

Oracle Database 11g: 面向管理员的新增功能

学生指南第 3 册

D50081CN21

版本 2.1

2010 年 5 月

D66318

ORACLE®

作者

Christian Bauwens
Maria Billings
Mark Fuller
Peter Fusek
Richard Green
Christine Jeal
Donna Keesling
Deidre Matishak
James Spiller
Jenny Tsai
Jean-Francois Verrier
Jim Womack
Marcie Young

技术撰稿人和审稿人

Maqsood Alam
Kalyan Bitra
John Boyle
Harald Van Breederode
Sharath Bhujani
Immanuel Chan
Timothy Chien
Edward Choi
Jacco Draaijer
Al Flournoy
Steve Fogel
Andy Fortunak
Gerlinde Frenzen
Greg Gagnon
GP Gongloor
Joel Goodman
Hansen Han
Uwe Hesse
Sunil Hingorani
Magnus Isaksson
Susan Jang
Martin Jensen
Dominique Jeunot
Pete Jones
Yash Kapani
Pierre Labrousse
Richard.W.Lewis
Hakan Lindfors
Russ Lowenthal
Kurt Lysy
Isabelle Marchand
Silvia Marrone
Heejin Park
Srinivas Putrevu
Jagannath Poosarla
Surya Rekha
Tim Shetler
Eric Siglin
Ranbir Singh
Jeff Skochil
George Spears
Kesavan Srinivasan
Birgitte Taagholt
Glenn Tripp
Branislav Valny
Anthony Woodell

版权所有 © 2010, Oracle。保留所有权利。

免责声明

本文档包含专有权信息，并受版权法和其它知识产权法的保护。您可以复制和打印本文档，但只能在 Oracle 培训课程中使用。不得以任何方式修改或变更本文档。除了在依照版权法中制定的“合理使用”范围内使用本文档外，在未经 Oracle 明确授权的情况下，您不得以全部或部分的形式使用、共享、下载、上载、复制、打印、显示、展示、再版、发布、许可、张贴、传播或散布本文档。

本文档中包含的信息如有更改，恕不另行通知。如果您在本文档中发现任何问题，请书面通知：Oracle University, 500 Oracle Parkway, Redwood Shores, California 94065 USA。Oracle 不保证本文档中没有错误。

有限权利声明

如果将本文档交付给美国政府或代表美国政府使用本文档的任何人，则适用以下通知中的规定：

U.S. GOVERNMENT RIGHTS

The U.S. Government's rights to use, modify, reproduce, release, perform, display, or disclose these training materials are restricted by the terms of the applicable Oracle license agreement and/or the applicable U.S. Government contract.

商标声明

Oracle 是 Oracle 公司和（或）其分公司的注册商标。其它名称可能是其各自拥有者的商标。

编辑

Aju Kumar
Amitha Narayan

出版商

Sujatha Nagendra
Michael Sebastian Almeida
Jobi Varghese

目录

I 简介

- 概览 I-2
- Oracle DB 创新 I-3
- 企业网格计算 I-4
- Oracle Database 11g: 侧重的领域 I-5
- 管理自动化 I-7
- 自助管理数据库: 下一代 I-8
- 其它建议课程 I-9
- 更多信息 I-10
- 建议计划 I-11

1 Oracle Grid Infrastructure

- 课程目标 1-2
- Oracle Grid Infrastructure 1-3
- 自动存储管理技术堆栈 1-4
- Oracle Grid Infrastructure 和 Oracle DB 安装: 系统要求 1-5
- 准备操作系统 1-6
- 设置环境变量 1-7
- 检查系统要求 1-9
- 定义用于 ASM 的 OS 设备的所有权 1-10
- 安装场景 1-11
- 第一部分: 安装用于独立服务器的 Oracle Grid Infrastructure 1-12
- 选择产品语言 1-13
- 创建 ASM 磁盘组 1-14
- 定义 ASM 口令 1-15
- 定义授权的操作系统组 1-16
- 指定安装位置 1-17
- 创建清单 1-18
- 执行先决条件检查 1-19
- 验证安装概要数据 1-20
- 监视安装进度 1-21

- 执行 root 配置脚本 1-22
- 执行 Configuration Assistant 1-23
- 完成安装 1-24
- 配置 FRA 磁盘组 1-25
- Oracle 本地注册表 1-26
- 测验 1-28
- 练习 1-1: 概览 1-29
- ASM 文件和卷 1-30
- ASM 动态卷管理器 (ADVM) 概念 1-31
- ASM 动态卷条带化 1-32
- ADVM 限制 1-33
- 创建 ASM 动态卷 1-34
- 管理 ASM 动态卷 1-36
- 使用 SQL 管理 ASM 动态卷 1-37
- 用于 ASM 动态卷的 ASMCMD 扩展命令 1-39
- ASM 集群文件系统 (ACFS) 1-40
- ACFS 体系结构 1-41
- 使用 Enterprise Manager 管理 ACFS 1-42
- Linux/UNIX 文件系统 API 1-43
- Linux/UNIX 扩展命令 1-44
- Windows 文件系统 API 1-45
- Windows 扩展命令 1-46
- ACFS 独立于平台的命令 1-47
- ASM 集群文件系统限制 1-48
- 创建 ASM 集群文件系统 1-49
- ACFS 快照 1-50
- 管理 Oracle ACFS 快照 1-51
- 测验 1-52
- 小结 1-53
- 练习 1-2: 概览 1-54

2 安装增强功能

课程目标 2-2

Oracle Database 11g 安装：更改 2-3

第二部分：安装 Oracle DB 软件 2-6

选择安装类型 2-7

选择网格安装选项 2-8

选择语言设置 2-9

选择数据库版本 2-10

指定安装位置 2-11

选择操作系统组 2-12

执行先决条件检查 2-13

安装概要页 2-14

安装产品页 2-15

安装完成页 2-16

测验 2-17

练习 2-1：概览 2-18

Oracle Database 11g 发行版 2 升级路径 2-19

Oracle Database 11g 发行版 1 和发行版 2 中废弃的功能 2-20

初始化参数信息 2-21

练习 2-2：概览 2-23

练习 2-3：概览 2-24

Direct NFS 客户机：概览 2-25

Direct NFS 配置 2-27

监视 Direct NFS 2-29

联机打补丁：概览 2-30

安装联机补丁程序 2-31

联机打补丁的优点 2-32

常规打补丁和联机打补丁 2-33

联机打补丁注意事项 2-34

测验 2-36

小结 2-37

练习 2-4：概览 2-38

3 Oracle Restart

- 课程目标 3-2
- Oracle Restart 3-3
- Oracle Restart 进程的启动 3-5
- 控制 Oracle Restart 3-6
- 选择正确的 SRVCTL 实用程序 3-8
- Oracle Restart 配置 3-9
- 使用 SRVCTL 实用程序 3-10
- 获取有关 SRVCTL 实用程序的帮助 3-11
- 使用 SRVCTL 实用程序启动组件 3-12
- 使用 SRVCTL 实用程序停止组件 3-13
- 查看组件状态 3-14
- 显示组件的 Oracle Restart 配置 3-15
- 手动向 Oracle Restart 配置添加组件 3-16
- 测验 3-17
- 小结 3-18
- 练习 3-1: 概览 3-19

4 ASM 增强功能

- 课程目标 4-2
- 无 ASM 快速镜像重新同步时 4-3
- ASM 快速镜像重新同步: 概览 4-4
- 使用 Enterprise Manager 执行快速镜像重新同步 4-5
- 设置 ASM 快速镜像重新同步 4-7
- ASM 首选镜像读取: 概览 4-9
- ASM 首选镜像读取: 设置 4-10
- Enterprise Manager 的 ASM 配置页 4-11
- ASM 首选镜像读取: 最佳实践 4-12
- 测验 4-13
- ASM 可伸缩性和性能增强 4-14
- Oracle Database 11g 中的 ASM 可伸缩性 4-16
- SYSASM 的权限 4-17
- 使用 Oracle Enterprise Manager 管理 ASM 用户 4-18
- ASM 磁盘组兼容性 4-19
- ASM 磁盘组属性 4-21
- 使用 Enterprise Manager 编辑磁盘组属性 4-22

- 增强的磁盘组检查 4-23
- 适用于快速重新平衡的受限装载磁盘组 4-24
- 强制装载磁盘组 4-25
- 强制删除磁盘组 4-27
- Oracle Database 11g 发行版 1 中的 ASMCMD 扩展 4-28
- ASMCMD 命令扩展 4-30
- ASMCMD 扩展：示例 4-31
- 测验 4-32
- ASM ACL 4-33
- ASM ACL 的先决条件 4-34
- 使用 SQL 命令管理 ASM ACL 4-36
- 使用 ASMCMD 命令管理 ASM ACL 4-37
- 使用 Enterprise Manager 管理 ASM ACL 4-38
- ASM 智能数据放置 4-40
- 管理 ASM 智能数据放置的示例 4-41
- 管理 ASM 智能数据放置 4-42
- ASM 智能数据放置的最佳实践 4-44
- 查看 ASM 智能数据放置的相关信息 4-45
- 测验 4-47
- 小结 4-48
- 练习 4：概览 4-49

5 存储增强功能

- 课程目标 5-2
- 支持 4 KB 扇区磁盘 5-3
- 使用 4 KB 扇区磁盘 5-4
- 指定磁盘扇区大小 5-5
- 使用 SECTOR_SIZE 子句 5-6
- 创建使用 4 KB 扇区磁盘的数据库 5-7
- 指定 BLOCKSIZE 5-8
- 确定日志文件块大小 5-9
- 执行到 4 KB 磁盘的脱机移植 5-10
- 测验 5-12
- 表压缩：概览 5-13
- 表压缩的概念 5-14
- 压缩表数据 5-15

- 使用 OLTP 压缩 5-17
- 使用压缩指导 5-18
- 查看表压缩信息 5-19
- SQL 访问指导：概览 5-20
- SQL 访问指导：使用模型 5-21
- 可能的建议 5-23
- 使用 SQL 访问指导 5-24
- SQL 访问指导：PL/SQL 过程流程 5-25
- SQL 访问指导：PL/SQL 示例 5-26
- 临时表空间收缩 5-27
- DBA_TEMP_FREE_SPACE 5-28
- 用于创建临时表的表空间选项 5-29
- 按需创建段 5-30
- 创建没有段的表 5-31
- 控制延迟创建段功能 5-32
- 限制和例外 5-33
- 其它自动功能 5-34
- 测验 5-35
- 小结 5-36
- 练习 5：概览 5-37

6 数据仓库和分区增强功能

- 课程目标 6-2
- 为 ORACLE_LOADER 访问驱动程序预处理外部表中的数据 6-3
- Windows 示例 6-4
- 预处理示例 6-5
- 对目录对象的 EXECUTE 权限 6-7
- 隐含的安全问题 6-8
- 确保 ORACLE_LOADER 访问驱动程序的安全 6-10
- 注意事项和使用说明 6-12
- 测验 6-13
- 回顾：并行度 (DOP) 6-14
- 回顾：PARALLEL 子句 6-15
- 目标：简化并行执行 6-16
- 自动确定并行度 6-17
- 参数变更概要 6-18

使用 PARALLEL_MIN_TIME_THRESHOLD	6-19
使用 PARALLEL_DEGREE_POLICY	6-20
使用 PARALLEL_DEGREE_LIMIT	6-22
使用 PARALLEL_FORCE_LOCAL	6-24
对现有参数的更改	6-25
并行提示现在位于语句级别	6-26
语句级并行提示的意义	6-27
EXPLAIN PLAN 增强功能	6-28
增强的 Explain Plan 示例	6-29
内存中并行查询	6-30
测验	6-31
Oracle 分区	6-32
分区增强功能	6-33
间隔分区	6-34
间隔分区：示例	6-35
移动转换点：示例	6-36
系统分区	6-38
系统分区：示例	6-39
系统分区：准则	6-40
用于列表分区的系统管理的索引	6-41
基于虚拟列的分区	6-42
基于虚拟列的分区：示例	6-43
引用分区	6-44
引用分区：优势	6-45
引用分区：示例	6-46
组合分区增强功能	6-48
范围-范围分区：示例	6-49
小结	6-50
练习 6：概览	6-51

7 Oracle SecureFiles

课程目标 7-2

管理企业信息 7-3

现有 LOB 实施中的问题 7-4

Oracle SecureFiles 7-5

启用 SecureFiles 存储 7-6

SecureFiles: 存储选项 7-8

SecureFiles: 高级功能 7-9

创建 SecureFiles 7-10

使用 Enterprise Manager 创建 SecureFiles 7-11

共享 I/O 池 7-12

更改 SecureFiles 7-13

测验 7-14

什么是 SecureFile 压缩 7-15

压缩模式 7-16

使用 LOW 压缩创建 SecureFile LOB 7-17

配置参数 7-18

测验 7-19

访问 SecureFiles 元数据 7-20

移植到 SecureFiles 7-21

SecureFiles 移植: 示例 7-22

SecureFiles 监视 7-23

小结 7-24

练习 7: 概览 7-25

8 安全和联网增强功能

课程目标 8-2

安全口令支持 8-3

自动安全配置 8-4

口令配置 8-5

启用内置口令复杂性检查器 8-6

管理默认审计 8-7

审计的权限 8-8

设置安全性参数 8-9

设置数据库管理员验证 8-11

测验 8-12

- 透明数据加密 8-13
- 硬件安全模块 8-14
- 使用表空间加密 8-15
- 表空间加密的新增功能 8-16
- 重新生成表空间主密钥 8-17
- 测验 8-18
- Enterprise Manager 安全管理 8-19
- 使用 RMAN 安全性增强功能 8-20
- 管理对外部网络服务的细粒度访问 8-21
- 支持 IPv6 地址通知 8-23
- 连接到 Oracle DB 8-25
- Java 接口对 IPv6 的支持 8-26
- 小结 8-27
- 练习 8: 概览 8-28

9 SQL 性能分析器

- 课程目标 9-2
- DBA 执行更改时面临的挑战 9-3
- 计划永远赶不上变化 9-4
- Oracle Database 11g 中的更改管理 9-5
- 更改管理的生命周期 9-6
- SQL 性能分析器: 概览 9-8
- SQL 性能分析器: 用例 9-9
- 步骤 1: 捕获 SQL 工作量 9-11
- 步骤 2: 传送到测试系统 9-12
- 步骤 3: 构建更改前性能数据 9-13
- 步骤 4: 实施计划的更改以及步骤 5: 构建更改后性能数据 9-14
- 步骤 6: 比较和分析性能以及步骤 7: 优化回归的 SQL 9-15
- 测验 9-16
- 访问 SQL 性能分析器 9-17
- 使用 Oracle Enterprise Manager 访问 SQL 性能分析器 9-18
- SQL 性能分析: PL/SQL 示例 9-19
- 优化回归的 SQL 语句 9-22

测试数据库升级：Oracle9i Database 和 Oracle Database 10g 发行版 1 9-23

测试数据库升级：Oracle Database 10g 发行版 2 及更高版本 9-26

SQL 性能分析器：数据字典视图 9-29

小结 9-30

练习 9：概览 9-31

10 SQL 计划管理

课程目标 10-2

SQL 计划管理：概览 10-3

SQL 计划基线：体系结构 10-4

加载 SQL 计划基线 10-6

演化 SQL 计划基线 10-7

查看重要的基线 SQL 计划属性 10-8

重要的基线 SQL 计划属性 10-9

SQL 计划选择 10-11

测验 10-13

可能的 SQL 计划管理场景 10-14

SQL 性能分析器和 SQL 计划基线场景 10-15

自动加载 SQL 计划基线 10-16

清除 SQL 管理库策略 10-17

Enterprise Manager 和 SQL 计划基线 10-18

使用 MIGRATE_STORED_OUTLINE 函数 10-19

小结 10-20

练习 10：概览 10-21

11 数据库重放

课程目标 11-2

为什么使用数据库重放 11-3

使用数据库重放 11-4

整体情况 11-5

系统体系结构：捕获 11-6

系统体系结构：预处理工作量 11-7

系统体系结构：重放 11-8

更改前生产系统 11-9

支持的工作量 11-10

捕获注意事项 11-11

重放注意事项	11-12
重放选项	11-14
工作量重放过滤器	11-16
重放分析	11-18
测验	11-20
Enterprise Manager 中的数据库重放 workflow	11-21
访问数据库重放	11-22
程序包和过程	11-24
数据库重放: PL/SQL 示例	11-25
数据字典视图: 数据库重放	11-28
校准重放客户机	11-29
小结	11-31
练习 11: 概览	11-32

12 自动 SQL 优化

课程目标	12-2
Oracle Database 10g 中的 SQL 优化	12-3
Oracle Database 11g 中的自动 SQL 优化	12-4
Oracle Database 11g 中的自动化功能概要	12-5
选择可能的 SQL 语句进行优化	12-6
维护窗口时间表	12-7
自动优化进程	12-8
控制自动 SQL 优化任务	12-10
示例: 控制自动 SQL 优化任务	12-11
自动 SQL 优化任务	12-12
配置自动 SQL 优化	12-13
自动 SQL 优化结果概要	12-14
测验	12-15
自动 SQL 优化: 微调	12-16
使用 PL/SQL 接口生成报表	12-18
自动 SQL 优化注意事项	12-19
小结	12-20
练习 12: 概览	12-21

13 智能基础结构增强功能

- 课程目标 13-2
- 使用新增的和增强的自动工作量资料档案库视图 13-3
- 使用 AWR 基线进行比较性能分析 13-4
- 自动工作量资料档案库基线 13-5
- 移动窗口基线 13-6
- 基线显示选项 13-7
- 基线模板 13-8
- 创建 AWR 基线 13-9
- DBMS_WORKLOAD_REPOSITORY 程序包 13-10
- 针对单一时段生成基线模板 13-11
- 创建重复基线模板 13-12
- 基线视图 13-13
- 测验 13-14
- 性能监视和基线 13-15
- 使用静态基线定义预警阈值 13-17
- 使用 Enterprise Manager 快速配置自适应阈值 13-18
- 练习 13-1: 概览 13-19
- 维护窗口 13-20
- 默认维护计划 13-21
- 自动维护任务优先级 13-22
- 控制自动维护任务 13-23
- 测验 13-24
- Oracle DB 的重要 I/O 度量 13-25
- I/O 校准和 Enterprise Manager 13-27
- I/O 校准和 PL/SQL 接口 13-28
- I/O 统计信息: 概览 13-30
- I/O 统计信息和 Enterprise Manager 13-32
- 练习 13-2 和 13-3: 概览 13-34
- 资源管理器增强功能 13-35
- 资源管理器增强功能: 数据库合并 13-36
- 采用固定策略的 CPU 资源管理 13-37
- 限制 CPU 使用率 13-38
- 资源管理器增强功能: 服务器合并 13-39
- 实例封锁 13-40
- 实例封锁示例 13-41

启用实例封锁	13-42
监视实例封锁	13-43
测验	13-44
资源管理器：对 DBMS_RESOURCE_MANAGER 的更改	13-45
资源管理器：新的 Enterprise Manager 界面	13-46
默认创建的资源计划	13-47
默认计划	13-48
I/O 资源限制阈值	13-49
资源管理器统计信息	13-50
小结	13-51

14 易诊断增强功能

课程目标	14-2
Oracle Database 11g 故障管理	14-3
简化诊断：自动诊断工作流	14-4
自动诊断资料档案库	14-5
ADRCI：ADR 命令行工具	14-7
V\$DIAG_INFO	14-8
诊断跟踪的位置	14-9
使用 Enterprise Manager 查看预警日志	14-10
使用 ADRCI 查看预警日志	14-11
测验	14-13
问题和意外事件	14-14
意外事件打包服务 (IPS)	14-17
意外事件程序包	14-18
Enterprise Manager 支持工作台：概览	14-20
Enterprise Manager 支持工作台流程	14-21
在 Enterprise Manager 中查看严重错误预警	14-22
将诊断数据打包并上载到 Oracle 技术支持	14-23
跟踪 SR 并实施修复	14-24
创建用户报告的问题	14-25
Enterprise Manager 针对 ASM 的支持工作台	14-26
使用 ADRCI 调用 IPS	14-27
测验	14-29
健康状况监视器：概览	14-30
手动运行健康状况检查：Enterprise Manager 示例	14-32

手动运行健康状况检查：PL/SQL 示例 14-33
 使用 ADRICI 实用程序查看 HM 报告 14-34
 SQL 修复指导：概览 14-35
 使用 Enterprise Manager 访问 SQL 修复指导 14-36
 查看、禁用或删除 SQL 补丁程序 14-37
 从 PL/SQL 使用 SQL 修复指导：示例 14-38
 使用 SQL Test Case Builder 14-39
 测验 14-40
 小结 14-41
 练习 14：概览 14-42

15 实时 SQL 监视

课程目标 15-2
 SQL 监视 15-3
 Oracle Database 11g 发行版 2 中的 SQL 监视 15-5
 使用 Enterprise Manager Database Control 进行 SQL 监视 15-6
 监视的 SQL 执行 15-7
 SQL 监视列表 15-8
 监视的 SQL 执行详细资料 15-9
 并行查询的 SQL 执行详细资料 15-10
 并行执行的详细资料 15-11
 并行执行的活动详细资料 15-12
 查看会话详细资料 15-13
 SQL 详细资料 15-14
 查看 SQL 监视报表 15-15
 测验 15-16
 小结 15-17
 练习 15-1：概览 15-18

16 性能增强功能

课程目标 16-2
 使用 DBMS_ADDM 程序包 16-3
 已命名的指导查找结果和指令 16-6
 做了修改的指导视图 16-7
 新增的 ADDM 视图 16-8
 测验 16-9

回顾：Oracle Database 10g SGA 参数	16-10
回顾：Oracle Database 10g PGA 参数	16-11
Oracle DB 内存指导	16-13
自动内存管理：概览	16-15
Oracle Database 11g 内存参数	16-17
自动内存参数依赖性	16-18
启用自动内存管理	16-20
监视自动内存管理	16-21
DBCA 和自动内存管理	16-23
测验	16-24
DB 智能闪存高速缓存概览	16-25
使用 DB 智能闪存高速缓存	16-26
DB 智能闪存高速缓存体系结构概览	16-27
配置 DB 智能闪存高速缓存	16-29
调整 DB 智能闪存高速缓存的大小	16-31
为表指定 DB 智能闪存高速缓存	16-32
统计信息首选项：概览	16-33
使用新增的统计信息首选项功能	16-34
使用 Oracle Enterprise Manager 设置全局首选项	16-35
分区表和增量统计信息：概览	16-36
基于散列的列统计信息采样	16-38
多列统计信息：概览	16-39
表达式统计信息：概览	16-41
延迟统计信息发布：概览	16-42
延迟统计信息发布：示例	16-44
测验	16-45
锁定增强功能	16-46
识别前台进程事件和后台进程事件	16-47
小结	16-48
练习 16：概览	16-49

17 应用程序性能增强功能

- 课程目标 17-2
- 联机重新定义增强功能 17-3
- 细粒度依赖性管理 17-4
- 将从属对象重新编译减至最少 17-5
- 更精确的依赖性元数据 17-6
- 管理依赖性 17-7
- 减少失效的使用准则 17-9
- 不可见的索引：概览 17-10
- 不可见的索引：示例 17-11
- 自适应游标共享：概览 17-12
- 自适应游标共享：体系结构 17-13
- 自适应游标共享视图 17-15
- 与自适应游标共享交互 17-17
- SQL 查询结果高速缓存：概览 17-18
- 设置 SQL 查询结果高速缓存 17-19
- 管理 SQL 查询结果高速缓存 17-20
- 使用 RESULT_CACHE 提示 17-21
- 内嵌视图：示例 17-22
- 使用表注释控制结果高速缓存 17-23
- 使用 DBMS_RESULT_CACHE 程序包 17-24
- 查看 SQL 结果高速缓存字典信息 17-25
- SQL 查询结果高速缓存：注意事项 17-26
- 测验 17-28
- OCI 客户机查询高速缓存 17-29
- 使用客户端查询高速缓存 17-30
- PL/SQL 函数高速缓存 17-32
- 使用 PL/SQL 函数高速缓存 17-33
- PL/SQL 函数高速缓存：注意事项 17-35
- PL/SQL 和 Java 本机编译增强功能 17-36
- 设置和测试 PL/SQL 本机编译 17-37
- 为 PL/SQL 本机编译重新编译整个数据库 17-38
- 小结 17-40
- 练习 17：概览 17-41

18 备份和恢复增强功能

课程目标 18-2

使用新增的 SET NEWNAME 子句 18-3

SET NEWNAME 的替代变量 18-4

使用 SET NEWNAME FOR DATABASE 18-5

使用 SET NEWNAME FOR TABLESPACE 18-6

性能增强功能 18-7

增强了块损坏的检测功能 18-8

使用二进制压缩的新设置 18-9

使用新的压缩算法设置 18-10

优化的备份 18-11

对超大型文件应用并行备份和还原 18-13

使用 RMAN 多节备份 18-14

测验 18-15

复制数据库 18-16

执行活动数据库复制 18-17

RMAN DUPLICATE 命令 18-18

使用 DUPLICATE 命令创建备用数据库 18-19

对数据库复制所做的增强 18-20

无目标 DUPLICATE 18-21

使用无目标 DUPLICATE 18-22

使用新增的 DUPLICATE 命令选项 18-23

DUPLICATE ... [SKIP] TABLESPACE 增强功能 18-24

使用 RMAN 创建归档备份 18-25

管理归档数据库备份 18-26

创建归档备份 18-27

丢失服务器参数文件后可以更轻松地恢复此文件 18-28

对 TSPITR 所做的增强和修改 18-29

确定跨越恢复集边界的关系 18-30

使用映像副本提高 TSPITR 性能 18-31

测验 18-32

管理恢复目录 18-33

IMPORT CATALOG 命令 18-35

创建和使用虚拟专用目录 18-37

使用 RMAN 虚拟专用目录 18-38
 测验 18-40
 小结 18-41
 练习 18: 概览使用 RMAN 增强功能 18-42

19 闪回技术、LogMiner 和数据泵增强功能

课程目标 19-2
 闪回数据归档: 概览 19-3
 闪回数据归档: 体系结构 19-5
 准备数据库 19-6
 闪回数据归档: 工作流 19-10
 使用闪回数据归档 19-11
 配置默认闪回数据归档 19-12
 填充闪回数据归档空间 19-13
 维护闪回数据归档 19-14
 闪回数据归档: 示例 19-15
 闪回数据归档: DDL 限制 19-16
 闪回数据归档: 支持透明的方案演化 19-17
 闪回数据归档: 支持完整的方案演化 19-18
 查看闪回数据归档 19-19
 准则和使用提示 19-20
 测验 19-21
 闪回事务处理回退 19-22
 先决条件 19-23
 闪回事务处理 19-24
 使用闪回事务处理向导 19-25
 使用 DBMS_FLASHBACK.TRANSACTION_BACKOUT 过程 19-26
 查看相关性报表 19-27
 查看闪回事务处理元数据 19-28
 测验 19-29
 闪回数据库增强功能 19-30
 使用 LogMiner 19-31
 回顾: 数据泵导出和导入 19-33
 以数据泵旧模式提供移植支持 19-34

数据泵旧模式	19-35
管理文件位置	19-37
测验	19-38
小结	19-39
练习 19: 概览使用闪回技术	19-40

20 数据恢复指导

课程目标	20-2
修复数据故障	20-3
数据恢复指导	20-4
评估数据故障	20-7
数据故障	20-8
数据故障: 示例	20-9
数据恢复指导 RMAN 命令行界面	20-10
列出数据故障	20-11
修复建议	20-13
执行修复	20-14
分类 (和关闭) 故障	20-15
测验	20-16
数据恢复指导视图	20-17
最佳实践: 预防性检查	20-18
设置检测损坏的参数	20-19
小结	20-21
练习 20: 概览修复故障	20-22

附录 A: 练习与解答

附录 B: 调度程序增强功能

课程目标	B-2
轻量作业	B-3
选择正确的作业	B-4
调度程序电子邮件通知: 概览	B-5
使用调度程序电子邮件通知	B-6
设置 email_server 属性	B-7
设置 email_sender 属性	B-8
使用 ADD_JOB_EMAIL_NOTIFICATION	B-9

使用 REMOVE_JOB_EMAIL_NOTIFICATION B-11

电子邮件通知：新增的字典视图 B-12

调度程序文件监视器：概览 B-13

创建文件监视器和基于事件的作业 B-14

步骤 1：创建调度程序身份证明对象 B-15

步骤 2：创建文件监视器 B-16

使用 DBMS_SCHEDULER.CREATE_FILE_WATCHER B-17

步骤 3：创建调度程序程序对象 B-18

使用 SCHEDULER_FILEWATCHER_RESULT 对象类型 B-20

使用 SCHEDULER_FILEWATCHER_REQUEST 对象类型 B-21

步骤 4：创建引用文件监视器的基于事件的作业 B-22

步骤 5：启用对象 B-23

启用来自远程系统的文件到达事件 B-24

使用 DBMS_SCHEDULER.DROP_FILE_WATCHER B-25

文件监视器：对现有 DBMS_SCHEDULER 过程的更改 B-26

文件监视器：对现有属性的更改 B-27

文件监视器：新增的字典视图 B-28

文件监视器：现有视图中新增的列 B-30

调度程序远程数据库作业：概览 B-31

使用 CREATE_DATABASE_DESTINATION B-32

使用 DROP_DATABASE_DESTINATION B-33

创建远程数据库作业：配置 B-34

为远程作业设置数据库 B-35

创建远程数据库作业 B-36

调度程序多目标作业：概览 B-37

调度程序多目标作业：术语 B-38

创建多目标作业 B-40

在多目标作业中使用目标组 B-41

使用 CREATE_GROUP B-42

使用 ADD_GROUP_MEMBER B-43

使用 DROP_GROUP B-44

使用 REMOVE_GROUP_MEMBER B-45

对现有 DBMS_SCHEDULER 过程的修改 B-46

多目标作业：新增的字典视图 B-47

多目标作业：现有字典视图中新增的列 B-48

小结 B-49

附录 C: Oracle Secure Backup 云模块

课程目标 C-2

在云中备份 Oracle 数据库 C-3

Oracle Secure Backup 云模块简介 C-4

备份至云的优点 C-5

什么是 Amazon Simple Storage Service (S3) C-6

安装 Oracle Secure Backup 云模块 C-7

小结 C-8

These eKit materials are to be used ONLY by you for the express purpose SELF STUDY. SHARING THE FILE IS STRICTLY PROHIBITED.

