Oracle ASM 配置助手 (ASMCA)	585
26.6.4	ASMCMD 增强	585
26.6.5	智能数据布局	586
26.6.6	为磁盘驱动器指定扇区大小	586
26.6.7	磁盘组更名	586
26.6.8	Oracle ASM 文件访问控制	586
26.6.9	Oracle ASM 上的 Oracle Cluster Registry 和 Voting 文件	586
26.6.10	Oracle Restart	586
26.6.11	支持 Oracle ACFS 的 Oracle 企业管理器	586
26.7	本章小结	587

## 第 27 章 容灾与数据卫士

588

本章主要介绍了 Oracle 11g 数据卫士的概述与入门，如何创建物理与逻辑备用数据库、日志传输服务、日志应用服务、角色管理，以及使用 RMAN 来备份和恢复文件，在基础知识之上涉及 Oracle 11g 的一些新的相关特性，如 Oracle 数据库企业版新的数据库选件 Oracle 动态数据卫士、实时查询功能以及压缩的日志传出，同时读取和恢复单个备用数据库，对生产数据库的报告、备份、测试和“滚动”升级。

27.1	Oracle 数据卫士概述	588
27.1.1	数据卫士简介	588
27.1.2	数据卫士体系结构	589
27.1.3	数据卫士配置	590
27.1.4	数据卫士服务	591
27.1.5	数据卫士代理	592
27.1.6	数据卫士保护模式	594
27.1.7	数据卫士进程结构	595
27.1.8	数据卫士功能	596
27.1.9	Oracle 数据卫士提供的好处	597
27.1.10	数据卫士和其他的相关技术	598
27.2	数据卫士入门	599
27.2.1	备用数据库类型	599
27.2.2	管理数据卫士配置的用户接口	602
27.2.3	数据卫士操作的前提条件	606
27.2.4	备用数据库目录结构	607
27.2.5	在线 Redo 日志、归档 Redo 日志和备用 Redo 日志	609
27.3	创建物理备用数据库	610
27.3.1	为备用数据库准备主数据库	610
27.3.2	实施创建操作	613

27.3.3	创建后的步骤	618
27.4	创建逻辑备用数据库	618
27.4.1	准备工作	618
27.4.2	实施创建操作	620
27.4.3	创建后的步骤	624
27.5	数据保护模式	624
27.5.1	数据卫士保护模式	624
27.5.2	设置主数据库的保护模式	625
27.6	日志传输服务	626
27.6.1	概述	626
27.6.2	配置 Redo 传输服务	627
27.6.3	级联 Redo 传输目的地	632
27.6.4	监控 Redo 传输服务	635
27.7	应用服务	639
27.7.1	概述	639
27.7.2	日志应用服务配置选项	642
27.7.3	将 Redo 数据应用到物理备用数据库	644
27.7.4	将 Redo 数据应用到逻辑备用数据库	645
27.8	角色转换	646
27.8.1	角色转换简介	646
27.8.2	包含物理备用数据库的角色转换	651
27.8.3	向逻辑备用数据库的角色转换	656
27.8.4	在故障转移后使用闪回技术	662
27.9	管理物理与快照备用数据库	665
27.9.1	启动与关闭物理备用数据库	665
27.9.2	打开物理备用数据库	666
27.9.3	在物理备用数据库上需要手工干预的主数据库变化	669
27.9.4	通过 OPEN RESETLOGS 语句进行还原	675
27.9.5	监控主、物理备用和快照备用数据库	676
27.9.6	优化 Redo 应用	678
27.9.7	管理快照备用数据库	678
27.10	管理逻辑备用数据库	679
27.10.1	概览 SQL 应用架构	680
27.10.2	控制用户访问逻辑备用数据库上的表格	683
27.10.3	管理与监控和逻辑备用数据库相关的视图	684
27.10.4	监控逻辑备用数据库	689
27.10.5	定制逻辑备用数据库	692
27.10.6	在逻辑备用数据库环境中管理特定工作	698
27.10.7	优化逻辑备用数据库	703
27.10.8	在逻辑备用数据库环境中备份与还原	708
27.11	使用 RMAN 来备份和恢复文件	709
27.11.1	关于数据卫士配置中的 RMAN 文件管理	710

27.11.2	关于数据卫士环境中的 RMAN 配置	711
27.11.3	推荐的 RMAN 与 Oracle 数据库配置	711
27.11.4	备份过程	714
27.11.5	在数据卫士环境下注册与注销数据库	717
27.11.6	数据卫士环境下的报告	717
27.11.7	数据卫士环境下的备份维护	717
27.11.8	数据卫士环境下的还原脚本	719
27.11.9	例外的备份环境	723
27.11.10	运用 RMAN 增量备份来滚动前移物理备用数据库	724
27.12	运用 SQL 应用升级 Oracle 数据库	726
27.12.1	运用 SQL 应用滚动升级的好处	727
27.12.2	运用 SQL 应用执行滚动升级的前提条件	727
27.12.3	升级操作中使用的数据与惯例	727
27.12.4	通过创建新逻辑备用数据库执行滚动升级	728
27.12.5	用已有的逻辑备用数据库执行滚动升级	730
27.12.6	用已有的物理备用数据库执行滚动升级	735
27.13	数据卫士案例分享	737
27.13.1	在故障转移之后配置逻辑备用数据库	737
27.13.2	用闪回数据库把失败的主数据库转换成备用数据库	739
27.13.3	在输入 OPEN RESETLOGS 语句之后使用闪回数据库	742
27.13.4	在指定 NOLOGGING 从句之后进行还原	744
27.13.5	创建使用 OMF 或 Oracle ASM 的备用数据库	746
27.13.6	在主数据库上从 Lost-write 错误进行还原	748
27.13.7	用 RMAN 备份把失败的主数据库转换成备用数据库	749
27.13.8	在不重建物理备用数据库的情况下变更主数据库的字符集	753
27.14	本章小结	753

## 第 28 章 故障诊断

754

本章主要介绍了故障诊断的一些方法和工具，并提出了相关建议。

28.1	数据库挂死管理器	754
28.1.1	IPS 事件打包服务	754
28.1.2	自动诊断知识库 (ADR)	755
28.1.3	故障处理的工作流	756
28.1.4	增强网络诊断	756
28.1.5	增强 OCI 诊断	756
28.1.6	智能解决方案	756
28.2	SQL 修复建议器	757
28.3	问题避免	757
28.3.1	自动健康监控	757
28.3.2	健康监控	758
28.4	支撑平台	759
28.5	如何得到故障信息	759

28.6 企业管理器 Support Workbench 支持 ASM	760
28.7 本章小结	760

## 第 29 章 真正应用测试 761

本章主要介绍了 Oracle 11g 新增功能——真正应用测试的原理和步骤，包括数据库重放和 SQL 性能分析器。

29.1 数据库重放	761
29.2 SQL 性能分析器	768
29.2.1 SQL 性能分析器方法论	769
29.2.2 运行 SQL 性能分析器	770
29.3 本章小结	772

# 第八部分 商业智能与数据仓库

## 第 30 章 ETL 相关的功能 774

本章主要介绍了 ETL 的基础概念，并描述了与 ETL 相关的几个 Oracle 特性，包括变化数据捕捉、物化视图、查询重写、OWB 等。

30.1 概 述	774
30.2 变化数据捕捉	774
30.2.1 发布变化数据	778
30.2.2 订阅变化数据	785
30.2.3 Oracle 11g 变化数据捕捉功能增强	787
30.3 物化视图	788
30.4 查询重写	790
30.4.1 概述	790
30.4.2 查询重写的先决条件	790
30.4.3 保证查询重写有效	791
30.4.4 与查询重写相关的初始化参数	791
30.4.5 控制查询重写	792
30.4.6 查询重写的准确性	792
30.4.7 激活查询重写的权限	793
30.4.8 示例方案和物化视图	793
30.4.9 如何验证发生过查询重写	794
30.4.10 查询重写示例	795
30.4.11 内嵌视图	795
30.4.12 远程表	796
30.5 Oracle Warehouse Builder	796
30.5.1 OWB 方法论	797
30.5.2 OWB 的架构图	800
30.5.3 安装与配置	801
30.5.4 数据转换功能	802
30.5.5 缓慢变化维	803
30.6 与数据仓库有关的 SQL 操作增强	803

## 第 31 章 多维数据库

本章主要介绍了Oracle OLAP多维数据库的基础知识，包括OLAP的访问和创建，并且详细说明了Oracle OLAP的管理工具AWM。本章最后介绍了Oracle 11g中OLAP新特性。

31.1	OLAP 选件概述.....	805
31.2	OLAP 对外接口.....	806
31.2.1	Java OLAP API.....	807
31.2.2	关系视图与 SQL 查询.....	808
31.2.3	OLAP DML 与 PL/SQL 包.....	808
31.3	分析工作空间管理器.....	809
31.3.1	简化维模型的构建.....	810
31.3.2	实施物理存储模型.....	810
31.3.3	映射关系源.....	812
31.3.4	管理生命周期.....	813
31.3.5	保存模型设计.....	813
31.4	分析工作区管理器演示.....	813
31.4.1	授权.....	814
31.4.2	创建分析工作区.....	814
31.4.3	创建维.....	815
31.4.4	创建多维数据集.....	818
31.4.5	加载和聚合数据.....	820
31.5	11gr1 OLAP 增强功能.....	821
31.5.1	SQL 优化器的增强.....	821
31.5.2	允许利用 Java 插件.....	821
31.5.3	分区和存储顾问 ( Advisors ).....	821
31.5.4	基于成本的汇总.....	822
31.5.5	立方脚本.....	822
31.5.6	数据库管理的立方刷新.....	823
31.5.7	自动重写到按照立方组织的物化视图.....	825
31.5.8	数据字典.....	825
31.5.9	计算模板.....	825
31.5.10	安全增强.....	825
31.6	本章小结.....	827

## 第 32 章 数据挖掘

本章主要介绍了Oracle数据挖掘的基础知识，包括数据挖掘方法论和Oracle数据库提供的主要挖掘算法，并介绍了Oracle 11g中数据挖掘的新特性。

32.1	数据挖掘过程.....	828
32.2	监督式学习算法.....	830
32.2.1	朴素贝叶斯.....	830
32.2.2	自适应贝叶斯网络.....	830

32.2.3	支持向量机	831
32.2.4	属性重要性排序	831
32.2.5	决策树	831
32.3	无监督式学习算法	831
32.3.1	聚类	832
32.3.2	关联规则	832
32.3.3	特征选择	832
32.3.4	文本挖掘和非结构化数据	833
32.4	数据挖掘接口	833
32.5	数据挖掘功能增强	834
32.5.1	自动和嵌入的数据转换	834
32.5.2	数据挖掘方案对象	836
32.5.3	数据挖掘的 Java API	837
32.5.4	广义线性模型	837
32.5.5	预测分析：PROFILE	837
32.5.6	SQL 预测增强	837
32.5.7	新增视图	838
32.5.8	安全性	839
32.5.9	数据挖掘模型的升级	839
32.5.10	11g 中不赞成使用的特性	839
32.5.11	不被支持的特性	840
32.6	本章小结	840

## 第九部分 非结构化数据库

### 第 33 章 空间数据库

842

本章主要介绍了 Oracle 空间数据库的基础知识，包括矢量几何体数据和栅格数据的存取方法，并介绍了 Oracle 11g 中空间数据库的新特性。

33.1	使空间数据库概述	842
33.2	几何实体数据	843
33.2.1	几何实体类型	843
33.2.2	空间数据模型	843
33.2.3	SDO_GEOMETRY	844
33.2.4	几何实体的元数据	850
33.2.5	空间 R 树索引	851
33.2.6	过滤和空间关系	852
33.2.7	空间操作符	854
33.2.8	SDO_GEOM 包	856
33.2.9	空间聚集函数	857
33.2.10	空间参考坐标系	858
33.2.11	其他	863
33.3	Oracle Spatial 选项	863

33.4	GeoRaster 数据	864
33.5	Map Viewer	867
33.6	空间数据库 11gR1 增强特性	868
33.7	空间数据操作演示	870
33.8	本章小结	873

## 第 34 章 XML 数据库

874

本章主要介绍了 Oracle XML 数据库的基础知识，包括 XMLType 的存储方式和创建方法，并介绍了常用的 XML 存取操作，用示例说明了 XML DB 的开发。

34.1	Oracle XML DB 概述	874
34.1.1	XMLType 存储	874
34.1.2	XML DB 资料库	875
34.1.3	协议结构	876
34.1.4	API 访问	876
34.2	XML DB 开发	877
34.2.1	XMLType 类型	877
34.2.2	存储索引	880
34.2.3	XML 模式	880
34.2.4	注册 XML 模式	881
34.2.5	XML 模式进化	883
34.2.6	XML DB 应用开发工具	884
34.3	XML 存取操作	884
34.3.1	XQuery	884
34.3.2	SQL/XML 二元性	886
34.3.3	XML 与关系数据互操作	887
34.3.4	XML 运算符	887
34.3.5	XML DB 维护工具	888
34.3.6	XML DB 全文检索	889
34.3.7	XML DB 11gR1 增强	889
34.4	Oracle XML DB 示例	892
34.5	本章小结	896

## 第十部分 Oracle 11g 数据库的其他新特性

### 第 35 章 Oracle 11g 应用增强

898

本章主要介绍了 Oracle 11g 中一些主要的新增应用特性，包括结果缓存、执行计划管理、高级压缩和 SQL 方面的一些增强特性。

35.1	结果缓存	898
35.1.1	概述	898
35.1.2	应用演示	900
35.2	执行计划管理	907



35.2.1	概述	907
35.2.2	应用演示	909
35.3	高级压缩	910
35.3.1	概述	910
35.3.2	11g 中新压缩特性	911
35.4	数据库控制	914
35.4.1	高级复制变革	915
35.4.2	ASM 增强	915
35.4.3	转变管理者-同步与传播	915
35.4.4	增强数据库克隆	915
35.4.5	增强数据库配置 ( ECM 收集 )	915
35.4.6	改进数据库主页和性能页	915
35.4.7	增强的自适应测度 ( metric ) 阈值	916
35.4.8	融合需求 ( Fusion )	917
35.4.9	管理 Oracle 文本索引	917
35.4.10	迁移数据库到 ASM——企业管理器中的增强	920
35.4.11	增强存储/审计报表和测度	920
35.4.12	增强存储、计划、安全和配置	920
35.4.13	增强等待事件详述	921
35.4.14	工作空间管理	921
35.5	整体数据库管理——真正应用集群的 ADDM	921
35.6	LOBs 增强	922
35.7	SQL 增强	924
35.7.1	SQL 指令	924
35.7.2	函数	941
35.7.3	其他增强	944
35.8	本章小结	945