附录 A

练习与解答

目录

第 1 课的练习.....	4
练习 1-1: 安装用于独立服务器的 Oracle Grid Infrastructure	5
练习 1-2: 配置 ASM 集群文件系统.....	11
第 2 课的练习.....	16
练习 2-1: 安装 Oracle Database 11g 软件	17
练习 2-2: 新建数据库	20
练习 2-3: 使用 ASM 集群文件系统快照.....	22
练习 2-4: 使用联机打补丁	28
第 3 课的练习.....	40
练习 3-1: 使用 Oracle Restart 管理组件	41
第 4 课的练习.....	48
练习 4-1: 使用 ASM 快速镜像重新同步.....	49
第 5 课的练习.....	56
练习 5-1: 使用表压缩	57
第 6 课的练习.....	64
练习 6-1: 使用引用分区	65
练习 6-2: 使用间隔分区	84
练习 6-3: 使用扩展的组合分区	96
练习 6-4: 使用基于虚拟列的分区	102
练习 6-5: 使用 SQL 访问指导获取分区建议.....	115
第 7 课的练习.....	119
练习 7-1: 使用 SecureFiles.....	120
第 8 课的练习.....	144
练习 8-1: 使用口令安全功能	145
练习 8-2: 使用加密的表空间	155
第 9 课的练习.....	163
练习 9-1: 使用 SQL 性能分析器.....	164
第 10 课的练习.....	190
练习 10-1: 使用 SQL 计划管理.....	191
第 11 课的练习.....	231
练习 11-1: 使用数据库重放	232
练习 11-2: 使用数据库回放 — 异步模式	243
第 12 课的练习.....	248
练习 12-1: 使用自动 SQL 优化.....	249
第 13 课的练习.....	262
练习 13-1: 创建基线	263
练习 13-2: 控制自动任务作业	266
练习 13-3: 校准 I/O 资源.....	269

第 14 课的练习.....	270
练习 14-1: 使用支持工作台分析损坏	271
练习 14-2: 使用 SQL 修复指导	283
第 15 课的练习.....	298
练习 15-1: 使用 Enterprise Manager 监视 SQL 命令	299
第 16 课的练习.....	304
练习 16-1: 使用自动内存管理	305
练习 16-2: 使用优化程序延迟统计信息	318
第 17 课的练习.....	340
练习 17-1: 使用结果高速缓存	341
第 18 课的练习.....	360
练习 18-1: 复制活动数据库	361
练习 18-2: 管理目录	368
第 19 课的练习.....	378
练习 19-1: 使用闪回数据归档	379
练习 19-2: 执行闪回事务处理回退	399
第 20 课的练习.....	409
练习 20-1: 使用 Enterprise Manager 修复数据库.....	410
练习 20-2: 使用 Enterprise Manager 修复块损坏.....	420
练习 20-3: 使用 RMAN 修复数据库	424

注：在本课程的所有练习中，使用了简单易记的口令，目的是为了将重点放在练习上。在真实的开发和生产环境中，请遵循《Oracle Database Security Guide》中的准则使用强口令。

第 1 课的练习

在本课的练习中，您将安装用于独立服务器的 Oracle Grid Infrastructure 并配置 ASM 集群文件系统。

练习 1-1: 安装用于独立服务器的 Oracle Grid Infrastructure

在本练习中，您将创建用于在计算机上安装和配置 Oracle Grid Infrastructure 的 ASM 磁盘。

- 1) 在终端窗口中，以 root 用户身份连接，将 oracleasm 组件配置为归 oracle 和组 dba 所有。此外，请确保 oracleasm 组件在系统启动时启动。

```
# lsmod | grep oracleasm
# oracleasm configure -i
Configuring the Oracle ASM library driver.

This will configure the on-boot properties of the Oracle ASM
library
driver. The following questions will determine whether the
driver is
loaded on boot and what permissions it will have. The current
values
will be shown in brackets ('[]'). Hitting <ENTER> without
typing an
answer will keep that current value. Ctrl-C will abort.

Default user to own the driver interface []: oracle
Default group to own the driver interface []: dba
Start Oracle ASM library driver on boot (y/n) [n]: y
Scan for Oracle ASM disks on boot (y/n) [y]: y
Writing Oracle ASM library driver configuration: done
#
```

- 2) 启动 oracleasm 组件确保其配置生效。

```
# oracleasm exit
# oracleasm init
Creating /dev/oracleasm mount point: /dev/oracleasm
Loading module "oracleasm": oracleasm
Mounting ASMLib driver filesystem: /dev/oracleasm
# oracleasm status
Checking if ASM is loaded: yes
Checking if /dev/oracleasm is mounted: yes
# lsmod | grep oracleasm
oracleasm                46356    1
#
```

练习 1-1：安装用于独立服务器的 Oracle Grid Infrastructure（续）

- 3) 使用 `oracleasm` 创建 ASM 磁盘。在您的计算机上有 13 个块设备可供使用：
`/dev/xvdb`、`/dev/xvdc` ... `/dev/xvdn`。请按照下列形式命名每个 ASM
磁盘：ASMDISK01、ASMDISK02 ... ASMDISK13。

```
# oracleasm createdisk ASMDISK01 /dev/xvdb
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK02 /dev/xvdc
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK03 /dev/xvdd
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK04 /dev/xvde
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK05 /dev/xvdf
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK06 /dev/xvdg
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK07 /dev/xvdh
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK08 /dev/xvdi
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK09 /dev/xvdj
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK10 /dev/xvdk
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK11 /dev/xvdl
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK12 /dev/xvdm
Writing disk header: done
Instantiating disk: done
# oracleasm createdisk ASMDISK13 /dev/xvdn
Writing disk header: done
Instantiating disk: done
#
```

练习 1-1：安装用于独立服务器的 Oracle Grid Infrastructure（续）

- 4) 使用 `listdisks` 命令，检查现在的确有 13 个 ASM 磁盘。

```
# oracleasm scandisks
Reloading disk partitions: done
Cleaning any stale ASM disks...
Scanning system for ASM disks...
# oracleasm listdisks
ASMDISK01
ASMDISK02
ASMDISK03
ASMDISK04
ASMDISK05
ASMDISK06
ASMDISK07
ASMDISK08
ASMDISK09
ASMDISK10
ASMDISK11
ASMDISK12
ASMDISK13
#
```

- 5) 在图形终端会话中以 `oracle` 用户身份在计算机的 `/u01/app/oracle/product/11.2.0/grid` 目录下安装 Oracle Grid Infrastructure。请确保在之前创建的 ASM 磁盘中，选用前四个 ASM 磁盘 (ASMDISK01 ... ASMDISK04) 创建常规冗余磁盘组 DATA。
- a) 设置环境变量，将其指向 Oracle Grid Infrastructure 安装的 Oracle 主目录：
`/u01/app/oracle/product/11.2.0/grid`。

```
$ . oraenv
ORACLE_SID = [db10g] ? +ASM
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/product/11.2.0/grid
$ echo $ORACLE_BASE
/u01/app/oracle
$
```

练习 1-1：安装用于独立服务器的 Oracle Grid Infrastructure（续）

- b) 从 Oracle Grid Infrastructure 软件的暂存目录执行 Oracle Universal Installer。

```
$ cd /stage/11*/g*
$ ./runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 9119
MB       Passed
Checking swap space: must be greater than 150 MB.   Actual
4000 MB       Passed
Checking monitor: must be configured to display at least 256
colors.      Actual 16777216       Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2009-09-28_10-00-29AM. Please wait ...
```

- c) 在 “Select Installation Option（选择安装选项）” 页上，选择 “Install and Configure Grid Infrastructure for a Standalone Server（安装并配置用于独立服务器的 Grid Infrastructure）”。单击 “Next（下一步）”。
- d) 在 “Select Product Languages（选择产品语言）” 页上，单击双箭头按钮选择所有语言。然后，单击 “Next（下一步）”。
- e) 在 “Create ASM DiskGroup（创建 ASM 磁盘组）” 页上，在 “Disk Group Name（磁盘组名）” 中输入 DATA，在 “Redundancy（冗余）” 字段中选择 “Normal（常规）”，然后在 “Candidate Disks（候选磁盘）” 中选择下列磁盘：ORCL:ASMDISK01、ORCL:ASMDISK02、ORCL:ASMDISK03、ORCL:ASMDISK04。然后，单击 “Next（下一步）”。
- f) 在 “Specify ASM Password（指定 ASM 口令）” 页上，选择 “Use same passwords for these accounts（对这些帐户使用相同口令）”，然后在 “Specify Password（指定口令）” 和 “Confirm Password（确认口令）” 两个字段中输入 oracle_4U。单击 “Next（下一步）”。
- g) 在 “Privileged Operating System Groups（已授权的操作系统组）” 页上，为每个组选择 dba。单击 “Next（下一步）”。
- h) 显示 “Warning（警告）” 窗口时，请单击 “Yes（是）” 继续安装。
- i) 在 “Installation Location（安装位置）” 页上，确保将 “Oracle Base（Oracle 基目录）” 设置为 /u01/app/oracle，并将 “Software Location（软件位置）” 设置为 /u01/app/oracle/product/11.2.0/grid。然后，单击 “Next（下一步）”。
- j) 此时将显示 “Perform Prerequisite Checks（执行先决条件检查）” 页。请耐心等待，直到检查完成。
- k) 在 “Summary（概要）” 页上，单击 “Finish（完成）”。
- l) 此时将显示 “Setup（设置）” 页。

练习 1-1：安装用于独立服务器的 Oracle Grid Infrastructure（续）

- m) 请耐心等待，直到要求您执行脚本。
- n) 显示“Execute Configuration Scripts（执行配置脚本）”窗口后，请以 root 用户身份从终端窗口进行连接，然后执行所需的脚本：

```
# /u01/app/oracle/product/11.2.0/grid/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/product/11.2.0/grid

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
    Copying dbhome to /usr/local/bin ...
    Copying oraenv to /usr/local/bin ...
    Copying coraenv to /usr/local/bin ...

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
2009-09-28 10:19:44: Checking for super user privileges
2009-09-28 10:19:44: User has super user privileges
2009-09-28 10:19:44: Parsing the host name
Using configuration parameter file:
/u01/app/oracle/product/11.2.0/grid/crs/install/
crsconfig_params
Creating trace directory
LOCAL ADD MODE
Creating OCR keys for user 'oracle', privgrp 'oinstall'..
Operation successful.
CRS-4664: Node edrsr22p1 successfully pinned.
ohasd is starting
Adding daemon to inittab
CRS-4123: Oracle High Availability Services has been started.
ohasd is starting
Successfully configured Oracle Grid Infrastructure for a
Standalone Server
Updating inventory properties for clusterware
Starting Oracle Universal Installer...
Checking swap space: must be greater than 500 MB. Actual
4000 MB Passed
The inventory pointer is located at /etc/OraInst.loc
The inventory is located at /u01/app/oracle/oraInventory
'UpdateNodeList' was successful.
#
```

练习 1-1：安装用于独立服务器的 Oracle Grid Infrastructure（续）

- o) 返回 “Execute Configuration Scripts（执行配置脚本）” 窗口，然后单击 “OK（确定）”。
- p) 请耐心等待，直到所有 Configuration Assistant 执行完毕。
- q) 在 “Finish（完成）” 页上，单击 “Close（关闭）”。
- 6) 请使用接下来的四个 ASM 磁盘 (ASMDISK05 ... ASMDISK08) 通过 ASMCA 创建 FRA 外部磁盘组。
 - a) 在以 oracle 用户身份连接的图形终端窗口中，确保已正确设置环境变量，然后调用 ASMCA。

```
$ . oraenv
ORACLE_SID = [+ASM] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ asmca
```

- b) 在 “Disk Groups（磁盘组）” 选项卡页上，单击 “Create（创建）”。
- c) 请在 “Create Disk Group（创建磁盘组）” 页的 “Disk Group Name（磁盘组名）” 字段中输入 FRA。在 “Redundancy（冗余）” 中，为该磁盘组选择 “External (None)（外部（无））”。请选择下列 ASM 磁盘：
ORCL:ASMDISK05、ORCL:ASMDISK06、ORCL:ASMDISK07 和
ORCL:ASMDISK08。
- d) 单击 “Show Advanced Options（显示高级选项）”，确保将 ASM 和
“Database Compatibility（数据库兼容性）” 字段设置为 11.2.0.0.0。
然后单击 “OK（确定）”。
- e) 此时将显示 “DiskGroup: Creation（磁盘组：创建）” 窗口。创建磁盘组后，
在 “DiskGroup: Creation（磁盘组：创建）” 窗口中单击 “OK（确认）”。
- f) 在 “Disk Group（磁盘组）” 选项卡页上，请确保可以看见 FRA 磁盘组已
装载。单击 “Exit（退出）”。
- g) 在 “ASM Configuration Assistant” 窗口中，单击 “Yes（是）”。

练习 1-2: 配置 ASM 集群文件系统

在本练习中，您将配置在系统启动时自动加载 ACFS 驱动程序。然后，新建一个 ASM 磁盘组，稍后将使用该磁盘组在 ASM 集群文件系统中存储 Oracle DB 二进制文件。

- 1) 以 root 用户身份从终端会话中进行连接，确定当前已加载到系统中的 ASM 驱动程序。

```
# lsmod | grep oracle
oracleacfs          787460  0
oracleadvn         177792  0
oracleoks          226656  2 oracleacfs,oracleadvn
oracleasm           46356  1
#
```

- 2) 重新启动系统。

```
# reboot

Broadcast message from root (pts/2) (Mon Sep 28 12:21:04
2009):

The system is going down for reboot NOW!
#
```

- 3) 重新启动后，以 root 用户身份重新连接至系统，并再次确定当前已加载的 ASM 驱动程序。您会得出什么结论？

- a) 很明显，此次未加载任何 ACFS 驱动程序。

```
# lsmod | grep oracle
oracleasm           46356  1
#
```

- 4) 手动重新启动 ACFS 驱动程序。

```
# /u01/app/oracle/product/11.2.0/grid/bin/acfsload start -s
# lsmod | grep oracle
oracleacfs          787460  0
oracleadvn         177792  0
oracleoks          226656  2 oracleacfs,oracleadvn
oracleasm           46356  1
#
```

练习 1-2: 配置 ASM 集群文件系统（续）

5) 如何确保系统启动时能自动加载这些驱动程序？

- a) 在 /etc/init.d 目录下创建一个脚本，用于启动 ACFS 驱动程序。该脚本需包含下列命令：

```
#!/bin/sh

# chkconfig: 2345 30 21

# description: Load Oracle ACFS drivers at system boot
/u01/app/oracle/product/11.2.0/grid/bin/acfsload start -s
```

```
# vi /etc/init.d/acfsload
```

将上述命令插入文件。

插入后，文件应类似下面的示例：

```
# cat /etc/init.d/acfsload
#!/bin/sh

# chkconfig: 2345 30 21
# description: Load Oracle ACFS drivers at system boot
/u01/app/oracle/product/11.2.0/grid/bin/acfsload start -s

#
```

- b) 使用 chkconfig 命令将该脚本添加为服务。

```
# chmod u+x /etc/init.d/acfsload
# chkconfig --add acfsload
# chkconfig --list acfsload
acfsload          0:off   1:off   2:on    3:on    4:on    5:on
6:off
#
```

6) 验证该解决方案，确保在系统启动时所有 ACFS 驱动程序都加载。

```
# reboot

Broadcast message from root (pts/2) (Mon Sep 28 12:55:25
2009):

The system is going down for reboot NOW!
# login as: root
root@... 's password:
Last login: Mon Sep 28 12:20:58 2009 from ...
# lsmod | grep oracle
oracleacfs          787460  0
oracleadvn          177792  0
oracleoks           226656  2 oracleacfs,oracleadvn
oracleasm           46356  1
#
```

练习 1-2: 配置 ASM 集群文件系统（续）

- 7) 使用 ASMCA 创建具有以下特征的 ACFS 磁盘组：使用外部冗余、使用 ASMDISK09 到 ASMDISK11 磁盘。此外，请确保该磁盘组可以托管 ASM 卷和 Oracle DB 主目录。

- a) 确保在上次重新启动系统后，系统已自动启动了 ASM 实例。

```
# ps -ef | grep lgwr
oracle      5799      1    0 19:08 ?          00:00:00 asm_lgwr_+ASM
root        5839    5600    0 19:12 pts/2      00:00:00 grep lgwr
#
```

- b) 在以 oracle 用户身份连接的图形终端窗口中，将环境变量设置为指向 ASM 实例，然后执行 asmca 命令。

```
$ . oraenv
ORACLE_SID = [db10g] ? +ASM
$ asmca
```

- c) 在 “Disks Groups (磁盘组)” 页上，单击 “Create (创建)”。
 - d) 在 “Create Disk Group (创建磁盘组)” 页的 “Disk Group Name (磁盘组名)” 字段中输入 ACFS。在 “Redundancy (冗余)” 中选择 “External (None) (外部 (无))”。然后选择下列 ASM 磁盘：ORCL:ASMDISK09、ORCL:ASMDISK10 和 ORCL:ASMDISK11。单击 “Show Advanced Options (显示高级选项)”，确保将全部三个磁盘的 “Diskgroup Compatibility (磁盘组兼容性)” 属性都设置为 11.2.0.0.0。单击 “OK (确定)”。
 - e) 创建该磁盘组将花费一些时间。
 - f) 该磁盘组创建后，将显示 “DiskGroup: Creation (磁盘组：创建)” 窗口。单击 “OK (确定)”。
 - g) 在 “Disk Group (磁盘组)” 选项卡页上，现在应看见 ACFS 磁盘组已装载。
- 8) 创建 ACFS 磁盘组后，请创建一个由该 ACFS 磁盘组托管的文件系统。创建后，请确保将该文件系统装载至 /u01/app/oracle/acfsmounts。
 - a) 右键单击 “ACFS Disk Group Name (ACFS 磁盘组名)”。
 - b) 在显示的菜单中选择 “Create ACFS for Database Home (为数据库主目录创建 ACFS)”。
 - c) 此时将显示 “Create ACFS Hosted Database Home (创建 ACFS 托管的数据库主目录)” 窗口。保留所有默认设置：acfs_db1、6、oracle、oinstall，然后在 “Database Home Mountpoint (数据库主目录装载点)” 字段输入 /u01/app/oracle/acfsmounts/acfs_db1。单击 “OK (确定)”。

练习 1-2: 配置 ASM 集群文件系统（续）

- d) 此时将自动创建新的 ASM 卷和文件系统。
- e) 完成后，将显示 “Database Home: Run ACFS script（数据库主目录：运行 ACFS 脚本）” 窗口。
- f) 以 root 用户身份打开一个终端窗口并执行
/u01/app/oracle/cfgtoollogs/asmca/scripts/acfs_script.sh
脚本。

```
# /u01/app/oracle/cfgtoollogs/asmca/scripts acfs_script.sh
#
```

- g) 然后在 “Database Home: Run ACFS script（数据库主目录：运行 ACFS 脚本）” 窗口中单击 “Close（关闭）”。
 - h) 请注意，在 “Disk Group（磁盘组）” 选项卡页上，目前 ACFS 磁盘组已接近填满。
 - i) 单击 “Volumes（卷）” 选项卡，查看自动创建的 ASM 卷。
 - j) 单击 “ASM Cluster File Systems（ASM 集群文件系统）” 选项卡，查看自动创建的文件系统。
 - k) 单击 “Exit（退出）” 从 ASMCA 中退出。
 - l) 在显示的 “ASM Configuration Assistant” 中，单击 “Yes（是）”。
- 9) 在终端会话窗口中，以 root 用户身份连接，浏览新的文件系统。

```
# ls -l /u01/app/oracle
total 36
drwxr-x--- 3 oracle oinstall 4096 Sep 28 13:06 acfsmounts
drwxr-x--- 4 oracle oinstall 4096 Sep 28 10:21 admin
drwxr-xr-x 4 oracle oinstall 4096 Sep 28 10:21 cfgtoollogs
drwxr-xr-x 2 oracle oinstall 4096 Sep 28 10:22 checkpoints
drwxrwxr-x 4 oracle oinstall 4096 Sep 28 10:21 diag
drwxr-x--- 3 oracle oinstall 4096 May  8 18:01
flash_recovery_area
drwxr-x--- 3 oracle oinstall 4096 May  8 18:00 oradata
drwxrwx--- 7 oracle oinstall 4096 Sep 28 10:20 oraInventory
drwxrwx--- 4 oracle oinstall 4096 Sep 28 10:12 product
# ls -l /u01/app/oracle/acfsmounts
total 4
drwxrwxr-x 4 oracle oinstall 4096 Sep 28 13:08 acfs_db1
# ls -la /u01/app/oracle/acfsmounts/*
total 76
drwxrwxr-x 4 oracle oinstall  4096 Sep 28 13:08 .
drwxr-x--- 3 oracle oinstall  4096 Sep 28 13:06 ..
drwxr-xr-x 5 root    root      4096 Sep 28 13:08 .ACFS
drwx----- 2 root    root      65536 Sep 28 13:08 lost+found
[root@edrsr22p1 ~]#
# ls -l /dev/asm
```

练习 1-2: 配置 ASM 集群文件系统（续）

```
total 0
brwxrwx--- 1 root dba 252, 246273 Sep 28 13:06 acfs_db1-481
# mount
/dev/sda5 on / type ext3 (rw)
proc on /proc type proc (rw)
sysfs on /sys type sysfs (rw)
devpts on /dev/pts type devpts (rw,gid=5,mode=620)
/dev/sda2 on /boot type ext3 (rw)
tmpfs on /dev/shm type tmpfs (rw,size=1152M)
none on /proc/sys/fs/binfmt_misc type binfmt_misc (rw)
sunrpc on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw)
oracleasmfs on /dev/oracleasm type oracleasmfs (rw)
/dev/sda1 on /mnt/cdrive type vfat (rw)
/dev/asm/acfs_db1-481 on /u01/app/oracle/acfsmounts/acfs_db1
type acfs (rw)
# acfsutil info fs
/u01/app/oracle/acfsmounts/acfs_db1
  ACFS Version: 11.2.0.1.0.0
  flags:          MountPoint,Available
  mount time:     Mon Sep 28 13:08:48 2009
  volumes:        1
  total size:     6442450944
  total free:     6391668736
  primary volume: /dev/asm/acfs_db1-481
    label:
    flags:          Primary,Available,ADVM
    on-disk version: 39.0
    allocation unit: 4096
    major, minor:   252, 246273
    size:           6442450944
    free:           6391668736
    ADVM diskgroup   ACFS
    ADVM resize increment: 268435456
    ADVM redundancy: unprotected
    ADVM stripe columns: 4
    ADVM stripe width: 131072
  number of snapshots: 0
  snapshot space usage: 0
#
```

第 2 课的练习

在本课的练习中，您将安装 Oracle Database 11g 软件、创建数据库、使用 ASM 集群文件系统快照并通过联机打补丁功能为数据库打补丁。

练习 2-1：安装 Oracle Database 11g 软件

在本练习中，您将在装载到 /u01/app/oracle/acfsmounts/acfs_db1 上的 ASM 集群文件系统中安装 Oracle DB 软件。

- 1) 在图形终端会话中，请确保将环境变量设置为指向 ORACLE_HOME 的 ASFC 装载点，并确保将 ORACLE_BASE 设置为 /u01/app/oracle。

```
$ . oraenv
ORACLE_SID = [+ASM] ? orcl
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/acfsmounts/acfs_db1
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
```

- 2) 将当前目录更改为软件暂存目录，并调用 Oracle Universal Installer。

```
$ cd /stage/11*/d*
$ ./runInstaller
Starting Oracle Universal Installer...

Checking Temp space: must be greater than 80 MB.   Actual 6589
MB           Passed
Checking swap space: must be greater than 150 MB.   Actual
3868 MB      Passed
Checking monitor: must be configured to display at least 256
colors.      Actual 16777216   Passed
Preparing to launch Oracle Universal Installer from
/tmp/OraInstall2009-05-25_02
Please wait ...
```

- 3) 在“Configure Security Updates（配置安全更新）”页上，确保未选中任何选项，然后单击“Next（下一步）”。
- 4) 在“Email Address Not Specified（未指定电子邮件地址）”预警窗口中，单击“Yes（是）”。
- 5) 在“Installation Option（安装选项）”页上，请选择“Install Database Software Only（仅安装数据库软件）”，然后单击“Next（下一步）”。
- 6) 在“Grid Options（网格选项）”页上，选择“Single Instance Database Installation（单实例数据库安装）”选项，然后单击“Next（下一步）”。
- 7) 在“Product Languages（产品语言）”页上，使用双箭头按钮选择所有语言。然后，单击“Next（下一步）”。
- 8) 在“Database Edition（数据库版本）”页上，选择“Enterprise Edition（企业版）”并单击“Next（下一步）”。

练习 2-1: 安装 Oracle Database 11g 软件（续）

- 9) 在 “Installation Location（安装位置）” 页上，确保将 “Oracle Base（Oracle 基目录）” 字段设置为 /u01/app/oracle，并将 “Software Location（软件位置）” 设置为 /u01/app/oracle/acfsmounts/acfs_db1。然后，单击 “Next（下一步）”。
- 10) 如果显示一个警告窗口指出选定的 Oracle 主目录不为空，请单击 “Yes（是）”。
- 11) 在 “Operating System Groups（操作系统组）” 页中，请确保已分别为 OSDBA 和 OSOPER 选定了 dba 和 dba。然后，单击 “Next（下一步）”。
- 12) 完成先决条件检查后，将显示 “Summary（概要）” 页。单击 “Finish（完成）”。
- 13) 安装流程随即启动。
- 14) 请耐心等待，直到要求您以 root 用户身份运行脚本。
- 15) 要求您执行 /u01/app/oracle/acfsmounts/acfs_db1/root.sh 脚本时，请以 root 用户身份打开一个终端窗口，然后执行该脚本：

```
$ su -
Password:
# /u01/app/oracle/acfsmounts/acfs_db1/root.sh
Running Oracle 11g root.sh script...

The following environment variables are set as:
    ORACLE_OWNER= oracle
    ORACLE_HOME=  /u01/app/oracle/acfsmounts/acfs_db1

Enter the full pathname of the local bin directory:
[/usr/local/bin]:
The file "dbhome" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "oraenv" already exists in /usr/local/bin.  Overwrite
it? (y/n)
[n]:
The file "coraenv" already exists in /usr/local/bin.
Overwrite it? (y/n)
[n]:

Entries will be added to the /etc/oratab file as needed by
Database Configuration Assistant when a database is created
Finished running generic part of root.sh script.
Now product-specific root actions will be performed.
Finished product-specific root actions.
#
```

练习 2-1：安装 Oracle Database 11g 软件（续）

- 16) 完成后，请返回 OUI 窗口，然后在“Execute Configuration Scripts（执行配置脚本）”页上单击“OK（确定）”。
- 17) 在“Finish（完成）”页上，单击“Close（关闭）”。

练习 2-2: 新建数据库

在本练习中，您将在 ASM 磁盘组中新建一个数据库。

1) 使用 DBCA 新建一个数据库。

- a) 以 oracle 用户身份在图形终端会话中进行连接，确保为 Oracle Database 11g 正确设置了环境。

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
ORACLE_HOME = [/home/oracle] ?
/u01/app/oracle/acfsmounts/acfs_db1
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
```

- b) 从同一终端窗口中执行 dbca 命令：

```
$ dbca
```

- c) 在 “Welcome (欢迎使用)” 页上，单击 “Next (下一步)”。
- d) 在 “Operations (操作)” 页上，选择 “Create a Database (创建数据库)”，然后单击 “Next (下一步)”。
- e) 在 “Database Templates (数据库模板)” 页上，选择 “General Purpose (一般用途)” 模板或 “Transaction Processing (事务处理)” 模板，然后单击 “Next (下一步)”。
- f) 在 “Database Identification (数据库标识)” 页的 “Global Database Name (全局数据库名)” 字段中输入 orcl (不使用域)，然后单击 “Next (下一步)”。
- g) 在 “Management Options (管理选项)” 页上，确认以下选择，然后单击 “Next (下一步)”：
- 配置 Enterprise Manager
 - Configure Database Control for local management (配置 Database Control 以进行本地管理)
- h) 在 “Database Credentials (数据库身份证明)” 页上，选择 “Use the Same Password for All Accounts (所有帐户使用同一口令)”，然后在 “Password (口令)” 和 “Confirm Password (确认口令)” 两个字段中输入 oracle_4U，然后单击 “Next (下一步)”。

练习 2-2: 新建数据库（续）

- i) 在“Database File Locations（数据库文件位置）”页上，从“Storage Type（存储类型）”下拉列表中选择“Automatic Storage Management (ASM)（自动存储管理 (ASM)）”，并确保选定“Use Oracle-Managed Files（使用 Oracle 管理的文件）”。
- j) 在同一页中，单击“Database Area（数据库区）”字段旁的“Browse（浏览）”按钮。
- k) 在“Select Disk Group（选择磁盘组）”页上，选择 DATA，然后单击“OK（确定）”。
- l) 单击“Next（下一步）”。
- m) 在“ASM Credentials（ASM 身份证明）”页上，输入 oracle_4U 作为 ASMSNMP 用户的口令。然后单击“OK（确定）”。
- n) 在“Recovery Configuration（恢复配置）”页上，选择“Specify Flash Recovery Area（指定快速恢复区）”。
- o) 单击“Flash Recovery Area（快速恢复区）”字段旁的“Browse（浏览）”。
- p) 在“Select Disk Group（选择磁盘组）”页上，选择 FRA，然后单击“OK（确定）”。
- q) 单击“Next（下一步）”。
- r) 在“Database Content（数据库内容）”页上，选择“Sample Schemas（示例方案）”，然后单击“Next（下一步）”。
- s) 在“Initialization Parameters: Memory（初始化参数：内存）”页上，选择“Custom（定制）”，然后将“SGA Size（SGA 大小）”设置为 404 M Bytes，将“PGA Size（PGA 大小）”设置为 150 M Bytes。然后单击“Character Sets（字符集）”选项卡。
- t) 在“Character Sets（字符集）”页上，选择“Use Unicode (AL32UTF8)（使用 Unicode (AL32UTF8)）”。单击“Next（下一步）”。
- u) 在“Database Storage（数据库存储）”页上，单击“Next（下一步）”。
- v) 在“Creation Options（创建选项）”页上，选择“Create Database（创建数据库）”，然后单击“Finish（完成）”。
- w) 在“Confirmation（确认）”页上，单击“OK（确定）”。
- x) 等待直到完成数据库创建。
- y) DBCA 完成操作后，请记下下列信息以供将来参考：安装日志文件的位置、全局数据库名、系统标识符 (SID)、服务器参数文件名和位置、Enterprise Manager Database Control URL。请注意，Enterprise Manager Database Control 默认启用 SSL。在“Database Configuration Assistant”对话框中，单击“Exit（退出）”。

练习 2-3: 使用 ASM 集群文件系统快照

在本练习中，您将使用 ACFS 快照从文件系统中恢复文件。

- 1) 在终端窗口中，确定 /u01/app/oracle/acfsmounts/acfs_db1 文件系统的目录结构。可以看到什么？（注：请勿在完成步骤 1 后退出该终端窗口，因为在步骤 3 中还将会用到。）
 - a) 此时，您应看到一个名为 .ACFS 的目录，其中包含 ACFS 快照。

```
$ cd /u01/app/oracle/acfsmounts/acfs_db1
$
$ ls -la
total 520
drwxrwxr-x 76 oracle oinstall 8192 Sep 28 14:37 .
drwxr-x--- 3 oracle oinstall 4096 Sep 28 13:06 ..
drwxr-xr-x 5 root root 4096 Sep 28 13:08 .ACFS
drwxr-xr-x 8 oracle oinstall 8192 Sep 28 13:51 apex
drwxr-xr-x 8 oracle oinstall 4096 Sep 28 13:48 assistants
drwxr-xr-x 2 oracle oinstall 20480 Sep 28 14:34 bin
...
drwxr-xr-x 2 oracle oinstall 4096 Sep 28 13:49 utl
drwxr-xr-x 3 oracle oinstall 4096 Sep 28 13:48 wwg
drwxr-xr-x 7 oracle oinstall 4096 Sep 28 13:49 xdk
$

$ ls -la .ACFS
total 32
drwxr-xr-x 5 root root 4096 Sep 28 13:08 .
drwxrwxr-x 76 oracle oinstall 8192 Sep 28 14:37 ..
drwx----- 2 root root 4096 Sep 28 13:08 .fileid
drwx----- 6 root root 4096 Sep 28 13:08 repl
drwxr-xr-x 2 root root 4096 Sep 28 13:08 snaps
$

$ ls -la .ACFS/snaps
total 8
drwxr-xr-x 2 root root 4096 Sep 28 13:08 .
drwxr-xr-x 5 root root 4096 Sep 28 13:08 ..
$
```

练习 2-3: 使用 ASM 集群文件系统快照（续）

- 2) 使用 Enterprise Manager Database Control 为您的 ASM 集群文件系统新建一个快照。
 - a) 在浏览器中输入下列 URL，调用 Enterprise Manager Database Control:
https://<hostname>:5500/em。以 SYS 用户身份使用 `oracle_4U` 口令进行连接。从“Connect As（连接身份）”菜单中选择 **SYSDBA**。（在 Oracle University 课堂中，您可能会接收到一条关于安全证书的预警消息。如果是这样，请单击“OK（确定）”关闭“Alert（预警）”窗口。依次单击“Or you can add an exception link（或者添加一条异常错误链接）”、“Add Exception（添加异常错误）”、“Get Certificate（获取证书）”和“Confirm Security Exception（确认安全性异常错误）”，然后继续按前面所述登录 Enterprise Manager。）
 - b) 在数据库主页上，单击“General（一般信息）”部分中的“+ASM”链接。
 - c) 在“Automatic Storage Management: +ASM（自动存储管理：+ASM）”页上，单击“ASM Cluster File System（ASM 集群文件系统）”选项卡。
 - d) 在“Automatic Storage Management Login（自动存储管理登录）”页上，输入 SYS 用户身份证明（`sys/oracle_4U`），选择“Save as Preferred Credential（另存为首选身份证明）”，然后单击“Login（登录）”。
 - e) 在“Automatic Storage Management: +ASM（自动存储管理：+ASM）”页的“ASM Cluster File System（ASM 集群文件系统）”选项卡中，选择文件系统 `/u01/app/oracle/acfsmounts/acfs_db1` 装载点，然后单击“Create Snapshot（创建快照）”。
 - f) 在“Create Snapshot on ASM Cluster File System:
`/u01/app/oracle/acfsmounts/acfs_db1`（在 ASM 集群文件系统上创建快照：
`/u01/app/oracle/acfsmounts/acfs_db1`）”页中，输入 `snapshot_1` 作为快照名。
 - g) 单击“Show Command（显示命令）”查看用于创建该快照的 `acfsutil` 命令。此命令应为：`/sbin/acfsutil snap create "snapshot_1" /u01/app/oracle/acfsmounts/acfs_db1`
 - h) 在“Show Command（显示命令）”页上，单击“Return（返回）”。
 - i) 返回“Create Snapshot on ASM Cluster File System:
`/u01/app/oracle/acfsmounts/acfs_db1`（在 ASM 集群文件系统上创建快照：
`/u01/app/oracle/acfsmounts/acfs_db1`）”页，然后单击“OK（确定）”。
 - j) 在“ASM Cluster File System Host Credentials（ASM 集群文件系统主机身份证明）”页上，输入 `oracle/oracle` 作为您的主机身份证明，选择“Save as Preferred Credential（另存为首选身份证明）”并单击“Continue（继续）”。

练习 2-3: 使用 ASM 集群文件系统快照（续）

- k) 返回 “Automatic Storage Management: +ASM（自动存储管理: +ASM）” 页，此时会出现一条确认状态消息，表明已成功创建快照 “snapshot_1”。
- l) 在 “ASM Cluster File System（ASM 集群文件系统）” 选项卡上，应看到该文件系统当前的快照数量为 1。
- 3) 在终端窗口中（仍在 /u01/app/oracle/acfsmounts/acfs_db1 目录下），验证是否已创建快照。
 - a) 尽管快照中包含所有的文件，但快照的创建速度非常快！

```
$ ls -la .ACFS/snaps
total 24
drwxr-xr-x  3 root    root      4096 Sep 29 11:24 .
drwxr-xr-x  5 root    root      4096 Sep 28 13:08 ..
drwxrwxr-x 76 oracle oinstall 8192 Sep 29 11:24 snapshot_1
$
$ ls -la .ACFS/snaps/*
total 456
drwxrwxr-x 76 oracle oinstall 8192 Sep 29 11:24 .
drwxr-xr-x  3 root    root      4096 Sep 29 11:24 ..
drwxr-xr-x  5 root    root      4096 Sep 28 13:08 .ACFS
drwxr-xr-x  8 oracle oinstall 8192 Sep 28 13:51 apex
drwxr-xr-x  8 oracle oinstall 4096 Sep 28 13:48 assistants
drwxr-xr-x  2 oracle oinstall 20480 Sep 28 14:34 bin
...
drwxr-xr-x  2 oracle oinstall 4096 Sep 28 13:49 utl
drwxr-xr-x  3 oracle oinstall 4096 Sep 28 13:48 wwg
drwxr-xr-x  7 oracle oinstall 4096 Sep 28 13:49 xdk
$
```

- 4) 使用 Enterprise Manager Database Control 验证已创建快照且快照包含必要的文件。
 - a) 返回到 “ASM Cluster File System（ASM 集群文件系统）” 选项卡页，在您的文件系统对应的 “Snapshots（快照）” 列中单击 **1**。
 - b) 在 “ASM Cluster File System: /u01/app/oracle/acfsmounts/acfs_db1（ASM 集群文件系统: /u01/app/oracle/acfsmounts/acfs_db1）” 页的 “Snapshots（快照）” 选项卡中，单击 “Snapshots（快照）” 表的 “Name（名称）” 列中的 **snapshot_1** 链接。
 - c) 如果需要，请输入 oracle/oracle 作为您的主机身份证明，然后单击 “Continue（继续）”。您可以选择 “Save as Preferred Credential（另存为首选身份证明）”。
 - d) 此时将转到 “Search and List: Directory（搜索并列出: 目录）” 页，您可以从该页搜索快照中的文件。

练习 2-3: 使用 ASM 集群文件系统快照（续）

- 5) 在终端窗口中，从 Oracle DB 主目录中删除 demo 目录。这样做是为了模拟一种人为错误。