## 第 36 章 高级数据管理 946

本章主要介绍了 Oracle 11g 增强的一些数据库管理和维护特性，并介绍了新增的信息生命周期管理的概念和使用方法。

36.1	增强高级管理特性	946
36.2	信息生命周期管理	949
36.2.1	信息生命周期管理概念	949
36.2.2	Oracle 信息生命周期管理方案	950
36.2.3	信息生命周期管理实施	951
36.2.4	信息生命周期管理助手	953
36.2.5	在线数据归档的好处	957
36.3	本章小结	958

## 第 37 章 数据库升级 959

本章主要介绍了 Oracle 数据库升级为 11g 的方法和步骤，并说明了升级完成后的任务。

37.1 概述.....	959
37.2 升级过程.....	960
37.3 11g新增的后台进程.....	961
37.4 升级准备.....	962
37.5 升级到新版本.....	963
37.6 升级后的任务.....	964
37.7 11gr1兼容性和互操作性问题.....	964
37.8 数据库降级.....	971
37.9 本章小结.....	972

## 参考文献

973

# 第 22 章 Oracle 闪回技术 ( Flashback )

本章介绍与数据库闪回相关的知识，闪回技术从 Oracle 9i 版本的闪回查询开始，在 Oracle 10g 版本中系统使用，在 Oracle 11g 中进行了增强。

## 22.1 闪回技术概述

为了使 Oracle 数据库能够从任何的逻辑误操作中迅速恢复，Oracle 推出了闪回技术。该技术首先以闪回查询 (Flashback Query) 出现在 Oracle 9i 版本中，后来在 Oracle 10g 中对该技术进行了全面扩展，提供了闪回数据库、闪回删除、闪回表、闪回事务及闪回版本查询等功能，在 Oracle 11g 中，Oracle 继续对该技术进行了改进和增强，增加了闪回数据归档功能。

在 Oracle 11g 中，闪回技术包括以下各项。

- ◎ 闪回数据库 (Flashback Database): 该特性允许用户通过 `flashback database` 语句，使数据库迅速回滚到以前的某个时间点或者某个 SCN (系统更改号) 上，而不需要进行时间点的恢复操作。该功能并不基于撤销数据 (undodata)，而是基于闪回日志。
- ◎ 闪回丢弃 (Flashback Drop): 类似于操作系统的垃圾回收站，可以从中恢复被 `drop` 的表或索引。该功能基于撤销数据 (undodata)。
- ◎ 闪回版本查询 (Flashback Version Query): 通过该功能，可以看到特定的表在某个时间段内所进行的任何修改操作，如同电影回放一般，表在该时间段内的变化一览无余。该功能基于撤销数据 (undodata)。
- ◎ 闪回事务查询 (Flashback Transaction Query): 使用该特性，可以在事务级别上检查数据库的任何改变，大大方便了对数据库的性能优化、事务审计及错误诊断等操作。该功能基于撤销数据 (undodata)。
- ◎ 闪回表 (Flashback Table): 使用该特性，可以确保数据库表能够被恢复到之前的某一个时间点上。注意，该功能与最早的 Oracle 9i 中的 Flashback Query 不同，Flashback Query 仅是得到了表在之前某个时间点上的快照而已，并不改变当前表的状态；而 Flashback Table 却能够将表及附属对象一起回到以前的某个时间点。该功能基于撤销数据 (undodata)。
- ◎ 闪回数据归档 (Flashback Data Archive): 通过 Flashback Data Archive 可以查询指定对象的任何时间点 (只要满足保护策略) 的数据，而且不需要利用到 `undo`，在有审计需要的环境，或者是安全性特别重要的高可用数据库中，是一个非常好的特性。缺点是如果该表变化很频繁，

则对空间的要求可能很高。Flashback Data Archive 是对对象的保护，是 Flashback Database 的一个强力补充。

## 22.2 闪回恢复区

与闪回技术密切相关的是闪回恢复区，下面对闪回恢复区进行详细介绍。

### 22.2.1 闪回恢复区的作用

闪回恢复区在 Oracle 10g 中首次出现，是为了支持 Oracle 数据库的一个新的功能：自动基于磁盘的备份和恢复（Automatic Disk-Based Backup and Recovery）。简单地说，闪回恢复区是一块用来集中存储所有与数据库恢复相关文件的存储空间区域。

闪回恢复区可以放在以下几种存储形式上。

- ◎ 目录
- ◎ 文件系统
- ◎ 自动存储管理（ASM）磁盘组

需要注意的是，如果在 RAC 环境中，该位置必须为集群文件系统（Cluster File System）ASM 磁盘组或是通过 NFS 控制的文件共享目录。此外，所有实例的位置和操作系统的磁盘限额（Disk Quota）必须一致。

以下几种文件可以放到闪回恢复区中。

- ◎ 控制文件；
- ◎ 归档的日志文件；
- ◎ 闪回日志；
- ◎ 控制文件和 SPFILE 自动备份；
- ◎ 通过 RMAN 的 BACKUP 命令产生的备份集（Backup sets）；
- ◎ 通过 RMAN 的 COPY 或 BACKUP AS COPY 命令产生的图像副本。

闪回恢复区为数据恢复提供了一个集中化的存储区域，这在很大程度上减小了管理开销。另外，随着硬盘存储容量的增大，读写的速度越来越快，使自动基于磁盘备份与恢复技术实现成为可能。而闪回恢复区正是磁盘备份与恢复的基础。当然，在实际应用中，往往都是将闪回恢复区与 OMF、ASM 结合起来运用，达到满意的效果。

### 22.2.2 配置闪回恢复区

配置闪回恢复区是一个很简单的过程。只要在初始化参数文件中指定恢复区的位置和大小即可。下面两种情况下都可以进行指定。



(1) 使用 DBCA 创建数据库的过程中, 会有专门的页面指定闪回恢复区的位置和大小, 如图 22-1 所示。



图 22-1 选择数据的恢复选项界面

(2) 如果在创建数据库时没有指定快速闪回区 (即没有启动闪回恢复区), 则可以在数据库创建完成后, 通过修改以下两个初始化参数设定闪回恢复区。

- ◎ DB\_RECOVERY\_FILE\_DEST
- ◎ DB\_RECOVERY\_FILE\_DEST\_SIZE

这两个参数分别用来指定闪回恢复区的位置与闪回恢复区的大小 (默认值为空)。

下面通过实例具体说明, 注意在设定这两个动态初始化参数时, 不需要重启实例。

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=2g SCOPE=BOTH;
SQL> ALTER SYSTEM SET db_recovery_file_dest='/u01/app/Oracle/flash_recovery_area'
  2 SCOPE=BOTH;
```

当然, 也可以通过以下命令查看修改后的参数是否生效。

```
SQL> SHOW parameter db_recovery_file_dest
NAME                                TYPE                                VALUE
-----                                -
db_recovery_file_dest                string                              /u01/app/Oracle/flash_recovery_area
db_recovery_file_dest_size            big integer                          2G
```

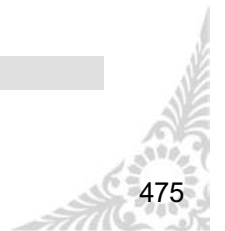
还可以用专门的命令修改闪回恢复区的大小, 以及停用闪回恢复区。这些命令都使用 ALTER SYSTEM 语句执行, 具体如下。

(1) 将闪回恢复区的大小设置为 4GB。

```
SQL> ALTER SYSTEM SET db_recovery_file_dest_size=4g SCOPE=BOTH;
```

(2) 要停用闪回恢复区, 只要将参数 db\_recovery\_file\_dest 置空即可。

```
SQL>ALTER SYSTEM SET db_recovery_file_dest='';
```



### 22.2.3 闪回恢复区的文件保留策略

闪回恢复区是一块集中存储所有与数据库恢复相关文件的存储空间区域，其大小和位置在数据库参数中定义。当文件被增加到闪回恢复区域或从闪回恢复区域被删除时，相关信息会自动记录到数据库的警告日志中。

文件在闪回恢复区中保留的时间长度是由 RMAN 的保留策略决定的，RMAN 通过 RMAN CONFIGURE RETENTION POLICY 命令规定保留备份的天数。超过 RMAN 保留策略或废弃的文件将从闪回恢复区中删除。复合 RMAN 保留策略的文件也不会被删除，而将被用于重建。

- ⊙ 持久文件会被删除；
- ⊙ 通过 RMAN 配置策略过期的文件会被删除；
- ⊙ 已经拷贝到磁带上的临时性文件会被删除；
- ⊙ 在 Data Guard 环境中，当已归档的重做日志文件可以从闪回恢复区中删除时，可以应用归档的重做日志删除策略进行自动删除（在恢复管理器中 CONFIGURE ARCHIVELOG DELETION POLICY TO...）。

当闪回恢复区中的空间率使用率超过 85% 时，数据库会向 alert 文件中写入告警信息。而当超过 97% 时则会写入严重告警信息。当闪回恢复区空间全部耗尽时，Oracle 将报告如下错误。

```
ORA-19809: limit exceeded for recovery files
ORA-19804: cannot reclaim 52428800 bytes disk space from 1258291200 limit
```

此时查询视图 dba\_outstanding\_alerts，将会给出错误的原因及操作建议，如图 22-2 所示。

```
SQL> select reason,object_type,suggested_action from dba_outstanding_alerts;
```

REASON	OBJECT_TYPE	SUGGESTED_ACTION
db_recovery_file_dest_size of 1258291200 bytes is 88.20% used and has 148509184 remaining bytes available.	RECOVERY AREA	Add disk space and increase db_recovery_file_dest_size, backup files to tertiary device, delete files from recovery area using RMAN, consider changing RMAN retention policy or consider changing RMAN archive log deletion policy.

```
1 rows selected.
SQL>
```

图 22-2 给出错误原因代码

### 22.2.4 使用闪回恢复区

如前面所介绍的，闪回恢复区是一块用来集中存储所有与数据库恢复相关文件的存储空间区域，这在很大程度上减少了对备份文件管理的开销。事实上，在 Oracle 的早期版本中，归档日志文件可能无法与备份同步，此时需要手工进行删除。而闪回恢复区的一个特性是，这些文件和数据文件及控制文件相关联，这样就避免了手工清除不必要的归档日志带来的麻烦。

闪回恢复区设置完成后，备份的过程就非常简单，首先需要规定闪回恢复区是使用 CONFIGURE 备份的默认配置。然后，执行 BACKUP 或 BACKUP AS COPY 命令，备份就会直接送往闪回恢复区中进行集中存储。





用户可以通过动态视图 `V$RECOVERY_FILE_DEST` 查看闪回恢复区消耗的空间及其他统计信息。

下面是一个数据库备份的例子，读者可以在自己的试验环境中运行，查看相关的结果，这里仅给出命令和步骤。

```
C:\Documents and Settings\Administrator>rman
RMAN>connect target; //连接到目标数据库
RMAN>BACKUP AS COPY DATABASE; //完成数据库的图像副本备份
.....//观察结果，图像副本是否存放到闪回区域中
```

备份完成后，查询 `V$RECOVERY_FILE_DEST` 视图，确定 `SPACE_USED` 列值是否增加。

```
SQL>select * from v$recovery_file_dest;
... ..
```

### 22.2.5 与闪回恢复区有关的视图

与闪回恢复区有关的视图有以下几种：

- ◎ `DBA_OUTSTANDING_ALERTS`，通过该视图可以得到相关告警信息；
- ◎ `V$RECOVERY_FILE_DEST`，通过该视图可以监控恢复区的使用空间；
- ◎ `V$FLASH_RECOVERY_AREA_USAGE`，与文件类型及空间使用有关的视图；
- ◎ 此外，`V$CONTROLFILE`、`V$LOGFILE`、`V$ARCHIVED_LOG`、`V$DATAFILE_COPY` 等视图都增加了新的列 `IS_RECOVERY_DEST_FILE`，指明相关的文件是否在恢复区内。

## 22.3 闪回数据库

设置闪回恢复区后，就可以配置和启用闪回数据库的功能。本节将详细介绍闪回数据库的配置和使用。

### 22.3.1 闪回数据库概述

闪回数据库能够使数据库迅速回滚到以前的某个时间点或某个 SCN（系统更改号）上，这对于数据库从逻辑错误中恢复特别有用，而且也是大多数逻辑损害时恢复数据库的最佳选择。该功能不基于撤销数据（`undodata`），而是基于闪回日志。

Oracle 系统为使用数据库的闪回功能，特别创建了另外一组日志，即 `Flashback_logs`（闪回日志），记录数据库的闪回操作。

使用闪回数据库恢复比使用传统的恢复方法要快得多，这是因为恢复不再受数据库大小的影响。也就是说，传统的恢复时间（`MTTR`）是由所需重建的数据文件的大小和所要应用的归档日志的大小所决定的，而使用闪回数据库恢复，其恢复时间是由恢复过程中需要备份的变化的数量，而不是数据文件和归档日志的大小决定。

闪回数据库由恢复写入器（`RVWR`）后台进程和闪回数据库日志组成。如果要启用闪回数据库功能，`RVWR` 进程也要启动。



闪回数据库日志是一种新的日志文件类型，包括物理数据块先前的“图像”。

闪回恢复区是闪回数据库的先决条件，由于 RVWR 进程要将闪回日志写入该区域中，所以在使用闪回数据库功能时，必须要启用该区。

对于逻辑损害和用户的误操作，闪回数据库是不完全恢复的优秀替代。但是必须指出的是，闪回数据库有其自身的局限性。

- ◎ 使用闪回数据库恢复不能解决媒介故障。若要从媒介故障中恢复，仍然需要重建数据文件和恢复归档日志文件；
- ◎ 截短数据文件（缩小数据文件到较小的尺寸），用闪回数据库不能恢复此类操作；
- ◎ 如果控制文件已被重建，不能使用闪回数据库；
- ◎ 不能完成删除表空间的恢复；
- ◎ 最多只能将数据库闪回到在闪回日志中最早可用的那个 SCN，并不能将数据库闪回到任意的 SCN 值。

### 22.3.2 配置闪回数据库

配置闪回恢复区后，要启用闪回数据库功能，还需要进行进一步的配置，需要注意以下几点。

- ◎ 配置闪回恢复区；
- ◎ 数据库需要运行在归档模式下（Archive log）；
- ◎ 通过数据库参数 DB\_FLASHBACK\_RETENTION\_TARGET，指定可以在多长时间内闪回数据库。该值以分钟为单位，默认为 1440（1 天），更大的值对应更大的闪回恢复空间，类似于闪回数据库的基线；
- ◎ 需要在 MOUNT 状态下使用 ALTER DATABASE FLASHBACK ON 命令启动闪回数据库功能。

下面通过具体的实例演示如何启动闪回数据库功能。

#### 1. 登录系统

```
SQL> conn /as sysdba;
Connected.
SQL> show parameter db_recovery_file_dest;
NAME                                TYPE                                VALUE
-----
db_recovery_file_dest                string                              /oratest/app/Oracle/flash_recovery_area
db_recovery_file_dest_size           big integer                          2G
SQL>

SQL> show parameter flashback;
NAME                                TYPE                                VALUE
-----
db_flashback_retention_target        integer                              1440
SQL>
```





## 2. 确认实例是否为归档模式 (需要设置在归档模式下运行)

```
SQL> archive log list;
Database log mode           No Archive Mode
Automatic archival         Disabled
Archive destination        USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 29
Current log sequence       31
SQL>
当前的数据库处于非归档模式下。
```

## 3. 将数据库改为在归档模式下运行, 并且打开 flashback 功能

```
SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup mount;
ORACLE instance started.
Total System Global Area  418484224 bytes
Fixed Size                 1300324 bytes
Variable Size             310380700 bytes
Database Buffers          100663296 bytes
Redo Buffers              6139904 bytes
Database mounted.
SQL> alter database archivelog;
Database altered.
```

## 4. 设置参数 DB\_FLASHBACK\_RETENTION\_TARGET 为希望的值, 单位为分钟, 本例设置为 3 天, $1440 \times 3 = 4320$

```
SQL> alter db_flash_back_retention_target=4320;
Database altered.
```

## 5. 启动闪回数据库, 将数据库置为 open 状态

```
SQL> alter database flashback on;
Database altered.
SQL> alter database open;
Database altered.
```

## 6. 查看更改后的参数

```
SQL> archive log list;
Database log mode           Archive Mode
Automatic archival         Enabled
Archive destination        USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 29
Next log sequence to archive 31
Current log sequence       31
SQL>
```

经过以上对数据库闪回功能的设置, Oracle 11g 的 FLASHBACK DATABASE 功能可自动搜集数据, 只要确保数据库是归档运行即可, 所归档区由 DB\_RECOVERY\_FILE\_DEST 参数指定。



## 22.3.3 使用闪回数据库

## 1. FLASHBACK DATABASE 语法说明

Oracle 11g 的 FLASHBACK 命令可以对表级进行恢复，也可以对数据库级进行恢复。要对数据库级进行恢复，就要用到 FLASHBACK DATABASE 命令，其语法如下：

```
FLASHBACK [STANDBY] DATABASE <database>
{ TO [ SCN | TIMESTAMP ] <exp> | TO BEFORE [ SCN | TIMESTAMP ] <exp> }
```

其中：

- ◎ STANDBY：指定恢复备用的数据库到某个 SCN 或某个时间点上。如果没有备用数据库，则系统会返回一个错误。如果省略 STANDBY 子项，则数据库可能是主数据库，也可能是备用数据库；
- ◎ TO SCN <exp>：指定一个系统改变号 SCN；
- ◎ TO BEFORE SCN <exp>：恢复到之前的 SCN；
- ◎ TO TIMESTATMP：需要恢复的时间表达式；
- ◎ TO BEFORE TIMESTAMP：恢复数据库到之前的时间表达式。

用户可以从 sysdate 得到系统的当前时间。

当用户发出 FLASHBACK DATABASE 语句后，数据库会首先检查所需要的归档文件与联机重做日志文件的可用性。如果可用，则会将数据库恢复到指定的 SCN 或时间点上。

在数据库中，闪回数据库的总数和大小由 DB\_FLASHBACK\_RETENTION\_TARGET 初始化参数控制，可通过查询 V\$FLASHBACK\_DATABASE\_LOG 视图确定能恢复到的时间点。

如果数据库中所保留的数据不能执行恢复，可使用标准的恢复过程恢复到过去的某个时间点上。

如果数据文件集没有保留足够的数据库，则数据库会返回一个错误，这种情况下，可先使数据文件脱机，然后再发出语句恢复剩余的部分，最后再用标准方法恢复这些脱机的数据文件。

## 2. 闪回数据库例子

FLASHBACK 语句既可以在 SQL>下使用，也可以在 RMAN>中使用，还可以在 EM 中使用，下面给出一个具体实例。

下面的例子是使用闪回数据库完成到某个 SCN 的恢复。

(1) 首先，查询 V\$FLASHBACK\_DATABASE\_LOG 视图以获得 OLDEST\_FLASHBACK\_SCN。

```
C:\Documents and Settings\Administrator>sqlplus /nolog
SQL*Plus: Release 11.1.0.6.0 - Production on 星期一 11月 19 07:12:38 2007
Copyright (c) 1982, 2007, Oracle. All rights reserved.
SQL> conn /as sysdba;
已连接。
```

```
SQL> select oldest_flashback_scn,oldest_flashback_time from v$flashback_database_log;
OLDEST_FLASHBACK_SCN OLDEST_FLASH
```



5540916

19-NOV-07

(2) 关闭数据库，并在 MOUNT 模式下启动数据库。

```
SQL> shutdown immediate;
数据库已经关闭。
已经卸载数据库。
ORACLE 例程已经关闭。
SQL> startup mount;
ORACLE 例程已经启动。
```

```
Total System Global Area      426852352 bytes
Fixed Size                      1333648 bytes
Variable Size                   327157360 bytes
Database Buffers                92274688 bytes
Redo Buffers                    6086656 bytes
数据库装载完毕。
SQL>
```

(3) 使用 FLASHBACK DATABASE 闪回数据库到 SCN 5540916。

```
SQL> flashback database to scn 5540916;
Flashback complete.
SQL>
```

(4) 最后，用 resetlogs 选项打开数据库。

```
SQL> alter database open resetlogs;
Database altered.
SQL>
```

### 22.3.4 与闪回数据库有关的视图

在监控闪回数据库时，常用的重要视图有以下几种。

- ◎ V\$DATABASE，用来显示闪回数据库是启动还是关闭，即闪回数据库是否被激活；
- ◎ V\$ FLASHBACK\_ DATABASE \_LOG，用来查看与闪回数据库有关的 SCN，TIME 及闪回数据库的时间，闪回数据的大小；
- ◎ V\$FLASHBACK\_ DATABASE \_STAT，显示闪回数据库日志的情况，用来估算闪回数据库潜在的需求空间。

## 22.4 闪回表

本节介绍 Oracle 11g 闪回技术中 Flashback Table 的概念和具体实例。

### 22.4.1 闪回表概念

闪回表是一种能够恢复表或设置表到过去某个特定的时间点而又不需要进行不完全恢复的闪回技术。使用闪回表时，所有的相关对象都能得到恢复。



首先要说明的是，Oracle 11g 的 Flashback Table 与 Oracle 9i 的 Flashback Query 不同，Oracle 9i 的 Flashback Query 只是查询以前的一个快照而已，并不改变当前表的状态，而 Oracle 11g 的 Flashback Table 将改变当前表及附属对象，使其一起闪回到以前的某个时间点上。

Oracle 11g 的 Flashback Table 有以下特性。

- ◎ 在线操作；
- ◎ 恢复到指定时间点或 SCN 的任何数据；
- ◎ 自动恢复相关属性，如索引、触发器等；
- ◎ 满足分布式的一致性；
- ◎ 满足数据一致性，所有相关对象将自动一致；
- ◎ 闪回表技术是基于撤销数据（undodata）来实现的，因此，要想闪回表到过去的某个时间点上，必须确保与撤销表空间有关的参数设置合理。与撤销表空间相关的参数有 `undo_management`、`undo_tablespace` 与 `undo_retention`。

下面的例子显示了与闪回表有关的撤销表空间参数的查询结果。

### 示例 1：与闪回有关的撤销表空间及参数查询。

```
SQL> conn /as sysdba;
Connected.
SQL> show parameter undo;
NAME                                TYPE                                VALUE
-----
undo_management                      string                              AUTO
undo_retention                       integer                             900
undo_tablespace                      string                              UNDOTBS1
SQL>

SQL> alter system set undo_retention=1800 scope=both;
System altered.

SQL> show parameter undo;
NAME                                TYPE                                VALUE
-----
undo_management                      string                              AUTO
undo_retention                       integer                             1800
undo_tablespace                      string                              UNDOTBS1
SQL>
```

从显示的参数来看，`undo_management` 表示系统采用撤销表空间记录增、删、改数据，而不是早期版本（如 Oracle 9i）的回滚段。`undo_tablespace` 表示撤销表空间是 UNDOTBS1；`undo_retention` 是一个关键的参数，表示所做的增、删、改操作被提交后，记录在撤销表空间内的增、删、改数据还保留一段时间（秒数），这里为 900 秒。

如果撤销表空间足够大，可以将增、删、改操作数据保留得长一些，如果发现某个操作是错误的，就可以使用 `FLASHBACK TABLE` 命令进行恢复。

一般在创建撤销表空间时，就要考虑数据保存时间的长短、每秒产生的块数据量及块的大小等。



Undo=UR \* UPS \* db\_block\_size +冗余量

UR 表示 undo 中保持的最长时间数 (秒), 由数据库参数 undo\_retention 决定。

UPS 表示在 undo 中每秒产生的数据块数量。例如: 在数据库中保留 2 小时的回退数据, 假定每小时产生 200 个数据库块, 数据库块大小为 8kb, 则:

Undo=2\*60\*60\*200\*8192=11GB

## 22.4.2 使用闪回表

为了使读者对 FLASHBACK TABLE 命令有一个初步的认识, 下面先给出语法介绍, 然后再举例说明。

### 1. 闪回表的语法介绍

```
FLASHBACK TABLE [schema.]<table_name>
TO
{
[ BEFORE DROP [RENAME TO table ] |
[ SCN | TIMESTAMP] expr [ENABLE | DISABLE] TRIGGERS
}
```

其中:

- ◎ Schema: 模式名, 一般为用户名;
- ◎ TO TIMESTAMP: 系统邮戳, 包含年月日时分秒;
- ◎ TO SCN: 系统更改号, 可从 flashback\_transaction\_query 数据字典中查到, 如: flashback table employe TO SCN 3698893;
- ◎ ENABLE TRIGGERS: 表示触发器恢复以后为 enable 状态, 而默认为 disable 状态;
- ◎ TO BEFORE DROP: 表示恢复到删除之前;
- ◎ Rename to table: 表示更换表名。

另外, 需要注意的是, 如果需要 FLASHBACK 一个表, 则需要保证:

- ◎ 具备 FLASHBACK ANY TABLE 的系统权限或是该表的 FLASHBACK 对象权限;
- ◎ 有该表的 SELECT、INSERT、DELETE、ALTER 权限;
- ◎ 必须保证该表 ROW MOVEMENT。

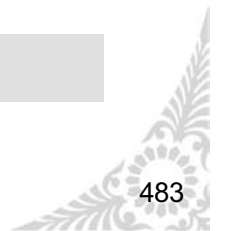
### 2. 闪回表使用的具体实例

从 FLASHBACK TABLE 命令的语法来看, 可恢复到之前的某个时间戳、SCN 号或之前的任何 DROP 动作。下面分别给出几种情况的恢复例子。

示例 2: 首先创建样例表, 然后删除某些行, 再用 FLASHBACK TABLE 命令恢复。

(1) 登录到 SCOTT 并创建一个样例表 EMP2。

```
[oracle@localhost ~]$ sqlplus scott/tiger;
SQL*Plus: Release 11.1.0.6.0 - Production on Sat Oct 20 15:43:21 2007
```



```

Copyright (c) 1982, 2007, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> set time on;
15:43:25 SQL> show user;
USER is "SCOTT"
15:43:27 SQL> create table emp2 as select * from emp;
Table created.

15:43:53 SQL> set linesize 120;
15:44:11 SQL> select * from emp2;
EMPNO ENAME      JOB      MGR      HIREDATE      SAL      COMM      DEPTNO
-----
7369  SMITH      CLERK      7902      17-DEC-80      800              20
7499  ALLEN      SALESMAN   7698      20-FEB-81      1600      300      30
7521  WARD      SALESMAN   7698      22-FEB-81      1250      500      30
7566  JONES      MANAGER    7839      02-APR-81      2975              20
7654  MARTIN    SALESMAN   7698      28-SEP-81      1250      1400     30
7698  BLAKE      MANAGER    7839      01-MAY-81      2850              30
7782  CLARK      MANAGER    7839      09-JUN-81      2450              10
7788  SCOTT      ANALYST    7566      19-APR-87      3000              20
7839  KING      PRESIDENT          17-NOV-81      5000              10
7844  TURNER    SALESMAN   7698      08-SEP-81      1500      0        30
7876  ADAMS      CLERK      7788      23-MAY-87      1100              20
7900  JAMES      CLERK      7698      03-DEC-81      950              30
7902  FORD      ANALYST    7566      03-DEC-81      3000              20
7934  MILLER    CLERK      7782      23-JAN-82      1300              10

14 rows selected.

```

(2) 使用 DELETE 语句删除 empno=7369 数据行并提交。

```

15:44:15 SQL> delete from emp2 where empno=7369;
1 row deleted.
15:49:51 SQL> commit;
Commit complete.
15:49:54 SQL>

```

(3) 使用 “ALTER TABLE ... ENABLE ROW MOVEMENT;” 语句进行恢复。

```

15:49:54 SQL> alter table emp2 enable row movement;
Table altered.
15:52:13 SQL>

15:56:11 SQL> flashback table emp2 to timestamp to_timestamp('2007-10-20 15:44:15' ,
'yyyy-mm-dd hh24:mi:ss');
Flashback complete.

15:59:13 SQL> select * from emp2;

EMPNO  ENAME      JOB      MGR      HIREDATE      SAL      COMM      DEPTNO
-----
7369   SMITH      CLERK      7902      17-DEC-80      800              20
7499   ALLEN      SALESMAN   7698      20-FEB-81      1600      300      30
7521   WARD      SALESMAN   7698      22-FEB-81      1250      500      30

```



```

7566 JONES MANAGER 7839 02-APR-81 2975 20
7654 MARTIN SALESMAN 7698 28-SEP-81 1250 1400 30
7698 BLAKE MANAGER 7839 01-MAY-81 2850 30
7782 CLARK MANAGER 7839 09-JUN-81 2450 10
7788 SCOTT ANALYST 7566 19-APR-87 3000 20
7839 KING PRESIDENT 17-NOV-81 5000 10
7844 TURNER SALESMAN 7698 08-SEP-81 1500 0 30
7876 ADAMS CLERK 7788 23-MAY-87 1100 20
7900 JAMES CLERK 7698 03-DEC-81 950 30
7902 FORD ANALYST 7566 03-DEC-81 3000 20
7934 MILLER CLERK 7782 23-JAN-82 1300 10
14 rows selected.
    