```
$ rm -r /u01/app/oracle/acfsmounts/acfs_db1/demo
$ ls
apex          emcli          md             precomp
assistants    EMStagePatches_orcl  mesg           racg
bin           has            mgw            rdbms
ccr           hs             network        relnotes
cdata         ide            nls            root.sh
cfgtoollogs   install        oc4j           scheduler
clone         install.platform  odbc           slax
config        instantclient  olap           sqldeveloper
crs           inventory      ons           sqlj
csmig         j2ee           OPatch         sqlplus
css           javavm         opmn           srvn
ctx           jdbc           oracore        sysman
cv            jdev           oraInst.loc    timingframework
dbs           jdk            ord            ucp
dc_ocm        jlib           oui            uix
deinstall     ldap           owb            utl
diagnostics   lib            owm            wwg
dv            log            perl           xdk
edrsr         lost+found     plssql
$
```

- 6) 使用 Enterprise Manager Database Control，确保您能恢复该丢失的目录。
- 返回“Search and List: Directory（搜索并列出：目录）”页，单击“Search（搜索）”表中的 **demo** 链接。
 - 可以看到其中包含 schema 目录。
 - 单击“Close（关闭）”按钮。
- 7) 使用操作系统命令还原 demo 目录。

```
$ cp -r
/u01/app/oracle/acfsmounts/acfs_db1/.ACFS/snaps/snapshot_1/
demo /u01/app/oracle/acfsmounts/acfs_db1
$
$ ls -ls /u01/app/oracle/acfsmounts/acfs_db1
total 496
16 drwxr-xr-x  8 oracle oinstall  8192 Sep 28 13:51 apex
 4 drwxr-xr-x  8 oracle oinstall  4096 Sep 28 13:48 assistants
52 drwxr-xr-x  2 oracle oinstall 20480 Sep 28 14:34 bin
 4 drwxr-xr-x  7 oracle oinstall  4096 Sep 28 13:52 ccr
 4 drwxr-xr-x  3 oracle oinstall  4096 Sep 28 13:53 cdata
 4 drwxr-xr-x  4 oracle oinstall  4096 Sep 28 14:09
cfgtoollogs
 4 drwxr-xr-x  4 oracle oinstall  4096 Sep 28 13:50 clone
 4 drwxr-xr-x  2 oracle oinstall  4096 Sep 28 13:53 config
```

练习 2-3: 使用 ASM 集群文件系统快照 (续)

```

4 drwxr-xr-x 6 oracle oinstall 4096 Sep 28 13:48 crs
4 drwxr-xr-x 3 oracle oinstall 4096 Sep 28 13:48 csmig
4 drwxr-xr-x 6 oracle oinstall 4096 Sep 28 13:53 css
4 drwxr-xr-x 10 oracle oinstall 4096 Sep 28 13:54 ctx
4 drwxr-xr-x 6 oracle oinstall 4096 Sep 28 13:48 cv
4 drwxr-xr-x 4 oracle oinstall 4096 Sep 28 14:34 dbs
4 drwxr-xr-x 2 oracle oinstall 4096 Sep 28 13:51 dc_ocm
4 drwxr-xr-x 4 oracle oinstall 4096 Sep 28 13:56 deinstall
4 drwxr-xr-x 3 oracle oinstall 4096 Sep 29 11:41 demo
4 drwxr-xr-x 3 oracle oinstall 4096 Sep 28 13:48
diagnostics
...
4 drwxr-xr-x 2 oracle oinstall 4096 Sep 28 13:49 utl
4 drwxr-xr-x 3 oracle oinstall 4096 Sep 28 13:48 wwg
4 drwxr-xr-x 7 oracle oinstall 4096 Sep 28 13:49 xdk
$

```

- 8) 使用终端窗口以 root 用户身份连接，检索有关 ASM 集群文件系统的信息数据。

```

# acfsutil info fs /u01/app/oracle/acfsmounts/acfs_db1
/u01/app/oracle/acfsmounts/acfs_db1
ACFS Version: 11.2.0.1.0.0
flags:          MountPoint,Available
mount time:     Mon Sep 28 13:08:48 2009
volumes:        1
total size:     6442450944
total free:     1360973824
primary volume: /dev/asm/acfs_db1-481
  label:
  flags:          Primary,Available,ADVM
  on-disk version: 39.0
  allocation unit: 4096
  major, minor:   252, 246273
  size:           6442450944
  free:           1360973824
  ADVM diskgroup  ACFS
  ADVM resize increment: 268435456
  ADVM redundancy: unprotected
  ADVM stripe columns: 4
  ADVM stripe width: 131072
  number of snapshots: 1
  snapshot space usage: 23502848
You have new mail in /var/spool/mail/root
#

```

练习 2-3: 使用 ASM 集群文件系统快照（续）

- 9) 使用 Enterprise Manager Database Control 删除您的 ACFS 快照。
 - a) 返回 “ASM Cluster File System: /u01/app/oracle/acfsmounts/acfs_db1（ASM 集群文件系统: /u01/app/oracle/acfsmounts/acfs_db1）” 页的 “Snapshots（快照）” 选项卡。
 - b) 在 “Snapshots（快照）” 选项卡页上，选择 **snapshot_1**，然后单击 “Delete（删除）”。
 - c) 在 “Confirmation（确认）” 页上，展开 “Show Command（显示命令）” 链接。该命令应为：`/sbin/acfsutil snap delete "snapshot_1" /u01/app/oracle/acfsmounts/acfs_db1`
 - d) 在 “Confirmation（确认）” 页上，单击 “Yes（是）”。
 - e) 输入主机身份证明 `oracle/oracle`，然后选择 “Save as Preferred Credential（另存为首选身份证明）”。单击 “Continue（继续）”。
 - f) 返回 “ASM Cluster File System: /u01/app/oracle/acfsmounts/acfs_db1（ASM 集群文件系统: /u01/app/oracle/acfsmounts/acfs_db1）” 的页 “Snapshots（快照）” 选项卡，此时会出现一条确认消息，表明已成功删除选中的快照。

练习 2-4：使用联机打补丁

在本练习中，将对实时运行的数据库实例应用联机补丁程序，而不先将其关闭。

- 1) 在以 oracle 用户身份连接的终端窗口中，执行 OP_setup.sh 脚本。此脚本在 Oracle DB 软件中会生成 bug。

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ cd /home/oracle/solutions/OP
$ ./OP_setup.sh
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.
SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29
13:08:29 2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
chmod 755 /u01/app/oracle/acfsmounts/acfs_db1/bin - Linking
Oracle
rm -f /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
gcc -o /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
-m32 -L/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/
-L/u01/app/oracle/acfsmounts/acfs_db1/lib/
-L/u01/app/oracle/acfsmounts/acfs_db1/lib/stubs/
-L/u01/app/oracle/acfsmounts/acfs_db1/lib/ -lirc -lipgo
-Wl,-E /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/opimai.o
/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/ssoraed.o
/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/ttcsoi.o
-Wl,--whole-archive -lperfsvl1 -Wl,--no-whole-archive
/u01/app/oracle/acfsmounts/acfs_db1/lib/nautab.o
/u01/app/oracle/acfsmounts/acfs_db1/lib/naeet.o
/u01/app/oracle/acfsmounts/acfs_db1/lib/naect.o
```

练习 2-4：使用联机打补丁（续）

```

/u01/app/oracle/acfsmounts/acfs_db1/lib/naedhs.o
/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/config.o
-lserver11 -lodm11 -lcell111 -lnnet11 -lskgxp11 -lsnls11
-lnls11 -lcore11 -lsnls11 -lnls11 -lcore11 -lsnls11 -lnls11
-lxml11 -lcore11 -lunls11 -lsnls11 -lnls11 -lcore11 -lnls11
-lclient11 -lvsn11 -lcommon11 -lgeneric11 -lknlopt `if
/usr/bin/ar tv
/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/libknlopt.a |
grep xsyeolap.o > /dev/null 2>&1 ; then echo "-loraolap11" ;
fi` -ls1ax11 -lp1s11 -lrt -lp1p11 -lserver11 -lclient11
-lvs11 -lcommon11 -lgeneric11 `if [ -f
/u01/app/oracle/acfsmounts/acfs_db1/lib/libavserver11.a ] ;
then echo "-lavserver11" ; else echo "-lavstub11"; fi` `if
[ -f /u01/app/oracle/acfsmounts/acfs_db1/lib/libavclient11.a ] ;
then echo "-lavclient11" ; fi` -lknlopt -ls1ax11 -lp1s11 -lrt
-lp1p11 -ljavavm11 -lserver11 -lwwg `cat
/u01/app/oracle/acfsmounts/acfs_db1/lib/ldflags` -lncrypt11
-lnsgr11 -lnzjs11 -ln11 -ln111 -lnrol1 `cat
/u01/app/oracle/acfsmounts/acfs_db1/lib/ldflags` -lncrypt11
-lnsgr11 -lnzjs11 -ln11 -ln111 -lnnz11 -lzt11 -lmm -lsnls11
-lnls11 -lcore11 -lsnls11 -lnls11 -lcore11 -lsnls11 -lnls11
-lxml11 -lcore11 -lunls11 -lsnls11 -lnls11 -lcore11 -lnls11
-lztkg11 `cat /u01/app/oracle/acfsmounts/acfs_db1/lib/ldflags`
-lncrypt11 -lnsgr11 -lnzjs11 -ln11 -ln111 -lnrol1 `cat
/u01/app/oracle/acfsmounts/acfs_db1/lib/ldflags` -lncrypt11
-lnsgr11 -lnzjs11 -ln11 -ln111 -lnnz11 -lzt11 -lsnls11
-lnls11 -lcore11 -lsnls11 -lnls11 -lcore11 -lsnls11 -lnls11
-lxml11 -lcore11 -lunls11 -lsnls11 -lnls11 -lcore11 -lnls11
`if /usr/bin/ar tv
/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/libknlopt.a |
grep "kxmnsd.o" > /dev/null 2>&1 ; then echo " " ; else echo
"-lordsd011"; fi` -
L/u01/app/oracle/acfsmounts/acfs_db1/ctx/lib/ -lctxc11 -lctx11
-lzx11 -lgx11 -lctx11 -lzx11 -lgx11 -lordimt11 -lclsrall
-ldbcfg11 -lhasgen11 -lskgxn2 -lnnz11 -lzt11 -lxml11 -locr11
-locrb11 -locrut11 -lhasgen11 -lskgxn2 -lnnz11 -lzt11 -lxml11
-lasmclnt11 -lcommon11 -lcell111 -lskgxp11 -lgeneric11
-lcommon11 -lgeneric11 -loraz -llzopro -lorabz2 -lsnls11
-lnls11 -lcore11 -lsnls11 -lnls11 -lcore11 -lsnls11 -lnls11
-lxml11 -lcore11 -lunls11 -lsnls11 -lnls11 -lcore11 -lnls11
-lsnls11 -lunls11 -lsnls11 -lnls11 -lcore11 -lsnls11 -lnls11
-lcore11 -lsnls11 -lnls11 -lxml11 -lcore11 -lunls11 -lsnls11
-lnls11 -lcore11 -lnls11 -lasmlnt11 -laio `cat
/u01/app/oracle/acfsmounts/acfs_db1/lib/sysliblist` -Wl,
-rpath,/u01/app/oracle/acfsmounts/acfs_db1/lib -lm `cat
/u01/app/oracle/acfsmounts/acfs_db1/lib/sysliblist` -ldl -lm
-L/u01/app/oracle/acfsmounts/acfs_db1/lib
test !-f /u01/app/oracle/acfsmounts/acfs_db1/bin/oracle ||\
mv -f

```

练习 2-4：使用联机打补丁（续）

```

/u01/app/oracle/acfsmounts/acfs_db1/bin/oracle
/u01/app/oracle/acfsmounts/acfs_db1/bin/oracleO
mv /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
/u01/app/oracle/acfsmounts/acfs_db1/bin/oracle
chmod 6751 /u01/app/oracle/acfsmounts/acfs_db1/bin/oracle

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 13:09:30
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to an idle instance.
SQL> SQL> ORACLE instance started.
Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 180357504 bytes
Database Buffers 234881024 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 2) 调用 SQL*Plus 并以 SYS 用户身份进行连接。执行以下命令：

```
select * from mysql_is_great;
```

会看到什么结果？

- a) 会出现 ORA-600 错误。该错误是由在本练习中专门引入的 bug 导致的。
退出 SQL*Plus。

```

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 13:14:27
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
SQL> select * from mysql_is_great;
select * from mysql_is_great
*
ERROR at line 1:
ORA-00600: internal error code, arguments: [HAH!], [ORACLE],
[IS], [BETTER!], [], [], [], [], [], [], []

```

练习 2-4：使用联机打补丁（续）

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
$
```

- 3) 在 \$HOME/solutions/OP 中解压缩 hotpatch.zip 文件。这就是修复该 bug 需要应用的补丁程序。

```
$ rm -r 5943776
$ unzip 5943776.zip
Archive: 5943776.zip
  creating: 5943776/
  creating: 5943776/files/
  creating: 5943776/files/hpatch/
  inflating: 5943776/files/hpatch/bug5943776.pch
  creating: 5943776/etc/
  creating: 5943776/etc/config/
  inflating: 5943776/etc/config/inventory.xml
  inflating: 5943776/etc/config/actions.xml
  creating: 5943776/etc/xml/
  inflating: 5943776/etc/xml/GenericActions.xml
  inflating: 5943776/etc/xml/ShiphomeDirectoryStructure.xml
  extracting: 5943776/README.txt
$
```

- 4) 验证提供的补丁程序是否处于联机状态，然后在运行数据库时应用该补丁程序。安装只需要花费几秒钟的时间。**注：**请确保使用 `，而非直引号！

```
$ cd /home/oracle/solutions/OP/5943776
$ $ORACLE_HOME/OPatch/patch query -is_online_patch `pwd`
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.
Oracle Home      : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory : /u01/app/oracle/oraInventory
   from           : /etc/oraInst.loc
OPatch version   : 11.1.0.6.6
OUI version      : 11.2.0.1.0
OUI location     : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/patch/patch2
009-09-29_13-18-23PM.log
```

练习 2-4：使用联机打补丁（续）

```

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt
-----

Patch is an online patch: true
OPatch succeeded.

$ $ORACLE_HOME/OPatch/opatch lsinventory
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.

Oracle Home           : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory     : /u01/app/oracle/oraInventory
   from               : /etc/oraInst.loc
OPatch version        : 11.1.0.6.6
OUI version           : 11.2.0.1.0
OUI location          : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location     :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch2
009-09-29_13-19-27PM. log

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt

Lsinventory Output file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/lsinv/
lsinventory2009-09-29_13-19-27PM.txt
-----

Installed Top-level Products (1):

Oracle Database 11g                               11.2.0.1.0
There are 1 products installed in this Oracle Home.
There are no Interim patches installed in this Oracle Home.
-----

OPatch succeeded.

$ $ORACLE_HOME/OPatch/opatch apply
/home/oracle/solutions/OP/5943776 -connectString orcl:::
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.
Oracle Home           : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory     : /u01/app/oracle/oraInventory

```


练习 2-4：使用联机打补丁（续）

```
from           : /etc/oraInst.loc
OPatch version  : 11.1.0.6.6
OUI version     : 11.2.0.1.0
OUI location    : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch2
009-07-07_16-07-24PM.log

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt

ApplySession applying interim patch '5555555' to OH
'/u01/app/oracle/acfsmounts/acfs_db1'

Running prerequisite checks...

OPatch detected non-cluster Oracle Home from the inventory and
will patch the local system only.

Backing up files and inventory (not for auto-rollback) for the
Oracle Home
Backing up files affected by the patch '5555555' for
restore. This might take a while...
Backing up files affected by the patch '5555555' for
rollback. This might take a while...

Patching component oracle.rdbms, 11.2.0.1.0...
The patch will be installed on active database instances.
Running mkpatch on 'bug5943776.pch' to make it compatible with
the Oracle binary...

Installing and enabling the online patch 'bug5943776.pch', on
database 'orcl'.

ApplySession adding interim patch '5555555' to inventory

Verifying the update...
Inventory check OK: Patch ID 5555555 is registered in Oracle
Home inventory with proper meta-data.
Files check OK: Files from Patch ID 5555555 are present in
Oracle Home.

OPatch succeeded.

$ $ORACLE_HOME/OPatch/opatch lsinventory
Invoking OPatch 11.1.0.6.6
```

练习 2-4：使用联机打补丁（续）

```

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.

Oracle Home      : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory : /u01/app/oracle/oraInventory
   from           : /etc/oraInst.loc
OPatch version   : 11.1.0.6.6
OUI version      : 11.2.0.1.0
OUI location     : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch2
009-09-29_13-27-06PM.log

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt

Lsinventory Output file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/lsinv/
lsinventory2009-09-29_13-27-06PM.txt
-----
Installed Top-level Products (1):

Oracle Database 11g                               11.2.0.1.0
There are 1 products installed in this Oracle Home.
Interim patches (1) :

Patch (online) 5555555: applied on Tue Jul 07 16:07:31
GMT+07:00 2009
Unique Patch ID: 1
   Created on 6 Jul 2009, 16:53:30 hrs PST8PDT
   Bugs fixed:
     5943776
-----

OPatch succeeded.
$

```

- 5) 调用 SQL*Plus 并以 SYS 用户身份进行连接。请再次执行以下命令：

```
select * from mysql_is_great;
```

会看到什么结果？

- a) 这次出现了标准 ORA-00942 错误。退出 SQL*Plus。

```

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 13:30:59
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:

```

练习 2-4：使用联机打补丁（续）

```

Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select * from mysql_is_great;
select * from mysql_is_great
          *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

6) 在数据库实例仍在运行时，回退补丁程序。会看到什么结果？

a) 补丁程序被快速删除。

```

$ $ORACLE_HOME/OPatch/opatch rollback -id 5555555 -
connectString orcl:::
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.
Oracle Home      : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory : /u01/app/oracle/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch2
009-09-29_13-31-53PM.log

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt

RollbackSession rolling back interim patch '5555555' from OH
'/u01/app/oracle/acfsmounts/acfs_db1'

Running prerequisite checks...

OPatch detected non-cluster Oracle Home from the inventory and
will patch the local system only.

```

练习 2-4：使用联机打补丁（续）

```
Backing up files affected by the patch '5555555' for
restore. This might take a while...

Patching component oracle.rdbms, 11.2.0.1.0...
The patch will be removed from active database instances.
Disabling and removing online patch 'bug5943776.pch', on
database 'orcl'

RollbackSession removing interim patch '5555555' from
inventory

OPatch succeeded.

$ $ORACLE_HOME/OPatch/opatch lsinventory
Invoking OPatch 11.1.0.6.6

Oracle Interim Patch Installer version 11.1.0.6.6
Copyright (c) 2009, Oracle Corporation. All rights reserved.

Oracle Home      : /u01/app/oracle/acfsmounts/acfs_db1
Central Inventory : /u01/app/oracle/oraInventory
   from           : /etc/oraInst.loc
OPatch version    : 11.1.0.6.6
OUI version       : 11.2.0.1.0
OUI location      : /u01/app/oracle/acfsmounts/acfs_db1/oui
Log file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch2
009-09-29_13-33-15PM.log

Patch history file:
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/opatch_
history.txt

Lsinventory Output file location :
/u01/app/oracle/acfsmounts/acfs_db1/cfgtoollogs/opatch/lsinv/
lsinventory2009-09-29_13-33-15PM.txt
-----
Installed Top-level Products (1):

Oracle Database 11g                               11.2.0.1.0
There are 1 products installed in this Oracle Home.
There are no Interim patches installed in this Oracle Home.
-----

OPatch succeeded.
$
```

练习 2-4：使用联机打补丁（续）

- 7) 调用 SQL*Plus 并以 sys 用户身份进行连接。执行以下命令：

```
select * from mysql_is_great;
```

会看到什么结果？

- a) 此时会再次出现内部错误。退出 SQL*Plus。

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Sep 29 13:34:51
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select * from mysql_is_great;
select * from mysql_is_great
          *
ERROR at line 1:
ORA-00600: internal error code, arguments: [HAH!], [ORACLE],
[IS], [BETTER!], [], [], [], [], [], [], [], []

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 8) 在以 oracle 用户身份连接的终端窗口中，执行 OP_cleanup.sh 脚本以还原环境。

```
$ cd /home/oracle/solutions/OP
$ ./OP_cleanup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 13:37:28
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
```

练习 2-4：使用联机打补丁（续）

```
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
chmod 755 /u01/app/oracle/acfsmounts/acfs_db1/bin

- Linking Oracle
rm -f /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
gcc -o /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
-m32 -L/u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/
-L/u01/app/oracle/acfsmounts/acfs_db1/lib/ -
...
/u01/app/oracle/acfsmounts/acfs_db1/bin/oracle
/u01/app/oracle/acfsmounts/acfs_db1/bin/oracleO
mv /u01/app/oracle/acfsmounts/acfs_db1/rdbms/lib/oracle
/u01/app/oracle/acfsmounts/acfs_db1/bin/oracle
chmod 6751 /u01/app/oracle/acfsmounts/acfs_db1/bin/oracle

SQL*Plus: Release 11.2.0.1.0 Production on Tue Sep 29 13:38:22
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.

SQL> SQL> ORACLE instance started.

Total System Global Area  422670336 bytes
Fixed Size                  1337116 bytes
Variable Size              167774436 bytes
Database Buffers           247463936 bytes
Redo Buffers                6094848 bytes
Database mounted.
Database opened.
SQL> SQL> select * from mysql_is_great
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 2-4：使用联机打补丁（续）

- 9) 在以 oracle 用户身份连接的终端窗口中，使用以下命令重新启动 Enterprise Manager Database Control:

```
emctl start dbconsole
```

```
$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control
..... started.
-----
Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/edrsr40p1.us.oracle.com_
orcl/sysman/log
$
```

第 3 课的练习

在本练习中，您将使用 Oracle Restart 来管理组件。

练习 3-1：使用 Oracle Restart 管理组件

在本练习中，您将使用 Oracle Restart 管理自动重新启动的数据库组件。

- 1) 使用 Enterprise Manager 确定 orcl 数据库实例当前是否由 Oracle Restart 管理。
 - a) 单击 “Database（数据库）” 选项卡转到数据库主页。
 - b) 在数据库主页上，查看 “High Availability（高可用性）” 部分，确定您的数据库实例是否已 “Enabled（启用）” Oracle Restart。
- 2) 要确定 Oracle Restart 的效果，请中止 orcl 数据库实例的 LGWR 进程。会看到什么结果？
 - a) 为 orcl 数据库实例设置环境变量。中止 LGWR 进程后，该实例将关闭。稍等片刻，Oracle Restart 自动重新启动了该实例。

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
$ srvctl config database -d orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/acfsmounts/acfs_db1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: automatic
Disk Groups: DATA,FRA
Services:
$
$ ps -ef | grep ora_lgwr_orcl
oracle      8502          1   0 Sep29 ?           00:00:03 ora_lgwr_orcl
oracle     10115      5384   0 07:01 pts/2    00:00:00 grep ora_lgwr_orcl
$ kill -9 8502
$ ps -ef | grep ora_lgwr_orcl
oracle     10277      5384   0 07:03 pts/2    00:00:00 grep ora_lgwr_orcl
$ ps -ef | grep ora_lgwr_orcl
oracle     10336          1   0 07:03 ?           00:00:00 ora_lgwr_orcl
oracle     10519      5384   0 07:04 pts/2    00:00:00 grep ora_lgwr_orcl
$
```

练习 3-1：使用 Oracle Restart 管理组件（续）

- 3) 从 Oracle Restart 配置中删除 orcl 数据库实例。

```
$ srvctl remove database -d orcl
PRKO-3141 : Database orcl could not be removed because it was
running
$ srvctl stop database -d orcl
$ srvctl remove database -d orcl
Remove the database orcl? (y/[n]) y
$ srvctl remove database -d orcl
PRCD-1120 : The resource for database orcl could not be found.
PRCR-1001 : Resource ora.orcl.db does not exist
$
```

- 4) 重新启动该 orcl 数据库实例。

```
$ srvctl start database -d orcl
PRCD-1120 : The resource for database orcl could not be found.
PRCR-1001 : Resource ora.orcl.db does not exist
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 07:14:20
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 222300544 bytes
Database Buffers 192937984 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 3-1：使用 Oracle Restart 管理组件（续）

5) 再次中止该 orcl 数据库实例的 LGWR 进程。会看到什么结果？

a) 此时，数据库实例将关闭，不会再自动重新启动。

```
$ ps -ef | grep ora_lgwr_orcl
oracle 11500      1  0 07:14 ?        00:00:00 ora_lgwr_orcl
oracle 11794  5384  0 07:17 pts/2  00:00:00 grep ora_lgwr_orcl
$ kill -9 11500
$
$ ps -ef | grep ora_lgwr_orcl
oracle 11841  5384  0 07:18 pts/2  00:00:00 grep ora_lgwr_orcl
$
```

6) 重新启动该 orcl 数据库实例。

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 07:19:51
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.

SQL> startup
ORACLE instance started.

Total System Global Area  422670336 bytes
Fixed Size                  1336960 bytes
Variable Size              222300544 bytes
Database Buffers           192937984 bytes
Redo Buffers                6094848 bytes
Database mounted.
Database opened.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 3-1：使用 Oracle Restart 管理组件（续）

- 7) 使用 Enterprise Manager 将 orcl 数据库实例重新添加至 Oracle Restart 配置中。会看到什么结果？
 - a) 以 SYS 用户身份连接 Database Console，然后转到数据库主页。
 - b) 刷新几次，直到数据库不再显示为脱机为止。您也可以尝试将 Database Console 停止，再重新启动，确保其为最新状态。在终端窗口中使用以下命令：

```
$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://...:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.
$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://...:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control
..... started.
-----
Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/..._orcl/sysman/log
$
```

- c) 转到数据库主页的“High Availability（高可用性）”部分，单击“Oracle Restart”字段旁的“Disabled（禁用）”链接。（注：如果 Enterprise Manager 页显示 Oracle Restart 未禁用，但无链接可单击，则您需要手动执行步骤 f 中的 SRVCTL 命令，然后继续执行步骤 i。）
- d) 在“Specify Host Credentials（指定主机身份证明）”页的“Username（用户名）”和“Password（口令）”两个字段中输入 oracle，然后选择“Save as Preferred Credential（另存为首选身份证明）”。
- e) 单击“Login（登录）”。
- f) 在“Confirmation: Register Database（确认：注册数据库）”页上，可以看见用于在 Oracle Restart 中注册 orcl 的命令：


```
srvctl add database -d orcl -o
/u01/app/oracle/acfsmounts/acfs_db1
```
- g) 单击“Continue（继续）”。
- h) 返回数据库主页后，应看见已对 orcl 数据库实例启用了 Oracle Restart。

练习 3-1：使用 Oracle Restart 管理组件（续）

- i) 不过，只有数据库资源重新添加回 Oracle Restart 配置中。您可以使用 SRVCTL 对此进行验证：

```
$ srvctl config database -d orcl
Database unique name: orcl
Database name:
Oracle home: /u01/app/oracle/acfsmounts/acfs_db1
Oracle user: oracle
Spfile:
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: automatic
Disk Groups:
Services:
```

- 8) 将 orcl 数据库实例及其从属对象重新添加回 Oracle Restart 配置中。

```
$ srvctl status database -d orcl
Database is not running.

$ srvctl start database -d orcl
$ srvctl status database -d orcl
Database is running.
$
$ srvctl config database -d orcl
Database unique name: orcl
Database name:
Oracle home: /u01/app/oracle/acfsmounts/acfs_db1
Oracle user: oracle
Spfile:
Domain:
Start options: open
Stop options: immediate
Database role: PRIMARY
Management policy: automatic
Disk Groups:
Services:
$
$ srvctl modify database -d orcl -n orcl -p
+DATA/orcl/spfileorcl.ora -a DATA,FRA
$ srvctl config database -d orcl
Database unique name: orcl
Database name: orcl
Oracle home: /u01/app/oracle/acfsmounts/acfs_db1
Oracle user: oracle
Spfile: +DATA/orcl/spfileorcl.ora
Domain:
Start options: open
Stop options: immediate
```

练习 3-1：使用 Oracle Restart 管理组件（续）

```
Database role: PRIMARY
Management policy: automatic
Disk Groups: DATA,FRA
Services:
$
```

- 9) 确保 Oracle Restart 对该 orcl 数据库实例起作用。再次中止 LGWR 进程并验证其是否自动重新启动。

```
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     12033      1    0 07:20 ?          00:00:00 ora_lgwr_orcl
oracle     13712    5384    0 07:31 pts/2      00:00:00 grep lgwr
$ kill -9 12033
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     13844    5384    0 07:33 pts/2      00:00:00 grep lgwr
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     13905      1    0 07:33 ?          00:00:00 ora_lgwr_orcl
oracle     14036    5384    0 07:34 pts/2      00:00:00 grep lgwr
$
```

- 10) 在 orcl 数据库实例上禁用 Oracle Restart。

```
$ srvctl disable database -d orcl
$ srvctl status database -d orcl -f
Database is disabled
Database is running.
$
```

- 11) 验证 Oracle Restart 不再自动重新启动 orcl 数据库实例。再次中止 LGWR 进程，然后验证该实例不再自动重新启动。

```
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     13905      1    0 07:33 ?          00:00:00 ora_lgwr_orcl
oracle     14273    5384    0 07:36 pts/2      00:00:00 grep lgwr
$ kill -9 13905
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     14414    5384    0 07:38 pts/2      00:00:00 grep lgwr
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     14414    5384    0 07:38 pts/2      00:00:00 grep lgwr
$ ps -ef | grep lgwr
oracle      4848      1    0 Sep28 ?          00:00:00 asm_lgwr_+ASM
oracle     14414    5384    0 07:38 pts/2      00:00:00 grep lgwr
$
```

练习 3-1：使用 Oracle Restart 管理组件（续）

12) 在 orcl 数据库实例上再次启用 Oracle Restart。

```
$ srvctl enable database -d orcl
$ srvctl status database -d orcl -f
Database is not running.
$ srvctl status database -d orcl -f
Database is running.
$
```

第 4 课的练习

此练习包括使用 ASM 快速镜像重新同步。

练习 4-1：使用 ASM 快速镜像重新同步

在本练习中，将比较在出现非数据问题（与磁盘上的数据不相关的故障）后添加脱机磁盘（已脱机的磁盘）所用的时间。比较相同的操作，一次不使用 ASM 快速镜像重新同步，一次使用 ASM 快速镜像重新同步。

- 1) 确定现有 ASM 磁盘组的兼容值。会看到什么结果？
 - a) 将 ASM 和数据库兼容性属性都设置为默认值。

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 07:59:27
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.
Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> col name format a10
SQL> col compatibility format a20
SQL> col database_compatibility format a25
SQL> select name,compatibility,database_compatibility from
v$asm_diskgroup;
```

NAME	COMPATIBILITY	DATABASE_COMPATIBILITY
DATA	11.2.0.0.0	10.1.0.0.0
FRA	11.2.0.0.0	11.2.0.0.0
ACFS	11.2.0.0.0	11.2.0.0.0

- 2) 使用 Enterprise Manager Database Control，导航至 DATA 磁盘组页，确保此处 DATA 磁盘组的兼容值与之前所见的兼容值相同。
 - a) 以 SYS 用户身份连接到 Enterprise Manager Database Control。
 - b) 在主页上，在“General（一般信息）”部分中单击 ASM 字段旁的 +ASM 链接。
 - c) 在 ASM 主页上，单击“Disk Groups（磁盘组）”选项卡。

练习 4-1：使用 ASM 快速镜像重新同步（续）

- d) 如果此时显示 “Automatic Storage Management Login（自动存储管理登录）” 页，则在 “Username（用户名）” 字段中输入 `sys`，在 “Password（口令）” 字段中输入 `oracle_4U`，在 “Connect As（连接身份）” 字段中选择 `SYSASM`。选中 “Save as Preferred Credential（另存为首选身份证明）” 复选框，然后单击 “Login（登录）”。
 - e) 在 “Disk Groups（磁盘组）” 页上，单击表中的 **DATA** 链接。
 - f) 此时应看到 “Database Compatibility（数据库兼容性）” 设置为 `10.1.0.0.0`，“ASM Compatibility（ASM 兼容性）” 设置为 `11.2.0.0.0`。此外，请注意，“Disk Repair Time（磁盘修复时间）” 属性设置为 3.6 小时。
- 3) 使用 Enterprise Manager Database Control 更改 DATA 磁盘组的数据库兼容性属性。将其设置为 `11.2.0.0.0`。
 - a) “Disk Group: DATA General（磁盘组：DATA 一般信息）” 子页上，单击 “Advanced Attributes（高级属性）” 部分中的 “Edit（编辑）”。
 - b) 在 “Edit Advanced Attributes for Disk Group: DATA（编辑磁盘组的高级属性：DATA）” 页上，在 “Database Compatibility（数据库兼容性）” 和 “ASM Compatibility（ASM 兼容性）” 字段中输入 `11.2.0.0.0`。单击 “OK（确定）”。
 - 4) 使用 SQL*Plus 验证上述更新已得到正确执行。退出 SQL*Plus。

```
SQL> select name,compatibility,database_compatibility from
v$asm_diskgroup;
```

NAME	COMPATIBILITY	DATABASE_COMPATIBILITY
DATA	11.2.0.0.0	11.2.0.0.0
FRA	11.2.0.0.0	11.2.0.0.0
ACFS	11.2.0.0.0	11.2.0.0.0

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 4-1：使用 ASM 快速镜像重新同步（续）

- 5) 执行 `asm_setup.sh` 脚本（位于 `/home/oracle/solutions/ASM` 目录中）以设置此练习的环境。该脚本使用一个 50 MB 的文件在 DATA 磁盘组中新建一个名为 TBSJFV 的表空间。然后，在新的表空间中新建一个名为 SYSTEM.JFV 的表。接下来，该脚本在新创建的表中插入几行。

```
$ cd /home/oracle/solutions/ASM
$ ./asm_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 08:23:01
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
Tablespace created.

SQL> SQL> drop table jfv purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL>
Table created.

SQL> SQL>
1 row created.

SQL> SQL>
Commit complete.