```

在上例中，empno=7369 数据行是恢复之后的数据行。

在上面的 FLASHBACK TABLE 语句中，我们采用 TO\_TIMESTAMP 指定恢复时间。在 flashback table 语句中也可以采用系统更改号 (SCN)，但是在操作中，时间比较容易把握，而不容易知道误操作时的 SCN，Oracle 可以用 TIMESTAMP\_TO\_SCN 函数实现将时间戳转换成 SCN (SCN\_TO\_TIMESTAMP 函数则将 SCN 转换成时间戳)。需要注意的是，在每个系统中，所返回的 SCN 是不同的。

**示例 3：用 TIMESTAMP\_TO\_SCN 函数将时间戳转换成 SCN。**

(1) 在本例中，要使用安装 Oracle 时的样例用户 OE 的 ORDERS 表，所以，首先应将表的相关权限授予 scott 用户。

```

SQL> conn /as sysdba;
Connected.
SQL> show user;
USER is "SYS"
SQL> grant select ,insert,delete on oe.orders to scott;
Grant succeeded.
    
```

(2) 使用 scott 用户，查看 orders 表的结构。

```

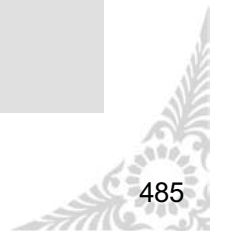
SQL> conn scott/tiger;
Connected.
SQL> show user;
USER is "SCOTT"
SQL> desc scott.orders;
Name                Null?              Type
-----
ORDER_ID            NOT NULL          NUMBER(12)
ORDER_DATE          NOT NULL          TIMESTAMP(6) WITH LOCAL TIME ZONE
ORDER_MODE          VCHAR2(8)
CUSTOMER_ID         NOT NULL          NUMBER(6)
ORDER_STATUS        NUMBER(2)
ORDER_TOTAL         NUMBER(8,2)
SALES_REP_ID        NUMBER(6)
PROMOTION_ID        NUMBER(6) SQL>
    
```

(3) 插入一行数据，然后利用转换函数查看 SCN 号。

```

SQL> insert into oe.orders(order_id,order_date,customer_id,order_total)
  2 values(5000,systimestamp,188,2345);
1 row created.
SQL> commit;
Commit complete.

SQL> select timestamp_to_scn(order_date) from oe.orders
    
```





```

2 where order_id=5000;
TIMESTAMP_TO_SCN(ORDER_DATE)
-----
1233785
SQL>

```

## 示例 4：用 ORA\_ROWSCN 查询 SCN，再将 SCN 转换成 TIMESTAMP。

```

SQL> show user;
USER is "SCOTT"
SQL> set pagesize 50;
SQL> select ora_rowscn ,empno,ename,sal from emp;
ORA_ROWSCN      EMPNO      ENAME      SAL
-----
518834      7369      SMITH      800
518834      7499      ALLEN      1600
518834      7521      WARD      1250
518834      7566      JONES      2975
518834      7654      MARTIN     1250
518834      7698      BLAKE      2850
518834      7782      CLARK      2450
518834      7788      SCOTT      3000
518834      7839      KING      5000
518834      7844      TURNER     1500
518834      7876      ADAMS      1100
518834      7900      JAMES      950
518834      7902      FORD      3000
518834      7934      MILLER     1300
14 rows selected.

SQL> set linesize 140;
SQL> col timestamp for a40;
SQL> select scn_to_timestamp(ora_rowscn) timestamp ,empno,ename,sal
2 from emp;

TIMESTAMP                                EMPNO      ENAME      SAL
-----
03-AUG-07 05.12.20.000000000 PM          7369      SMITH      800
03-AUG-07 05.12.20.000000000 PM          7499      ALLEN      1600
03-AUG-07 05.12.20.000000000 PM          7521      WARD      1250
03-AUG-07 05.12.20.000000000 PM          7566      JONES      2975
03-AUG-07 05.12.20.000000000 PM          7654      MARTIN     1250
03-AUG-07 05.12.20.000000000 PM          7698      BLAKE      2850
03-AUG-07 05.12.20.000000000 PM          7782      CLARK      2450
03-AUG-07 05.12.20.000000000 PM          7788      SCOTT      3000
03-AUG-07 05.12.20.000000000 PM          7839      KING      5000
03-AUG-07 05.12.20.000000000 PM          7844      TURNER     1500
03-AUG-07 05.12.20.000000000 PM          7876      ADAMS      1100
03-AUG-07 05.12.20.000000000 PM          7900      JAMES      950
03-AUG-07 05.12.20.000000000 PM          7902      FORD      3000
03-AUG-07 05.12.20.000000000 PM          7934      MILLER     1300
14 rows selected.
SQL>

```





## 22.5 闪回丢弃

闪回丢弃是将被丢弃的数据库对象及其相依对象的拷贝保存在回收站中，以便在必要时能够及时恢复这些对象。在回收站被清空以前，被丢弃的对象并没有从数据库中删除，这就使得数据库能够恢复被意外或误操作而删除的表。

为了更好地理解闪回丢弃，需要理解回收站的工作细节、丢弃对象在回收站中的存储、查询与清除等，下面几节将对其一一介绍。

### 22.5.1 回收站概念

回收站 (Recycle Bin) 是所有丢弃表及其相依对象的逻辑存储容器。当一个表被丢弃时 (DROP)，回收站会将该表及其相应对象存储在回收站中。存储在回收站中的表的相应对象包括索引、约束、触发器、嵌套表、大的二进制对象 (LOB) 段和 LOB 索引段。

Oracle 回收站将用户所做的 DROP 语句的操作记录在一个系统表里，即将被删除的对象写到一个数据字典表中，当不再需要被删除的对象时，可以使用 PURGE 命令对回收站空间进行清除。

为了避免被删除表与同类对象名称的重复，被删除表 (以及相依对象) 放到回收站中后，Oracle 系统对被删除的对象名做了转换。被删除对象 (如表) 的名字转换格式如下：

```
BIN$globalUID$version
```

- ◎ globalUID 是一个全局唯一的、24 个字符长的标识对象，它是 Oracle 内部使用的标识，对用户来说没有任何实际意义，该标识与对象未删除前的名称没有关系。
- ◎ \$version 是 Oracle 数据库分配的版本号。

### 22.5.2 使用回收站

如果要对 DROP 过的表进行恢复操作，可以使用以下语句。

```
SQL>FLASHBACK TABLE table_name TO BEFORE DROP
```

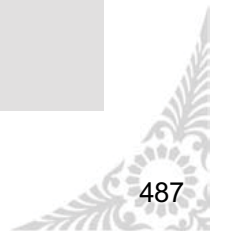
为了帮助读者理解回收站在使用中的操作过程，下面给出较详细的回收站操作步骤。

**示例 1：**本例给出数据准备、删除表、查询回收站信息、恢复及查询恢复后的情况。

(1) 连接 Oracle。

```
[oracle@localhost ~]$ sqlplus scott/tiger
SQL*Plus: Release 11.1.0.6.0 - Production on Sat Oct 20 16:32:00 2007
Copyright (c) 1982, 2007, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.1.0.6.0 - Production
With the Partitioning, OLAP, Data Mining and Real Application Testing options

SQL> set line 120;
SQL> set pagesize 50;
SQL> show user;
```





## Oracle 11g 权威指南（第 2 版）

```
USER is "SCOTT"
SQL>
```

### (2) 准备数据。

```
SQL> create table my_emp as select * from emp;
Table created.
SQL> select count(*) from my_emp;
   COUNT(*)
-----
        14
SQL>
```

### (3) 删除表结构。

```
SQL> select * from tab;
TNAME                                TABTYPE      CLUSTERID
-----
BONUS                                TABLE
DEPT                                  TABLE
EMP                                   TABLE
EMP2                                  TABLE
MY_EMP                                TABLE
SALGRADE                              TABLE
SYS_TEMP_FBT                          TABLE
7 rows selected.

SQL> drop table my_emp;
Table dropped.
SQL>
```

### (4) 删除（DROP）表后的数据字典

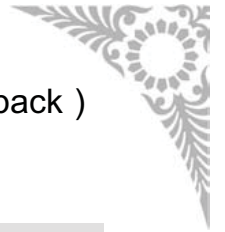
```
SQL> select * from tab;
TNAME                                TABTYPE      CLUSTERID
-----
BIN$POiMOEfPgU3gQAB/AQASlg==$0      TABLE
BONUS                                TABLE
DEPT                                  TABLE
EMP                                   TABLE
EMP2                                  TABLE
SALGRADE                              TABLE
SYS_TEMP_FBT                          TABLE
7 rows selected.
```

需要说明的是，当 MY\_EMP 表被删除后，在数据库回收站里变成了 BIN\$POiMOEfPgU3gQAB/AQASlg==\$0，version 是 0。

### (5) 查看 user\_recyclebin 回收站，可以看到删除的表对应的记录。

```
SQL> col object_name for a30;
SQL> col original_name for a20;
SQL> select object_name,original_name from user_recyclebin;
OBJECT_NAME                          ORIGINAL_NAME
-----
BIN$POiMOEfPgU3gQAB/AQASlg==$0 MY_EMP
SQL>
```





(6) 利用 `user_recyclebin` 中的记录，使用 `FLASHBACK` 从回收站恢复 `MY_EMP`。

```
SQL> flashback table my_emp to before drop;
Flashback complete.
SQL> select count(*) from my_emp;
   COUNT(*)
-----
         14
SQL>
```

以上是恢复完成后的查询结果。

### 22.5.3 回收站与空间利用

回收站是丢弃对象的逻辑存储容器，它以表空间中现有的已经分配的空间为基础，这意味着系统并没有给回收站预留空间。这使得回收站空间依赖于现有表空间中的可用空间（即丢弃表占据的空间仍然需要计入表空间配额），因此，并不能总是保证丢弃对象在回收站中的最小时间。

如果不对回收站进行清除操作，丢弃对象会一直保存在回收站内，一直到丢弃对象所属的表空间无法再分配新的存储区域，这种状态称为空间压力。有时，用户的表空间限额也会导致空间压力状态的出现，即使表空间中仍然存在自由空间。

当空间压力出现时，Oracle 会覆写回收站对象从而自动回收表空间。Oracle 根据先进先出的原则选择丢弃对象进行删除，所以最先被丢弃的对象也最先被清除。而对象的清除仅仅是为了解决产生的空间压力问题，所以会尽可能清除少的对象以满足空间压力的要求。这样既最大限度地保证了对象在回收站中的可用时间，又减少了 Oracle 在事务处理时的性能影响。

DBA 需要关注回收站的空间利用情况，掌握清除回收站对象从而释放空间的办法，这可以用 `PURGE` 命令完成，`PURGE` 命令可从回收站中删除表或索引，并释放有关表和索引所占用的空间；用 `PURGE` 命令也可清除整个回收站或清除被删除的表空间的所有部分。

需要注意的是，当用 `PURGE` 命令清除掉被删除的对象后，该对象即被完全清除而不能重建。

要清除回收站中的对象、释放空间，可以有以下几种方式。

(1) 使用 `PURGE TABLE original_table_name`。

这里的 `original_table_name` 表示表未 `drop` 以前的名称（源名称），使用该操作可以从回收站中永久删除对象并释放空间。

(2) 使用 `PURGE TABLE recyclebin_object_name`。

这里的 `recyclebin_object_name` 表示回收站中的对象名称，使用该操作可以从回收站中永久删除对象并释放空间。

(3) 使用 `PURGE TABLESPACE tablespace_name` 从回收站清除一个特定表空间的所有对象。

该命令从指定的表空间中清除所有的丢弃对象，以及相依对象。因为相依对象（比如 `LOB`、嵌套表、索引、分区等）未必与基表存储在同一个表空间，该命令会将相依对象从其所在的表空间中清除。



(4) 使用 `PURGE TABLESPACE tablespace_name USER user_name` 会从回收站中清除属于某个特定用户的所有丢弃对象（当然也包括基表的相依对象）。

(5) 使用命令 `DROP USER user_name CASCADE` 会直接删除指定用户及其所属的全部对象。

也就是说，`DROP USER` 命令会绕过回收站进行直接删除。同时，如果回收站中也有该用户的所属对象，则也会从回收站中清除。

(6) 使用 `PURGE RECYCLEBIN` 命令可以清除用户自己的回收站。

该命令从用户回收站中清除所有的对象并释放与这些对象关联的空间。

(7) `PURGE DBA_RECYCLEBIN` 从所有用户的回收站清除所有对象。

该命令能高效地完全清空回收站，当然，执行该命令必须具有 `SYSDBA` 系统管理权限。

**示例 2：查询当前用户回收站中的内容，再用 `PURGE` 清除。**

```
SQL> show user;
USER is "SCOTT"
SQL> create table orcltest as select * from emp;
Table created.
SQL> select * from tab;

```

TNAME	TABTYPE	CLUSTERID
BONUS	TABLE	
DEPT	TABLE	
EMP	TABLE	
EMP2	TABLE	
MY_EMP	TABLE	
ORCLTEST	TABLE	
SALGRADE	TABLE	
SYS_TEMP_FBT	TABLE	

```

8 rows selected.

SQL> drop table orcltest;
Table dropped.
SQL> select object_name,original_name from user_recyclebin;
OBJECT_NAME          ORIGINAL_NAME
-----
BIN$POiMOEfQgU3gQAB/AQASlg==$0 ORCLTEST
SQL>
SQL> show user;
USER is "SCOTT"
SQL> select object_name,original_name from user_recyclebin;
OBJECT_NAME          ORIGINAL_NAME
-----
BIN$POiMOEfQgU3gQAB/AQASlg==$0 ORCLTEST

SQL> purge table orcltest;
Table purged.
SQL> select object_name,original_name from user_recyclebin;
no rows selected
SQL>
```



## 22.6 闪回版本查询

许多安全和保密要求高的应用系统，都希望对表等重要对象的操作进行登记，以便进行监督。本节介绍的闪回版本查询可满足这一要求。

### 22.6.1 闪回版本查询概念

Oracle 的闪回版本查询功能 (Flashback Version Query) 提供了一个审计行改变的查询功能，它能找到所有已经提交了行的记录。借助此特殊功能，我们可以清楚地看到何时执行了何操作。使用该功能，可以很轻松地实现对应用系统的审计，而没有必要使用细粒度的审计功能或 LOGMNR 了。

闪回版本查询功能依赖于 AUM (Automatic Undo Management)，AUM 是指采用撤销表空间记录增、删、改数据的方法。

要用 Flashback Version Query 实现对数据行改变的记录进行查询，主要采用 SELECT 语句带 FLASHBACK\_QUERY 子句实现，FLASHBACK\_QUERY 子句的语法格式如下：

```
SELECT <column1>,... FROM <table>
...
VERSION BETWEEN [ SCN | TIMESTAMP ]
[ <expr> | MAXVALUE] AND <expr> | MINVALUE]
| AS OF [SCN |TIMESTAMP ] <expr>
```

其中：

- ⊙ AS OF：表示恢复单个版本；
- ⊙ SCN：系统更改号；
- ⊙ TIMESTAMP：时间。

### 22.6.2 使用闪回版本查询

下面我们通过具体的实例来演示闪回版本查询的使用。

**示例：闪回版本查询内容。**

(1) 系统初始情况的查询，用 to\_timestamp 函数转换时间格式。

```
SQL> set time on
17:21:07 SQL> show user;
USER is "SCOTT"
17:21:12 SQL> select * from emp2 as of timestamp to_timestamp('2007-10-20
16:00:00','yyyy-mm-dd hh24:mi:ss');
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30



```

7566 JONES  MANAGER      7839    02-APR-81    2975          20
7654 MARTIN SALESMAN      7698    28-SEP-81    1250    1400    30
7698 BLAKE  MANAGER      7839    01-MAY-81    2850          30
7782 CLARK  MANAGER      7839    09-JUN-81    2450          10
7788 SCOTT  ANALYST      7566    19-APR-87    3000          20
7839 KING   PRESIDENT          17-NOV-81    5000          10
7844 TURNER SALESMAN      7698    08-SEP-81    1500     0    30
7876 ADAMS  CLERK        7788    23-MAY-87    1100          20
7900 JAMES  CLERK        7698    03-DEC-81    950           30
7902 FORD   ANALYST      7566    03-DEC-81    3000          20
7934 MILLER CLERK        7782    23-JAN-82    1300          10
    
```

14 rows selected.

(2) 在 17:23:10 时刻删除一行。

```

17:21:28 SQL> delete from emp2 where empno=7369;
1 row deleted.
17:23:10 SQL> commit;
Commit complete.
    
```

(3) 在 17:23:10 时刻删除一行，所以在 17:23:13 时刻只看到了 13 行。

```

17:23:20 SQL> select * from emp2 as of timestamp to_timestamp('2007-10-20 17:23:13',
'yyyy-mm-dd hh24:mi:ss');
EMPNO ENAME  JOB            MGR      HIREDATE          SAL        COMM        DEPTNO
-----
7499 ALLEN  SALESMAN      7698      20-FEB-81         1600         300         30
7521 WARD   SALESMAN      7698      22-FEB-81         1250         500         30
7566 JONES  MANAGER      7839      02-APR-81         2975          20
7654 MARTIN SALESMAN      7698      28-SEP-81         1250        1400         30
7698 BLAKE  MANAGER      7839      01-MAY-81         2850          30
7782 CLARK  MANAGER      7839      09-JUN-81         2450          10
7788 SCOTT  ANALYST      7566      19-APR-87         3000          20
7839 KING   PRESIDENT          17-NOV-81         5000          10
7844 TURNER SALESMAN      7698      08-SEP-81         1500     0         30
7876 ADAMS  CLERK        7788      23-MAY-87         1100          20
7900 JAMES  CLERK        7698      03-DEC-81         950           30
7902 FORD   ANALYST      7566      03-DEC-81         3000          20
7934 MILLER CLERK        7782      23-JAN-82         1300          10
    
```

13 rows selected.

(4) 在 17:30:21 时刻删除另外一行。

```

17:29:55 SQL> delete from emp2 where empno=7499;
1 row deleted.
17:30:21 SQL> commit;
Commit complete.
17:30:23 SQL>
    
```

(5) 在 17:23:10 和 17:30:21 之间时，是 13 行，在 17:30:21 以后是 12 行。

```

17:35:20 SQL> select * from emp2 as of timestamp to_timestamp('2007-10-20 17:35:13',
'yyyy-mm-dd hh24:mi:ss');
EMPNO ENAME  JOB            MGR      HIREDATE          SAL        COMM        DEPTNO
-----
7521 WARD   SALESMAN      7698      22-FEB-81         1250         500         30
7566 JONES  MANAGER      7839      02-APR-81         2975          20
    
```



```

7654 MARTIN SALESMAN      7698      28-SEP-81      1250      1400      30
7698 BLAKE  MANAGER      7839      01-MAY-81      2850      30
7782 CLARK  MANAGER      7839      09-JUN-81      2450      10
7788 SCOTT  ANALYST      7566      19-APR-87      3000      20
7839 KING   PRESIDENT          17-NOV-81      5000      10
7844 TURNER SALESMAN      7698      08-SEP-81      1500      0      30
7876 ADAMS  CLERK        7788      23-MAY-87      1100      20
7900 JAMES  CLERK        7698      03-DEC-81      950      30
7902 FORD   ANALYST      7566      03-DEC-81      3000      20
7934 MILLER CLERK        7782      23-JAN-82      1300      10
12 rows selected.

```

(6) 对 EMP2 表的操作进行审计，查询结果如下。

```

17:30:23 SQL> select empno,ename,sal,versions_operation,versions_xid,versions_starttime
                from emp2
17:36:54 2 versions between timestamp minvalue and maxvalue order by empno,versions_starttime;
EMPNO ENAME      SAL      VERSIONS_OPERATION  VERSIONS_XID      VERSIONS_STARTTIME
7369 SMITH      800.00  D              04000500AF030000  20-10月-07 05.23.09 下午
7369 SMITH      800.00
7499 ALLEN     1600.00  D              04000400AF030000  20-10月-07 05.30.18 下午
7499 ALLEN     1600.00
7521 WARD      1250.00
7566 JONES     2975.00
7654 MARTIN   1250.00
7698 BLAKE     2850.00
7782 CLARK     2450.00
7788 SCOTT     3000.00
7839 KING     5000.00
7844 TURNER   1500.00
7876 ADAMS    1100.00
7900 JAMES     950.00
7902 FORD     3000.00
7934 MILLER   1300.00
16 rows selected.

```

## 22.7 闪回事务查询

事务是关系数据库系统常用到的一个概念，具体来说，事务就是访问数据库时一系列的逻辑相关动作。比如列车订票系统中选择目的地（或车次）、日期和时间、选择座位（上下铺）、出票（总票数减一张）、付费，这一系列处理构成一个事务。Oracle 11g 的闪回事务查询就是对过去某段时间内所完成的事务的查询和撤销。

### 22.7.1 闪回事务查询概念

闪回事务查询是一种诊断工具，帮助识别数据库发生的事务级变化，可以用于事务审计的数据分析。通过闪回事务分析，可以识别在一个特定的时间段内所发生的所有变化，也可以对数据库表进行事务级恢复。

闪回事务查询的基础是依赖于撤销数据（undodata），它也是利用初始化的数据库参数 UNDO\_





RETENTION 来确定已经提交的撤销数据在数据库中的保存时间。

另外，我们在前面介绍的 Flashback Version Query 可以实现审计一段时间内表的所有改变，但是这仅仅是发现在某个时间段内所做过的操作，对于错误的事务还不能进行撤销处理。而 Flashback Transaction Query 可实现撤销处理，因为可以从 FLASHBACK\_TRANSACTION\_QUERY 中获得事务的历史操作及撤销语句 (undo\_sql)，也就是说，我们可以审计一个事务到底做了什么，也可以撤销一个已经提交的事务。

### 22.7.2 使用闪回事务查询

要使用 Flashback Version Query 功能，需要用到 FLASHBACK\_TRANSACTION\_QUERY 视图，下面是 FLASHBACK\_TRANSACTION\_QUERY 的列的解释。

列名称	类型	解释
XID	RAW(8)	--事务标识
START_SCN	NUMBER	--事务起始 SCN
START_TIMESTAMP	DATE	--事务起始时间
COMMIT_SCN	NUMBER	--事务提交 SCN
COMMIT_TIMESTAMP	DATE	--事务提交时间戳
LOGON_USER	VARCHAR2(30)	--登录的用户名
UNDO_CHANGE#	NUMBER	--撤销改变号
OPERATION	VARCHAR2(32)	--前滚操作
TABLE_NAME	VARCHAR2(256)	--表名
TABLE_OWNER	VARCHAR2(32)	--表拥有者
ROW_ID	VARCHAR2(19)	--唯一的行标识
UNDO_SQL	VARCHAR2(4000)	--撤销的 SQL 语句

下面继续以 Oracle 系统 SCOTT 模式下的 EMP 表为例，来演示闪回事务查询功能。

#### 示例：闪回事务。

(1) 查看某个时间段内的操作发生情况。

```
SQL> select empno ,ename, sal , versions_operation , versions_xid, versions_starttime
        from scott.my_emp
2 versions between timestamp minvalue and maxvalue order by empno,versions_starttime;
```

EMPNO	ENAME	SAL	Version_operation	ERSIONS_XID	VERSIONS_STARTTIME
7369	SMITH	800	D	08001F00B2040000	21-OCT-07 02.21.12 PM
7369	SMITH	800			
7499	ALLEN	1600			
7521	WARD	1250			
7566	JONES	2975			
7654	MARTIN	1250			
7698	BLAKE	2850			
7782	CLARK	2450			
7788	SCOTT	3000			
7839	KING	5000			
7844	TURNER	1500			
7876	ADAMS	1100			
7900	JAMES	950			
7902	FORD	3000			
7934	MILLER	1300			





```
15 rows selected.
```

(2) 找出 XID 为 08001F00B2040000 的语句的撤销操作 (UNDO\_SQL) (SCOTT 默认下不能查询 flashback\_transaction\_query 表)。

```
SQL> conn /as sysdba;
Connected.
SQL> col table_name for a18;
SQL> col table_owner for a10;
SQL> col logon_user for a12;
SQL> set long 1000;
SQL> select logon_user,table_name,table_owner,undo_sql
  2 from flashback_transaction_query where table_owner='SCOTT' and xid='08001F00B2040000';
```

LOGON_USER	TABLE_NAME	TABLE_OWNE	UNDO_SQL
SYS	MY_EMP	SCOTT	insert into "SCOTT"."MY_EMP"("EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM", "DEPTNO") values ('7369','SMITH','CLERK','7902',TO_DATE('17-12月-80', 'DD-MON-RR'),'800', NULL,'20');

```
SQL>
```

(3) 运行上面找出的 SQL 语句,即可将以前删除的数据恢复回来。

```
SQL> conn scott/tiger;
Connected.
SQL>insert into "SCOTT"."MY_EMP"("EMPNO","ENAME","JOB","MGR","HIREDATE","SAL","COMM",
  "DEPTNO") values ('7369','SMITH','CLERK','7902',TO_DATE('17-12月-80',
  'DD-MON-RR'),'800', NULL,'20');
1 row created.

SQL> commit;
Commit complete.

SQL>
```

## 22.8 闪回数据归档

在 Oracle 11g 中,对闪回技术再次进行了扩展,提供了一个全新的 flashback 方式,称为闪回数据归档,本节我们将对闪回数据库归档进行介绍。

### 22.8.1 闪回数据归档概念

这里我们从 Oracle 9i 开始引进的 Flashback Query 开始介绍,这是 Oracle 第一次引入闪回技术,该技术使得一些逻辑误操作不再需要利用归档日志和数据库备份进行时间点恢复。

而在 Oracle10g 中,引入了 Flashback Version Query、Flashback Transaction Query、Flashback Database、Flashback Table、Flashback Drop 等特性,大大简化了 Flashback Query 的使用。

在上面的诸多闪回技术中,除了 Flashback Database (依赖于闪回日志)外,其他闪回技术都是依



赖于撤销数据，都与数据库初始化参数 UNDO\_RETENTION 密切相关（该参数决定了撤销数据在数据库中的保存时间），都是从撤销数据中读取信息来构造旧数据的。但是存在一个限制，就是 undo 中的信息不能被覆盖。而 undo 段是循环使用的，只要事务提交，之前的 undo 信息就可能被覆盖，虽然可以通过 undo\_retention 等参数来延长 undo 的存活期，但该参数会影响所有的事务，设置过大，可能导致 undo tablespace 快速膨胀。

Oracle 11g 则为 Flashback 家族又带来一个新的成员：Flashback Data Archive。该技术与上面所说的诸多闪回技术的实现机制不同，通过将变化数据存储到创建的闪回归档区（Flashback Archive）中，以与 undo 区别开来，这样就可以通过为闪回归档区单独设置存储策略，使得可以闪回到指定时间之前的旧数据而不影响 undo 策略。并且，可以根据需要指定哪些数据库对象需要保存历史变化数据，而不是将数据库中所有对象的变化数据都保存下来，这样可以极大地减少空间需求。

**注意** Flashback Data Archive 并不是记录数据库的所有变化，而只是记录了指定表的数据变化。所以，Flashback Data Archive 是针对象的保护，是 Flashback Database 的有力补充。

通过 Flashback Data Archive 可以查询指定对象的任何时间点（只要满足保护策略）的数据，而且不需要利用 undo，这在有审计需要的环境，或是安全性特别重要的高可用数据库中是一个非常好的特性。缺点是如果该表变化很频繁，那么对空间的要求可能很高。

## 22.8.2 闪回数据归档区

闪回数据归档区是闪回数据归档的历史数据存储区域，在一个系统中，可以有一个默认的闪回数据归档区，也可以创建其他闪回数据归档区域。

每一个闪回数据归档区都可以有一个唯一的名称，同时，每一个闪回数据归档区都对应了一定的数据保留策略。例如，可以配置归档区 FLASHBACK\_DATA\_ARCHIVE\_1 中的数据保留期为 1 年，而归档区 FLASHBACK\_DATA\_ARCHIVE\_2 的数据保留期为 2 天或者更短。以后如果将表放到对应的闪回数据归档区，那就按照该归档区的保留策略来保存历史数据。

闪回数据归档区是一个逻辑概念，是从一个或多个表空间中抽出一定的空间，保存表的修改历史，这样就摆脱了对撤销数据的依赖，方便表不必利用 undo 就可以闪回到归档策略内的任何一个时间点上。

创建闪回数据归档区可以使用 CREATE FLASHBACK ARCHIVE ... 命令完成。

下面我们通过实例演示如何创建闪回数据归档区。

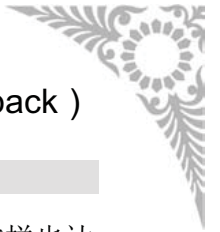
(1) 创建一个系统默认的、磁盘限额为 100MB、保留策略为 1 年的闪回数据归档区。

```
SQL>create flashback archive default fbar_1 tablespace "USERS" 2 quota 100M
2 retention 1 year;
```