SQL> SQL> 2 3 4 5 6 7
PL/SQL procedure successfully completed.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 4-1：使用 ASM 快速镜像重新同步（续）

- 6) 使第二个磁盘脱机，该磁盘是 DATA 磁盘组的一部分，确保已将 “Disk Repair Time（磁盘修复时间）” 属性设置为 0：
 - a) 返回 “Disk Group: DATA General（磁盘组：DATA 一般信息）” 页，选择第二个磁盘 (ASMDISK02)，然后单击 “Offline（脱机）”。
 - b) 在 “Confirmation（确认）” 页上，将 “Disk Repair Time（磁盘修复时间）” 从默认值（3.6 个小时）更改为 0，然后单击 “Show SQL（显示 SQL）”。该命令应为：ALTER DISKGROUP DATA OFFLINE DISK ASMDISK02 DROP AFTER 0 h
 - c) 然后单击 “Return（返回）”。
 - d) 返回 “Confirmation（确认）” 页，单击 “Yes（是）”。
- 7) 会看到什么结果？
 - a) 返回 “Disk Group: DATA General（磁盘组：DATA 一般信息）” 页，可以看见 ASM 正在使 ASMDISK02 脱机。由于执行了重新平衡操作，因此脱机操作将花费一些时间。刷新浏览器页，直至看不到 ASM 磁盘 ASMDISK02。
 - b) 返回 “Disk Group: DATA General（磁盘组：DATA 一般信息）” 页，此时应看到剩余的三个磁盘有大约 53% 的空间已占满。这会强制丢失的镜像区在正常磁盘内重新获得平衡。
- 8) 修改 SYSTEM.JFV 表中的一些行（删除 499 行）。是否仍能更新此表？
 - a) 仍可以修改 JFV 表：

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
[oracle@edrsr40p1-orcl ASM]$ sqlplus system

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 11:33:03
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password:

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> delete from jfv where rownum<500;
```

练习 4-1：使用 ASM 快速镜像重新同步（续）

```
499 rows deleted.

SQL> commit;

Commit complete.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 9) 将删除的 ASM 磁盘再添加回 DATA 磁盘组：
 - a) 返回 “Disk Group: DATA General（磁盘组：DATA 一般信息）” 页，单击 “Add（添加）”。
 - b) 在 “Add Disks（添加磁盘）” 页上，从 “Select Member Disks（选择成员磁盘）” 下拉列表中选择 “All Disks（所有磁盘）”。
 - c) 从 “Member Disks（成员磁盘）” 表中，选择 ORCL:ASMDISK02 并单击 “Show SQL（显示 SQL）”。该命令应为：ALTER DISKGROUP DATA ADD DISK 'ORCL:ASMDISK02' SIZE 2304 M
 - d) 单击 “Return（返回）”。
 - e) 在 “Add Disks（添加磁盘）” 页上，单击 “OK（确定）”。
 - f) 返回 “Add Disks（添加磁盘）” 页，此时将会接收到一条错误消息 “Failed to commit: ORA-15032: not all alterations performed ORA-15033: disk 'ORCL:ASMDISK02' belongs to diskgroup 'DATA'.”。这是因为 ASMDISK02 仍被看做是 DATA 磁盘组的一部分，所以在默认情况下，ASM 不希望将其覆盖。
 - g) 现在，请确保选中 ORCL:ASMDISK02 及其对应的 “Force Reuse（强制重用）” 复选框。
 - h) 单击 “Show SQL（显示 SQL）”。该命令应为：ALTER DISKGROUP DATA ADD DISK 'ORCL:ASMDISK2' SIZE 2304 M FORCE
 - i) 单击 “Return（返回）”。
 - j) 返回至 “Add Disks（添加磁盘）” 页上，单击 “OK（确定）”。
- 注：**如果接收到内部服务器错误，请刷新浏览器或单击 “Back（返回）” 按钮，然后继续执行步骤 10。

练习 4-1：使用 ASM 快速镜像重新同步（续）

10) 会看到什么结果？

- a) 返回 “Disk Group: DATA (磁盘组: DATA)” 页，持续刷新该页面几次，直到重新平衡操作完成。
- b) 在重新平衡操作结束后，应看到每个 ASM 磁盘有 40% 的空间已占满。

11) 使 DATA 磁盘组中的第二个磁盘脱机，确保已将 “Disk Repair Time (磁盘修复时间)” 属性设置为默认值。然后，再次修改 SYSTEM.JFV 表（再删除 499 行）。会看到什么结果？

- a) 返回 “Disk Group: DATA General (磁盘组: DATA 一般信息)” 页，选择第二个磁盘 (ASMDISK02)，然后单击 “Offline (脱机)”。
- b) 在 “Confirmation (确认)” 页上，保留字段 “Disk Repair Time (磁盘修复时间)” 的默认值 3.6 个小时，然后单击 “Yes (是)”。
- c) 返回 “Disk Group: DATA General (磁盘组: DATA 一般信息)” 页，可以看到 ASMDISK02 的大小与脱机操作之前相同。即使刷新浏览器页，也不会发生重新平衡活动。要对此进行检查，只需查看该页 “General (一般信息)” 部分的 “Pending Operations (暂挂操作)” 字段。其值应该仍为 0。
- d) 仍可以修改 SYSTEM.JFV 表：

```
$ . oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
[oracle@edrsr40p1-orcl ASM]$ sqlplus system

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 11:44:02
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Enter password: *****

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> delete from system.jfv where rownum<500;

499 rows deleted.

SQL> commit;
```

练习 4-1：使用 ASM 快速镜像重新同步（续）

```
Commit complete.
```

```
SQL> exit
```

```
Disconnected from Oracle Database 11g Enterprise Edition
```

```
Release 11.2.0.1.0 - Production
```

```
With the Partitioning, Automatic Storage Management, OLAP,
```

```
Data Mining and Real Application Testing options
```

```
$
```

12) 现在，如何将脱机磁盘重新添加回 DATA 磁盘组？

- a) 无需清除已删除的磁盘或强制再次创建该磁盘。
- b) 返回“Disk Group: DATA General（磁盘组：DATA 一般信息）”页，选择脱机磁盘 (ASMDISK02) 并单击“Online（联机）”。
- c) 在确认页中，单击“Yes（是）”。
- d) 返回“Disk Group: DATA General（磁盘组：DATA 一般信息）”页，应看到该磁盘回到其原来的占用程度（占满大约 39% 的空间），无需执行任何重新平衡操作。您可以看到状态是 SYNCING（将鼠标划过对应的红色十字）。几乎会立即将该磁盘重新添加回磁盘组。刷新浏览器窗口，查看模式的更改。

第 5 课的练习

在本课的练习中，您将使用表压缩。

练习 5-1：使用表压缩

在构建和维护大型数据仓库的开销中，磁盘系统的开销可能占相当大的一部分。Oracle Database 11g 通过对数据进行压缩降低了此项开销，并且对于常规查询操作未因节约空间而导致数据访问时间延长。压缩带来的开销通常与数据的初始压缩以及修改或更改数据的操作 (DML) 相关。

- 1) 转到 /home/oracle/solutions/TC 目录。调用 SQL*Plus，并以 SYS 用户身份登录。执行 setup.sql 脚本。

```
$ cd /home/oracle/solutions/TC

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 11:59:26
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @setup
Connected.
SQL> alter user sh identified by sh account unlock
      2 quota unlimited on users;

User altered.

SQL> grant create tablespace to sh;

Grant succeeded.

SQL> grant drop tablespace to sh;

Grant succeeded.

SQL>
```

练习 5-1：使用表压缩（续）

- 2) 可以比较压缩表的存储要求和未压缩表的存储要求。首先创建 SALES 表的两个副本，一个进行了压缩，另一个未进行压缩。在 SQL*Plus 会话中，执行 create_sales_tbls.sql 脚本。

```
SQL> @create_sales_tbls
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> REM
SQL> REM
SQL> connect sh/sh
Connected.
SQL> drop table sales_nocompress purge
  2  /
drop table sales_nocompress purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> drop table sales_compress purge
  2  /
drop table sales_compress purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> set echo on
SQL> set timing on
SQL> create table sales_nocompress
  2  as select * from sales
  3  /

Table created.

Elapsed: 00:00:16.09
SQL>
SQL> create table sales_compress compress for all operations
  2  as select * from sales where 1=0
  3  /

Table created.

Elapsed: 00:00:00.14
SQL>
SQL> select count(*)
  2  from sales_compress
  3  /
```

练习 5-1：使用表压缩（续）

```

COUNT(*)
-----
          0

Elapsed: 00:00:00.00
SQL>
SQL> @oltp_insert
SQL> set timing on
SQL> declare
  2
  3     commit_after integer := 0 ;
  4     loop_variable integer ;
  5
  6     cursor c_sales is
  7     select prod_id
  8           , cust_id
  9           , time_id
 10           , channel_id
 11           , promo_id
 12           , quantity_sold
 13           , amount_sold
 14     from sales ;
 15
 16 begin
 17
 18     for r_sales in c_sales
 19     loop
 20
 21         if commit_after = 0
 22         then
 23
 24             loop_variable := 0 ;
 25
 26             commit_after := round(dbms_random.value(1,1)) ;
 27
 28         end if ;
 29
 30         insert into sales_compress
 31         (prod_id, cust_id, time_id, channel_id, promo_id,
 32          quantity_sold, amount_sold)
 33         values
 34         ( r_sales.prod_id
 35         , r_sales.cust_id
 36         , r_sales.time_id
 37         , r_sales.channel_id
 38         , r_sales.promo_id
 39         , r_sales.quantity_sold
 40         , r_sales.amount_sold
 41         ) ;
 41

```

练习 5-1：使用表压缩（续）

```

42     if loop_variable = commit_after
43     then
44         commit ;
45         commit_after := 0 ;
46     end if ;
47
48     loop_variable := loop_variable + 1 ;
49
50 end loop ;
51
52 end ;
53 /

```

PL/SQL procedure successfully completed.

Elapsed: 00:03:38.49

SQL> set timing off

SQL>

SQL> set timing off

```

SQL> select count(*) from sales_compress
2 /

```

```

COUNT(*)
-----
918843

```

```

SQL> select count(*) from sales_nocompress
2 /

```

```

COUNT(*)
-----
918843

```

SQL>

- 3) 现在即可比较刚刚创建的两个表的存储要求。在 SQL*Plus 会话中，执行 examine_storage.sql 脚本：

```

SQL> @examine_storage
SQL> set echo off
SQL>
SQL> select segment_name, sum(bytes)/1024/1024 mb
2   from dba_segments
3   where owner = user
4   and segment_name in ('SALES_COMPRESS','SALES_NOCOMPRESS')
5   group by segment_name
6   order by segment_name
7   /

```

练习 5-1：使用表压缩（续）

```

SEGMENT_NAME                                MB
-----
SALES_COMPRESS                             15
SALES_NOCOMPRESS                           36

2 rows selected.

SQL>
SQL> select table_name,compression,compress_for from
user_tables where table_name like '%COMPRESS%';

TABLE_NAME                                COMPRESS COMPRESS_FOR
-----
SALES_NOCOMPRESS                           DISABLED
SALES_COMPRESS                             ENABLED  OLTP

2 rows selected.

SQL>

```

- 4) 尽管您可能预计修改压缩表的数据要比修改未压缩表的数据慢得多，但其差别并不明显。在 SQL*Plus 会话中，执行 dml_perf.sql 脚本。

```

SQL> @dml_perf
SQL> set echo off
Connected.
SQL> set timing on
SQL> delete from sales_nocompress
2 /

918843 rows deleted.

Elapsed: 00:01:18.07
SQL>
SQL> commit;

Commit complete.

Elapsed: 00:00:04.72
SQL>
SQL> delete from sales_compress
2 /

918843 rows deleted.

Elapsed: 00:01:12.94
SQL>
SQL> commit;

Commit complete.

```

练习 5-1：使用表压缩（续）

```

Elapsed: 00:00:00.10
SQL>
SQL> set timing off
SQL> select count(*) from sales_nocompress
 2 /

COUNT(*)
-----
          0

1 row selected.

SQL> select count(*) from sales_compress
 2 /

COUNT(*)
-----
          0

1 row selected.

SQL>

1 row selected.

SQL>

```

- 5) 要清理环境，请执行 cleanup.sql 脚本：

```

SQL> @cleanup
SQL> connect sh/sh
Connected.
SQL> drop table sales_nocompress purge
 2 /

Table dropped.

SQL> drop table sales_compress purge
 2 /

Table dropped.

SQL> drop table q_sales purge;
drop table q_sales purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

```

练习 5-1：使用表压缩（续）

```
SQL> drop table q_sales2 purge;
drop table q_sales2 purge
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

第 6 课的练习

在本课的练习中，您将使用新的分区方案。还将使用 SQL 访问指导获取分区建议。

练习 6-1：使用引用分区

使用引用分区时，通过从父表继承分区键（而无需复制键列），可以从逻辑上对具有父/子关系的表均匀分区。在本练习中，您将创建名为 `ORDERS` 的范围分区表，然后创建名为 `ORDER_ITEMS` 的表，按 `ORDERS` 表中包含的外键进行引用分区。

- 1) 以 `oracle` 用户身份打开一个终端窗口，转到 `$HOME/solutions/part` 目录，启动以 `SYS` 用户身份连接的 `SQL*Plus` 会话。在 `SQL*Plus` 会话中，执行 `setup.sql` 脚本。

```
$ $ . oraenv
ORACLE_SID = [db10g] ? orcl
$
$ cd /home/oracle/solutions/part
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 12:15:02
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @setup

SQL> alter user sh identified by sh account unlock;

User altered.

SQL> grant create tablespace to sh;

Grant succeeded.

SQL> grant drop tablespace to sh;

Grant succeeded.

SQL> alter user hr identified by hr account unlock;

User altered.

SQL> alter user oe identified by oe account unlock;

User altered.
```

练习 6-1：使用引用分区（续）

```
SQL> grant select on oe.orders to sh;

Grant succeeded.

SQL> grant select on oe.order_items to sh;

Grant succeeded.

SQL> grant select on hr.employees to sh;

Grant succeeded.
SQL>
```

- 2) 仍在 SQL*Plus 会话中，以 SH 用户身份进行连接。执行 create_orders.sql 脚本，创建范围分区的 ORDERS 表。

```
SQL> connect sh/sh
Connected.
SQL> @create_orders
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL> -- cleanup
SQL>
SQL> drop table order_items
  2 /
drop table order_items
      *

ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> drop table orders
  2 /
drop table orders
      *

ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL>
SQL> -- create initial tablespaces
SQL>
SQL> drop tablespace x;
drop tablespace x
      *
```

练习 6-1：使用引用分区（续）

```
ERROR at line 1:
ORA-00959: tablespace 'X' does not exist

SQL> create tablespace x datafile '/tmp/x.f' size 10m reuse;

Tablespace created.

SQL>
SQL> drop tablespace y;
drop tablespace y
*
ERROR at line 1:
ORA-00959: tablespace 'Y' does not exist

SQL> create tablespace y datafile '/tmp/y.f' size 10m reuse;

Tablespace created.

SQL>
SQL> -- create a range-partitioned table orders
SQL>
SQL>
SQL> create table orders
  2  ( order_id number(12) not null
  3    , order_date date not null
  4    , order_mode varchar2(8)
  5    , order_status varchar2(1)
  6  )
  7  partition by range (order_date)
  8  ( partition p_before_jan_2006 values less than
(to_date('01-JAN-2006','dd-MON-yyyy'))
  9    , partition p_2006_jan values less than (to_date('01-FEB-
2006','dd-MON-yyyy'))
 10    , partition p_2006_feb values less than (to_date('01-MAR-
2006','dd-MON-yyyy'))
 11    , partition p_2006_mar values less than (to_date('01-APR-
2006','dd-MON-yyyy'))
 12    , partition p_2006_apr values less than (to_date('01-MAY-
2006','dd-MON-yyyy'))
 13    , partition p_2006_may values less than (to_date('01-JUN-
2006','dd-MON-yyyy'))
 14    , partition p_2006_jun values less than (to_date('01-JUL-
2006','dd-MON-yyyy'))
 15    , partition p_2006_jul values less than (to_date('01-AUG-
2006','dd-MON-yyyy'))
 16    , partition p_2006_aug values less than (to_date('01-SEP-
2006','dd-MON-yyyy'))
 17    , partition p_2006_sep values less than (to_date('01-OCT-
2006','dd-MON-yyyy'))
```

练习 6-1：使用引用分区（续）

```

18  , partition p_2006_oct values less than (to_date('01-NOV-
2006','dd-MON-yyyy'))
19  , partition p_2006_nov values less than (to_date('01-DEC-
2006','dd-MON-yyyy'))
20  , partition p_2006_dec values less than (to_date('01-JAN-
2007','dd-MON-yyyy'))
21  ) parallel
22  /
Table created.

SQL>
SQL> alter table orders add constraint orders_pk
      2 primary key (order_id)
      3 /

Table altered.

SQL>

```

- 3) 执行 create_order_items.sql 脚本，创建引用分区的 ORDER_ITEMS 表。

```

SQL>
SQL> @create_order_items
SQL> REM
SQL> REM REF Partitioning
SQL> REM 3.Create a REF partitioned table that depends on the
SQL> REM first created one (e.g. Lineitems, pk-fk order_id)
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL> -- Create a table order_items with a reference to the
SQL> REM orders table partition scheme
SQL>
SQL>
SQL> Rem see ORA-14652 for restrictions around REF
SQL> Partitioning FKs
SQL>
SQL> create table order_items
      2 ( order_id number(12) not null
      3   , product_id number not null
      4   , quantity number not null

```

练习 6-1：使用引用分区（续）

```

5  , sales_amount number not null
6  , constraint order_items_orders_fk foreign key (order_id)
references orders(order_id)
7  )
8  partition by reference (order_items_orders_fk)
9  parallel
10 /

Table created.

SQL>

```

- 4) 执行 query_dict_1.sql 脚本，查看关于引用分区的 ORDER_ITEMS 表的信息。

```

SQL> @query_dict_1
SQL> REM
SQL> REM REF Partitioning
SQL> REM 4.show the partition setup for the REF partitioned
table in the dictionary
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set echo on
SQL>
SQL>
SQL> -- show how the partitions were created
SQL>
SQL>
SQL> select table_name, partitioning_type,
ref_ptn_constraint_name
2  from user_part_tables
3  where table_name in ('ORDERS','ORDER_ITEMS');

```

TABLE_NAME	PARTITION	REF_PTN_CONSTRAINT_NAME
ORDERS	RANGE	
ORDER_ITEMS	REFERENCE	ORDER_ITEMS_ORDERS_FK

```

2 rows selected.

```

练习 6-1：使用引用分区（续）

```
SQL>
SQL> select table_name, partition_name, high_value
       2   from user_tab_partitions
       3   where table_name in ('ORDERS','ORDER_ITEMS')
       4   order by partition_position, table_name;
```

TABLE_NAME	PARTITION_NAME	HIGH_VALUE
ORDERS	P_BEFORE_JAN_2006	TO_DATE(' 2006-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_BEFORE_JAN_2006	
ORDERS	P_2006_JAN	TO_DATE(' 2006-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_JAN	
ORDERS	P_2006_FEB	TO_DATE(' 2006-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_FEB	
ORDERS	P_2006_MAR	TO_DATE(' 2006-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_MAR	
ORDERS	P_2006_APR	TO_DATE(' 2006-05-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_APR	
ORDERS	P_2006_MAY	TO_DATE(' 2006-06-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_MAY	
ORDERS	P_2006_JUN	TO_DATE(' 2006-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_JUN	
ORDERS	P_2006_JUL	TO_DATE(' 2006-08-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_JUL	
ORDERS	P_2006_AUG	TO_DATE(' 2006-09-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_AUG	
ORDERS	P_2006_SEP	TO_DATE(' 2006-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_SEP	

练习 6-1：使用引用分区（续）

```

ORDER_ITEMS                P_2006_SEP
ORDERS                      P_2006_OCT
TO_DATE(' 2006-11-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                P_2006_OCT
ORDERS                      P_2006_NOV
TO_DATE(' 2006-12-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                P_2006_NOV
ORDERS                      P_2006_DEC
TO_DATE(' 2007-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                P_2006_DEC

26 rows selected.

SQL>
SQL> select up.table_name, up.partitioning_type, uc.table_name
ref_table
   2   from user_part_tables up,
   3   (select r.table_name, r.constraint_name from
user_constraints uc, user_constraints r
   4   where uc.constraint_name=r.constraint_name and
uc.owner=r.owner) uc
   5   where up.ref_ptn_constraint_name = uc.constraint_name(+)
   6   and up.table_name in ('ORDERS','ORDER_ITEMS');

TABLE_NAME                PARTITION REF_TABLE
-----
ORDERS                    RANGE
ORDER_ITEMS              REFERENCE ORDER_ITEMS

2 rows selected.

SQL>

```

- 5) 执行 insert_orders.sql 脚本，在 ORDERS 和 ORDER_ITEMS 表中插入数据。

```

SQL> @insert_orders
SQL> REM
SQL> REM REF Partitioning
SQL> REM 5.insert data into orders. Show data placement in an
explicit partition
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on

```

练习 6-1：使用引用分区（续）

```
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL> -- insert some data into the orders and order_items
tables
SQL>
SQL>
SQL> insert into orders values (1, to_date('23-OCT-2006','dd-
MON-yyyy'), 'manual', 'U') ;

1 row created.

SQL> insert into orders values (2, to_date('24-OCT-2006','dd-
MON-yyyy'), 'auto', 'I') ;

1 row created.

SQL> insert into orders values (3, to_date('25-OCT-2006','dd-
MON-yyyy'), 'manual', 'P') ;

1 row created.

SQL> insert into orders values (4, to_date('26-OCT-2006','dd-
MON-yyyy'), 'manual', 'U') ;

1 row created.

SQL> insert into orders values (5, to_date('26-NOV-2006','dd-
MON-yyyy'), 'manual', 'U') ;

1 row created.

SQL> insert into order_items values (1, 12, 40, 12000) ;

1 row created.

SQL> insert into order_items values (1, 2, 1, 2000) ;

1 row created.

SQL> insert into order_items values (1, 8, 20, 8400) ;

1 row created.

SQL> insert into order_items values (2, 1, 4, 900) ;

1 row created.
```


练习 6-1：使用引用分区（续）

```
SQL> insert into order_items values (2, 87, 4, 6200) ;
1 row created.
SQL> insert into order_items values (2, 2, 4, 3400) ;
1 row created.
SQL> insert into order_items values (2, 6, 2, 70) ;
1 row created.
SQL> insert into order_items values (3, 1, 90, 1200) ;
1 row created.
SQL> insert into order_items values (4, 90, 10, 9000) ;
1 row created.
SQL> insert into order_items values (4, 74, 12, 8000) ;
1 row created.
SQL> insert into order_items values (4, 21, 1, 900) ;
1 row created.
SQL> insert into order_items values (4, 90, 4, 39000) ;
1 row created.
SQL> insert into order_items values (4, 32, 6, 700) ;
1 row created.
SQL> insert into order_items values (4, 45, 2, 80) ;
1 row created.
SQL> insert into order_items values (4, 72, 6, 72000) ;
1 row created.
SQL> insert into order_items values (5, 1, 40, 42000) ;
1 row created.
```

练习 6-1：使用引用分区（续）

```
SQL> insert into order_items values (5, 2, 40, 800) ;

1 row created.

SQL> insert into order_items values (5, 52, 80, 100) ;

1 row created.

SQL> insert into order_items values (5, 90, 4, 1200) ;

1 row created.

SQL> commit
      2  /

Commit complete.
SQL>
```

- 6) 执行 show_data_placement.sql 脚本，观察数据在分区中的分配情况。

```
SQL> @show_data_placement
SQL> REM
SQL> REM REF Partitioning
SQL> REM 5.insert data into orders. Show data placement in an
explicit partition
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set echo on
SQL>
SQL> set feedback on
SQL>
SQL> -- see how the data was co-located in the same partitions
SQL>
SQL> alter session set nls_date_format='dd-mon-yyyy' ;

Session altered.

SQL>
SQL> select order_id, order_date
      2  from orders partition (p_2006_oct)
      3  /
```

练习 6-1：使用引用分区（续）

```

ORDER_ID ORDER_DATE
-----
1 23-oct-2006
2 24-oct-2006
3 25-oct-2006
4 26-oct-2006

4 rows selected.

SQL>
SQL>
SQL> select oi.order_id, o.order_date, sum(oi.sales_amount)
sum_sales
2   from orders o
3   , order_items partition (p_2006_oct) oi
4  where o.order_id = oi.order_id
5  group by oi.order_id, o.order_date
6  order by oi.order_id
7  /

ORDER_ID ORDER_DATE    SUM_SALES
-----
1 23-oct-2006         22400
2 24-oct-2006         10570
3 25-oct-2006          1200
4 26-oct-2006        129680

4 rows selected.

SQL>
SQL>
SQL> select order_id, order_date
2   from orders partition (p_2006_nov)
3  /

ORDER_ID ORDER_DATE
-----
5 26-nov-2006

1 row selected.

SQL>
SQL>
SQL> select oi.order_id, o.order_date, sum(oi.sales_amount)
sum_sales
2   from orders o
3   , order_items partition (p_2006_nov) oi
4  where o.order_id = oi.order_id

```

练习 6-1：使用引用分区（续）

```

5  group by oi.order_id, o.order_date
6  order by oi.order_id
7  /

ORDER_ID ORDER_DATE    SUM_SALES
-----
5 26-nov-2006      44100

1 row selected.

SQL>

```

- 7) 执行 drop_partition.sql 脚本，删除 ORDERS 表中的 p_before_jan_2006 分区。

```

SQL> @drop_partition
SQL> REM
SQL> REM REF Partitioning
SQL> REM 6.insert data into the REF partitioned table and show
SQL> REM that co-location of the dependent data is taken place
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set feedback on
SQL>
SQL>
SQL> -- show the manageability aspects of inheriting the
SQL> partitioning scheme (drop a partition for the ORDERS table)
SQL>
SQL> set echo on
SQL> alter table orders drop partition p_before_jan_2006
SQL> update global indexes
SQL> 2 /

Table altered.

SQL>

```

练习 6-1：使用引用分区（续）

8) 执行 query_dict_2.sql 脚本，查看这些表的相关信息。

```
SQL> @query_dict_2
SQL> REM
SQL> REM REF Partitioning
SQL> REM 6.insert data into the REF partitioned table and show
SQL> REM that co-location of the dependent data is taken place
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set feedback on
SQL>
SQL>
SQL> -- again, show the partitions
SQL>
SQL> set echo on
SQL> select table_name, partition_name, high_value
2  from user_tab_partitions
3  where table_name in ('ORDERS','ORDER_ITEMS')
4  order by partition_position, table_name
5  /
```

TABLE_NAME	PARTITION_NAME	HIGH_VALUE
ORDERS	P_2006_JAN	TO_DATE(' 2006-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_JAN	
ORDERS	P_2006_FEB	TO_DATE(' 2006-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_FEB	
ORDERS	P_2006_MAR	TO_DATE(' 2006-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_MAR	
ORDERS	P_2006_APR	TO_DATE(' 2006-05-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_APR	

练习 6-1：使用引用分区（续）

```

ORDER_ITEMS                                P_2006_APR
ORDERS                                      P_2006_MAY
TO_DATE(' 2006-06-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_MAY
ORDERS                                      P_2006_JUN
TO_DATE(' 2006-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_JUN
ORDERS                                      P_2006_JUL
TO_DATE(' 2006-08-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_JUL
ORDERS                                      P_2006_AUG
TO_DATE(' 2006-09-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_AUG
ORDERS                                      P_2006_SEP
TO_DATE(' 2006-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_SEP
ORDERS                                      P_2006_OCT
TO_DATE(' 2006-11-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_OCT
ORDERS                                      P_2006_NOV
TO_DATE(' 2006-12-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_NOV
ORDERS                                      P_2006_DEC
TO_DATE(' 2007-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS                                P_2006_DEC

24 rows selected.

SQL>

```

- 9) 执行 add_partition.sql 脚本，添加 p2007_01/02 分区。

```

SQL> @add_partition
SQL> REM
SQL> REM REF Partitioning
SQL> REM 6.insert data into the REF partitioned table and show
that co-location of the dependent data is taken place
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000

```

练习 6-1：使用引用分区（续）

```

SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set echo on
SQL> set feedback on
SQL>
SQL> -- manageability aspects and flexibility to combine PMOPs
but still have the flexibility to place various partitions
SQL>
SQL> alter table orders
  2  add partition p2007_01 values less than (to_date('01-feb-
2007','dd-mon-yyyy'))
  3  tablespace x;

Table altered.

SQL>
SQL> -- show co-location of partitions for parent and child
SQL>
SQL> select table_name, partition_name, tablespace_name,
high_value
  2  from user_tab_partitions where table_name in
('ORDERS','ORDER_ITEMS')
  3  order by partition_position, table_name;

```

TABLE_NAME	PARTITION_NAME	TABLESPACE_NAME	HIGH_VALUE
ORDERS	P_2006_JAN	USERS	TO_DATE(' 2006-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_JAN	USERS	
ORDERS	P_2006_FEB	USERS	TO_DATE(' 2006-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_FEB	USERS	
ORDERS	P_2006_MAR	USERS	TO_DATE(' 2006-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_MAR	USERS	
ORDERS	P_2006_APR	USERS	TO_DATE(' 2006-05-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS	P_2006_APR	USERS	

练习 6-1：使用引用分区（续）

```

ORDER_ITEMS          P_2006_APR          USERS
ORDERS                P_2006_MAY          USERS
TO_DATE(' 2006-06-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_MAY          USERS
ORDERS                P_2006_JUN          USERS
TO_DATE(' 2006-07-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_JUN          USERS
ORDERS                P_2006_JUL          USERS
TO_DATE(' 2006-08-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_JUL          USERS
ORDERS                P_2006_AUG          USERS
TO_DATE(' 2006-09-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_AUG          USERS
ORDERS                P_2006_SEP          USERS
TO_DATE(' 2006-10-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_SEP          USERS
ORDERS                P_2006_OCT          USERS
TO_DATE(' 2006-11-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_OCT          USERS
ORDERS                P_2006_NOV          USERS
TO_DATE(' 2006-12-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_NOV          USERS
ORDERS                P_2006_DEC          USERS
TO_DATE(' 2007-01-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_DEC          USERS
ORDERS                P2007_01          X
TO_DATE(' 2007-02-01 00:00:00', 'SYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P2007_01          X

26 rows selected.

SQL>
SQL> -- second PMOP
SQL>
SQL> alter table orders
  2  add partition p2007_02 values less than (to_date('01-mar-
2007','dd-mon-yyyy'))
  3  tablespace x
  4  dependent tables ( order_items (partition foo tablespace
y));

```


练习 6-1：使用引用分区（续）

Table altered.		
SQL>		
SQL> select table_name, partition_name, tablespace_name,		
high_value		
2 from user_tab_partitions		
3 where table_name in ('ORDERS','ORDER_ITEMS') order by		
partition_position, table_name;		
TABLE_NAME	PARTITION_NAME	
TABLESPACE_NAME	HIGH_VALUE	

ORDERS	P_2006_JAN	USERS
TO_DATE(' 2006-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_JAN	USERS
ORDERS	P_2006_FEB	USERS
TO_DATE(' 2006-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_FEB	USERS
ORDERS	P_2006_MAR	USERS
TO_DATE(' 2006-04-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_MAR	USERS
ORDERS	P_2006_APR	USERS
TO_DATE(' 2006-05-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_APR	USERS
ORDERS	P_2006_MAY	USERS
TO_DATE(' 2006-06-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_MAY	USERS
ORDERS	P_2006_JUN	USERS
TO_DATE(' 2006-07-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_JUN	USERS
ORDERS	P_2006_JUL	USERS
TO_DATE(' 2006-08-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_JUL	USERS
ORDERS	P_2006_AUG	USERS
TO_DATE(' 2006-09-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		
ORDER_ITEMS	P_2006_AUG	USERS
ORDERS	P_2006_SEP	USERS
TO_DATE(' 2006-10-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',		
'NLS_CALENDAR=GREGORIAN')		

练习 6-1：使用引用分区（续）

```

ORDER_ITEMS          P_2006_SEP          USERS
ORDERS                P_2006_OCT          USERS
TO_DATE(' 2006-11-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_OCT          USERS
ORDERS                P_2006_NOV          USERS
TO_DATE(' 2006-12-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_NOV          USERS
ORDERS                P_2006_DEC          USERS
TO_DATE(' 2007-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P_2006_DEC          USERS
ORDERS                P2007_01          X
TO_DATE(' 2007-02-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          P2007_01          X
ORDERS                P2007_02          X
TO_DATE(' 2007-03-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN')
ORDER_ITEMS          FOO                Y

28 rows selected.

SQL>

```

- 10) 执行 cleanup_1.sql 脚本，删除在此练习中创建的分区和表空间。退出 SQL*Plus。

```

SQL> @cleanup_1
SQL> REM
SQL> REM REF Partitioning
SQL> REM
SQL>
SQL> set echo on
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> set echo on
SQL> set feedback on
SQL>
SQL> -- cleanup
SQL>
SQL> alter table orders drop partition p2007_01;

Table altered.

```

练习 6-1：使用引用分区（续）

```
SQL> alter table orders drop partition p2007_02;

Table altered.

SQL>
SQL> drop tablespace x;

Tablespace dropped.

SQL> drop tablespace y;

Tablespace dropped.

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 6-2: 使用间隔分区

间隔分区可以完全自动地创建范围分区。管理新分区的创建可能是一项重复性很高的繁重任务。对于可预测性地增加涵盖小范围的分区，如添加新的每日分区，这种情况尤其突出。间隔分区可以通过按需创建分区来自动完成此类操作。假设您的当前目录为 /home/oracle/solutions/part。

- 1) 执行 create_newsales.sql 脚本，创建 NEWSALES 间隔分区表。

```
$ . oraenv
ORACLE_SID = [db10g] ? orcl
$ cd /home/oracle/solutions/part
$
$ sqlplus sh/sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 12:33:12
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @create_newsales
SQL>
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 1.Introduce the concept of Interval Partitioning
SQL> REM 2.Create an interval partitioned table
SQL> REM
SQL>
SQL> -- cleanup
SQL> set echo on
SQL>
SQL> drop table historical_newsales
      2 /
drop table historical_newsales
      *
ERROR at line 1:
ORA-00942: table or view does not exist
```

练习 6-2: 使用间隔分区 (续)

```

SQL>
SQL> drop table newsales
      2  /
drop table newsales
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> -- create interval partitioned table
SQL>
SQL> create table newsales
      2  ( prod_id number(6) not null
      3    , cust_id number not null
      4    , time_id date not null
      5    , channel_id char(1) not null
      6    , promo_id number(6) not null
      7    , quantity_sold number(3) not null
      8    , amount_sold number(10,2) not null
      9  )
     10 partition by range (time_id)
     11 interval (numtodsinterval(1,'DAY'))
     12 ( partition p_before_1_jan_2005 values
     13   less than (to_date('01-01-2005','dd-mm-yyyy')))
     14 /

Table created.

SQL>
SQL>

```

- 2) 执行 query_dict_3.sql 脚本, 查询 USER_TAB_PARTITIONS 字典视图以了解有关 NEWSALES 表的信息。

```

SQL>
SQL> @query_dict_3
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 3.Show the segment allocation of it
SQL> REM
SQL> set echo on
SQL> -- show partitions and partition boundaries for the
interval partitioned table
SQL> select partition_name, high_value

```

练习 6-2: 使用间隔分区 (续)

```

2  from user_tab_partitions
3  where table_name = 'NEWSALES'
4  order by partition_position
5  /

```

```

PARTITION_NAME                HIGH_VALUE
-----

```

```

P_BEFORE_1_JAN_2005          TO_DATE(' 2005-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')

```

```

1 row selected.

```

```

SQL>

```

- 3) 执行 insert_newsales.sql 脚本, 将新数据插入强制创建新的分区 (段) 的 NEWSALES 表中。

```

SQL>
SQL> @insert_newsales
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 4.Insert new data that forces the creation of a new
partition (segment)
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> -- insert some data after January 1 2005
SQL>
SQL> insert into newsales values (11160,17450,to_date('01-jan-
2005','dd-mon-yyyy'),'I',9999,19,798) ;

1 row created.

SQL> insert into newsales values (1340,33710,to_date('02-jan-
2005','dd-mon-yyyy'),'S',9999,16,1264) ;

1 row created.

```

练习 6-2: 使用间隔分区 (续)

```
SQL> insert into newsales values (25270,65880,to_date('05-jan-2005','dd-mon-yyyy'),'I',9999,5,210) ;

1 row created.

SQL> insert into newsales values (1615,73480,to_date('05-jan-2005','dd-mon-yyyy'),'I',9999,8,96) ;

1 row created.

SQL> insert into newsales values (1900,84910,to_date('06-jan-2005','dd-mon-yyyy'),'I',9999,42,378) ;

1 row created.

SQL> insert into newsales values (8085,37900,to_date('09-jan-2005','dd-mon-yyyy'),'S',9999,1,68) ;

1 row created.

SQL> insert into newsales values (755,26590,to_date('09-jan-2005','dd-mon-yyyy'),'I',9999,11,132) ;

1 row created.

SQL> insert into newsales values (10,68060,to_date('09-jan-2005','dd-mon-yyyy'),'P',9999,28,4900) ;

1 row created.

SQL> insert into newsales values (13425,109310,to_date('10-jan-2005','dd-mon-yyyy'),'I',9999,1,68) ;

1 row created.

SQL> insert into newsales values (1955,65190,to_date('10-jan-2005','dd-mon-yyyy'),'S',9999,28,1512) ;

1 row created.

SQL>
SQL> set feedback on
SQL>
SQL> commit
    2  /

Commit complete.

SQL>
```

练习 6-2: 使用间隔分区 (续)

4) 执行 query_dict_4.sql 脚本, 查看有关该新分区的信息。

```
SQL>
SQL> @query_dict_4
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 5.Show the segment allocation again-you will see a
new segment
SQL> REM
SQL> set echo on
SQL> -- see how new partitions were created
SQL>
SQL> select partition_name, high_value
2   from user_tab_partitions
3   where table_name = 'NEWSALES'
4   order by partition_position
5   /
```

PARTITION_NAME	HIGH_VALUE
P_BEFORE_1_JAN_2005	TO_DATE(' 2005-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P21	TO_DATE(' 2005-01-02 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P22	TO_DATE(' 2005-01-03 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P23	TO_DATE(' 2005-01-06 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P24	TO_DATE(' 2005-01-07 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P25	TO_DATE(' 2005-01-10 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')
SYS_P26	TO_DATE(' 2005-01-11 00:00:00',
'SYYYY-MM-DD HH24:MI:SS',	'NLS_CALENDAR=GREGORIAN')

7 rows selected.

SQL>

练习 6-2: 使用间隔分区 (续)

- 5) 执行 merge_partition.sql 脚本, 合并两个分区。

```
SQL>
SQL> @merge_partition
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 6.Do a partition maintenance operation on older
Partitions (e.g. MERGE). This will demonstrate the fact that
an interval partitioned table can have manually managed older
range partition components
SQL> REM
SQL> -- merge 2 partitions: traditional functionality of
course works
SQL> set echo on
SQL> alter table newsales
  2 merge partitions for(to_date('01-JAN-2005','dd-MON-
YYYY'))
  3 , for(to_date('02-JAN-2005','dd-MON-yyyy'))
  4 into partition p_before_3_jan_2005
  5 /

Table altered.

SQL>
```

- 6) 执行 query_dict_5.sql 脚本, 查看有关合并分区的信息。

```
SQL>
SQL> @query_dict_5
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 7.take a standard range partitioned table and extend
it to an interval partitioned table. This demonstrates the
migration path to this new functionality.
SQL> REM
SQL> set echo on
SQL> -- show the list of partitions again
```

练习 6-2: 使用间隔分区 (续)

```
SQL> select partition_name, high_value
       2   from user_tab_partitions
       3   where table_name = 'NEWSALES'
       4   order by partition_position
       5   /
```

PARTITION_NAME	HIGH_VALUE
P_BEFORE_1_JAN_2005	TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_BEFORE_3_JAN_2005	TO_DATE(' 2005-01-03 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
SYS_P23	TO_DATE(' 2005-01-06 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
SYS_P24	TO_DATE(' 2005-01-07 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
SYS_P25	TO_DATE(' 2005-01-10 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
SYS_P26	TO_DATE(' 2005-01-11 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')

6 rows selected.

SQL>

- 7) 执行 create_hist_newsales.sql 脚本, 创建一个范围分区的表。

```
SQL>
SQL> @create_hist_newsales
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 7.take a standard range partitioned table and extend
SQL> REM it to an interval partitioned table. This demonstrates the
SQL> REM migration path to this new functionality.
SQL> REM
SQL> set echo on
SQL> -- create a range partitioned table, that will be
SQL> REM migrated to interval partitioning
SQL> create table historical_newsales
       2   ( prod_id number(6) not null
       3     , cust_id number not null
       4     , time_id date not null
       5     , channel_id char(1) not null
```

练习 6-2: 使用间隔分区 (续)

```

6  , promo_id number(6) not null
7  , quantity_sold number(3) not null
8  , amount_sold number(10,2) not null
9  )
10 partition by range (time_id)
11 (partition p_previous_century values
12 less than (to_date('01-JAN-2000','dd-MON-yyyy')))
13 , partition p_2000 values less than (to_date('01-JAN-
2001','dd-MON-yyyy'))
14 , partition p_2001 values less than (to_date('01-JAN-
2002','dd-MON-yyyy'))
15 , partition p_2002 values less than (to_date('01-JAN-
2003','dd-MON-yyyy'))
16 , partition p_2003 values less than (to_date('01-JAN-
2004','dd-MON-yyyy'))
17 , partition p_2004 values less than (to_date('01-JAN-
2005','dd-MON-yyyy'))
18 , partition p_2005 values less than (to_date('01-JAN-
2006','dd-MON-yyyy'))
19 )
20 /

```

Table created.

SQL>

- 8) 执行 query_dict_6.sql 脚本, 查看有关范围分区的信息。

```

SQL>
SQL> @query_dict_6
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM 7.take a standard range partitioned table and extend
it to an interval partitioned table. This demonstrates the
migration path to this new functionality.
SQL> REM
SQL> set echo on
SQL> -- show the list of partitions
SQL> select partition_name, high_value
2  from user_tab_partitions
3  where table_name = 'HISTORICAL_NEWSALES'
4  order by partition_position
5  /

```

练习 6-2: 使用间隔分区 (续)

PARTITION_NAME	HIGH_VALUE
-----	-----
P_PREVIOUS_CENTURY	TO_DATE(' 2000-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2000	TO_DATE(' 2001-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2001	TO_DATE(' 2002-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2002	TO_DATE(' 2003-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2003	TO_DATE(' 2004-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2004	TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2005	TO_DATE(' 2006-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
7 rows selected.	
SQL>	

- 9) 执行 insert_histnewsales_row_1.sql 脚本, 在 HISTORICAL_NEWSALES 表中插入一行。由于该分区表目前还不是间隔分区的表, 插入操作将失败。

```
SQL>
SQL> @insert_histnewsales_row_1
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM Interval Partitioning
SQL> REM 7.take a standard range partitioned table and extend
it to an interval partitioned table. This demonstrates the
migration path to this new functionality.
SQL> REM
SQL> set echo on
SQL> -- try to insert a new row in 2006 (THIS WILL FAIL)
SQL> insert into historical_newsales
  2 values (11160,17450,to_date('02-jan-2006','dd-mon-
yyy''), 'I', 9999,19,798)
  3 /
insert into historical_newsales
*
```

练习 6-2: 使用间隔分区 (续)

```
ERROR at line 1:
ORA-14400: inserted partition key does not map to any
partition
```

```
SQL>
```

10) 执行 alter_hist_newsales.sql 脚本，将分区表更改为间隔分区的表。

```
SQL>
SQL> @alter_hist_newsales
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM 7.take a standard range partitioned table and extend
it to an interval partitioned table. This demonstrates the
migration path to this new functionality.
SQL> REM
SQL> set echo on
SQL> -- change the partitioned table to become interval
partitioned
SQL> alter table historical_newsales
  2  set interval (NUMTOYMINTERVAL(1, 'YEAR'))
  3  /

Table altered.

SQL>
```

11) 执行 insert_histnewsales_row_2.sql 脚本，尝试再次向该表中插入一行。

```
SQL> @insert_histnewsales_row_2
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM
SQL> set echo on
SQL> -- try to insert a new row in 2006 again (NOW IT WILL
SUCCEED)
```

练习 6-2: 使用间隔分区 (续)

```
SQL> insert into historical_newsales
  2 values (11160,17450,to_date('02-jan-2006','dd-mon-
  3 yyyy'),'I',9999,19,798)

1 row created.

SQL>
SQL> commit
  2 /

Commit complete.
```

12) 执行 query_dict_7.sql 脚本, 查看有关分区的信息。退出 SQL*Plus。

```
SQL> @query_dict_7
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM
SQL> REM Interval Partitioning
SQL> REM
SQL> set echo on
SQL> -- show the list of partitions again
SQL> select partition_name, high_value
  2 from user_tab_partitions
  3 where table_name = 'HISTORICAL_NEWSALES'
  4 order by partition_position
  5 /
```

PARTITION_NAME	HIGH_VALUE
P_PREVIOUS_CENTURY	TO_DATE(' 2000-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2000	TO_DATE(' 2001-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2001	TO_DATE(' 2002-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2002	TO_DATE(' 2003-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2003	TO_DATE(' 2004-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
P_2004	TO_DATE(' 2005-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')

练习 6-2: 使用间隔分区 (续)

```
P_2005                TO_DATE(' 2006-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')
SYS_P27              TO_DATE(' 2007-01-01 00:00:00',
'SYYYY-MM-DD HH24:MI:SS', 'NLS_CALENDAR=GREGORIAN')

8 rows selected.

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 6-3：使用扩展的组合分区

范围-范围组合分区支持沿两个维进行逻辑范围分区；例如按 `ORDER_DATE` 的分区以及按 `SHIPPING_DATE` 的范围子分区。组合范围-范围分区提供了另外一种建模策略，可以将业务需求映射到对象的分区。

在此示例中，与客户签订的服务级协议声明，订单将在下达订单的那个日历月内交付。确定的订单类型包括：

EARLY：在订单下达之后的下一个日历月中旬之前交付的订单。这些订单的交付时间往往早于客户预期的时间。

AGREED：在下达订单的那个日历月交付的订单（但不是 **EARLY** 订单）。

LATE：在下达订单之后的第二个日历月才开始交付的订单。

假设您的当前目录为 `/home/oracle/solutions/part`。

执行以下步骤，创建一个范围-范围组合分区的表以满足业务需求：

- 1) 以 SH 用户身份连接，然后执行 `create_shipments.sql` 脚本创建 SHIPMENTS 表，将其创建为按 `ORDER_DATE` 和 `SHIPDATE` 分区的范围-范围组合分区表。

```
$ . oraenv
ORACLE_SID = [db10g] ? orcl
$ cd /home/oracle/solutions/part
$
$ sqlplus sh/sh
SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 12:56:55
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
SQL>
SQL> @create_shipments
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> REM New extended composite partitions
SQL> REM 1.Recap the concept of composite partitioned tables
SQL> REM 2.Introduce the new extension based on business cases
(e.g. RANGE-RANGE on oder_date/ship_date
```


练习 6-3: 使用扩展的组合分区 (续)

```

SQL> REM 3.Demonstrate the data placement for a business
scenario
SQL> REM
SQL> REM EXTENDED COMPOSITE PARTITIONING
SQL> REM
SQL>
SQL> drop table shipments
2 /
drop table shipments
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> set echo on
SQL>
SQL> -- create table shipments, composite partition by
order_date, shipdate
SQL>
SQL> create table shipments
2 ( order_id number not null
3 , order_date date not null
4 , ship_date date not null
5 , customer_id number not null
6 , sales_amount number not null
7 )
8 partition by range (order_date)
9 subpartition by range (ship_date)
10 ( partition p_2006_jul values
11 less than (to_date('01-AUG-2006','dd-MON-yyyy'))
12 ( subpartition p_2006_jul_early values
13 less than (to_date('15-AUG-2006','dd-MON-yyyy'))
14 , subpartition p_2006_jul_agreed values
15 less than (to_date('01-SEP-2006','dd-MON-yyyy'))
16 , subpartition p_2006_jul_late values less than
(maxvalue)
17 )
18 , partition p_2006_aug values
19 less than (to_date('01-SEP-2006','dd-MON-yyyy'))
20 ( subpartition p_2006_aug_early values
21 less than (to_date('15-SEP-2006','dd-MON-yyyy'))
22 , subpartition p_2006_aug_agreed values
23 less than (to_date('01-OCT-2006','dd-MON-yyyy'))
24 , subpartition p_2006_aug_late values less than
(maxvalue)
25 )
26 , partition p_2006_sep values
27 less than (to_date('01-OCT-2006','dd-MON-yyyy'))
28 ( subpartition p_2006_sep_early values
29 less than (to_date('15-OCT-2006','dd-MON-yyyy'))

```

练习 6-3: 使用扩展的组合分区 (续)

```

30      , subpartition p_2006_sep_agreed values
31        less than (to_date('01-NOV-2006','dd-MON-yyyy'))
32      , subpartition p_2006_sep_late values less than
(maxvalue)
33    )
34  , partition p_2006_oct values
35    less than (to_date('01-NOV-2006','dd-MON-yyyy'))
36    ( subpartition p_2006_oct_early values
37      less than (to_date('15-NOV-2006','dd-MON-yyyy'))
38    , subpartition p_2006_oct_agreed values
39      less than (to_date('01-DEC-2006','dd-MON-yyyy'))
40    , subpartition p_2006_oct_late values less than
(maxvalue)
41    )
42  , partition p_2006_nov values
43    less than (to_date('01-DEC-2006','dd-MON-yyyy'))
44    ( subpartition p_2006_nov_early values
45      less than (to_date('15-DEC-2006','dd-MON-yyyy'))
46    , subpartition p_2006_nov_agreed values
47      less than (to_date('01-JAN-2007','dd-MON-yyyy'))
48    , subpartition p_2006_nov_late values less than
(maxvalue)
49    )
50  , partition p_2006_dec values
51    less than (to_date('01-JAN-2007','dd-MON-yyyy'))
52    ( subpartition p_2006_dec_early values
53      less than (to_date('15-JAN-2007','dd-MON-yyyy'))
54    , subpartition p_2006_dec_agreed values
55      less than (to_date('01-FEB-2007','dd-MON-yyyy'))
56    , subpartition p_2006_dec_late values less than
(maxvalue)
57    )
58  )
59  /

```

Table created.

SQL>

- 2) 使用 insert_shipments.sql 脚本, 在 SHIPMENTS 表中插入多个行。

```

SQL>
SQL> @insert_shipments
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>

```

练习 6-3: 使用扩展的组合分区 (续)

```
SQL> REM New extended composite partitions
SQL> REM 1.Recap the concept of composite partitioned tables
SQL> REM 2.Introduce the new extension based on business cases
(e.g. RANGE-RANGE on oder_date/ship_date
SQL> REM 3.Demonstrate the data placement for a business
scenario
SQL> REM
SQL>
SQL> REM
SQL> REM EXTENDED COMPOSITE PARTITIONING
SQL> REM
SQL>
SQL> set echo on
SQL> -- insert some data into the shipments table
SQL>
SQL> insert into shipments select o.order_id
   2  , to_date('12-oct-2006','dd-mon-yyyy')
   3  , to_date('12-oct-2006','dd-mon-yyyy') + trunc(70 *
abs(dbms_random.value)) -- generate some early, agreed, late
orders
   4  , nvl(o.sales_rep_id,trunc(105 * dbms_random.value))
   5  , sum(oi.unit_price * oi.quantity)
   6  from oe.orders o
   7  , oe.order_items oi
   8  where o.order_id = oi.order_id
   9  group by o.order_id
  10  , o.sales_rep_id
  11  /

105 rows created.

SQL>
SQL> commit
   2  /

Commit complete.

SQL>
```

练习 6-3: 使用扩展的组合分区（续）

- 3) 执行 count_shipments.sql 脚本，查看数据在 SHIPMENTS 表中的分布方式。**注：**由于在前一步执行 INSERT 语句时随机生成值，所以您的结果可能会有所不同。

```
SQL> @count_shipments
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL> REM New extended composite partitions
SQL> REM 1.Recap the concept of composite partitioned tables
SQL> REM 2.Introduce the new extension based on business cases
(e.g. RANGE-RANGE on oder_date/ship_date
SQL> REM 3.Demonstrate the data placement for a business
scenario
SQL> REM
SQL>
SQL> REM
SQL> REM EXTENDED COMPOSITE PARTITIONING
SQL> REM
SQL> set echo on
SQL>
SQL> -- show how data was distributed for this business case
SQL>
SQL>
SQL> select count(1) from shipments subpartition
(P_2006_OCT_EARLY) ;

COUNT(1)
-----
         42

1 row selected.

SQL> select count(1) from shipments subpartition
(P_2006_OCT_AGREED) ;

COUNT(1)
-----
        30

1 row selected.
```

练习 6-3：使用扩展的组合分区（续）

```
SQL> select count(1) from shipments subpartition
(P_2006_OCT_LATE) ;

COUNT(1)
-----
          33

1 row selected.

SQL>
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 6-4：使用基于虚拟列的分区

假设您的当前目录为 `/home/oracle/solutions/part`。

可使用虚拟列分区对表的虚拟列上定义的键列进行分区。通过计算表达式来定义虚拟列。可以在创建或修改表时定义虚拟列。对逻辑分区对象的业务要求经常与现有列不一一对应。Oracle 分区功能得到了增强，可以在虚拟列上定义分区策略，因而可以更加全面地匹配业务要求。

在此示例中，雇员的分类如下：

差：薪酬总额（薪金 + 佣金）少于 1000

一般：薪酬在 1000 到 3000 之间

小康：薪酬为 3000

相当富裕：薪酬大于 3000

要了解如何使用基于虚拟列的分区，请执行以下步骤：

- 1) 执行 `create_employees.sql` 脚本，创建包含虚拟列的 `EMPLOYEES` 表。

```
$ . oraenv
ORACLE_SID = [db10g] ? orcl
$ cd /home/oracle/solutions/part
$
$ sqlplus sh/sh
SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 13:03:10
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
SQL>
SQL> @create_employees
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
Extract the month out of a date)
```

练习 6-4：使用基于虚拟列的分区（续）

```
SQL> REM 2.Create a simple virtual column and select from it
      (you can also highlight that you can collect stats on such a
      column)
SQL> REM 3.Create a virtual column-based partitioned table
      (for demonstration purposes this can be something goofy like
      adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
      on the virtual column definitions
SQL> REM
SQL>
SQL> set echo on
SQL> drop table employees
      2 /
drop table employees
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> drop table product_returns
      2 /
drop table product_returns
      *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL>
SQL> -- create a table with a virtual column
SQL>
SQL>
SQL> create table employees
      2 ( employee_id number(6) not null
      3 , first_name varchar2(20)
      4 , last_name varchar2(25)
      5 , email varchar2(25)
      6 , phone_number varchar2(20)
      7 , hiredate date
      8 , job_id varchar2(10)
      9 , salary number(8,2)
     10 , commission_pct number(2,2)
     11 , manager_id number(6)
     12 , department_id number(4)
     13 , well_off as (case sign(1000 - (salary +
nvl(commission_pct,0)))
     14 when 1 then 'poor'
     15 else case sign(3000 - (salary +
nvl(commission_pct,0)))
     16 when 1 then 'not doing too bad'
     17 when 0 then 'on target'
```

练习 6-4：使用基于虚拟列的分区（续）

```

18             else 'filthy rich'
19             end
20         end
21     ) virtual
22 )
23 /

```

Table created.

SQL>

- 2) 执行 insert_employees.sql 脚本，在 EMPLOYEES 表中 INSERT 多个行。

```

SQL> @insert_employees
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
SQL> REM Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
SQL> REM (you can also highlight that you can collect stats on such a
SQL> REM column)
SQL> REM 3.Create a virtual column-based partitioned table
SQL> REM (for demonstration purposes this can be something goofy like
SQL> REM adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
SQL> REM on the virtual column definitions
SQL> REM
SQL>
SQL> set echo on
SQL> -- insert data into the table
SQL>
SQL>
SQL> insert into employees
SQL> 2 (employee_id, first_name, last_name, email, phone_number,
SQL> hiredate, job_id, salary, commission_pct, manager_id,
SQL> department_id)
SQL> 3 select employee_id, first_name, last_name, email,
SQL> phone_number, hire_date, job_id, salary, commission_pct,
SQL> manager_id, department_id from hr.employees
SQL> 4 /

```


练习 6-4：使用基于虚拟列的分区（续）

```

107 rows created.SQL>

SQL>
SQL> commit
    2  /

Commit complete.

SQL>

```

3) 执行 select_employees.sql 脚本，查询 EMPLOYEES 表。

```

SQL> @select_employees
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL> column name format a45
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
SQL> REM Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
SQL> REM (you can also highlight that you can collect stats on such a
SQL> REM column)
SQL> REM 3.Create a virtual column-based partitioned table
SQL> REM (for demonstration purposes this can be something goofy like
SQL> REM adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
SQL> REM on the virtual column definitions
SQL> REM
SQL> set echo on
SQL> -- select from the virtual column
SQL>
SQL>
SQL> select first_name ||' '|| last_name name, well_off,
SQL> salary + nvl(commission_pct,0) total_salary
    2  from employees
    3  /

```

NAME	WELL_OFF	TOTAL_SALARY
-----	-----	-----
Donald OConnell	not doing too bad	2600
Douglas Grant	not doing too bad	2600
Jennifer Whalen	filthy rich	4400
Michael Hartstein	filthy rich	13000
...		

练习 6-4：使用基于虚拟列的分区（续）

Vance Jones	not doing too bad	2800
Alana Walsh	filthy rich	3100
Kevin Feeney	on target	3000

107 rows selected.

SQL>

- 4) 执行 create_prod_ret.sql 脚本，创建包含虚拟列的 PRODUCT_RETURNS 表，此虚拟列用于对表进行分区。

```
SQL>
SQL> @create_prod_ret
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
SQL> REM Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
SQL> REM (you can also highlight that you can collect stats on such a
SQL> REM column)
SQL> REM 3.Create a virtual column-based partitioned table
SQL> REM (for demonstration purposes this can be something goofy like
SQL> REM adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
SQL> REM on the virtual column definitions
SQL> REM
SQL> set echo on
SQL> -- now, create a table product returns with a virtual
SQL> column, and partition by this column
SQL>
SQL>
SQL> create table product_returns
2  ( order_id number not null
3  , product_id number not null
4  , customer_id number not null
5  , quantity number not null
6  , unit_price number not null
7  , total_value as (quantity * unit_price) virtual
8  )
9  partition by range(total_value)
10 ( partition dont_even_bother values less than (100)
```

练习 6-4：使用基于虚拟列的分区（续）

```

11 , partition perhaps_the_accountant_cares values less than
(1000)
12 , partition now_it_becomes_worrying values less than
(10000)
13 , partition oops_somebody_be_fired values less than
(100000)
14 , partition this_means_bankruptcy values less than
(maxvalue)
15 )
16 /

```

Table created.

SQL>

- 5) 执行 insert_prod_ret.sql 脚本，在 PRODUCT_RETURNS 表中插入多个行。

```

SQL> @insert_prod_ret
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
(you can also highlight that you can collect stats on such a
column)
SQL> REM 3.Create a virtual column-based partitioned table
(for demonstration purposes this can be something goofy like
adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
on the virtual column definitions
SQL> REM
SQL>
SQL> set echo on
SQL> -- insert some data into the table
SQL>
SQL>
SQL> set feedback off
SQL>
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1, 36, 109, 10,
120) ;

```

练习 6-4：使用基于虚拟列的分区（续）

```
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (2, 46, 129, 20, 10)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (3, 73, 159, 15, 1)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (4,136,1109, 16, 2)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (5, 27, 309, 20,
420) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (6, 96, 129, 10,
1020) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (7, 39, 104, 1,
12000) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (8,436,6109, 8, 200)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (9,789, 709, 2,
2400) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (11,276, 809,9,
10000) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (12, 34, 199, 10,
20000) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (15, 24, 100,5, 10)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (17, 76,1109, 1, 20)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (19, 55, 209, 1,
300000) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (21, 76, 139, 1, 10)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (31, 90, 104, 2,
4020) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (51,284,5109, 9, 60)
;
```

练习 6-4：使用基于虚拟列的分区（续）

```
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1,912, 169, 1,
10980) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (31,536, 179, 2, 60)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (13,752, 108, 1,
400) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (12,986,1109, 3,
700) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1, 39, 129, 4, 6) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1, 32, 103, 1, 30)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (12, 34, 104, 1,
98710) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1,173,6109, 1,
9650) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (17,853,7109, 1, 60)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (19,981, 809, 5, 80)
;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1, 84, 103, 16,
1000) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1, 21,2109, 1000,
90) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (13,782,2359, 102,
1850) ;
SQL> insert into product_returns (order_id, product_id,
customer_id, quantity, unit_price) values (1,941,5309, 10, 40)
;
SQL> commit ;
SQL>
```

练习 6-4：使用基于虚拟列的分区（续）

6) 执行 select_prod_return.sql 脚本，查询 PRODUCT_RETURNS 表。

```
SQL> @select_prod_return
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
SQL> REM Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
SQL> REM (you can also highlight that you can collect stats on such a
SQL> REM column)
SQL> REM 3.Create a virtual column-based partitioned table
SQL> REM (for demonstration purposes this can be something goofy like
SQL> REM adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
SQL> REM on the virtual column definitions
SQL> REM
SQL> set feedback on
SQL> set echo on
SQL> -- select from the different partitions
SQL>
SQL> select order_id, quantity, unit_price, total_value
2  from product_returns partition (dont_even_bother)
3  order by total_value
4  /
```

ORDER_ID	QUANTITY	UNIT_PRICE	TOTAL_VALUE
21	1	10	10
3	15	1	15
17	1	20	20
1	4	6	24
1	1	30	30
4	16	2	32
15	5	10	50
17	1	60	60

8 rows selected.

练习 6-4：使用基于虚拟列的分区（续）

```

SQL>
SQL> select order_id, quantity, unit_price, total_value
  2   from product_returns partition
(perhaps_the_accountant_cares)
  3   order by total_value
  4   /

```

ORDER_ID	QUANTITY	UNIT_PRICE	TOTAL_VALUE
31	2	60	120
2	20	10	200
1	10	40	400
13	1	400	400
19	5	80	400
51	9	60	540

6 rows selected.

```

SQL>
SQL> select order_id, quantity, unit_price, total_value
  2   from product_returns partition (now_it_becomes_worrying)
  3   order by total_value
  4   /

```

ORDER_ID	QUANTITY	UNIT_PRICE	TOTAL_VALUE
1	10	120	1200
8	8	200	1600
12	3	700	2100
9	2	2400	4800
31	2	4020	8040
5	20	420	8400
1	1	9650	9650

7 rows selected.

```

SQL>
SQL> select order_id, quantity, unit_price, total_value
  2   from product_returns partition (oops_somebody_be_fired)
  3   order by total_value
  4   /

```

ORDER_ID	QUANTITY	UNIT_PRICE	TOTAL_VALUE
6	10	1020	10200
1	1	10980	10980
7	1	12000	12000
1	16	1000	16000

练习 6-4：使用基于虚拟列的分区（续）

```

      11          9      10000      90000
      1         1000         90      90000
      12          1      98710      98710

7 rows selected.

SQL>
SQL> select order_id, quantity, unit_price, total_value
  2   from product_returns partition (this_means_bankruptcy)
  3   order by total_value
  4   /

  ORDER_ID    QUANTITY  UNIT_PRICE  TOTAL_VALUE
  -----
      13         102       1850      188700
      12          10      20000      200000
      19          1     300000      300000

3 rows selected.

SQL>

```

- 7) 执行 `xplan_prod_ret.sql` 脚本，对使用定义虚拟列的表达式的查询显示分区修剪。退出 SQL*Plus。

```

SQL> @xplan_prod_ret
SQL> set echo on
SQL> set pagesize 2000
SQL> set long 10000
SQL> set linesize 200
SQL> set feedback on
SQL> column partition_name format a25
SQL> column high_value format a85
SQL>
SQL>
SQL> REM
SQL> REM Virtual column-based partitioning
SQL> REM 1.Introduce the concept of a virtual column (e.g.
SQL> REM Extract the month out of a date)
SQL> REM 2.Create a simple virtual column and select from it
SQL> REM (you can also highlight that you can collect stats on such a
SQL> REM column)
SQL> REM 3.Create a virtual column-based partitioned table
SQL> REM (for demonstration purposes this can be something goofy like
SQL> REM adding two column values or so)
SQL> REM 4.insert some data and show the data placement based
SQL> REM on the virtual column definitions
SQL> REM
SQL>
SQL>
SQL>

```


练习 6-4：使用基于虚拟列的分区（续）

```
SQL> -- show partition pruning for queries that use the
expression that defines the virtual column (remember you can
also gather statistics on the virtual column)
SQL>
SQL> set echo on
SQL> explain plan for select * from product_returns where
quantity * unit_price = 60000
2 /

Explained.

SQL>
SQL>
SQL> select * from table(dbms_xplan.display) ;

PLAN_TABLE_OUTPUT
-----
-----
-----
-----
Plan hash value: 2847847071

-----
-----
| Id | Operation | Name | Rows |
Bytes | Cost (%CPU)| Time | Pstart| Pstop |
-----
-----
| 0 | SELECT STATEMENT | | 1 |
78 | 3 (0)| 00:00:01 | | |
| 1 | PARTITION RANGE SINGLE | | 1 |
78 | 3 (0)| 00:00:01 | 4 | 4 |
|* 2 | TABLE ACCESS FULL | PRODUCT_RETURNS | 1 |
78 | 3 (0)| 00:00:01 | 4 | 4 |
-----
-----

Predicate Information (identified by operation id):
-----

2 - filter("PRODUCT_RETURNS"."TOTAL_VALUE">=60000)

附注
-----
- dynamic sampling used for this statement

18 rows selected.
```

练习 6-4：使用基于虚拟列的分区（续）

```
SQL>
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
$
```

练习 6-5: 使用 SQL 访问指导获取分区建议

在此练习中，将使用 SQL 访问指导获得分区建议案。

- 1) 在以 oracle 用户身份连接的终端会话中，转到
/home/oracle/solutions/partadv 目录，执行 partadv_setup.sh 脚本。此脚本将构建归 SH 用户所有且名为 SQLSET_TEST_500 的 STS，并创建一个模板，用于指示分区指导对非常小的表（如 SH 方案中的表）进行分区。

```
$ cd /home/oracle/solutions/partadv
$ ./partadv_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Sep 30 13:34:05
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
SQL> Connected.
SQL> SQL> 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32 33 34 35 36 37
38 39 40 41 42 43 44 45 46 47 48 49 50
51 52 53 54 55 56 57 58 59 60 61 62 63
64 65 66 67 68 69 70 71 72 73 74 75 76
77 78 79 80
Function created.

SQL> SQL> No errors.
SQL> SQL>
Synonym created.

SQL> SQL>
Grant succeeded.

SQL> SQL>
Grant succeeded.

SQL> SQL> Connected.
SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> 2 3
4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
PL/SQL procedure successfully completed.
```

练习 6-5: 使用 SQL 访问指导获取分区建议 (续)

```
SQL> SQL> SQL> SQL> SQL> SQL> SQL> DROP TABLE temp_table purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
Table created.

SQL> SQL> SQL> SQL> SQL> SQL> SQL>
System altered.

SQL> BEGIN dbms_sqltune.drop_sqlset('SQLSET_TEST_500'); END;

*
ERROR at line 1:
ORA-13754: "SQL Tuning Set" "SQLSET_TEST_500" does not exist
for user "SH".
ORA-06512: at "SYS.DBMS_SQLTUNE_INTERNAL", line 11567
ORA-06512: at "SYS.DBMS_SQLTUNE", line 3118
ORA-06512: at line 1

SQL> SQL>      2      3      4      5      6      7      8      9      10     11
12  13  14  15  16  17  18  19  20  21  22  23  24
25  26  27  28  29  30  31  32  33  34  35  36  37
PL/SQL procedure successfully completed.

SQL> SQL> SQL>
COUNT(*)
-----
         499

1 row selected.

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>      2      3      4
PL/SQL procedure successfully completed.

SQL> SQL>      2      3      4      5  begin
*
ERROR at line 1:
ORA-13605: The specified task or object
SQLACCESS_PARTITION_SMALL does not exist for the current user.
ORA-06512: at "SYS.PRVT_ADVISOR", line 4635
```