**注意** 经过在 Oracle 11g 数据库上运行该语句发现，必须对 tablespace 关键字后面的表空间名称用 " " 引起来，否则无法运行成功。

(2) 在 TBS\_DATA1 上创建 fbar\_2 闪回数据归档区，保留策略为 2 天。

```
SQL>CREATE FLASHBACK ARCHIVE fbar_2 TABLESPACE "TBS_DATA1"
```



```
2 RETENTION 2 DAY;
```

一个归档区可以不仅对应一个表空间，而且可以增加或删除该归档区的表空间的个数，这样也达到了增加或是减少该归档区空间的目的。

(3) 给数据归档区 fbar\_2 增加一个表空间。

```
SQL>ALTER FLASHBACK ARCHIVE fbar_2 ADD TABLESPACE "TBS_DATA2"
2 QUOTA 100M;
```

也可以从归档区中删除表空间。注意：此删除仅表示从数据归档区删除，并不删除该表空间。

(4) 删除数据归档区 fbar\_2 的表空间 TBS\_DATA2。

```
SYS>ALTER FLASHBACK ARCHIVE fbar_2 REMOVE TABLESPACE "TBS_DATA2";
```

对已经分配给闪回数据归档区的表空间，可以修改归档区对应的磁盘限额。

(5) 修改归档区的磁盘限额。

```
SYS>ALTER FLASHBACK ARCHIVE fbar_2 MODIFY TABLESPACE "TBS_DATA2"
2 QUOTA 200M;
```

(6) 修改归档区的保留策略。

```
SYS>ALTER FLASHBACK ARCHIVE fbar_1 MODIFY RETENTION 1 month;
```

### 22.8.3 使用闪回数据归档

闪回数据归档区创建完成后，就可以指定特定的表，使其对应到特定的数据归档区。把表指定到对应的数据归档区有两种方法，一是在创建时直接指定归档区；二是对现有的表指定归档区。

**注意** 如果不指定归档区的名称，则指定到默认归档区，否则，就属于指定的数据归档区。

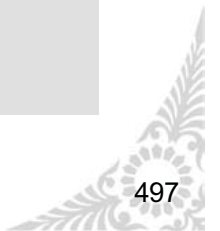
下面我们基于上一节所创建的两个闪回数据归档区 fbar\_1（默认的闪回数据恢复区）和 fbar\_2，创建 3 个表，一个指定到默认归档区 fbar\_1，一个指定到数据归档区 fbar\_2，另外一个为了作对比，没有指定到任何数据归档区。

(1) 创建表。

```
SYS@11gR1>connect scott/tiger
Connected.
scott@11gR1>create table test1(a int) flashback archive;
Table created.
scott@11gR1>create table test2(b int);
Table created.
scott@11gR1>alter table test2 flashback archive data_test2;
Table altered.
scott@11gR1>create table test3(c int);
Table created.
```

(2) 在表中插入数据后，进行 SELECT 查询，显示结果如下：

```
09:33:38 scott@11gR1>select * from test2;
      B
-----
      4
```



```

5
6

09:33:43 scott@11gR1>select * from test3;
C
-----
7
8
9

09:33:46 scott@11gR1>select to_char(sysdate,'yyyy-mm-dd hh24:mi:ss') time from dual;

TIME
----
2007-09-04 09:33:52

```

可以看到，这些数据是在 9:33 写进的，最新数据保留策略应当是，表 test1 对应的是默认的数据归档区 fbar\_1，数据保留策略是一个月，表 test2 对应的是数据归档区 fbar\_2，数据保留策略是 2 天，而表 test3 没有数据保留策略。然后，对这三个表再进行操作，如删除现有记录，并插入一些新记录，最后，我们不使用 undo 数据，查询时间点 2007-09-04 09:33:52，看是否能找回原来的数据。

(3) 对表进行更新操作，查询显示结果。

```

09:34:19 Piner@11gR1>delete from test1;
3 rows deleted.
09:34:23 Piner@11gR1>delete from test2;
3 rows deleted.
09:34:30 Piner@11gR1>delete from test3;
3 rows deleted.
09:34:35 Piner@11gR1>insert into test1 values (10);
1 row created.
09:34:47 Piner@11gR1>insert into test2 values (20);
1 row created.
09:34:53 Piner@11gR1>insert into test3 values (30);
1 row created.
09:34:58 Piner@11gR1>commit;
Commit complete.
09:36:32 Piner@11gR1>select * from test1;
A
-----
10
09:36:51 Piner@11gR1>select * from test2;
B
-----
20
09:36:56 Piner@11gR1>select * from test3;
C
-----
30

```

(4) 利用闪回功能查询数据，可以获得正确数据，但是，无法确认这些数据是经过 undo 获得的还是经过数据归档区获得的。

```

09:43:17 Piner@11gR1>select * from test1 as of timestamp
09:43:17 2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
A

```



```

-----
      1
      2
      3
09:43:17 Piner@11gR1>select * from test2 as of timestamp
09:43:24      2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
      B
-----
      4
      5
      6
09:43:25 Piner@11gR1>select * from test3 as of timestamp
09:43:30      2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
      C
-----
      7
      8
      9

```

(5) 为了证明查询使用的是闪回数据归档，创建新的 undo 表空间，切换 undo 表空间，为了确保生效，可以重新启动数据库例程。

切换到新的 undo 表空间，如果没有，则需要新创建。

```

SYS@11gR1>ALTER SYSTEM SET undo_tablespace=TBS_UNDO2;
System altered.

```

删除原来的 undo 表空间。

```

SYS@11gR1>drop tablespace UNDOTBS1;
Tablespace dropped.

```

(6) 最后，排除 undo 查询的可能后，再次执行查询。

```

scott@11gR1> select * from test3 as of timestamp
      2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
select * from test3 as of timestamp
      *
ERROR at line 1:
ORA-01555: snapshot too old: rollback segment number with name "" too small

scott@11gR1> select * from test1 as of timestamp
      2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
      A
-----
      1
      2
      3

scott@11gR1> select * from test2 as of timestamp
      2 to_timestamp('2007-09-04 09:33:52', 'yyyy-mm-dd hh24:mi:ss');
      B
-----
      4
      5
      6

```



可以看到，若没有设置数据归档策略的表 test3，则查询时会报 01555 错误，但是，设置过数据归档策略的表 test1 与表 test2，都能正常查询到数据，可以看到，数据归档已生效。

这里是一个简单的实验，时间可能远远没有达到设置的策略期，但是，却可以证明数据不是经过 undo 查询获得的。

### 22.8.4 清除闪回数据归档区数据

前面我们介绍过如何给闪回数据归档区增加空间，本节通过一些具体实例为大家介绍如何清除闪回归档区中的数据。

#### 1. 清除所有归档区的数据

```
SQL>ALTER FLASHBACK ARCHIVE data_test1 PURGE ALL;
```

#### 2. 清除一天以前的数据

```
SQL>ALTER FLASHBACK ARCHIVE data_test1  
2 PURGE BEFORE TIMESTAMP (SYSTIMESTAMP - INTERVAL '1' DAY);
```

#### 3. 清除特定 SCN 之前的数据

```
SQL>ALTER FLASHBACK ARCHIVE data_test1 PURGE BEFORE SCN 728969;
```

#### 4. 将指定的表不再设置数据归档

```
SQL>ALTER TABLE test1 NO FLASHBACK ARCHIVE;
```

#### 5. 删除数据归档区

```
SQL>DROP FLASHBACK ARCHIVE 删除、重命名，或者修改列；
```

**注意** 如果将表指定了闪回数据归档区，则不能对表进行如下操作。

- 转换 long 到 LOB 类型；
- ALTER TABLE...UPGRADE TABLE 操作；
- DROP、RENAME、TRUNCATE 表。

示例：

```
scott@11gR1> drop table test1;  
drop table test1  
*  
ERROR at line 1:  
ORA-55610: Invalid DDL statement on history-tracked table
```

### 22.8.5 与闪回数据归档有关的视图

可以通过以下视图得到与闪回数据归档有关的信息。



- ◎ DBA\_FLASHBACK\_ARCHIVE: DBA 视图, 闪回归档区信息;
- ◎ DBA\_FLASHBACK\_ARCHIVE\_TS: DBA 视图, 闪回归档有关表空间;
- ◎ DBA\_FLASHBACK\_ARCHIVE\_TABLES: DBA 视图, 对应表所对应的闪回归档信息;
- ◎ USER\_FLASHBACK\_ARCHIVE: 用户闪回归档区的创建信息;
- ◎ USER\_FLASHBACK\_ARCHIVE\_TABLES: 用户表对应的闪回归档区域。

### 22.9 本章小结

本章主要讲述了数据库的闪回技术, 闪回是数据库进行逻辑恢复的一个快捷工具。对 Oracle 11g 在闪回方面的最新技术闪回数据归档也进行了详细介绍。





## 第 32 章 数据挖掘

数据挖掘是发现潜在规律的过程。Oracle 数据挖掘 (Oracle Data Mining), 简称 ODM, 是内嵌在 Oracle 数据库中功能强大的数据挖掘软件, 它能够发现隐藏在数据中的新信息。Oracle 数据挖掘帮助企业业务人员确定其最佳客户, 发现并防止欺诈行为, 发现那些对主要绩效指标 (KPI) 影响最大的属性, 发现隐藏在其数据中的新信息。Oracle 数据挖掘帮助专业技术人员找出其数据中的模式, 识别关键属性, 发现新的聚类 and 关联, 揭示有用的信息。

利用 Oracle 数据挖掘, 一切工作都在 Oracle 数据库: 单一、安全、可伸缩的商务智能平台中进行。Oracle 数据挖掘将分析功能转移到数据库中, 即与数据在一起。

ODM 是 Oracle 数据库企业版的一个选项。ODM 在 Oracle 数据库中嵌入了分类和回归、关联和聚类模型、属性重要性排序、特性选择、文本挖掘等算法。可以通过 Java 和 PL/SQL 应用程序编程接口 (API) 和 Oracle Data Miner 图形用户界面访问, 方便了 ODM 模型构建和模型计分函数应用。Oracle Data Miner 的 GUI 与 ODM 的 PL/SQL 和 Java ODM API 相结合 使 Oracle 能够为数据分析人员和应用程序开发人员提供一个基础架构, 将数据挖掘与数据库应用程序集成在一起。

### 32.1 数据挖掘过程

Oracle 数据挖掘支持数据挖掘的跨行业标准流程, 即 CRISP-DM (CRoss-Industry Standard Process for Data Mining), CRISP-DM 目前已经成为事实上的数据挖掘行业标准和重要的数据挖掘过程模型。

CRISP-DM 支持 6 个阶段的数据挖掘过程, 如图 32-1 所示。

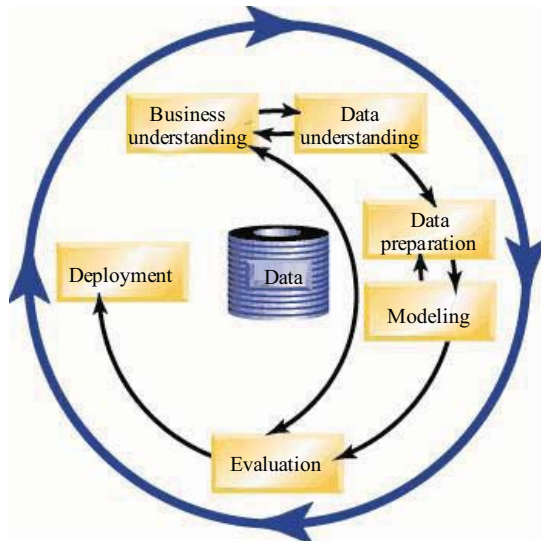




图 32-1 数据挖掘过程

图 32-1 的外圈象征数据挖掘自身的循环本质——在一个数据挖掘方案应用之后，会根据应用反馈以及业务变化，继续改善和调整模型。这个过程不断循环往复，持续下去。

#### (1) 业务理解 (Business Understanding)

非常重要的阶段，集中在理解项目目标和从业务的角度理解需求，同时将这个知识转化为数据挖掘问题的技术定义。

在这一阶段中，业务领域的专家确定将抽象业务目标（例如“我如何能够将更多产品销售给客户？”）转化为较切实有用的数据挖掘问题表述（例如“哪些客户最有可能购买产品 A？”）。要构建一个预测谁最有可能购买产品 A 的预测模型，首先必须拥有对过去已购买产品 A 的客户做出描述的数据。

#### (2) 数据理解 (Data Understanding)

数据理解阶段从数据收集开始，通过一些活动的处理，熟悉数据，识别数据的质量和可用性问题的，发现数据的内部规律，找出与目标相关的重要属性数据。

例如，查看数据的概要分析，包括空值、最大值、最小值、边界值等。

#### (3) 数据准备 (Data Preparation)

数据准备阶段包括从未处理数据中构造最终数据集的所有活动。活动包括表、记录和属性的选择，以及为模型工具转换和清洗、转换数据。准备后的数据将是模型工具的输入值。

准备阶段，执行某些数据处理过程来进行转换（例如“Date\_of\_Birth”字段转换为“AGE”）和获取字段（如“Number\_of\_times\_Amount\_Exceeds\_100”），以尝试“使隐藏的信息更接近于数据的表象”，以便进行更容易的挖掘。

#### (4) 建模 (Modeling)

在这个阶段，可以选择和应用不同的模型技术，模型参数被调整到最佳的数值。不同的算法和模型通常可以解决相同的数据挖掘问题。

利用数据挖掘算法在数据中进行筛选，以找出模式并构建预测模型。通常，数据分析人员会构建几个模型并更改挖掘参数，以尝试构建最好或最有用的模型。

#### (5) 评估 (Evaluation)

这个阶段，从数据分析的角度建立了一个高质量的分析模型。在部署模型之前，需要评估模型，检查构造模型的步骤，确保模型可以完成业务目标。核查查准率和查全率等指标。

例如，对多个模型的结果评估，找出最佳模型。

#### (6) 部署 (Deployment)

部署也称为打分阶段。模型的作用是从数据中找到知识，并将获得的知识应用于业务，完成预定目标。

例如，当 Oracle 数据挖掘找到对数据充分建模的有用模型之后，将新的信息和预测分发给其他人——经理、呼叫中心代表和管理人员。

## 32.2 监督式学习算法

大部分数据挖掘算法可以分成“监督式学习”和“无监督式学习”的数据挖掘技术。监督式学习需要数据分析人员确定一个目标属性或相关变量。监督式学习技术对数据进行筛选，试图找出独立属性（预测因子）与相关属性之间的模式和关系。