练习 6-5: 使用 SQL 访问指导获取分区建议 (续)

```
ORA-06512: at "SYS.DBMS_ADVISOR", line 848
ORA-06512: at line 2

SQL> SQL>      2      3      4      5      6      7      8      9     10
PL/SQL procedure successfully completed.

SQL> SQL> SQL> Connected.
SQL> SQL>
Synonym dropped.

SQL>
Function dropped.

SQL> SQL>
Revoke succeeded.

SQL> SQL>
Grant succeeded.

SQL>
Grant succeeded.

SQL>
Grant succeeded.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 2) 使用 Enterprise Manager Database Control 为使用 SQLACCESS_PARTITION_SMALL 模板的 SQLSET_TEST_500 生成建议。
 - a) 以 SH 用户身份和 SH 口令登录 Enterprise Manager。以 “Normal (常规)” 方式连接。如果执行步骤 1 时您已登录, 请注销, 然后在 Enterprise Manger 高速缓存该用户的权限后重新登录。
 - b) 单击该页底部的 “Advisor Central (指导中心)” 链接。
 - c) 在 “Advisor Central (指导中心)” 页上, 单击 “SQL Advisors (SQL 指导)” 链接。
 - d) 在 “SQL Advisors (SQL 指导)” 页上, 单击 “SQL Access Advisor (SQL 访问指导)” 。

练习 6-5: 使用 SQL 访问指导获取分区建议 (续)

- e) 在 “SQL Access Advisor: Initial Options (SQL 访问指导: 初始选项)” 页上, 选择 “Inherit Options from a previously saved Task or Template (从先前保存的任务或模板继承选项)”。等待页面刷新。在 “Tasks and Templates (任务和模板)” 表中, 确保已从 “View (视图)” 下拉列表中选择了 “Templates Only (仅模板)”。选择 **SQLACCESS_PARTITION_SMALL** 模板, 然后单击 “Continue (继续)”。
- f) 在 “SQL Access Advisor: Workload Source (SQL 访问指导: 工作量源)” 页上, 选择 “Use an existing SQL Tuning Set (使用现有的 SQL 优化集)”, 然后在 “SQL Tuning Set (SQL 优化集)” 字段中输入 **SH.SQLSET_TEST_500**。单击 “Next (下一步)”。
- g) 在 “SQL Access Advisor: Recommendation Options (SQL 访问指导: 建议选项)” 页上, 确保选择了下列设置: “Indexes (索引)”、“Materialized Views (实体化视图)” 和 “Partitioning (分区)” ; 指导应处于 “Comprehensive (综合)” 模式。单击 “Next (下一步)”。
- h) 在 “SQL Access Advisor: Schedule (SQL 访问指导: 调度)” 页上, 请确保将调度设置为 “Immediately (立即执行)”, 然后单击 “Next (下一步)”。
- i) 在 “Review (复查)” 页上, 单击 “Submit (提交)”。
- j) 返回 “Advisor Central (指导中心)” 页, 耐心等待任务完成。
- k) 任务完成后, 请在 “Advisor Central (指导中心)” 页的 “Results (结果)” 部分中选择 SQL 访问指导的任务名称, 然后单击 “View Result (查看结果)” 按钮。
- l) 在相应的任务页上, 单击 “Recommendations (建议)” 选项卡。单击 “Recommendation Details (建议详细资料)” 查看分区建议。
- m) 单击 “OK (确定)”, 返回到任务结果页。
- n) 单击 “Database (数据库)”, 返回到 “Database (数据库)” 主页。

第 7 课的练习

在本练习中，您将对 BasicFiles 和 SecureFiles 进行比较。

练习 7-1：使用 SecureFiles

在本练习中，您将了解 SecureFiles 在压缩、数据加密和性能方面的优势。

1) 要针对 SecureFiles 设置数据库和环境，请执行以下步骤：

- a) 请创建一个目录以存储透明数据加密 (TDE) Wallet。这是 SecureFiles LOB 加密所必需的。

```
$ mkdir /u01/app/oracle/admin/orcl/wallet
$
```

- b) 编辑 /u01/app/oracle/product/11.2.0/grid/network/admin/sqlnet.ora 文件指示 TDE Wallet 的位置，即指定在前一步骤中创建的目录。向该文件中添加以下参数：ENCRYPTION_WALLET_LOCATION=(SOURCE=(METHOD=FILE) (METHOD_DATA=(DIRECTORY=/u01/app/oracle/admin/orcl/wallet)))

```
$ cd /u01/app/oracle/product/11.2.0/grid/network/admin
$ vi sqlnet.ora
$ cat sqlnet.ora
# sqlnet.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/grid/network/admin/sqlnet.ora
# Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

ADR_BASE = /u01/app/oracle

ENCRYPTION_WALLET_LOCATION= (SOURCE=(METHOD=FILE) (METHOD_DATA=
(DIRECTORY=/u01/app/oracle/admin/orcl/wallet)))

$
```

2) 转到 /home/oracle/solutions/SecF 目录。使用 SQL*Plus，以 SYS 用户身份连接到数据库。

```
$ cd /home/oracle/solutions/SecF
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 07:32:51
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
```


练习 7-1：使用 SecureFiles（续）

```
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>
```

- 3) 在 SQL*Plus 会话中，执行 `secf_setup.sql` 脚本。第一个步骤是创建加密密钥并将其存储到 TDE Wallet 中。稍后，您将使用此加密密钥加密 SecureFiles。

```
SQL> @secf_setup
Connected.
SQL> set serveroutput on
SQL> -- set verify on;
SQL> set term on;
SQL> set lines 200
SQL> set pause on
SQL>
SQL> /*== Store the encryption key in the TDE wallet ==*/
SQL>
SQL> alter system set wallet open identified by "welcome1";
alter system set wallet open identified by "welcome1"
*
ERROR at line 1:
ORA-28367: wallet does not exist

SQL>
SQL> ALTER SYSTEM SET ENCRYPTION KEY AUTHENTICATED BY
"welcome1"
2 /

System altered.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 4) 按 [Enter] 继续执行脚本的下一步骤。脚本的下一步骤是为此练习创建数据库方案。该方案称为 `SF_DEMO`。将为该方案授予 `CONNECT` 和 `RESOURCE` 角色。它还需要具有 `EXECUTE ANY PROCEDURE` 和 `CREATE ANY DIRECTORY` 系统权限。

```
SQL>
SQL>
SQL> /*== Create a database account for this OBE ==*/
SQL>
SQL> DROP USER sf_demo CASCADE
2 /
DROP USER sf_demo CASCADE
*
ERROR at line 1:
ORA-01918: user 'SF_DEMO' does not exist
```

练习 7-1: 使用 SecureFiles (续)

```

SQL>
SQL> CREATE USER sf_demo IDENTIFIED BY oracle_4U
      2  DEFAULT TABLESPACE users
      3  TEMPORARY TABLESPACE temp
      4  QUOTA UNLIMITED ON users
      5  /

User created.

SQL>
SQL> GRANT connect, resource TO sf_demo
      2  /

Grant succeeded.

SQL>
SQL> GRANT EXECUTE ANY PROCEDURE, CREATE ANY DIRECTORY TO
sf_demo
      2  /

Grant succeeded.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```

- 5) 按 [Enter] 继续执行脚本的下一步骤。接下来，脚本将创建两个 150 MB 的表空间。SECF_TBS1 表空间用于以 BasicFiles 格式存储 LOB，而 SECF_TBS2 表空间用于以新的 SecureFiles 格式存储 LOB。要使表空间支持 SecureFiles，需要启用自动段空间管理 (ASSM)。

```

SQL>
SQL> /*== Create tablespaces to store the LOB tables      ==*/
SQL> /*==                                                ==*/
SQL> /*== Note that SecureFile LOB storage requires ASSM.==*/
SQL> /*== The segment allocation unit is 64MB.          ==*/
SQL>
SQL> DROP TABLESPACE secf_tbs1 INCLUDING CONTENTS
      2  /
DROP TABLESPACE secf_tbs1 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'SECF_TBS1' does not exist

SQL>
SQL> CREATE TABLESPACE secf_tbs1
      2  DATAFILE '+DATA' SIZE 150M REUSE
      3  EXTENT MANAGEMENT LOCAL
      4  UNIFORM SIZE 64M

```

练习 7-1: 使用 SecureFiles (续)

```

5  SEGMENT SPACE MANAGEMENT AUTO
6  /

Tablespace created.

SQL>
SQL> DROP TABLESPACE secf_tbs2 INCLUDING CONTENTS
2  /
DROP TABLESPACE secf_tbs2 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'SECF_TBS2' does not exist

SQL>
SQL> CREATE TABLESPACE secf_tbs2
2  DATAFILE '+DATA' SIZE 150M REUSE
3  EXTENT MANAGEMENT LOCAL
4  UNIFORM SIZE 64M
5  SEGMENT SPACE MANAGEMENT AUTO
6  /

Tablespace created.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```

- 6) 按 [Enter] 继续执行脚本的下一步骤。脚本的剩余步骤是在 SF_DEMO 方案中创建对象。首先创建名为 RESUMES 的表。此表包含以 BasicFiles 格式存储的 BLOB 列 (RESUME)。

```

SQL> /*== As SF_DEMO, create a table with a LOB that uses
BasicFile LOB storage ==*/
SQL>
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> CREATE TABLE resumes
2  (id NUMBER, first_name VARCHAR2(15),
3  last_name VARCHAR2(40), resume BLOB)
4  LOB(resume) STORE AS BASICFILE
5  (TABLESPACE secf_tbs1)
6  /

Table created.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```

练习 7-1：使用 SecureFiles（续）

- 7) 按 [Enter] 继续执行脚本的下一步骤。需要使用一个 DIRECTORY 对象来存储 MS Word 文件的路径名，之后将检索该文件并将其存储在 LOB 列中。系统提示输入目录名称时，请输入此练习中 SQL 脚本的完整路径。例如，该路径名可以是 /home/oracle/solutions/SecF。

```
SQL>
SQL> /*== Create a DIRECTORY object. It is required for
accessing the MS Word files ==*/
SQL> /*== to be loaded into the LOB column. Enter the full
path to the files, omit ==*/
SQL> /*== the final "/" at the end.
==*/
SQL>
SQL> CREATE OR REPLACE DIRECTORY cwd AS '&directory_name';
Enter value for directory_name: /home/oracle/solutions/SecF
old 1: CREATE OR REPLACE DIRECTORY cwd AS '&directory_name'
new 1: CREATE OR REPLACE DIRECTORY cwd AS
'/home/oracle/solutions/SecF'

Directory created.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 8) 按 [Enter] 继续执行脚本的下一步骤。为了读取文件系统上的 MS Word 文件并将其存储在 LOB 列中，创建了名为 LOADLOBFROMBFILE_PROC 的过程。此过程将调用 DBMS_LOB 程序包。请注意，对于 BasicFiles 和 SecureFiles LOB 存储格式都使用此过程。

```
SQL>
SQL> /*== Create the procedure to read the MS Word files and
load them into the LOB column ==*/
SQL>
SQL> CREATE OR REPLACE PROCEDURE loadLOBFromBFILE_proc
(dest_loc IN OUT BLOB, file_name IN VARCHAR2)
2  IS
3      src_loc          BFILE := BFILENAME('CWD', file_name);
4      amount           INTEGER := 4000;
5  BEGIN
6
7      DBMS_LOB.OPEN(src_loc, DBMS_LOB.LOB_READONLY);
8      amount := DBMS_LOB.GETLENGTH(src_loc);
9      DBMS_LOB.LOADFROMFILE(dest_loc, src_loc, amount);
10     DBMS_LOB.CLOSE(src_loc);
11
12 END;
13 /
```

练习 7-1：使用 SecureFiles（续）

```
Procedure created.
```

```
SQL>
SQL> show errors
No errors.
SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 9) 按 [Enter] 继续执行脚本的下一步骤。为了在 RESUMES 表中插入记录（包括 LOB 值），创建了名为 WRITE_LOB 的过程。此过程将调用在上一步骤中创建的 LOADLOBFROMBFILE_PROC 过程。

```
SQL>
SQL> /*== Create the procedure to insert LOBs into the table
==*/
SQL>
SQL> CREATE OR REPLACE PROCEDURE write_lob (p_file IN
VARCHAR2)
2  IS
3  i  NUMBER;
4  fn VARCHAR2(15);
5  ln VARCHAR2(40);
6  b  BLOB;
7
8  BEGIN
9  DBMS_OUTPUT.ENABLE;
10 DBMS_OUTPUT.PUT_LINE('Begin inserting rows...');
11  FOR i IN 1 .. 30 LOOP
12      fn:=SUBSTR(p_file,1,INSTR(p_file,'.')-1);
13      ln:=SUBSTR(p_file,INSTR(p_file,'.')+1,LENGTH(p_file)
-INSTR(p_file,'.')-4);
14      INSERT INTO resumes VALUES (i, fn, ln, EMPTY_BLOB())
15      RETURNING resume INTO b;
16      loadLOBfromBFILE_proc(b,p_file);
17      DBMS_OUTPUT.PUT_LINE('Row '|| i ||' inserted.');
```

```
Procedure created.
```

```
SQL>
SQL> show errors
No errors.
SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

练习 7-1: 使用 SecureFiles (续)

- 10) 按 [Enter] 继续执行脚本的下一步骤。为了从 RESUMES 表中检索记录（包括 LOB 值），创建了名为 READ_LOB 的过程。

```
SQL>
SQL> /*== Create the procedure to read LOBs from the table
==*/
SQL>
SQL> CREATE OR REPLACE PROCEDURE read_lob
2  IS
3      lob_loc          BLOB;
4      CURSOR resumes_cur IS
5          SELECT id, first_name, last_name, resume
6          FROM resumes;
7      resumes_rec      resumes%ROWTYPE;
8  BEGIN
9      OPEN resumes_cur;
10     LOOP
11         FETCH resumes_cur INTO resumes_rec;
12         lob_loc := resumes_rec.resume;
13         DBMS_OUTPUT.PUT_LINE('The length is: '||
DBMS_LOB.GETLENGTH(lob_loc));
14         DBMS_OUTPUT.PUT_LINE('The ID is: '||
resumes_rec.id);
15         -- just print out the first 200 bytes of the LOB
16         -- because DBMS_OUTPUT.PUT_LINE cannot display more
than 255 bytes
17         DBMS_OUTPUT.PUT_LINE('The blob is read: '||
18
UTL_RAW.CAST_TO_VARCHAR2(DBMS_LOB.SUBSTR(lob_loc,200,1)));
19         EXIT WHEN resumes_cur%NOTFOUND;
20     END LOOP;
21     CLOSE resumes_cur;
22 END;
23 /

Procedure created.

SQL> show errors
No errors.
SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

练习 7-1：使用 SecureFiles（续）

- 11) 按 [Enter] 继续执行脚本的下一步骤。创建了一个名为 CHECK_SPACE_BF 的过程，用于显示分配给 RESUMES 表中 BasicFiles LOB 段的磁盘块的空间使用情况。此过程将调用 DBMS_SPACE 程序包。**注：**DBMS_SPACE.SPACE_USAGE 过程在 Oracle Database 11g 中已被废弃。现在有两个 DBMS_SPACE.SPACE_USAGE 过程。这两个过程不能交替使用。调用 CHECK_SPACE_BF 中的 DBMS_SPACE.SPACE_USAGE 过程仅对 BasicFiles 有效。

```
SQL>
SQL> /*== Create a procedure for checking BasicFile LOB space
usage. ==*/
SQL> /*==
==*/
SQL> /*== Note that the DBMS_SPACE.SPACE_USAGE procedure has
been ==*/
SQL> /*== overloaded. There are 2 DBMS_SPACE.SPACE_USAGE
procedures. ==*/
SQL> /*== The two cannot be used interchangeably. The
invocation of ==*/
SQL> /*== the DBMS_SPACE.SPACE_USAGE procedure here is ONLY
valid ==*/
SQL> /*== for use against BasicFile LOBs.
==*/
SQL>
SQL> CREATE OR REPLACE PROCEDURE check_space_bf
2   IS
3
4       l_fs1_bytes NUMBER;
5       l_fs2_bytes NUMBER;
6       l_fs3_bytes NUMBER;
7       l_fs4_bytes NUMBER;
8       l_fs1_blocks NUMBER;
9       l_fs2_blocks NUMBER;
10      l_fs3_blocks NUMBER;
11      l_fs4_blocks NUMBER;
12      l_full_bytes NUMBER;
13      l_full_blocks NUMBER;
14      l_unformatted_bytes NUMBER;
15      l_unformatted_blocks NUMBER;
16      v_segname VARCHAR2(500);
17  BEGIN
18      SELECT segment_name
19      INTO v_segname
20      FROM user_lobs
21      WHERE table_name = 'RESUMES'
22      AND column_name = 'RESUME';
23
24      DBMS_SPACE.SPACE_USAGE(
25          segment_owner      => 'SF_DEMO',
26          segment_name       => v_segname,
```

练习 7-1: 使用 SecureFiles (续)

```

27      segment_type      => 'LOB',
28      fs1_bytes          => l_fs1_bytes,
29      fs1_blocks         => l_fs1_blocks,
30      fs2_bytes          => l_fs2_bytes,
31      fs2_blocks         => l_fs2_blocks,
32      fs3_bytes          => l_fs3_bytes,
33      fs3_blocks         => l_fs3_blocks,
34      fs4_bytes          => l_fs4_bytes,
35      fs4_blocks         => l_fs4_blocks,
36      full_bytes         => l_full_bytes,
37      full_blocks        => l_full_blocks,
38      unformatted_blocks => l_unformatted_blocks,
39      unformatted_bytes  => l_unformatted_bytes
40  );
41  DBMS_OUTPUT.ENABLE;
42  DBMS_OUTPUT.PUT_LINE(' FS1 Blocks = '||l_fs1_blocks||
Bytes = '||l_fs1_bytes);
43  DBMS_OUTPUT.PUT_LINE(' FS2 Blocks = '||l_fs2_blocks||
Bytes = '||l_fs2_bytes);
44  DBMS_OUTPUT.PUT_LINE(' FS3 Blocks = '||l_fs3_blocks||
Bytes = '||l_fs3_bytes);
45  DBMS_OUTPUT.PUT_LINE(' FS4 Blocks = '||l_fs4_blocks||
Bytes = '||l_fs4_bytes);
46  DBMS_OUTPUT.PUT_LINE('Full Blocks =
'||l_full_blocks||' Bytes = '||l_full_bytes);
47  DBMS_OUTPUT.PUT_LINE('Unformatted Blocks =
'||l_unformatted_blocks||' Bytes = '||l_unformatted_bytes);
48
DBMS_OUTPUT.PUT_LINE('=====
=====');
49  DBMS_OUTPUT.PUT_LINE('Total Blocks =
'||to_char(l_fs1_blocks + l_fs2_blocks +
50      l_fs3_blocks + l_fs4_blocks
+ l_full_blocks)|| ' || Total Bytes = '||
51      to_char(l_fs1_bytes +
l_fs2_bytes +
52      l_fs3_bytes + l_fs4_bytes +
l_full_bytes));
53  END;
54  /

Procedure created.

SQL>
SQL> show errors
No errors.
SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```


练习 7-1：使用 SecureFiles（续）

- 12) 按 [Enter] 继续执行脚本的下一步骤。创建了一个名为 CHECK_SPACE_SF 的过程，用于显示分配给 RESUMES 表中 SecureFiles LOB 段的磁盘块的空间使用情况。此过程将调用 DBMS_SPACE 程序包。调用 CHECK_SPACE_SF 中的 DBMS_SPACE.SPACE_USAGE 过程仅对 SecureFiles LOB 有效。

```
SQL>
SQL> /*== Create a procedure for checking SecureFiles LOB
space usage. ==*/
SQL> /*==
==*/
SQL> /*== Note that the DBMS_SPACE.SPACE_USAGE procedure has
been ==*/
SQL> /*== overloaded. There are 2 DBMS_SPACE.SPACE_USAGE
procedures. ==*/
SQL> /*== The two cannot be used interchangeably. The
invocation of ==*/
SQL> /*== the DBMS_SPACE.SPACE_USAGE procedure here is ONLY
valid ==*/
SQL> /*== for use against SecureFiles LOBs.
==*/
SQL>
SQL> CREATE OR REPLACE PROCEDURE check_space_sf
2  IS
3      l_segment_size_blocks NUMBER;
4      l_segment_size_bytes NUMBER;
5      l_used_blocks NUMBER;
6      l_used_bytes NUMBER;
7      l_expired_blocks NUMBER;
8      l_expired_bytes NUMBER;
9      l_unexpired_blocks NUMBER;
10     l_unexpired_bytes NUMBER;
11     v_segname varchar2(30);
12 BEGIN
13     SELECT segment_name
14     INTO v_segname
15     FROM user_lobs
16     WHERE table_name = 'RESUMES'
17     AND column_name = 'RESUME';
18     DBMS_SPACE.SPACE_USAGE(
19         segment_owner      => 'SF_DEMO',
20         segment_name       => v_segname,
21         segment_type       => 'LOB',
22         segment_size_blocks => l_segment_size_blocks,
23         segment_size_bytes => l_segment_size_bytes,
24         used_blocks        => l_used_blocks,
25         used_bytes         => l_used_bytes,
26         expired_blocks     => l_expired_blocks,
27         expired_bytes      => l_expired_bytes,
28         unexpired_blocks   => l_unexpired_blocks,
29         unexpired_bytes    => l_unexpired_bytes
```

练习 7-1：使用 SecureFiles（续）

```

30      );
31      DBMS_OUTPUT.ENABLE;
32      DBMS_OUTPUT.PUT_LINE(' Segment Blocks =
'||l_segment_size_blocks||' Bytes = '||l_segment_size_bytes);
33      DBMS_OUTPUT.PUT_LINE(' Used Blocks =
'||l_used_blocks||' Bytes = '||l_used_bytes);
34      DBMS_OUTPUT.PUT_LINE(' Expired Blocks =
'||l_expired_blocks||' Bytes = '||l_expired_bytes);
35      DBMS_OUTPUT.PUT_LINE(' Unexpired Blocks =
'||l_unexpired_blocks||' Bytes = '||l_unexpired_bytes);
36
DBMS_OUTPUT.PUT_LINE('=====
=====');
37  END;
38  /

Procedure created.

SQL> show errors
No errors.
SQL>

```

13) 对 **BasicFiles LOB** 存储性能进行取样。要捕获 BasicFiles LOB 存储的写入和读取性能统计信息，请执行以下步骤：

- a) 使用 SQL*Plus 会话，执行 write_data.sql 脚本。此脚本将 90 条记录插入 RESUMES 表中。每条记录都包括一个大小介于 35 KB 至 65 KB 之间的 LOB。将启动 SQL*Plus 计时器以捕获插入的总占用时间。
- b) 脚本完成后，记下以 BasicFiles 格式加载 LOB 的占用时间。**注：**由于硬件配置（如 CPU 速度或内存量）不同，您观察到的占用时间可能会不同。

```

SQL>
SQL> @write_data
SQL> connect sf_demo/oracle_4U
Connected.
SQL>
SQL> set serveroutput on
SQL> set verify on
SQL> set term on
SQL> set lines 200
SQL>
SQL> timing start load_data
SQL> exec write_lob('karl.brimmer.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

```

练习 7-1：使用 SecureFiles（续）

```

PL/SQL procedure successfully completed.

SQL> exec write_lob('monica.petera.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

PL/SQL procedure successfully completed.

SQL> exec write_lob('david.sloan.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

PL/SQL procedure successfully completed.

SQL> timing stop
timing for: load_data
Elapsed: 00:00:00.94
SQL>

```

14) 现在将检索在上一步骤中插入的记录。

- a) 执行 read_data.sql 脚本。此脚本读取 RESUMES 表中的 90 条记录。对于每条记录，屏幕上将显示 LOB 值的大小以及 LOB 的前 200 个字符。将启动 SQL*Plus 计时器以捕获检索的总占用时间。
- b) 脚本完成后，记下以 BasicFiles 格式读取 LOB 的占用时间。**注：**由于硬件配置（如 CPU 速度或内存量）不同，您观察到的占用时间可能会不同。

```

SQL>
SQL> @read_data
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> set serveroutput on
SQL> set verify on
SQL> set term on
SQL> set lines 200
SQL>
SQL> timing start read_data
SQL> exec read_lob;
The length is: 64000
The ID is: 1

```

练习 7-1：使用 SecureFiles（续）

```
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 2
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 3
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 4
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 5
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 6
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 7
The blob is read: ???          > ??          x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 8
The blob is read: ???          > ??          x
z      ???      w
...
```

练习 7-1：使用 SecureFiles（续）

```

The length is: 37376
The ID is: 30
The blob is read: ???          > ??          D
F          ???          C
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 37376
The ID is: 30
The blob is read: ???          > ??          D
F          ???          C
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????

PL/SQL procedure successfully completed.

SQL> timing stop
timing for: read_data
Elapsed: 00:00:00.56
SQL>

```

15) 通过执行以下步骤可确认 BasicFiles LOB 的空间使用情况：

- 使用 SQL*Plus，执行 space_usage.sql 脚本。
- 系统提示输入 LOB 类型时，输入 BF。脚本完成后，记下完整块的数目以及完整块字节。

```

SQL>
SQL> @space_usage
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> set serveroutput on
SQL> set pause off
SQL> set echo on
SQL>
SQL> exec check_space_&Enter_BF_or_SF_for_LOB_type
Enter value for enter_bf_or_sf_for_lob_type: BF
FS1 Blocks = 0 Bytes = 0
FS2 Blocks = 0 Bytes = 0
FS3 Blocks = 0 Bytes = 0
FS4 Blocks = 0 Bytes = 0
Full Blocks = 540 Bytes = 4423680
Unformatted Blocks = 7618 Bytes = 62406656
=====
Total Blocks = 540 || Total Bytes = 4423680

PL/SQL procedure successfully completed.

SQL>

```

练习 7-1: 使用 SecureFiles (续)

16) 要将 BasicFiles 存储迁移到 SecureFiles 存储, 请执行以下步骤:

- a) 在 SQL*Plus 会话中, 执行 mig_bf_to_sf.sql 脚本。该脚本首先查询 DBA_SEGMENTS 视图, 以查看 BasicFiles LOB 存储的 LOB 段子类型名称。
- b) 请注意它显示为 “ASSM”。

```
SQL> @mig_bf_to_sf
SQL> -- run this script as SYS
SQL> connect / as sysdba
Connected.
SQL>
SQL> set echo on;
SQL> set serveroutput on
SQL> set term on;
SQL> set lines 200
SQL>
SQL> set pause off
SQL>
SQL> /*== Check the current segment type of the LOB ==*/
SQL> /*==                                     ==*/
SQL> /*== Note that the segment subtype for      ==*/
SQL> /*== BasicFile LOB storage is ASSM          ==*/
SQL>
SQL> SELECT segment_name, segment_type, segment_subtype
2  FROM dba_segments
3  WHERE tablespace_name = 'SECF_TBS1'
4  AND segment_type = 'LOBSEGMENT'
5  /

SEGMENT_NAME
SEGMENT_TYPE          SEGMENT_SU
-----
SYS_LOB0000074197C00004$$
LOBSEGMENT            ASSM

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

练习 7-1：使用 SecureFiles（续）

- 17) 按 [Enter] 继续执行脚本的下一步骤。从 BasicFiles 到 SecureFiles LOB 存储格式的移植过程是联机执行的。这意味着在移植期间可继续访问 RESUMES 表。此类型的操作称为“联机重新定义”。联机重新定义要求使用临时表进行数据存储。在脚本的这个步骤中，使用 SecureFiles LOB 存储格式定义临时表，且将 LOB 存储在 SECF_TBS2 表空间中。完成移植后，RESUME LOB 存储在 SECF_TBS2 表空间中。

注：建议做法是在创建表时启用压缩和取消重复。另外，此版本的 Oracle DB 中的 SecureFiles 尚不支持 ALTER TABLE SHRINK 操作。

```
SQL>
SQL> set pause on
SQL>
SQL> /*== Create the interim table for online redefinition
==*/
SQL> /*==
SQL> /*== Note that the recommended practice is to enable
==*/
SQL> /*== COMPRESSION and DEDUPLICATION at table creation ==*/
SQL> /*== time. Also, the ALTER TABLE SHRINK operation is
==*/
SQL> /*== not yet supported for SecureFiles LOBs in
==*/
SQL> /*== release 1 of Oracle Database 11g.
==*/
SQL>
SQL> CREATE TABLE sf_demo.resumes_interim
2  (id NUMBER, first_name VARCHAR2(15),
3  last_name VARCHAR2 (40), resume BLOB)
4  LOB (resume) STORE AS SECUREFILE
5  (TABLESPACE secf_tbs2
6   COMPRESS HIGH
7   DEDUPLICATE)
8  /

Table created.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

练习 7-1: 使用 SecureFiles (续)

- 18) 按 [Enter] 继续执行脚本的下一步骤。此脚本中的下一个步骤调用 DBMS_REDEFINITION 程序包，以执行联机重新定义操作。

```
SQL>
SQL> /*== Perform the online redefinition ==*/
SQL>
SQL> DECLARE
  2   error_count PLS_INTEGER := 0;
  3   BEGIN
  4
  5   DBMS_REDEFINITION.START_REDEF_TABLE
  6   ('sf_demo', 'resumes', 'resumes_interim',
  7    'id id, first_name first_name, last_name last_name,
resume resume',
  8    OPTIONS_FLAG => DBMS_REDEFINITION.CONST_USE_ROWID);
  9
 10  DBMS_REDEFINITION.COPY_TABLE_DEPENDENTS
 11  ('sf_demo', 'resumes', 'resumes_interim',
 12   1, true,true,true,false, error_count);
 13
 14  DBMS_OUTPUT.PUT_LINE('Errors := ' ||
TO_CHAR(error_count));
 15
 16  DBMS_REDEFINITION.FINISH_REDEF_TABLE
 17  ('sf_demo', 'resumes', 'resumes_interim');
 18
 19  END;
 20  /
Errors := 0

PL/SQL procedure successfully completed.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 19) 按 [Enter] 继续执行脚本的下一步骤。联机重新定义操作完成后，可以删除该临时表。

```
SQL>
SQL> /*== Drop the interim table ==*/
SQL>
SQL> DROP TABLE sf_demo.resumes_interim
  2   /

Table dropped.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```


练习 7-1：使用 SecureFiles（续）

20) 按 [Enter] 继续执行脚本的下一步骤。移植的 LOB 数据现在位于 SECF_TBS2 表空间中。脚本的这一步骤将查询 DBA_SEGMENTS 视图，以查看 SecureFiles 存储的 LOB 段子类型名称。请注意它显示为 “SECUREFILE”。

```
SQL>
SQL> set pause off
SQL>
SQL> /*== Check the segment type of the migrated LOB ==*/
SQL> /*==                                     ==*/
SQL> /*== Note that the segment subtype for      ==*/
SQL> /*== SecureFiles LOB storage is SECUREFILE ==*/
SQL>
SQL> SELECT segment_name, segment_type, segment_subtype
2  FROM dba_segments
3  WHERE tablespace_name = 'SECF_TBS2'
4  AND segment_type = 'LOBSEGMENT'
5  /

SEGMENT_NAME
SEGMENT_TYPE          SEGMENT_SU
-----
SYS_LOB0000074206C00004$$
LOBSEGMENT            SECUREFILE

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

21) 此脚本的最后一步会查询 DBA_LOBS 视图，以查看 SecureFiles 段的压缩和取消重复设置。

```
SQL>
SQL> set pause off
SQL>
SQL> /*== Check the DBA_LOBS data dictionary view for ==*/
SQL> /*== information on the SecureFiles COMPRESSION ==*/
SQL> /*== and DEDUPLICATION settings                ==*/
SQL>
SQL> SELECT column_name, segment_name,
2  compression, deduplication, securefile
3  FROM dba_lobs
4  WHERE owner = 'SF_DEMO' and table_name = 'RESUMES'
5  /

COLUMN_NAME
-----
```

练习 7-1：使用 SecureFiles（续）

SEGMENT_NAME	COMPRE	DEDUPLICATION	SEC
-----	-----	-----	---
RESUME			
SYS_LOB0000074206C00004\$\$	HIGH	LOB	YES
SQL>			

22) 确认 SecureFiles 的空间使用情况。

要确认 SecureFiles 的空间使用情况，请执行以下步骤：

- 使用 SQL*Plus，执行 space_usage.sql 脚本。
- 系统提示输入 LOB 类型时，请输入 **SF**。脚本完成后，记录已使用块的数目和已使用块的字节。
- 将 BasicFiles 的完整块数和步骤 15b 中的字节与 SecureFiles 已使用的块数和字节进行对比。请注意，启用压缩和取消重复的 SecureFiles 空间使用量已显著减少。

```

SQL>
SQL> @space_usage
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> set serveroutput on
SQL> set pause off
SQL> set echo on
SQL>
SQL> exec check_space_&Enter_BF_or_SF_for_LOB_type
Enter value for enter_bf_or_sf_for_lob_type: SF
Segment Blocks = 8192 Bytes = 67108864
Used Blocks = 6 Bytes = 49152
Expired Blocks = 8162 Bytes = 66863104
Unexpired Blocks = 0 Bytes = 0
=====

PL/SQL procedure successfully completed.

SQL>

```

练习 7-1：使用 SecureFiles（续）

23) 对 SecureFiles 存储的性能进行采样。

要捕获 SecureFiles LOB 存储的读取和写入性能统计信息，请执行以下步骤。与前一个有关 BasicFiles 存储示例性能的部分有所不同，首先对 SecureFiles 存储的读取性能进行测试。这是为了确保对 BasicFiles 和 SecureFiles 进行有效的读取性能比较 — 从 BasicFiles 移植到 SecureFiles 后，需要立即对相同的记录集执行读取测试。

- a) 执行 read_data.sql 脚本。**注：**由于硬件配置（如 CPU 速度或内存量）不同，您观察到的占用时间可能会不同。

```
SQL> @read_data
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> set serveroutput on
SQL> set verify on
SQL> set term on
SQL> set lines 200
SQL>
SQL> timing start read_data
SQL> exec read_lob;
The length is: 64000
The ID is: 1
The blob is read: ???          >  ??                                x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 2
The blob is read: ???          >  ??                                x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 3
The blob is read: ???          >  ??                                x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 4
The blob is read: ???          >  ??                                x
z      ???      w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
```

练习 7-1：使用 SecureFiles（续）

```

The length is: 64000
The ID is: 5
The blob is read: ???          >  ??          x
z          ????          w
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 64000
The ID is: 6
The blob is read: ???          >  ??          x
z          ????          w
...
The length is: 37376
The ID is: 29
The blob is read: ???          >  ??          D
F          ????          C
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 37376
The ID is: 30
The blob is read: ???          >  ??          D
F          ????          C
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????
The length is: 37376
The ID is: 30
The blob is read: ???          >  ??          D
F          ????          C
????????????????????????????????????????????????????????????
????????????????????????????????????????????????????????????

PL/SQL procedure successfully completed.

SQL> timing stop
timing for: read_data
Elapsed: 00:00:00.61
SQL>

```

24) 现在插入其它记录。

- 执行 `write_data.sql` 脚本。此脚本将 90 条记录插入 `RESUMES` 表中。每条记录都包括一个大小介于 35 KB 至 65 KB 之间的 LOB。将启动 SQL*Plus 计时器以捕获插入的总占用时间。
- 脚本完成后，记下以 SecureFiles 格式写入 LOB 的占用时间。**注：**由于硬件配置（如 CPU 速度或内存量）不同，您观察到的占用时间可能会不同。

练习 7-1: 使用 SecureFiles (续)

```
SQL>
SQL> @write_data
SQL> connect sf_demo/oracle_4US
Connected.
SQL>
SQL> set serveroutput on
SQL> set verify on
SQL> set term on
SQL> set lines 200
SQL>
SQL> timing start load_data
SQL> exec write_lob('karl.brimmer.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

PL/SQL procedure successfully completed.

SQL> exec write_lob('monica.petera.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

PL/SQL procedure successfully completed.

SQL> exec write_lob('david.sloan.doc');
Begin inserting rows...
Row 1 inserted.
Row 2 inserted.
Row 3 inserted.
...
Row 30 inserted.

PL/SQL procedure successfully completed.

SQL> timing stop
timing for: load_data
Elapsed: 00:00:00.45
SQL>
SQL>
SQL>
```

练习 7-1: 使用 SecureFiles (续)

25) 启用加密。

要对以 SecureFiles 格式存储的 LOB 进行加密，请执行以下步骤：

- a) 使用 SQL*Plus，以 sf_demo/oracle_4U 用户身份连接到数据库。执行 encrypt_lob.sql 脚本。

```
SQL>
SQL> @encrypt_lob
SQL> set serveroutput on
SQL> set verify on
SQL> set term on
SQL> set lines 200
SQL> set pause off
SQL> set echo on
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> ALTER TABLE sf_demo.resumes
  2  MODIFY (resume ENCRYPT USING 'AES192')
  3  /

Table altered.

SQL>
```

26) 确认 LOB 现在已加密。执行 check_encrypt.sql 脚本。

注：列加密操作实质上是更新所有现有 LOB 值。然而，如果数据块中有可用空间，则原始未加密格式的 LOB 值可能仍然存在。为了确保磁盘上不存在未加密的值，最好在创建表时（即在插入任何值之前）启用加密。

```
SQL> @check_encrypt
SQL> connect sf_demo/oracle_4U
Connected.
SQL> set pause off
SQL> set echo on
SQL>
SQL> col TABLE_NAME format a18;
SQL> col COLUMN_NAME format a19;
SQL> col ENCRYPTION_ALG format a17;
SQL>
SQL> SELECT *
  2  FROM USER_ENCRYPTED_COLUMNS
  3  /
```

练习 7-1：使用 SecureFiles（续）

TABLE_NAME	COLUMN_NAME	ENCRYPTION_ALG	SAL
INTEGRITY_AL			

RESUMES	RESUME	AES 192 bits key	YES
SHA-1			
SQL>			

27) 要清除为此练习创建的数据库帐户和对象，请以 SYSDBA 用户身份连接，然后执行 cleanup.sql 脚本。退出 SQL*Plus。

```
SQL>
SQL> connect / as sysdba
Connected.
SQL> @cleanup
SQL> drop user sf_demo cascade
      2 /

User dropped.

SQL> drop tablespace secf_tbs1 including contents and
datafiles
      2 /

Tablespace dropped.

SQL> drop tablespace secf_tbs2 including contents and
datafiles
      2 /

Tablespace dropped.
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```

第 8 课的练习

在本练习中，您将应用几个安全性功能。

假设：监听程序已启动，实例名称为 `orcl`，SYS 口令为 `oracle_4U`，并且数据库实例已启动。

练习 8-1：使用口令安全功能

您已在 10.2 和 11.1 数据库的混合环境中安装了 Oracle 11g 数据库。使用 Enterprise Manager 中的“Evaluation Results: Secure Configuration for Oracle databases（评估结果：Oracle 数据库的安全配置）”页检查遵从性。根据遵从性级别，您决定通过强制授权用户的口令区分大小写来提高口令的安全性。由于没有检查口令复杂性，因此您可实施 `verify_function_11g` 口令复杂性检查。开始本练习之前，请以 `sys` 用户身份在 `SQL*Plus` 会话中进行连接，执行以下位于 `/home/oracle/solutions/SEC` 目录中的脚本：**`setup_sec.sql`**。

```
$ cd /home/oracle/solutions/SEC
$
$ . oraenv
ORACLE_SID = [db10g] ? orcl
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:30:37
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @setup_sec
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> ALTER USER HR identified by hr account unlock;

User altered.

SQL>
SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

练习 8-1：使用口令安全功能（续）

- 1) 使用 `emctl start dbconsole` 命令启动 Enterprise Manager Database Control（如果尚未启动）。提示：使用 `emctl status dbconsole` 检查进程状态。

```
$ emctl status dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://...:5500/em/console/aboutApplication
Oracle Enterprise Manager 11g is running.
-----
Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/ed.us.oracle.com_orcl/
sysman/log
$
```

- 2) 启动 Enterprise Manager Database Control 并以 SYS 用户身份登录。查看“Security Violations（安全违规）”和“Policy Trend Overview（策略趋势概览）”并记下配置级别。
 - a) 在“Database（数据库）”主页的“Policy Violation（策略违规）”部分上，策略级别将显示以下内容：

```
Policy Violations

All 7 Critical Rules Violated 7 Critical Security Patches 0
Compliance Score (%) 95
```

- b) 单击遵从性级别(%)的编号。您将进入“Policy Trend Overview（策略趋势概览）”页，在该页中可以看到对策略遵从性的总体可视化表示。
- 3) 查找违反的不同规则：
 - a) 在“Critical Severity（关键严重性）”行中，单击“Distinct Rules Violated（违反的不同规则）”列中的编号。
 - b) 您应看到有几个违规与 PUBLIC 对 PL/SQL 程序包有 EXECUTE 权限有关，还有一个与概要文件允许的失败登录次数过多有关。
- 4) 查找规则，确定登录失败尝试的值的允许范围。
 - a) 单击“Policy Rule（策略规则）”列中的“Profiles With Excessive Allowed Failed Login Attempts（允许的失败登录次数过多的概要文件）”。
 - b) 在此规则的“View Policy Rule Settings（查看策略规则设置）”页上，可以看到此规则的默认设置为 10。

练习 8-1：使用口令安全功能（续）

- 5) 可确定每个站点的策略为：五次登录尝试失败将导致帐户被锁定五分钟。检查“Password Profile Settings（口令概要文件设置）”并更改 DEFAULT 概要文件的设置以便与站点策略相匹配。
 - a) 返回到“Database（数据库）”主页。在“Database（数据库）”主页上，单击“Server（服务器）”选项卡，然后单击“Security（安全性）”部分中的“Profiles（概要文件）”。
 - b) 在“Profiles（概要文件）”页上，选择 DEFAULT 概要文件并单击“Edit（编辑）”。
 - c) 在“Edit Profile（编辑概要文件）”页上单击“Password（口令）”选项卡。
 - d) 在“Failed Login（失败登录）”部分中，将“Number of failed login attempts to lock after（锁定前允许的最大失败登录次数）”更改为 5 次，“Number of days to lock for（锁定天数）”更改为 0.0035。该值代表一天中的 5 分钟 (5/1440)。
 - e) 单击“Apply（应用）”。此时将显示更新消息。
- 6) 为数据库用户确认实例的口令区分大小写设置。查看 SEC_CASE_SENSITIVE_LOGIN 参数值。以 HR 用户身份使用 HR 作为口令连接到 orcl 实例。尝试使用以下用户名/口令组合进行连接：HR/hr、hr/hr 和 hr/HR。
 - a) 以 sysdba 用户身份连接 SQL*Plus，检查 SEC_CASE_SENSITIVE_LOGON 参数：

```

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:43:56
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> show parameter sec_case_sensitive_logon

NAME                                                    TYPE        VALUE
-----
sec_case_sensitive_logon                               boolean     TRUE
SQL> exit

```

练习 8-1：使用口令安全功能（续）

```
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- b) 使用 /nolog 选项启动 SQL*Plus。尝试以各种不同的用户名和口令组合进行连接。观察哪个组合允许连接。

注：请确保使用有效的连接字符串，如果五次尝试登录都失败，帐户会被锁定五分钟。

```
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:45:46
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> connect HR/HR
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> connect HR/hr
Connected.
SQL> connect hr/hr
Connected.
SQL> connect hr/HR
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL> exit
$
```

- 7) 由于口令文件可能位于升级的数据库中，因此应使其不区分大小写。使用以下命令：
- ```
orapwd file=$ORACLE_HOME/dbs/orapworcl entries=5
password=oracle_4U ignorecase=Y force=Y
```

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ orapwd file=$ORACLE_HOME/dbs/orapworcl entries=5
password=oracle_4U ignorecase=Y force=Y
$
```

## 练习 8-1：使用口令安全功能（续）

8) 为授权用户确认实例的口令区分大小写设置。尝试以 SYSDBA 用户身份，使用网络服务名和以下口令连接到 orcl 实例：

- 小写口令
- 大写口令
- 大小写混合的口令

网络服务名强制使用口令文件。哪种连接可以成功？

启动 SQL\*Plus 并尝试使用各种不同的口令组合进行连接。

```
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:48:25
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.
SQL> connect sys/oracle_4U@orcl AS SYSDBA
Connected.
SQL> connect sys/ORACLE_4U@orcl AS SYSDBA
Connected.
SQL> connect sys/OraCle_4u@orcl AS SYSDBA
Connected.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

9) 使口令文件区分大小写，以便与数据库口令用法匹配。使用以下命令：

```
orapwd file=$ORACLE_HOME/dbs/orapworcl entries=5
password=oracle_4U ignorecase=N force=Y
```

```
$ orapwd file=$ORACLE_HOME/dbs/orapworcl entries=5
password=oracle_4U ignorecase=N force=Y
$
```

10) 尝试以 SYSDBA 用户身份，使用网络服务名和以下口令连接到 orcl 实例：

- 小写口令
- 大写口令
- 大小写混合的口令

## 练习 8-1：使用口令安全功能（续）

网络服务名强制使用口令文件。哪种连接可以成功？

启动 SQL\*Plus 并尝试使用各种不同的口令组合进行连接。

```
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:51:01
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> connect sys/oracle_4U@orcl AS SYSDBA
Connected.
SQL> connect sys/ORACLE_4U@orcl AS SYSDBA
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL> connect sys/OraCle_4u@orcl AS SYSDBA
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> exit
$
```

- 11) 安装 Oracle Database 11g 软件提供的口令复杂性检查函数。请勿更改 DEFAULT 概要文件。utlpwdmg.sql 脚本将安装口令复杂性检查函数并更改 DEFAULT 概要文件。使用 cr\_verification\_func.sql 脚本。此脚本包含对已注释掉的 DEFAULT 概要文件的更改。该脚本将创建 verify\_function 和 verify\_function\_11g。  
从 \$HOME/solutions/SEC 目录执行 cr\_verification\_func.sql。

```
$ pwd
/home/oracle/solutions/SEC
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:53:59
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @cr_verification_func

Function created.

Function created.
```

## 练习 8-1：使用口令安全功能（续）

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 12) 创建 DEFAULT 概要文件。除了口令复杂性检查以外，按照 DEFAULT 概要文件设置所有内容。在 DBUSER 概要文件中，使用提供的名为 VERIFY\_FUNCTION\_11G 的验证函数。在 DBUSER 概要文件中查看口令限制设置。**提示：**创建新的概要文件时，未显式设置的所有值都将继承 DEFAULT 概要文件中的值。使用 \$HOME/solutions/SEC 目录下的 cr\_dbuser\_profile.sql 脚本。

```
$ pwd
/home/oracle/solutions/SEC
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 11:56:27
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @cr_dbuser_profile
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> -- create a DBUSER profile
SQL> CREATE PROFILE DBUSER
2 LIMIT
3 PASSWORD_VERIFY_FUNCTION verify_function_11g;

Profile created.

SQL>
SQL> -- View the settings of the DBUSER profile
SQL>
SQL> SELECT RESOURCE_NAME, LIMIT
2 FROM DBA_PROFILES
3 WHERE PROFILE = 'DBUSER'
4 AND RESOURCE_TYPE = 'PASSWORD';
```

## 练习 8-1：使用口令安全功能（续）

| RESOURCE_NAME            | LIMIT               |
|--------------------------|---------------------|
| -----                    | -----               |
| FAILED_LOGIN_ATTEMPTS    | DEFAULT             |
| PASSWORD_LIFE_TIME       | DEFAULT             |
| PASSWORD_REUSE_TIME      | DEFAULT             |
| PASSWORD_REUSE_MAX       | DEFAULT             |
| PASSWORD_VERIFY_FUNCTION | VERIFY_FUNCTION_11G |
| PASSWORD_LOCK_TIME       | DEFAULT             |
| PASSWORD_GRACE_TIME      | DEFAULT             |
| 7 rows selected.         |                     |
| SQL>                     |                     |

- 13) 创建使用简单口令 oracle 的名为 FRED 的用户。向 FRED 授予 CREATE SESSION 权限。以 SYS 用户身份连接，创建使用口令 oracle 的 FRED 用户。

```
SQL>
SQL> CREATE USER fred IDENTIFIED BY oracle;

User created.

SQL> GRANT CREATE SESSION TO FRED IDENTIFIED BY oracle;

Grant succeeded.

SQL>
```

- 14) 口令是否区分大小写？通过以 FRED 用户身份进行连接来进行测试。

```
SQL> connect fred/ORACLE
ERROR:
ORA-01017: invalid username/password; logon denied

Warning: You are no longer connected to ORACLE.
SQL> connect FRED/ORACLE
ERROR:
ORA-01017: invalid username/password; logon denied

SQL> connect fred/oracle
Connected.
SQL>
```



## 练习 8-1：使用口令安全功能（续）

- 15) 将 FRED 用户的概要文件设置为 DBUSER。

以 SYS 用户身份将 DBUSER 概要文件分配给 FRED 用户。

```
SQL> CONNECT / as sysdba
Connected.
SQL> ALTER USER fred PROFILE dbuser;

User altered.

SQL>
```

- 16) 以 FRED 用户身份连接并尝试将口令更改为 welcome4，而后再更改为 welcome1。会发生什么？

以 FRED 用户身份连接。使用 ALTER USER 命令更改口令。

```
SQL> connect fred/oracle
Connected.
SQL> ALTER USER fred IDENTIFIED BY welcome4 REPLACE oracle;

User altered.

SQL> ALTER USER fred IDENTIFIED BY welcome1 REPLACE welcome4;
ALTER USER fred IDENTIFIED BY welcome1 REPLACE welcome4
*
ERROR at line 1:
ORA-28003: password verification for the specified password
failed
ORA-20006: Password too simple

SQL>
```

- 17) 使用至少包含四个字符的口令，四个字符中必须有一个为字母，一个为数字。

以 FRED 用户身份将口令更改为所选的值。会发生什么情况？

再次使用 ALTER USER 命令将口令更改为更复杂的值。建议的口令是 Ft54red#。

```
SQL> ALTER USER fred IDENTIFIED BY Ft54red# REPLACE welcome4;

User altered.

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 8-1：使用口令安全功能（续）

- 18) 更改 “Security Profile Rules（安全概要文件规则）” 以减少严重违规次数。与 PL/SQL 程序包相关的网络需要使用 ACL 才能访问任何网络计算机。由于这些 ACL（如果存在）指定可以访问的用户和计算机，因此不存在允许这些程序包通过 PUBLIC 执行权限的安全性违规。
- a) 在 Enterprise Manager 中，单击数据库主页的 “Related Links（相关链接）” 部分中的相应链接导航至 “Metrics and Policy Settings（度量和策略设置）” 页。
  - b) 在 “Metrics and Policies（度量和策略）” 页上，单击 “Policies（策略）” 选项卡，然后查找以 “Restricted Privilege to Execute（执行 x 的权限受到限制）” 开始的策略。这些策略位于 76-100 行中。
  - c) 选择 “Restricted Privilege to Execute UTL\_HTTP（执行 UTL\_HTTP 的权限受到限制）”，然后单击右侧的 “Edit（编辑）” 图标。
  - d) 在 “Edit Policy Rule Settings: Restricted Privilege to Execute UTL\_HTTP（编辑策略规则设置：执行 UTL\_HTTP 的权限受到限制）” 页上，将 “Policy Evaluation（策略评估）” 更改为 “Disabled（禁用）”。然后单击 “Continue（继续）”。
  - e) 对于 UTL\_SMTP 和 UTL\_TCP 重复执行此操作。在 “Metrics and Policies（度量和策略）” 页上，单击 “OK（确定）”。
  - f) 在 “Confirmation（确认）” 页上，单击 “OK（确定）”。
- 19) 单击 “Database（数据库）”，返回到 “Database（数据库）” 主页。

## 练习 8-2：使用加密的表空间

如果您尚未完成练习 7-1，请确保在本练习前执行以下设置步骤。在 `sqlnet.ora` 文件中添加 `ENCRYPTION_WALLET_LOCATION` 条目：

```
$ mkdir /u01/app/oracle/admin/orcl/wallet
$ cd /u01/app/oracle/product/11.2.0/grid/network/admin
$ vi sqlnet.ora
$ cat
/u01/app/oracle/product/11.2.0/grid/network/admin/sqlnet.ora
sqlnet.ora Network Configuration File:
/u01/app/oracle/product/11.2.0/grid/network/admin/sqlnet.ora
Generated by Oracle configuration tools.

NAMES.DIRECTORY_PATH= (TNSNAMES, EZCONNECT)

ADR_BASE = /u01/app/oracle

ENCRYPTION_WALLET_LOCATION= (SOURCE=(METHOD=FILE) (METHOD_DATA=
(DIRECTORY=/u01/app/oracle/admin/orcl/wallet)))

$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 12:09:47
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>
SQL> alter system set wallet open identified by "welcome1";
alter system set wallet open identified by "welcome1"
*
ERROR at line 1:
ORA-28367: wallet does not exist

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 8-2：使用加密的表空间（续）

ORDERS 和 CUSTOMERS 表的 CUSTOMER\_ID 列包含敏感数据。为了使应用程序正常运行，该列必须编入索引并且 column\_id 必须是外键列。执行测试，检查更改的访问路径。通过将这两个表和 CUSTOMER\_ID 列中的索引添加到加密的表空间中来实现表空间加密。

- 1) 创建数据库主加密密钥。使用命令

```
ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "welcome1";
```

以 SYSDBA 用户身份连接 orcl 数据库，并发出 SET ENCRYPTION KEY 命令：

```
$ cd /home/oracle/solutions/SEC
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 1 12:15:39
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> ALTER SYSTEM SET ENCRYPTION KEY IDENTIFIED BY "welcome1";

System altered.

SQL>
```

- 2) 创建名为 ENCTS 的加密表空间，该加密的表空间使用默认加密算法，大小为 10 MB。请使用以下命令：

```
CREATE TABLESPACE encts
DATAFILE '+DATA' SIZE 10M
ENCRYPTION
DEFAULT STORAGE (ENCRYPT);
```

```
SQL> CREATE TABLESPACE encts
2 DATAFILE '+DATA' SIZE 10M
3 ENCRYPTION
4 DEFAULT STORAGE (ENCRYPT);

Tablespace created.

SQL>
```

## 练习 8-2：使用加密的表空间（续）

- 3) 索引是否适用于加密的表空间？以 OE 用户身份进行测试。  
query\_orders\_index.sql 文件中的查询使用名为 ORD\_CUSTOMER\_IX 的外键索引。在未加密的表空间中查看查询以验证是否正在使用外键索引。将已访问过的对象移动到加密的表空间中。确认执行计划未发生改变。
- a) 以 SYSDBA 用户身份连接到 SQL\*Plus。执行  
\$ORACLE\_HOME/sqlplus/admin 目录中的 plustrce.sql 脚本，创建 PLUSTRACE 角色。授予 OE 用户 PLUSTRACE 角色。解除 OE 帐户锁定并将口令更改为 OE。

```
SQL> connect / as sysdba
Connected.
SQL> @?/sqlplus/admin/plustrce
SQL>
SQL> drop role plustrace;
drop role plustrace
 *
ERROR at line 1:
ORA-01919: role 'PLUSTRACE' does not exist

SQL> create role plustrace;

Role created.

SQL> grant select on v_$sesstat to plustrace;

Grant succeeded.

SQL> grant select on v_$statname to plustrace;

Grant succeeded.

SQL> grant select on v_$mystat to plustrace;

Grant succeeded.

SQL> grant plustrace to dba with admin option;

Grant succeeded.

SQL> set echo off
SQL> GRANT PLUSTRACE to OE;

Grant succeeded.

SQL> ALTER USER oe IDENTIFIED BY oe ACCOUNT UNLOCK;

User altered.

SQL>
```

## 练习 8-2：使用加密的表空间（续）

- b) 在启用了 autotrace 的情况下，以 oe 用户身份执行  
/home/oracle/solutions/SEC 中的 query\_orders\_index.sql  
查询脚本并检查执行计划。请注意，正在访问 ORD\_CUSTOMER\_IX 索引。  
oe 用户所用的口令是 oe。

```
SQL>
SQL> connect oe/oe
Connected.
SQL> set autotrace traceonly explain
SQL> set echo on
SQL> @query_orders_index
SQL> select cust_last_name, order_id, order_total
 2 from customers, orders
 3 where customers.customer_id = orders.customer_id
 4 and account_mgr_id = 145;
```

Execution Plan

-----

Plan hash value: 3558567882

-----

| Id    | Operation                   | Name            | Rows |
|-------|-----------------------------|-----------------|------|
| Bytes | Cost (%)                    |                 |      |
| CPU)  | Time                        |                 |      |
| 0     | SELECT STATEMENT            |                 | 105  |
| 3150  | 8                           |                 |      |
| (13)  | 00:00:01                    |                 |      |
| * 1   | HASH JOIN                   |                 | 105  |
| 3150  | 8                           |                 |      |
| (13)  | 00:00:01                    |                 |      |
| 2     | TABLE ACCESS BY INDEX ROWID | ORDERS          | 105  |
| 1470  | 2                           |                 |      |
| (0)   | 00:00:01                    |                 |      |
| * 3   | INDEX RANGE SCAN            | ORD_CUSTOMER_IX | 105  |
|       | 1                           |                 |      |
| (0)   | 00:00:01                    |                 |      |

## 练习 8-2：使用加密的表空间（续）

```
|* 4 | TABLE ACCESS FULL | CUSTOMERS | 111
| 1776 | 5
(0) | 00:00:01 |

Predicate Information (identified by operation id):

 1 -
access("CUSTOMERS"."CUSTOMER_ID"="ORDERS"."CUSTOMER_ID")
 3 - access("ORDERS"."CUSTOMER_ID">0)
 4 - filter("ACCOUNT_MGR_ID"=145)

SQL>
SQL>
```

- c) 将访问过的对象（ORDERS 表、CUSTOMERS 表和 ORD\_CUSTOMER\_IX 索引）移动到加密表空间。
- 使用 ALTER TABLE ... MOVE TABLESPACE ENCTS 命令将 ORDERS 和 CUSTOMERS 表移至加密表空间。
- 可以使用 ALTER INDEX ... REBUILD TABLESPACE ENCTS 命令移动 ORD\_CUSTOMER\_IX 索引。

```
SQL>
SQL> connect oe/oe
Connected.
SQL> ALTER TABLE CUSTOMERS MOVE TABLESPACE ENCTS;

Table altered.

SQL> ALTER TABLE ORDERS MOVE TABLESPACE ENCTS;

Table altered.

SQL> ALTER INDEX ORD_CUSTOMER_IX REBUILD TABLESPACE ENCTS;

Index altered.

SQL>
```

练习 8-2：使用加密的表空间（续）

- d) 通过启用 autotrace，执行查询并检查执行计划。确认正在使用 ORD\_CUSTOMER\_IX 索引。以 OE 用户身份使用口令 oe 进行连接，启动自动跟踪，然后运行 query\_orders\_index.sql 脚本。检查执行计划。请注意，正在访问 ORD\_CUSTOMER\_IX 索引。

```
SQL> connect oe/oe
Connected.
SQL> set autotrace traceonly explain
SQL> set echo on
SQL> @query_orders_index
SQL> select cust_last_name, order_id, order_total
 2 from customers, orders
 3 where customers.customer_id = orders.customer_id
 4 and account_mgr_id = 145;
```

Execution Plan

-----

Plan hash value: 3558567882

-----

| Id | Operation                   | Name            | Rows |
|----|-----------------------------|-----------------|------|
| 0  | SELECT STATEMENT            |                 | 105  |
| 1  | HASH JOIN                   |                 | 105  |
| 2  | TABLE ACCESS BY INDEX ROWID | ORDERS          | 105  |
| 3  | INDEX RANGE SCAN            | ORD_CUSTOMER_IX | 105  |



**练习 8-2：使用加密的表空间（续）**

```

|* 4 | TABLE ACCESS FULL | CUSTOMERS | 111
| 1776 | 5
(0) | 00:00:01 |

Predicate Information (identified by operation id):

 1 -
access("CUSTOMERS"."CUSTOMER_ID"="ORDERS"."CUSTOMER_ID")
 3 - access("ORDERS"."CUSTOMER_ID">0)
 4 - filter("ACCOUNT_MGR_ID"=145)

SQL>

```

- e) 以 OE 用户身份使用 oe 口令进行连接。通过执行 cleanup.sql 脚本，将对象从 ENCTS 表空间移回 EXAMPLE 表空间。

```

SQL> connect oe/oe
Connected.
SQL> @cleanup
Connected.
SQL>
SQL> -- Move the OE.ORDERS table to EXAMPLE tablespace
SQL> ALTER TABLE orders MOVE TABLESPACE EXAMPLE;

Table altered.

SQL>
SQL> -- Move the OE.CUSTOMERS table to EXAMPLE tablespace
SQL> ALTER TABLE customers MOVE TABLESPACE EXAMPLE;

Table altered.

SQL>
SQL> -- Move the OE.ORD_CUSTOMER_IX index to the EXAMPLE
tablespace
SQL> ALTER INDEX ORD_CUSTOMER_IX REBUILD TABLESPACE EXAMPLE;

Index altered.

Connected.

SQL> DROP TABLESPACE encts including contents and datafiles;

```

## 练习 8-2：使用加密的表空间（续）

```
Tablespace dropped.
```

```
SQL>
```

```
SQL>
```

```
SQL> EXIT
```

```
Disconnected from Oracle Database 11g Enterprise Edition
```

```
Release 11.2.0.0.2 - Production
```

```
With the Partitioning, Automatic Storage Management, OLAP,
```

```
Data Mining and Real Application Testing options
```

```
$
```

## 第 9 课的练习

在本练习中，您将在一个升级场景中使用 SQL 性能分析器，在该场景中，您希望在升级 10g 数据库之前测试某些 SQL 语句。

## 练习 9-1：使用 SQL 性能分析器

在本练习中，您将从 10g 数据库中导出 SQL 优化集 (STS)，然后将其导回到 11g 测试环境。在该测试环境中，您将了解到在升级 10g 数据库之前导入的 SQL 语句的性能。

- 1) 在终端窗口（此窗口称为“第一个会话”）中执行 `setup_SPAbig10g.sh` 脚本，设置 10g 环境。

```
$ cd /home/oracle/solutions/SPAbig
$./setup_SPAbig10g.sh
rm: cannot remove
`/u01/app/oracle/product/10.2.0/dbhome_1/rdbms/log/apps.dmp':
No such file or directory
rm: cannot remove
`/u01/app/oracle/product/10.2.0/dbhome_1/rdbms/log/appsandstag
e.dmp': No such file or directory

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
08:53:20 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to an idle instance.

SQL> SQL> SQL> SQL> ORACLE instance started.

Total System Global Area 314572800 bytes
Fixed Size 1261540 bytes
Variable Size 117440540 bytes
Database Buffers 192937984 bytes
Redo Buffers 2932736 bytes
Database mounted.
Database opened.
SQL> SQL> BEGIN dbms_sqltune.drop_sqlset('STS_JFV','SYSTEM');
END;

*
ERROR at line 1:
ORA-13754: "SQL Tuning Set" "STS_JFV" does not exist for user
"SYSTEM".
ORA-06512: at "SYS.DBMS_SQLTUNE_INTERNAL", line 4672
ORA-06512: at "SYS.DBMS_SQLTUNE", line 2795
ORA-06512: at line 1

SQL> SQL> drop user apps cascade
*
ERROR at line 1:
ORA-01918: user 'APPS' does not exist
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
SQL> SQL>
Import: Release 10.2.0.3.0 - Production on Friday, 02 October,
2009 8:53:44

Copyright (c) 2003, 2005, Oracle. All rights reserved.

Connected to: Oracle Database 10g Enterprise Edition Release
10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully
loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_FULL_01": system/*****
directory=DATA_PUMP_DIR dumpfile=apps.dmp
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "APPS"."FACT_PD_OUT_ITM_293" 175.0
MB 1501663 rows
. . imported "APPS"."ADM_PG_FEATUREVALUE" 14.26
MB 172392 rows
. . imported "APPS"."LU_ELEMENTGROUP_REL" 3.477
MB 84468 rows
. . imported "APPS"."LU_OUTLET_293" 3.398
MB 22289 rows
. . imported "APPS"."LU_ITEM_293" 1.274
MB 5355 rows
. . imported "APPS"."LU_ELEMENTRANGE_REL" 779.3
KB 8155 rows
. . imported "APPS"."LU_ELEMENTRANGE_REL_Q2" 779.3
KB 8155 rows
. . imported "APPS"."ADM_CC_FEATUREVALUE" 248.5
KB 3047 rows
. . imported "APPS"."LU_PG_FEATUREVALUE_15_Q2" 122.8
KB 2505 rows
. . imported "APPS"."LU_PERIOD_293" 8.031
KB 31 rows
. . imported "APPS"."PLAN_TABLE" 0
KB 0 rows
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/FUNCTIONAL_AND_BITMAP/INDEX
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/FUNCTIONAL_AND_BITMAP/
INDEX_STATISTICS
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Job "SYSTEM"."SYS_IMPORT_FULL_01" successfully completed at
17:22:24

SQL> SQL>
LAST_ANAL

04-JUL-07

04-JUL-07
04-JUL-07
04-JUL-07
04-JUL-07
04-JUL-07
04-JUL-07

8 rows selected.

SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> Disconnected from Oracle Database 10g Enterprise
Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
```

- 2) 在该升级场景中使用 SQL 性能分析器之前，必须为 10g 数据库安装补丁程序 6903335。该补丁程序目前暂存在您的 /stage 目录下，名为 p6903335\_10203\_GENERIC.zip。请先为 db10g 数据库安装该补丁程序，然后再继续本练习。

```
$ cd /home/oracle/solutions/SPAbig
$ ls /stage
10.2.0 10.2.0.3_patchset 11.2.0 p6903335_10203_GENERIC.zip
$ rm -r 6903335
$ unzip /stage/p6903335_10203_GENERIC.zip
Archive: /stage/p6903335_10203_GENERIC.zip
 creating: 6903335/
 creating: 6903335/files/
 creating: 6903335/files/rdbms/
 creating: 6903335/files/rdbms/admin/
 inflating: 6903335/files/rdbms/admin/prvtspao.plb
 creating: 6903335/etc/
 creating: 6903335/etc/config/
 inflating: 6903335/etc/config/inventory
 inflating: 6903335/etc/config/actions
```

## 练习 9-1：使用 SQL 性能分析器（续）

```

 creating: 6903335/etc/xml/
 inflating: 6903335/etc/xml/GenericActions.xml
 inflating: 6903335/etc/xml/ShiphomeDirectoryStructure.xml
 inflating: 6903335/README.txt
$ ls
6903335 bench80.sh exportapps.sh movedmp.sh
setup_SPABig10g.sh stop_orcl.sh wrkl10g_jfv.sh
apps.dmp capsts10g.sh importapps.sh new_wrkl10g_jfv.sh
stage_sts.sh unpack_sts.sh
$ cd 6903335
$. oraenv
ORACLE_SID = [orcl] ? db10g
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/10.2.0/dbhome_1 is
/u01/app/oracle
$ export PATH=$PATH:$ORACLE_HOME/OPatch
$ echo $PATH
/u01/app/oracle/product/10.2.0/dbhome_1/bin:/usr/java/jdk1.5.0
_11/bin:/bin:/home/oracle/bin:/usr/bin:/usr/local/bin:/usr/
X11R6/bin:/u01/app/oracle/product/10.2.0/dbhome_1/OPatch:/u01/
app/oracle/product/10.2.0/dbhome_1/OPatch
$ opatch lsinventory
Invoking OPatch 10.2.0.3.0

Oracle interim Patch Installer version 10.2.0.3.0
Copyright (c) 2005, Oracle Corporation. All rights reserved..

Oracle Home : /u01/app/oracle/product/10.2.0/dbhome_1
Central Inventory : /u01/app/oraInventory
 from : /etc/oraInst.loc
OPatch version : 10.2.0.3.0
OUI version : 10.2.0.3.0
OUI location :
/u01/app/oracle/product/10.2.0/dbhome_1/oui
Log file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
opatch2009-10-02_08-59-45AM.log

Lsinventory Output file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
lsinventory2009-10-02_08-59-45AM.txt

Installed Top-level Products (2):

```

## 练习 9-1：使用 SQL 性能分析器（续）

```

Oracle Database 10g
10.2.0.1.0
Oracle Database 10g Release 2 Patch Set 2
10.2.0.3.0
There are 2 products installed in this Oracle Home.

Interim patches (2) :

Patch 5556081 : applied on Fri May 08 17:54:39 UTC 2009
 Created on 9 Nov 2006, 22:20:50 hrs PST8PDT
 Bugs fixed:
 5556081

Patch 5557962 : applied on Fri May 08 17:54:31 UTC 2009
 Created on 9 Nov 2006, 23:23:06 hrs PST8PDT
 Bugs fixed:
 4269423, 5557962, 5528974

OPatch succeeded.
$ opatch apply
Invoking OPatch 10.2.0.3.0

Oracle interim Patch Installer version 10.2.0.3.0
Copyright (c) 2005, Oracle Corporation. All rights reserved..

Oracle Home : /u01/app/oracle/product/10.2.0/dbhome_1
Central Inventory : /u01/app/oraInventory
 from : /etc/oraInst.loc
OPatch version : 10.2.0.3.0
OUI version : 10.2.0.3.0
OUI location :
/u01/app/oracle/product/10.2.0/dbhome_1/oui
Log file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
opatch2009-10-02_09-01-54AM.log

ApplySession applying interim patch '6903335' to OH
'/u01/app/oracle/product/10.2.0/dbhome_1'

OPatch detected non-cluster Oracle Home from the inventory and
will patch the local system only.

Please shutdown Oracle instances running out of this
ORACLE_HOME on the local system.
(Oracle Home = '/u01/app/oracle/product/10.2.0/dbhome_1')

```



## 练习 9-1：使用 SQL 性能分析器（续）

```
Is the local system ready for patching?

Do you want to proceed? [y|n]
n
User Responded with: N
ApplySession exits on request
ApplySession system modification phase did not start: Oracle
Home is not modified. ApplySession exits on request
System intact, OPatch will not attempt to restore the system

OPatch stopped on request.
$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
09:03:47 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition
Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
$ opatch apply
Invoking OPatch 10.2.0.3.0

Oracle interim Patch Installer version 10.2.0.3.0
Copyright (c) 2005, Oracle Corporation. All rights reserved..

Oracle Home : /u01/app/oracle/product/10.2.0/dbhome_1
Central Inventory : /u01/app/oraInventory
 from : /etc/oraInst.loc
OPatch version : 10.2.0.3.0
OUI version : 10.2.0.3.0
OUI location :
/u01/app/oracle/product/10.2.0/dbhome_1/oui
Log file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
opatch2009-10-02_09-04-59AM.log
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
ApplySession applying interim patch '6903335' to OH
'/u01/app/oracle/product/10.2.0/dbhome_1'

OPatch detected non-cluster Oracle Home from the inventory and
will patch the local system only.

Please shutdown Oracle instances running out of this
ORACLE_HOME on the local system.
(Oracle Home = '/u01/app/oracle/product/10.2.0/dbhome_1')

Is the local system ready for patching?

Do you want to proceed? [y|n]
y
User Responded with: Y
Backing up files and inventory (not for auto-rollback) for the
Oracle Home
Backing up files affected by the patch '6903335' for
restore. This might take a while...
Backing up files affected by the patch '6903335' for
rollback. This might take a while...

Patching component oracle.rdbms, 10.2.0.3.0...
Copying file to
"/u01/app/oracle/product/10.2.0/dbhome_1/rdbms/admin/prvtspao.
plb"
ApplySession adding interim patch '6903335' to inventory

Verifying the update...
Inventory check OK: Patch ID 6903335 is registered in Oracle
Home inventory with proper meta-data.
Files check OK: Files from Patch ID 6903335 are present in
Oracle Home.

The local system has been patched and can be restarted.

OPatch succeeded.
$
$ opatch lsinventory
Invoking OPatch 10.2.0.3.0

Oracle interim Patch Installer version 10.2.0.3.0
Copyright (c) 2005, Oracle Corporation. All rights reserved..

Oracle Home : /u01/app/oracle/product/10.2.0/dbhome_1
Central Inventory : /u01/app/oraInventory
 from : /etc/oraInst.loc
OPatch version : 10.2.0.3.0
OUI version : 10.2.0.3.0
```

## 练习 9-1：使用 SQL 性能分析器（续）

```

OUI location :
/u01/app/oracle/product/10.2.0/dbhome_1/oui
Log file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
opatch2009-10-02_09-06-47AM.log

Lsinventory Output file location :
/u01/app/oracle/product/10.2.0/dbhome_1/cfgtoollogs/opatch/
lsinventory2009-10-02_09-06-47AM.txt

Installed Top-level Products (2):

Oracle Database 10g
10.2.0.1.0
Oracle Database 10g Release 2 Patch Set 2
10.2.0.3.0
There are 2 products installed in this Oracle Home.

Interim patches (3) :

Patch 6903335 : applied on Fri Oct 02 09:05:29 UTC 2009
Created on 12 May 2008, 08:56:41 hrs US/Pacific
Bugs fixed:
6903335

Patch 5556081 : applied on Fri May 08 17:54:39 UTC 2009
Created on 9 Nov 2006, 22:20:50 hrs PST8PDT
Bugs fixed:
5556081

Patch 5557962 : applied on Fri May 08 17:54:31 UTC 2009
Created on 9 Nov 2006, 23:23:06 hrs PST8PDT
Bugs fixed:
4269423, 5557962, 5528974

OPatch succeeded.
$
$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
09:11:38 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to an idle instance.