在监督式学习应用程序中，定义一个目标属性，例如，描述哪些客户最近购买了新车——比如“1”表示“是”，“0”表示“否”。监督式数据挖掘算法在数据中进行筛选以找出模式，并构建能够最好地代表功能关系的“模型”。通常，对于数据挖掘过程，数据被分为两部分，即训练部分和测试部分两个样本。初始模型是使用第一个（通常较大）数据样本而构建的，随后将该模型应用到第二个样本，以评估模型预测的准确性，因为已经知道这些数据的结果（例如谁购买了车，谁没有购买车），所以第二个样本数据可以评估模型的准确性，并确定模型的有效性。其预测结果准确性比较高的那些模型经常具有很高的经济价值。二值/多值预测涵盖了众多重要业务问题，包括营销响应、销售建议、欺诈检测、收益预测、客户概要、信用评级、客户流失预测、库存需求、故障预测等。

Oracle 数据挖掘提供分类、回归、属性重要性等功能，其中监督式数据挖掘算法包括下面几种。

### 32.2.1 朴素贝叶斯

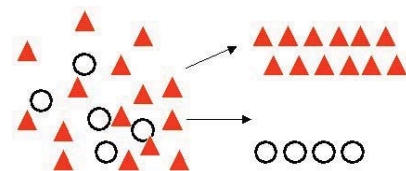
ODM 的朴素贝叶斯 Naïve Bayes (NB) 算法是一种用于分类和预测的监督式学习技术。ODM NB 构建的模型预测特定结果的可能性。NB 算法通过计算所观察到的不同情况的次数，找出数据中的模式和关系；然后构建一个数据挖掘模型来表示这些模式和关系。在 ODM NB 构建模型之后，可以将该模型用于预测。ODM NB 使应用程序开发人员能够构建对各种目标进行分类和预测的数据挖掘模型。例如：

- ◎ 识别哪些客户可能购买特定的产品；
- ◎ 识别哪些客户可能对营销活动做出响应；
- ◎ 识别哪些客户最有可能消费 3000 美元以上；
- ◎ 识别可能流失的客户；
- ◎ 预测某个成分出错的可能性等。

ODM 的 NB 算法使用 Bayes 定理进行预测，该定理假设每个属性都有条件地相互独立。NB 提供快速的建模和计分，可用于二级和多级分类问题。NB 交叉验证作为运行 NB 的一种可选方法而得到支持，它允许用户基于构建模型所用的同一批数据来测试模型准确性，而不是在一部分数据上构建模型，而在另一部分数据上测试该模型。如果用于构建模型的数据数量相对较少，则这种算法和构建方法特别有用。

### 32.2.2 自适应贝叶斯网络

ODM 的自适应贝叶斯网络 (ABN) 算法与朴素贝叶斯相似，并且根据所分析的数据，它可能生成更好的模型。ODM 还通过自适应





贝叶斯网络 (ABN) 提供一种高度可伸缩的类似于决策树的算法。ABN 结合了两个领域中最佳内容：简洁而易于解释的基于规则的模型以及贝叶斯推论。针对某一具体问题，分析人员可创建 NB 和 ABN 两种模型，从中择其优者。ABN 模型的另一个好处是它们能够生成简单的“规则”，可以提供关于所做预测原因的信息。典型的“预测”和“规则”可以是：

预测：是否购买“宝马”汽车

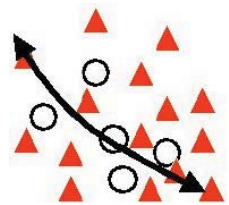
ABN 规则：30 < "年龄" < 40 并且 "收入" = "高"

可信度 (Confidence) = 85% (=539/634，总共 634 位客户适合此规则，其中的 539 位购买了“宝马”汽车)

支持度 (Support) = 0.54% (=539/99263，总共 99263 位客户参与调查，其中的 539 位购买了“宝马”汽车)

### 32.2.3 支持向量机

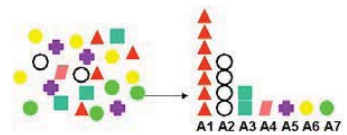
ODM 的支持向量机 (SVM) 算法支持二值/多值分类、预测和回归模型，即预测连续的目标属性。SVM 还擅长于发现隐藏在某些问题中的模式，这些问题具有大量的独立属性，却只有很少的数据记录或观察资料，例如异常情况检测功能、防范欺诈等。



### 32.2.4 属性重要性排序

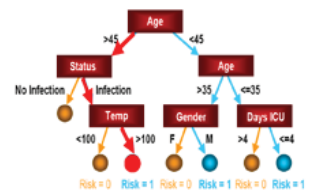
Oracle 数据挖掘的属性重要性算法帮助识别出对目标字段具有最大影响的属性。了解哪些属性最具影响力，帮助更好地了解和管理业务，并且能够帮助简化建模操作。

属性重要性可用于找出最相关的过程属性，用于预测生产部件的质量、与流失相关的因素或者最有可能与特定疾病治疗相关的基因。



### 32.2.5 决策树

决策树是一个类似于流程图的树结构，其中，每个内部结点表示在一个属性上的测试，每个分枝代表一个测试输出，而每个树叶结点代表类或类分布。树的顶层结点是根结点。内部结点用矩形表示，而树叶用椭圆表示。从根到叶子结点都有一条路径，这条路径就是一条“规则”。Oracle 支持决策树算法，快速预测结果。



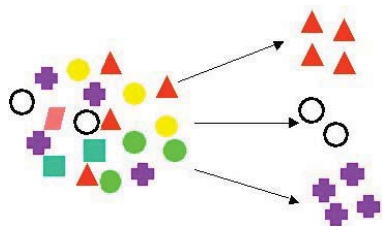
## 32.3 无监督式学习算法

在“无监督式学习”中，用户不为数据挖掘算法指定目标属性。无监督式学习技术（如关联和聚类算法）不对目标字段做出假定。相反，它们允许数据挖掘算法在独立于先前所定义业务目标的数据中找出关联和聚类。

Oracle 数据挖掘支持的无监督式数据挖掘功能包括聚类、关联、特征抽取和异常检测。



## 32.3.1 聚类



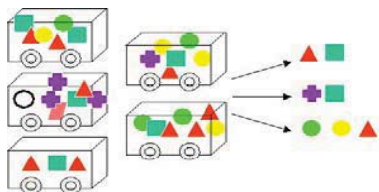
Oracle 数据挖掘提供两个聚类算法：增强的 k-Means 和正交分区聚类（O-Cluster）。

ODM 的增强 k-Means（EKM）和 O-Cluster 算法支持识别数据群中自然存在的分组。ODM 的 EKM 算法支持分级聚类，处理数字属性，并且会将数据群缩减为用户指定的聚类数量。

ODM 的 O-Cluster 算法既处理数字型属性又处理分类属性，并且会自动选择最佳的聚类定义。在这两种情况下，ODM 提供聚类详细信息、聚类规则、聚类质心值，并可根据数据群的从属关系进行“计分”。

例如，增强的 k-Means 聚类算法可用于揭示某疾病人群中的各个子群，自动对相关个体成员组成一个子群。

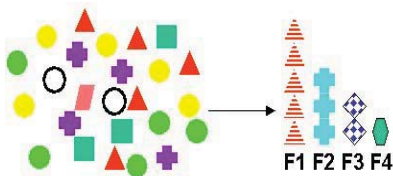
## 32.3.2 关联规则



ODM 的关联规则（AR）在数据中找出同时存在的项或事件关系。AR 经常被称为“市场组合分析”，它计算各项中每个可能的配对、三联、四联等组合的数量以找出模式。关联规则以前因后果的形式表示所发现的结果。在众多已发现的规则中，有一条 AR 规则可能是“顾客购买了牛奶、面包和果酱，则还购买黄油”的可信度为 78%，支持度为 12%。简单来说，该规则意味着如果发现“市场组合”具有前三项，则也会有很大的机会（78%的可信度）发现第四项，并且在所有研究的“市场组合”中有 12%的情况出现了这种组合。这样发现的关联或“规则”可用于设计专门的促销、产品捆绑和存货显示。

AR 可用于找出哪些生产部件和设备设置与故障事件相关、哪些患者及药品属性与哪些结果相关，或者购买了 A 项的人最有可能购买哪些项或产品。

## 32.3.3 特征选择



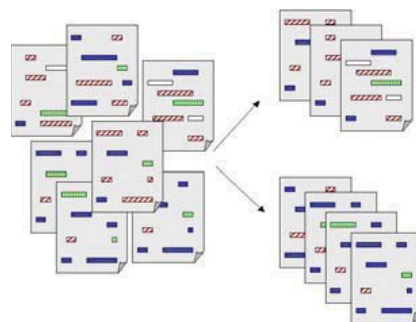
ODM 的非负矩阵因式分解（NMF）可用于将大型数据集缩减为有代表性的属性。NMF 与主成分分析（PCA）在概念上相似，但它能够处理更大量的属性，并且在附加的表示模型中，它是一种功能强大的优秀数据挖掘算法，可用于多种情况。

NMF 可用于将大量的数据（例如文本数据）缩减为更小、更稀疏的表示方法，这种表示方法减少了数据的维度，也就是说，可以使用很少的变量来保存相同的信息。可以使用监督式学习技术（如 SVM）或无监督式学习技术（如聚类技术）来分析 NMF 模型的输出。Oracle 数据挖掘使用 NMF 和 SVM 算法来挖掘非结构化的文本数据。



### 32.3.4 文本挖掘和非结构化数据

Oracle 数据挖掘作为 Oracle 数据库的一部分，用于挖掘结构化数据（也就是以行和列组织起来的分栏数据）和非结构化数据。ODM 可以挖掘非结构化数据，即作为单一文本属性的“文本”，可以与其他结构化数据（如年龄、身高和体重）相结合，以构建分类、预测和聚类模型。例如，ODM 可以将医师的注释添加到结构化的“临床”数据中，以提取更多信息并构建更好的数据挖掘模型。



这种将结构化数据与非结构化数据相结合的功能为挖掘数据提供了新的机遇。例如，执法人员可以构建基于年龄、前科数量、收入等来预测犯罪行为的模型，并结合警官关于个人的注释来构建更加准确的模型，这种模型利用了所有可用的信息。

## 32.4 数据挖掘接口

Oracle 数据挖掘在 Oracle 数据库中提供其数据挖掘功能的 3 个接口：

(1) ODM Java API: ODM 的 Java API 提供对数据挖掘函数的程序化控制，能够自动进行数据准备、建模和模型计分操作，以构建高级 BI 应用程序。

(2) ODM DBMS\_DM PL/SQL API: 应用程序开发人员可以使用 ODM 的 PL/SQL API 来构建高级 BI 应用程序，它利用一系列可以在 PL/SQL 程序块中调用的 SQL 元素。ODM 的 PL/SQL API 提供了一种大多数 Oracle 服务器开发人员和数据库管理员 (DBA) 所熟悉的语言和开发方法。

(3) Oracle Data Miner: Oracle Data Miner 为数据分析人员提供了一个图形用户界面，用于构建、评估和应用数据挖掘模型，如图 32-2 所示。Oracle Data Miner 指导数据分析人员完成数据挖掘过程。ODM Data Miner 支持同时构建模型和应用模型，因为它可以自动生成与数据挖掘操作相关联的 Java 和/或 SQL 代码。

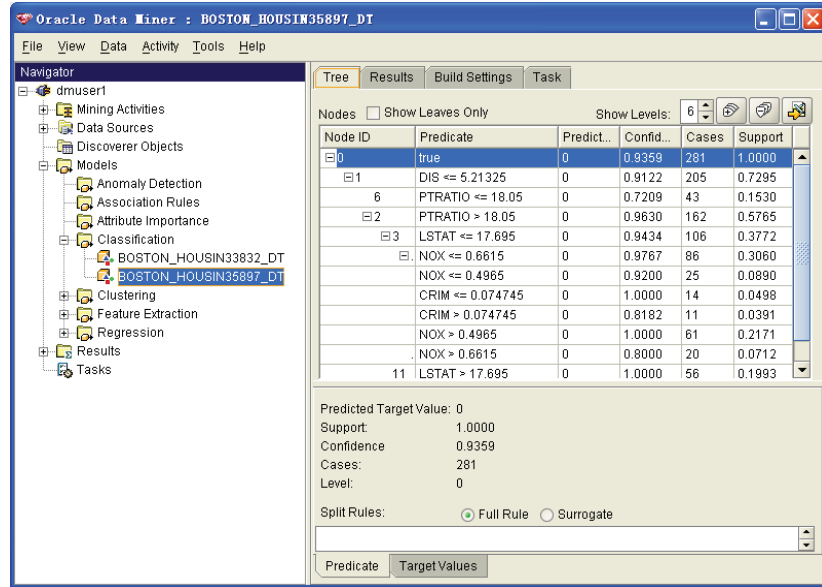


图 32-2 Oracle Data Miner



## 32.5 数据挖掘功能增强

在数据挖掘方面，Oracle 数据库 11gr1 主要简化了数据挖掘流程的管理，增强了数据挖掘与 SQL 的集成性，以及引入了新的挖掘算法。

### 32.5.1 自动和嵌入的数据转换

数据模型的质量依赖于数据的质量，数据挖掘项目中大量的时间用于数据准备。无论是训练（创建模型）数据还是被应用打分的数据必须认真地被查看、分析、转换和清洗。Oracle 数据挖掘提供了相关功能，显著简化了数据准备的流程。

#### 1. 自动数据准备（ADP）。

Oracle 数据挖掘有自动数据准备模式，当这个模式被激活时，Oracle 自动完成与算法相关的部分数据准备。这些数据转换指令与用户个性化的数据转换指令一起被嵌入到数据模型中。

##### （1）自动数据准备激活

自动数据准备是一个参数。默认方式是非激活的，通过在配置表中指出 `PREP_AUTO`，激活 ADP。配置表是一个用户自定义的表，一共两列，名称分别为 `SETTING_NAME` 和 `SETTING_VALUE`。为了激活 ADP，设置 `PREP_AUTO` 等于 `PREP_AUTO_ON`；为了关闭 ADP，设置 `PREP_AUTO` 等于 `PREP_AUTO_OFF`。

下面的语句激活 ADP：

```
BEGIN
INSERT into settings_tbl(
dbms_data_mining.prep_auto,
dbms_data_mining.prep_auto_on);
commit;
END;
```

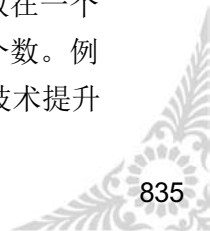
如果要在模型 `New_Model` 中使用这个激活的 ADP 设置，则执行如下语句：

```
BEGIN
dbms_data_mining.create_model(
model_name => 'clas_model',
mining_function => dbms_data_mining.classification,
data_table => 'my_data',
case_id_column_name => 'case_id',
target_column_name => 'class',
settings_table_name => 'settings_tbl');
END;
```

##### （2）算法特定的转换

分箱、规范化以及越界处理是数据挖掘算法通常都需要的转换操作。

**分箱：**也称为离散化，是一种减少连续或离散数据基数的方法，它通过分组将相关的值放在一个箱内（或集合内），不相关的值放在不同的箱内（或集合内），从而减少了用于算法的不同值个数。例如，创建模型的表中存在雇员工资一列，而工资不同值的个数为 1000 个，这样可以使使用分箱技术提升





挖掘性能，分为 10 个箱：[0—1000]，[1000—2000]，…，[8000—9000]，9000 以上等。常用的分箱方法包括等宽、有监督的、TOP-N 频繁项、分位数等。

分箱可以在几乎不丢失模型质量的情况下极大地提高创建和使用模型的性能。

**注：**分箱主要用于朴素贝叶斯和属性重要性算法。在 Oracle 数据挖掘决策树算法中，使用自己特有的分箱技术。

**规范化：**规范化是一种常用的技术，用来减少数据列数值的范围。最常用的规范化，是将一列值从一个大的取值范围映射为 0 和 1 之间的数值。例如，将收入值映射为（当前值-收入最小值）/（收入最大值-收入最小值）

**越界处理：**越界是指一个列值非常明显地偏离了这个列中的大多数数据。越界能严重影响模型的质量，同时也会影响分箱和规范化的效果。越界处理通常是采用修剪和 Winsorizing 的方式。越界处理应当谨慎，结合业务知识，判断偏离比较大的数值是否合理。

### （3）算法和自动数据准备

表 32-1 中列出了每种算法中 ADP 如何准备数据。

表 32-1 每种算法中 ADP 如何准备数据

算 法	功 能	ADP 的行为
朴素贝叶斯	分类	所有属性被有监督的分箱
决策树	分类	ADP 设置不起作用，算法会自身做数据准备
GLM	分类和回归	所有数值类型的属性被范式化
SVM	分类和回归	所有数值类型的属性被范式化
k-Means	聚类	所有数值类型的属性被范式化，使用越界敏感的范式化技术
O-Cluster	聚类	所有数值类型被分箱，使用等宽的分箱技术，数据列如果不同值的个数只有一个（或全是 NULL），则不被算法使用
MDL	属性重要性	所有属性被有监督的分箱
Apriori	关联规则	ADP 设置不起作用，算法会自身做数据准备
NMF	特征抽取	所有数值类型的属性被范式化

## 2. 嵌入数据准备

模型创建时使用的数据转换指令被嵌入到了模型中，不但在模型创建时使用，模型部署应用中也都可以自动使用。转换既可以嵌入到模型中被 ADP 自动激活，也可以作为用户指定的转换指令嵌入。

为了指明嵌入的是用户自定义转换，需要创建 TRANSFORMATION\_LIST 并且将它传递给 DBMS\_DATA\_MINING.CREATE\_MODEL。例如：

```
PROCEDURE create_model(
model_name IN VARCHAR2,
mining_function IN VARCHAR2,
data_table_name IN VARCHAR2,
case_id_column_name IN VARCHAR2,
target_column_name IN VARCHAR2 DEFAULT NULL,
settings_table_name IN VARCHAR2 DEFAULT NULL,
data_schema_name IN VARCHAR2 DEFAULT NULL,
settings_schema_name IN VARCHAR2 DEFAULT NULL,
xform_list IN TRANSFORM_LIST DEFAULT NULL);
```



### (1) 转换列表和 ADP

- ◎ 如果激活了 ADP，并在创建模型时指定了转换列表，则转换列表被嵌入到模型中，并且转换列表在系统自动转换之前被执行。
- ◎ 如果激活了 ADP，没有指定转换列表，则只有系统自动转换被嵌入到模型中。
- ◎ 如果没有激活 ADP，但指定了转换列表，则只有被个性化指定的转换被嵌入到模型中。
- ◎ 如果没有激活 ADP，也没有指定转换列表，则没有转换被嵌入到模型中，必须自己转换各种数据集（创建、测试、打分）。这种方法是 11g 之前版本采纳的。

### (2) 创建转换列表

转换列表是属性转换表达式的集合，每个属性转换表达式指定了一个属性的转换规则。属性表达式的部件如表 32-2 所示。

表 32-2 属性表达式的部件

字段名	数据类型	描述
attribute_name	VARCHAR2(30)	被构建模型的数据列名
attribute_subname	VARCHAR2(4000)	嵌套的属性名，如果这个列不是嵌套的，则值为 Null
Expression	EXPRESSION_REC	一个 SQL 表达式，指明了如何转换属性
reverse_expression	EXPRESSION_REC	一个 SQL 表达式，指明了如何反向转换。反向表达式支持模型的透明性
attribute_spec	VARCHAR2(4000)	Null 或 NORREP。当 ADP 激活时，NORREP 会阻止自动数据转换对此属性的数据转换。NORREP 不用于嵌套的属性。在嵌套属性中，如果指定了 NORREP 并且 ADP 激活，则返回错误

下面的例子显示了一个属性转换表达式，它相关的列名称为 INCOME，属于非嵌套列。

```
( ' INCOME,
  ' ',
  '(log(10, income) - 4.3)/0.7',
  'power(10, 0.7*income + 4.3)' )
```

### 3. 个性化数据准备的工具

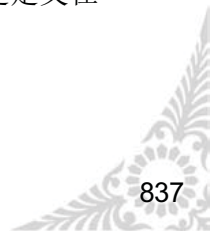
为了帮助用户建立自己的数据转换指令，Oracle 数据挖掘提供了数据转换的标准程序。借助这个新特性，可以做到：

- ◎ 为了减轻最终用户的工作量，提供了针对数据挖掘模型的自动的数据准备，并且提供了开箱即用的超模型；
- ◎ 将模型元数据与转换封装在一起，简化了模型的部署（数据转换与模型一起部署）

总之，通过增加对数据自动准备和嵌入的支持，简化了模型的建立和打分。

#### 32.5.2 数据挖掘方案对象

在以往的数据库版本中，数据挖掘模型并不同其他数据库对象定义在同一个层次上，而是定义在一个单独的方案中。11gr1 的增强表现在：



- ⊙ 删除了 DMSYS 方案，迁移必需的信息到 SYS 方案；
- ⊙ 增加的目录视图更适合输出数据挖掘模型的元数据；
- ⊙ 增加特定的权限控制数据，用于挖掘模型的建立、浏览和部署应用。

总之，数据挖掘模型现在可以和其他数据库对象一样，在 OEM 中管理，例如分配访问权限等。

### 32.5.3 数据挖掘的 Java API

JSR-247 支持 11g 提供的新的算法和功能，例如多变量线性回归和多变量逻辑回归。Oracle 数据挖掘增强的 Java API 符合 Java 数据挖掘标准 JSR-73 和 JSR-247。

### 32.5.4 广义线性模型

一种新的算法，广义线性模型在 11gr1 中引入。它支持两种数据挖掘功能：分类（二元逻辑回归）和回归（多变量线性回归）。一种新的数据挖掘函数，PREDICTION\_BOUNDS，被引入支持广义线性模型，它针对回归模型返回置信区间，针对分类模型返回预测概率。

二元变量逻辑回归算法，实现了广义线性模型，可以用于数据分类。广义线性模型分类算法利用线性模型技术生成了带有关联概率的预测。

多变量线性回归实现了针对回归的广义线性模型（GLM）。GLM 回归使用线性模型技术生成连续的预测值。GLM 回归支持预测概率的信用间隔。

### 32.5.5 预测分析：PROFILE

10gr2 引入的预测分析包，提供了一个面向业务人员的简单接口，不再需要更多统计知识背景，提供的功能包括 PREDICT（分类和回归）和 EXPLAIN（属性重要性）。11gr1 增加了一个新的功能 PROFILE，它利用决策树算法，给出目标属性的特征信息。

在已知目标属性的情况下，PROFILE 自动对记录排序并找到相似记录之间的关系，并且标识出一组被预测对象的属性上存在的共同特征。例如，如果零售商店办理了一次大型营销活动，可以用下面脚本找出响应营销活动的用户与不响应营销活动的用户各自特征：

```
--假定目标列为 affinity_card, 结果表名为 explain_res
BEGIN
DBMS_PREDICTIVE_ANALYTICS.PROFILE( data_table_name=> 'customers',
explain_column_name => 'affinity_card',
result_table_name=> 'explain_res');
END;
```

### 32.5.6 SQL 预测增强

3 个用于数据挖掘的 PL/SQL 包：

(1) DBMS\_DATA\_MINING，提供常规的创建、测试、打分、输入或输出数据挖掘模型功能。包里的过程包括 CREATE\_MODEL、DROP\_MODEL、RENAME\_MODEL、APPLY、EXPORT\_MODEL、



IMPORT\_MODEL 等。

(2) DBMS\_DATA\_MINING\_TRANSFORM, 提供常规的数据转换功能。因为从 11g 开始支持自动数据转换, 除非需要特殊的转换, 否则这个包不再使用。

(3) DBMS\_PREDICTIVE\_ANALYTICS, 包括完成预测分析以及数据挖掘流程自动化的功能。包含的过程有:

- ⊙ EXPLAIN——将各属性对目标的影响排名;
- ⊙ PREDICT——根据输入数据预测目标列的值;
- ⊙ PROFILE——对具有相同目标值的对象生成决策树规则。

数据挖掘应用打分功能嵌入 SQL 引入于 10gr2, 在 11gr1 中功能得到增强, 更方便了在 SQL 应用中嵌入数据挖掘。例如使用 SVM 模型打分, 其中 output 为输出表包含了预测值, SVM\_Clas\_Model 为 SVM 模型, svm\_data 是数据表。打分过程如下:

```
SQL>CREATE table output
(cust_id NUMBER,
prediction NUMBER,
probability NUMBER);
SQL>EXEC dbms_data_mining.apply
('SVM_Clas_Model','svm_data','cust_id','output');
```

Oracle 数据挖掘提供的预测类函数包括:

- ⊙ PREDICTION, 返回对目标的预测;
- ⊙ PREDICTION\_BOUNDS, 返回上限和下限;
- ⊙ PREDICTION\_COST, 返回错误预测目标的成本;
- ⊙ PREDICTION\_DETAILS, 返回包含打分解的 XML 字符串;
- ⊙ PREDICTION\_PROBABILITY, 返回预测概率;
- ⊙ PREDICTION\_SET, 返回相关的一系列对象。

此外, 11gr1 中允许成本矩阵用于预测模型, 提供基于成本的简单、准确的预测。它使用两个过程: DBMS\_DATA\_MINING.ADD\_COST\_MATRIX 和 DBMS\_DATA\_MINING.REMOVE\_COST\_MATRIX。

成本矩阵可以在分类模型中增加或删除。

### 32.5.7 新增视图

Oracle 11g 中新增的与数据挖掘相关的视图包括:

- ⊙ ALL/DBA/USER\_MINING\_MODELS: 提供数据挖掘模型的信息;
- ⊙ ALL/DBA/USER\_MINING\_ATTRIBUTES: 提供用于构建数据挖掘模型的数据列信息;
- ⊙ ALL/DBA/USER\_MINING\_MODEL\_SETTINGS: 提供数据挖掘模型的配置信息。



### 32.5.8 安全性

11g 之前的版本，数据挖掘使用 DMSYS 用户和方案，因此为了使用数据挖掘，需要给这个非系统账户赋予高级用户权限。

在 11gr1 中，数据挖掘使用的权限和角色使用方法，举例如下：

首先使用系统用户 SYS 或 SYSTEM 建立用户数据挖掘 dmuser，假定用户口令为 dmuser\_pwd，默认表空间和临时表空间分别为 tbl\_def 和 tbl\_tmp。

```
CREATE USER dmuser IDENTIFIED BY dmuser_pwd
DEFAULT TABLESPACE tbl_def
TEMPORARY TABLESPACE tbl_tmp
QUOTA UNLIMITED on tbl_def;
-- 建立数据挖掘模型必须有以下权限
GRANT create mining model TO dmuser;
GRANT create procedure TO dmuser;
GRANT create session TO dmuser;
GRANT create table TO dmuser;
GRANT create sequence TO dmuser;
GRANT create view TO dmuser;
GRANT create job TO dmuser;
GRANT create type TO dmuser;
GRANT create synonym TO dmuser;
```

dmuser 能够访问挖掘所使用的数据源也是必需的权限，如被访问的表是 CUSTOMERS。

```
GRANT SELECT ON customers TO dmuser;
```

如果使用文本数据挖掘，则下面的权限是必需的。

```
GRANT EXECUTE ON ctxsys.ctx_ddl TO dmuser;
```

### 32.5.9 数据挖掘模型的升级

- ◎ 10gr2 的元数据和模型都可以升级为 11gr1；
- ◎ 10gr1 的数据挖掘 PL/SQL API 可以升级到 11gr1，但是数据挖掘 Java API 不可以从 10gr1 升级到 11gr1，因为 10gr1 的 Java API 从 10gr2 开始就不被支持了。
- ◎ Oracle 9i 的数据挖掘模型不可以升级到 11gr1。

### 32.5.10 11g 中不赞成使用的特性

- (1) 自适应 Bayes 网络分类算法，被决策树算法取代。
  - (2) DM\_USER\_MODELS 视图和功能，被数据字典视图取代。
  - (3) 一些 PL/SQL 过程不再被赞成使用。
- ◎ GET\_DEFAULT\_SETTINGS: 被数据字典视图 USER/ALL/DBA\_MINING\_MODEL\_SETTINGS 代替；



- ◎ GET\_MODEL\_SETTINGS: 被数据字典视图 USER/ALL/DBA\_MINING\_MODEL\_SETTINGS 代替;
- ◎ GET\_MODEL\_SIGNATURE: 被数据字典视图 USER/ALL/DBA\_MINING\_MODEL\_ATTRIBUTES 代替。

### 32.5.11 不被支持的特性

- ◎ DMSYS 方案不再作为数据挖掘的元数据方案存在, 在 DBA\_REGISTRY 中也不再将数据挖掘作为一个部件;
- ◎ 数据挖掘打分引擎不再被支持;
- ◎ 之前版本的数据库助手 DBCA 可以安装数据挖掘选件, 11gr1 不再支持这种方式安装;
- ◎ 基本局部排比搜索工具 (BLAST 算法) 不再支持;
- ◎ DM\_USER\_MODELS 视图和功能被数据字典取代。

## 32.6 本章小结

本章主要介绍了 Oracle 数据挖掘的基础知识, 包括数据挖掘方法论和 Oracle 数据库提供的主要挖掘算法, 并介绍了 11g 中数据挖掘的新特性。