```

## 练习 9-1：使用 SQL 性能分析器（续）

```
SQL> startup
ORACLE instance started.

Total System Global Area 314572800 bytes
Fixed Size 1261540 bytes
Variable Size 138412060 bytes
Database Buffers 171966464 bytes
Redo Buffers 2932736 bytes
Database mounted.
Database opened.
SQL> drop table plan_table$;

Table dropped.

SQL> @?/rdbms/admin/catplan.sql
create type dbms_xplan_type
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

create type dbms_xplan_type_table
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

Grant succeeded.

Grant succeeded.

create type sql_plan_row_type
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

Synonym created.

Grant succeeded.

create type sql_plan_table_type
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

Synonym created.

Grant succeeded.
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
create sequence ora_plan_id_seq$
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

Table created.

Grant succeeded.

Synonym created.

create type sql_plan_stat_row_type
*
ERROR at line 1:
ORA-00955: name is already used by an existing object

Synonym created.

Grant succeeded.

Library created.

No errors.

Library created.

Package created.

Synonym created.

Grant succeeded.

Package body created.

No errors.
SQL>
SQL> @?/rdbms/admin/dbmsxpln.sql

Library created.

No errors.

Library created.

Package created.

Synonym created.

Grant succeeded.
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
SQL>
SQL> @?/rdbms/admin/prvtxp1n.plb

Package body created.

No errors.
SQL>
SQL> @?/rdbms/admin/prvtspao.plb

PL/SQL procedure successfully completed.

Package created.

No errors.

Synonym created.

No errors.

Package body created.

No errors.
SQL>
SQL> grant all on dbms_sqlpa to system;

Grant succeeded.

SQL>
SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition
Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
```

- 3) 在 10g 环境下创建 STS 之前，停止 Enterprise Manager Database Control 并关闭 11g 数据库实例 orcl。这样可节省系统上的资源。使用 stop\_orcl.sh 脚本执行此步骤。请注意，该脚本还将清除本练习之前可能对 orcl 数据库执行的操作。

```
$ cd /home/oracle/solutions/SPAbig
$./stop_orcl.sh
rm: cannot remove
`/u01/app/oracle/admin/orcl/dpdump/appsandstage.dmp': No such
file or directory
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
```

**练习 9-1：使用 SQL 性能分析器（续）**

```

https://ed.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 10:59:06
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> BEGIN
dbms_sqlpa.drop_analysis_task('SPA_JFV1'); END;

*
ERROR at line 1:
ORA-13605: The specified task or object SPA_JFV1 does not
exist for the current
user.
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 86
ORA-06512: at "SYS.PRVT_ADVISOR", line 5827
ORA-06512: at "SYS.DBMS_SQLPA", line 400
ORA-06512: at line 1

SQL> SQL> BEGIN dbms_sqlpa.drop_analysis_task('SPA_JFV2');
END;

*
ERROR at line 1:
ORA-13605: The specified task or object SPA_JFV2 does not
exist for the current
user.
ORA-06512: at "SYS.DBMS_SYS_ERROR", line 86
ORA-06512: at "SYS.PRVT_ADVISOR", line 5827
ORA-06512: at "SYS.DBMS_SQLPA", line 400
ORA-06512: at line 1

SQL> SQL> BEGIN dbms_sqltune.drop_sqlset('STS_JFV','SYSTEM');
END;

*
ERROR at line 1:
ORA-13754: "SQL Tuning Set" "STS_JFV" does not exist for user
"SYSTEM".
ORA-06512: at "SYS.DBMS_SQLTUNE_INTERNAL", line 11567
ORA-06512: at "SYS.DBMS_SQLTUNE", line 3118
ORA-06512: at line 1

```

## 练习 9-1：使用 SQL 性能分析器（续）

```
SQL> SQL> drop user apps cascade
*
ERROR at line 1:
ORA-01918: user 'APPS' does not exist

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 4) （同时执行步骤 4 和 5。）生成名为 STS\_JFV 的 SQL 优化集 (STS)，该优化集每隔五秒就会从游标高速缓存中捕获一次 SQL 语句，每次捕获大约持续十二分钟。确保只尝试捕获来自 APPS 方案的 SQL\_JFV 模块的语句。该 STS 也应属于 SYSTEM 用户。使用 capsts10g.sh 脚本执行该步骤。

```
$ cd /home/oracle/solutions/SPAbig
$./capsts10g.sh

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
12:13:57 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> SQL> SQL> SQL> SQL> 2 3 4 5 6 7 8
9 10 11
PL/SQL procedure successfully completed.

Elapsed: 00:09:04.90
SQL> No errors.
SQL> SQL> Disconnected from Oracle Database 10g Enterprise
Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
```



## 练习 9-1：使用 SQL 性能分析器（续）

- 5) （同时执行步骤 4 和 5。）在以 oracle 用户身份连接的第二个终端窗口中，使用 wrk110g\_jfv.sh 脚本执行工作量。该脚本运行 45 个语句的工作量，STS\_JFV 会自动捕获这些语句。

```
$ cd /home/oracle/solutions/SPAbig
$
$./wrk110g_jfv.sh

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
12:14:27 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> SQL> SQL>
System altered.

SQL>
System altered.

SQL> SQL> SQL> SQL> Connected.
SQL> SQL> SQL> SQL>
Session altered.

SQL> SQL> SQL> SQL> SQL> SQL> @Start 2009:OCT-02:12:15:26

SQL> SQL> SQL> SQL> SQL> SQL> SQL>
System altered.

SQL> SQL> SQL> @Statement 1
SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> 2 3 4 5 6 7 8 9 10
11 12 13 14
Elapsed: 00:00:00.39

Statistics

 957 recursive calls
 0 db block gets
 292 consistent gets
 146 physical reads
 188 redo size
 427 bytes sent via SQL*Net to client
 400 bytes received via SQL*Net from client
```

练习 9-1：使用 SQL 性能分析器（续）

```

 2 SQL*Net roundtrips to/from client
 18 sorts (memory)
 0 sorts (disk)
 1 rows processed

SQL> @1 Results in 15 Ticks
SQL> SQL> SQL> SQL> @Statement 2
SQL> SQL> SQL> 2 3 4 5 6 7 8 9 10
11 12 13 14
Elapsed: 00:00:00.01

Statistics

 8 recursive calls
 0 db block gets
 120 consistent gets
 9 physical reads
 0 redo size
 428 bytes sent via SQL*Net to client
 400 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 0 sorts (memory)
 0 sorts (disk)
 1 rows processed

SQL> @1 Results in 0 Ticks
SQL> SQL> SQL> SQL> @Statement 3
SQL> SQL> SQL> 2 3 4 5 6 7 8 9 10
11 12 13 14
Elapsed: 00:00:00.03

Statistics

 8 recursive calls
 0 db block gets
 120 consistent gets
 16 physical reads
 0 redo size
 428 bytes sent via SQL*Net to client
 400 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 0 sorts (memory)
 0 sorts (disk)
 1 rows processed

SQL> @1 Results in 16 Ticks
SQL> SQL> SQL> SQL> @Statement 4
SQL> SQL> SQL> 2 3 4 5 6 7 8 9 10
11 12 13 14
Elapsed: 00:00:00.30
```

## 练习 9-1：使用 SQL 性能分析器（续）

```

Statistics

 470 recursive calls
 0 db block gets
 598 consistent gets
 528 physical reads
 72 redo size
 427 bytes sent via SQL*Net to client
 400 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 14 sorts (memory)
 0 sorts (disk)
 1 rows processed

SQL> @1 Results in 0 Ticks
SQL> SQL> SQL> SQL> @Statement 5
SQL> SQL> SQL> 2 3 4 5 6 7 8 9 10
11 12 13 14
Elapsed: 00:00:00.02

Statistics

 8 recursive calls
 0 db block gets
 525 consistent gets
 9 physical reads
 0 redo size
 428 bytes sent via SQL*Net to client
 400 bytes received via SQL*Net from client
 2 SQL*Net roundtrips to/from client
 1 sorts (memory)
 0 sorts (disk)
 1 rows processed

...

SQL> @10 Results in 1188 Ticks
SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> we are done
SQL> SQL> SQL> SQL> @End 2009:OCT-02:12:18:16

SQL> SQL> SQL> Disconnected from Oracle Database 10g
Enterprise Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$

```

## 练习 9-1：使用 SQL 性能分析器（续）

- 6) 大约五分钟后，这两个会话就会完成。检查 STS\_JFV 的内容，并将其暂存在名为 APPS.STS\_JFV\_TAB 的表中。使用 stage\_sts.sh 脚本执行此步骤。

```
$ cd /home/oracle/solutions/SPAbig
$./stage_sts.sh

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
12:25:05 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> SQL> SQL> SQL>
NAME STATEMENT_COUNT

STS_JFV 33

SQL> SQL> drop table apps.sts_jfv_tab purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> Disconnected from Oracle Database 10g Enterprise
Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
```

- 7) 使用数据泵导出，将 APPS 方案导出到默认的数据泵 10g 目录。使用 exportapps.sh 脚本执行此步骤。

```
$ cd /home/oracle/solutions/SPAbig
$./exportapps.sh

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
12:26:40 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
```

# 练习 9-1：使用 SQL 性能分析器（续）

```

With the Partitioning, OLAP and Data Mining options

SQL> SQL>
Table dropped.

SQL> SQL> Disconnected from Oracle Database 10g Enterprise
Edition Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
rm: cannot remove
`/u01/app/oracle/product/10.2.0/dbhome_1/rdbms/log/appsandstag
e.dmp': No such file or directory

Export: Release 10.2.0.3.0 - Production on Friday, 02 October,
2009 12:26:41

Copyright (c) 2003, 2005, Oracle. All rights reserved.

Connected to: Oracle Database 10g Enterprise Edition Release
10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
Starting "SYSTEM"."SYS_EXPORT_SCHEMA_01": system/*****
directory=DATA_PUMP_DIR dumpfile=appsandstage schemas=apps
Estimate in progress using BLOCKS method...
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
Total estimation using BLOCKS method: 240.5 MB
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/FUNCTIONAL_AND_BITMAP/INDEX
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/FUNCTIONAL_AND_BITMAP/
INDEX_STATISTICS
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
. . exported "APPS"."FACT_PD_OUT_ITM_293" 175.0
MB 1501663 rows
. . exported "APPS"."ADM_PG_FEATUREVALUE" 14.26
MB 172392 rows
. . exported "APPS"."LU_ELEMENTGROUP_REL" 3.477
MB 84468 rows
. . exported "APPS"."LU_OUTLET_293" 3.398
MB 22289 rows

```

## 练习 9-1：使用 SQL 性能分析器（续）

```
. . exported "APPS"."LU_ITEM_293" 1.274
MB 5355 rows
. . exported "APPS"."LU_ELEMENTRANGE_REL" 779.3
KB 8155 rows
. . exported "APPS"."LU_ELEMENTRANGE_REL_Q2" 779.3
KB 8155 rows
. . exported "APPS"."STS_JFV_TAB_CPLANS" 254.8
KB 570 rows
. . exported "APPS"."STS_JFV_TAB" 80.05
KB 33 rows
. . exported "APPS"."ADM_CC_FEATUREVALUE" 248.5
KB 3047 rows
. . exported "APPS"."LU_PG_FEATUREVALUE_15_Q2" 122.8
KB 2505 rows
. . exported "APPS"."STS_JFV_TAB_CBINDS" 9.484
KB 0 rows
. . exported "APPS"."LU_PERIOD_293" 8.031
KB 31 rows
Master table "SYSTEM"."SYS_EXPORT_SCHEMA_01" successfully
loaded/unloaded

Dump file set for SYSTEM.SYS_EXPORT_SCHEMA_01 is:

/u01/app/oracle/product/10.2.0/dbhome_1/rdbms/log/appsandstage
.dmp
Job "SYSTEM"."SYS_EXPORT_SCHEMA_01" successfully completed at
12:27:24

$
```

- 8) 务必将生成的转储文件从默认 Data Pump 10g 目录复制到默认 Data Pump 11g 目录。使用 movedmp.sh 脚本执行此步骤。

```
$ cd /home/oracle/solutions/SPAbig
$./movedmp.sh
```

- 9) 启动包括数据库实例和 Enterprise Manager Database Control 的 11g 环境：

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 12:30:37
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to an idle instance.
```

## 练习 9-1：使用 SQL 性能分析器（续）

```
SQL> startup
ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 293603712 bytes
Database Buffers 121634816 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://...:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database
Control started.

Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/ed.us.oracle.com_orcl/sysm
an/log
$
```

- 10) 通过使用数据泵导入实用程序，将 APPS 方案导入到 11g 系统。使用 `importapps.sh` 脚本执行该步骤。可以忽略下列错误，不必担心。它们是由一些转换错误造成的。

```
$ cd /home/oracle/solutions/SPAbig
$./importapps.sh

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 12:39:21
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
```

**练习 9-1：使用 SQL 性能分析器（续）**

```

SQL> SQL> drop user apps cascade
*
ERROR at line 1:
ORA-01918: user 'APPS' does not exist

SQL> SQL>
Import: Release 11.2.0.1.0 - Production on Friday, 02 October,
2009 12:40:30

Copyright (c) 2003, 2007, Oracle. All rights reserved.

Connected to: Oracle Database 11g Enterprise Edition Release
11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Master table "SYSTEM"."SYS_IMPORT_FULL_01" successfully
loaded/unloaded
Starting "SYSTEM"."SYS_IMPORT_FULL_01": system/*****
directory=DATA_PUMP_DIR dumpfile=appsandstage
Processing object type SCHEMA_EXPORT/USER
Processing object type SCHEMA_EXPORT/SYSTEM_GRANT
Processing object type SCHEMA_EXPORT/ROLE_GRANT
Processing object type SCHEMA_EXPORT/DEFAULT_ROLE
Processing object type SCHEMA_EXPORT/PRE_SCHEMA/PROCACT_SCHEMA
Processing object type SCHEMA_EXPORT/TABLE/TABLE
Processing object type SCHEMA_EXPORT/TABLE/TABLE_DATA
. . imported "APPS"."FACT_PD_OUT_ITM_293" 175.0
MB 1501663 rows
. . imported "APPS"."ADM_PG_FEATUREVALUE" 14.26
MB 172392 rows
. . imported "APPS"."LU_ELEMENTGROUP_REL" 3.477
MB 84468 rows
. . imported "APPS"."LU_OUTLET_293" 3.398
MB 22289 rows
. . imported "APPS"."LU_ITEM_293" 1.274
MB 5355 rows
ORA-02374: conversion error loading table
"APPS"."LU_ELEMENTRANGE_REL"
ORA-12899: value too large for column ELEMENTRANGE_SDESC
(actual: 16, maximum: 15)

ORA-02372: data for row: ELEMENTRANGE_SDESC :
0X'32302C3031202D2033302C303020BF'

ORA-02374: conversion error loading table
"APPS"."LU_ELEMENTRANGE_REL"
ORA-12899: value too large for column ELEMENTRANGE_SDESC
(actual: 16, maximum: 15)

```



## 练习 9-1：使用 SQL 性能分析器（续）

```

ORA-02372: data for row: ELEMENTRANGE_SDESC :
0X'33302C3031202D2034302C303020BF'

...

. . imported "APPS"."LU_ELEMENTRANGE_REL_Q2" 779.3
KB 7957 out of 8155 rows
. . imported "APPS"."STS_JFV_TAB_CPLANS" 254.8
KB 570 rows
. . imported "APPS"."STS_JFV_TAB" 80.05
KB 33 rows
. . imported "APPS"."ADM_CC_FEATUREVALUE" 248.5
KB 3047 rows
. . imported "APPS"."LU_PG_FEATUREVALUE_15_Q2" 122.8
KB 2505 rows
. . imported "APPS"."STS_JFV_TAB_CBINDS" 9.484
KB 0 rows
. . imported "APPS"."LU_PERIOD_293" 8.031
KB 31 rows
Processing object type SCHEMA_EXPORT/TABLE/INDEX/INDEX
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/INDEX_STATISTICS
Processing object type SCHEMA_EXPORT/VIEW/VIEW
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/FUNCTIONAL_AND_BITMAP/INDEX
Processing object type
SCHEMA_EXPORT/TABLE/INDEX/STATISTICS/FUNCTIONAL_AND_BITMAP/
INDEX_STATISTICS
Processing object type
SCHEMA_EXPORT/TABLE/STATISTICS/TABLE_STATISTICS
Job "SYSTEM"."SYS_IMPORT_FULL_01" successfully completed at
12:42:30

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

## 练习 9-1：使用 SQL 性能分析器（续）

- 11) 解压缩先前在 SYSTEM 方案中导入的临时表。使用 `unpack_sts.sh` 脚本执行该步骤。

```
$ cd /home/oracle/solutions/SPAbig
$./unpack_sts.sh

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 12:46:29
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL>
System altered.

SQL>
System altered.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 12) 使用 NETCA 为 db10g 数据库实例创建一个 TNS 条目。您必须从 11g 数据库主目录执行该操作。

- a) 在图形终端会话中，将环境变量设置为指向 11g 安装。然后执行 `netca` 命令：

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ netca
```

- b) 在 “Welcome (欢迎使用)” 页上，选择 “Local Net Service Name configuration (本地网络服务名配置)”，然后单击 “Next (下一步)”。
- c) 在 “Net Service Name Configuration (网络服务名配置)” 页上，选择 “Add (添加)”，然后单击 “Next (下一步)”。
- d) 在 “Service Name Configuration (服务名配置)” 页的 “Service Name (服务名)” 字段中输入 db10g。然后，单击 “Next (下一步)”。

## 练习 9-1：使用 SQL 性能分析器（续）

- e) 在 “Select Protocol（选择协议）” 页上，选择 TCP 并单击 “Next（下一步）”。
  - f) 在 “TCP/IP Protocol（TCP/IP 协议）” 页的 “Host Name（主机名）” 字段中输入您的主机名。在终端窗口中使用 `hostname` 命令获取确切名称。请确保使用的端口号为 1521。然后单击 “Next（下一步）”。
  - g) 在 “Test（测试）” 页上，选择 “Yes（是）”，然后单击 “Next（下一步）”。
  - h) 在 “Connecting（连接）” 页上，单击 “Change Login（更改登录）”。
  - i) 在 “Change Login（更改登录）” 窗口中，输入 `system/oracle` 作为测试的身份证明。单击 “OK（确定）”。
  - j) 返回 “Connecting（连接）” 页，现在测试应成功完成。单击 “Next（下一步）”。
  - k) 在 “Net Service Name（网络服务名）” 页的 “Net Service Name（网络服务名）” 字段中输入 `db10g`。单击 “Next（下一步）”。
  - l) 在 “Another Net Service Name（是否配置另一个网络服务名）” 页上，选择 “No（否）”，然后单击 “Next（下一步）”。
  - m) 在 “Done（完成）” 页上，单击 “Next（下一步）”。
  - n) 在 “Welcome（欢迎使用）” 页上单击 “Finish（完成）”。
- 13) 使用 Enterprise Manager Database Control，以 SYSTEM 用户身份测试 STS\_JFV 在 11g 环境下的行为，并与其在 10g 环境下的执行进行对比。您会得出什么结论？
- a) 在主页中，单击 “Software and Support（软件和支持）” 选项卡。
  - b) 在 “Software and Support（软件和支持）” 子页上，单击 “Real Application Testing（实际应用程序测试）” 部分中的 “SQL Performance Analyzer（SQL 性能分析器）”。
  - c) 在 “SQL Performance Analyzer（SQL 性能分析器）” 页上，单击 “Upgrade from 10.2 or 11g（从 10.2 或 11g 升级）” 链接。
  - d) 在 “Upgrade from 10.2 or 11g（从 10.2 或 11g 升级）” 页上，在 “Task Name（任务名）” 字段中输入：SPA\_JFV1，在 “SQL Tuning Set（SQL 优化集）” 字段中输入 SYSTEM.STS\_JFV。从 “Creation Method（创建方法）” 下拉列表中选择 “Execute SQLs（执行 SQL）”。从 “Per-SQL Time Limit（每个 SQL 的时间限制）” 中选择 “Unlimited（无限制）”。

## 练习 9-1：使用 SQL 性能分析器（续）

- e) 在 “Pre-upgrade Trial（升级前试运行）” 部分中单击 “Create Database Link（创建数据库链接）” 按钮。
- f) 在 “Create Database Link（创建数据库链接）” 页上，在 “Name（名称）” 字段中输入 `db10g.us.oracle.com`。  
在 “Net Service Name（网络服务名）” 字段中输入 `db10g`。  
请确保已选中 “Public（公用）”。  
在 “Connect As（连接身份）” 部分中选择 “Fixed User（固定用户）”，  
“Username（用户名）” 使用 `system`，两个 “Password（口令）” 使用 `oracle`。  
然后单击 “OK（确定）”。
- g) 返回 “Upgrade from 10.2 or 11g（从 10.2 或 11g 升级）” 页，在  
“Pre-upgrade Trial（升级前试运行）” 部分的 “Database Link（数据库链接）” 字段中输入 `DB10G.US.ORACLE.COM`。
- h) 在 “Post-upgrade Trial（升级后试运行）” 部分中，取消选择 “Use the same system as in the pre-upgrade trial（与升级前试运行使用同一系统）”。然后  
在该部分中单击 “Create Database Link（创建数据库链接）” 按钮。
- i) 在 “Create Database Link（创建数据库链接）” 页上，在 “Name（名称）” 字段中输入 `orcl.us.oracle.com@loopback`。  
在 “Net Service Name（网络服务名）” 字段中输入 `orcl`。  
请确保已选中 “Public（公用）”。  
在 “Connect As（连接身份）” 部分中选择 “Fixed User（固定用户）”，  
“Username（用户名）” 使用 `system`，两个 “Password（口令）” 使用 `oracle_4U`。  
然后单击 “OK（确定）”。
- j) 返回 “Upgrade from 10.2 or 11g（从 10.2 或 11g 升级）” 页，在  
“Post-upgrade Trial（升级后试运行）” 部分的 “Database Link（数据库链接）” 字段中输入 `ORCL.US.ORACLE.COM@LOOPBACK`。
- k) 务必将 “Comparison Metric（比较度量）” 字段设置为 “Elapsed Time（用时）”，将 “Schedule（调度）” 设置为 “Immediately（立即执行）”。完成后，单击 “Submit（提交）”。
- l) 返回 “SQL Performance Analyzer（SQL 性能分析器）” 页，等待所提交任务（SPA\_JFV1）的 “Status（状态）” 变为 “Completed（已完成）”。执行该操作大约会花费 10 分钟的时间。
- m) 单击 SPA\_JFV1 链接。

## 练习 9-1：使用 SQL 性能分析器（续）

- n) 此时将转到 SQL 性能分析器的 “Task: SYS.SPA\_JFV1（任务：SYS.SPA\_JFV1）” 页在 “SQL Trial Comparisons（SQL 试运行比较）” 表的 “Comparison Report（比较报表）” 列中单击眼镜图标。该表中应该只有一个条目。
  - o) 在 “SQL Performance Analyzer Tasks（SQL 性能分析器任务）” 的 “Report: SYSTEM.SPA\_JFV1（报表：SYSTEM.SPA\_JFV1）” 页中，可以看出第二次试运行比第一次试运行快。第二次试运行对应于使用 Oracle DB 11g 的运行。这种改进来源于捕获和重新执行的语句。您还应该看到大部分语句都在使用实施了 11g 优化的新计划。
  - p) 在 “Top 10 SQL Statements Based on Impact on Workload（根据对工作量的影响列出的前 10 个 SQL 语句）” 表中，单击其中的一个 SQL ID 链接。
  - q) 此时将转到相应的 SQL 详细资料页，从中可查找语句文本以及在尝试这两种方法期间使用的不同执行计划。
  - r) 可以单击 “SQL Tuning Set Name（SQL 优化集名）” 字段右侧的 STS\_JFV 链接，调用 SQL 优化指导优化该语句。但是，在此练习中将不执行该操作。
- 14) 在终端窗口中，确保关闭了 db10g 数据库实例。

```
$. oraenv
ORACLE_SID = [orcl] ? db10g
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/10.2.0/dbhome_1 is
/u01/app/oracle
$ sqlplus / as sysdba

SQL*Plus: Release 10.2.0.3.0 - Production on Fri Oct 2
13:26:23 2009

Copyright (c) 1982, 2006, Oracle. All Rights Reserved.

Connected to:
Oracle Database 10g Enterprise Edition Release 10.2.0.3.0 -
Production
With the Partitioning, OLAP and Data Mining options

SQL> shutdown immediate
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> exit
Disconnected from Oracle Database 10g Enterprise Edition
Release 10.2.0.3.0 - Production
With the Partitioning, OLAP and Data Mining options
$
```

## 第 10 课的练习

SQL 计划管理 (SPM) 是 Oracle Database 11g 新增的一项功能，可以提供受控制的执行计划演变。

通过 SPM，优化程序可自动管理执行计划，并确保只使用已知或经过验证的计划。

如果发现 SQL 语句存在新计划，则仅在验证了该计划的性能相当于或优于当前计划的性能后，才使用此新计划。

SPM 包含三个主要组件：

1. 计划捕获
2. 计划选择
3. 计划验证

在本练习中，您将逐一了解这些组件。

## 练习 10-1：使用 SQL 计划管理

在此练习中，您会看到使用 SQL 计划管理的各个阶段。

- 1) 在开始本练习之前，需要设置一个新用户。执行 `spm_setup.sh` 脚本为本练习设置环境。此脚本将创建 SPM 用户，您在本练习中会始终使用该用户。

```
$ cd /home/oracle/solutions/SPM
$./spm_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 14:20:15
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> drop user spm cascade
*
ERROR at line 1:
ORA-01918: user 'SPM' does not exist

SQL> SQL>
User created.

SQL> SQL>
Grant succeeded.

SQL> SQL>
System altered.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

# 练习 10-1：使用 SQL 计划管理（续）

- SPM 的第一个组件是计划捕获。捕获计划的方法主要有两种：自动（即时）捕获或批量加载。首先，我们来看自动捕获。在本练习中，您需要启用自动计划捕获，以便为任何可重复的 SQL 语句自动填充 SPM 资料档案库。在以 SPM 用户身份连接的 SQL\*Plus 会话中，将初始化参数 `OPTIMIZER_CAPTURE_SQL_PLAN_BASELINES` 设置为 `TRUE`，启用自动计划捕获。连接到会话后，**请勿断开连接**。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ sqlplus spm/spm

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 2 14:22:34
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> show parameter baseline

NAME TYPE VALUE

optimizer_capture_sql_plan_baselines boolean FALSE
optimizer_use_sql_plan_baselines boolean TRUE
SQL> alter session set optimizer_capture_sql_plan_baselines =
TRUE;

Session altered.

SQL>
```

- 在 SQL\*Plus 会话中执行以下查询（/\*LOAD... 中没有空格）：  
`select /*LOAD_AUTO*/ * from sh.sales where quantity_sold > 40 order by prod_id` 使用 `query1.sql` 脚本执行该查询。

```
SQL> @query1
SQL>
SQL> select /*LOAD_AUTO*/ * from sh.sales where quantity_sold
> 40 order by prod_id;

no rows selected

SQL>
```



## 练习 10-1：使用 SQL 计划管理（续）

- 4) 由于是首次见到此 SQL 语句，并且该语句尚不可重复，因此，该语句没有计划基线。要确认这一点，可以检查是否未加载计划基线。请检查该语句是否存在计划基线（使用 check\_baselines.sql 脚本）。

```
SQL>
SQL> @check_baselines
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_AUTO*/%';

no rows selected

SQL>
```

- 5) 重新执行在步骤 3 中运行的查询：

```
SQL>
SQL> @query1
SQL> set echo on
SQL>
SQL> select /*LOAD_AUTO*/ * from sh.sales where quantity_sold
> 40 order by prod_id;

no rows selected

SQL>
```

- 6) 现在已知 SQL 语句是可重复的，并已自动捕获了计划基线。请检查是否为以前的语句加载了计划基线。会看到什么结果？
- a) 在输出中可以看到已为此 SQL 语句创建并启用计划基线。通过查看起始列中的值还可以识别出此计划是自动捕获的（使用 check\_baselines.sql 脚本）：

```
SQL>
SQL> @check_baselines
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_AUTO*/%';

SIGNATURE SQL_HANDLE

SQL_TEXT

```

# 练习 10-1：使用 SQL 计划管理（续）

| PLAN_NAME                                                     | ORIGIN       | ENA | ACC | FIX | AUT |
|---------------------------------------------------------------|--------------|-----|-----|-----|-----|
| 8.0622E+18 SYS_SQL_6fe28d438dfc352f                           |              |     |     |     |     |
| select /*LOAD_AUTO*/ * from sh.sales where quantity_sold > 40 |              |     |     |     |     |
| order by prod_id                                              |              |     |     |     |     |
| SQL_PLAN_6zsd8f6zsd9g54bc8843                                 | AUTO-CAPTURE | YES | YES | NO  | YES |

SQL>

7) 现在，将优化程序模式改为使用 FIRST\_ROWS 优化，然后重新执行该语句。描述发生的情况：

a) 这会导致使用另一个计划执行 SQL 语句：

|                                                               |
|---------------------------------------------------------------|
| SQL>                                                          |
| SQL> alter session set optimizer_mode = first_rows;           |
| Session altered.                                              |
| SQL> @query1                                                  |
| SQL> set echo on                                              |
| SQL>                                                          |
| SQL> select /*LOAD_AUTO*/ * from sh.sales where quantity_sold |
| > 40 order by prod_id;                                        |
| no rows selected                                              |
| SQL>                                                          |

b) 由于 SQL 语句将拥有一个新计划，因此将自动捕获另一计划基线。可以通过再次检查该计划基线对此进行确认：

```
SQL> @check_baselines
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_AUTO*/%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

8.0622E+18 SYS_SQL_6fe28d438dfc352f
select /*LOAD_AUTO*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_6zsd8f6zsd9g11df68d0 AUTO-CAPTURE YES NO NO YES
```

## 练习 10-1：使用 SQL 计划管理（续）

```

8.0622E+18 SYS_SQL_6fe28d438dfc352f
select /*LOAD_AUTO*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_6zsd8f6zsd9g54bc8843 AUTO-CAPTURE YES YES NO YES

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

SQL>

```

c) 现在可以看到查询具有两个计划基线，但请注意，第二个计划尚未被接受。这一新计划必须先通过验证，然后才能被接受作为有效计划使用。

8) 现在，将优化程序模式重置为默认值，并禁用自动捕获计划基线。

```

SQL> alter session set optimizer_mode = all_rows;

Session altered.

SQL> alter session set optimizer_capture_sql_plan_baselines =
FALSE;

Session altered.

SQL>

```

9) 现在清除计划基线，并确认此 SQL 计划基线为空。使用 `purge_auto_baseline.sql` 执行此操作：

```

SQL> @purge_auto_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> exec :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_6fe28d438dfc352f');

PL/SQL procedure successfully completed.

SQL> exec :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_396e2e2b6cb3e334');
BEGIN :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_396e2e2b6cb3e334');
END;

```

# 练习 10-1：使用 SQL 计划管理（续）

```

*
ERROR at line 1:
ORA-38131: specified SQL handle SYS_SQL_396e2e2b6cb3e334 does
not exist
ORA-06512: at "SYS.DBMS_SPM", line 2524
ORA-06512: at line 1

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
2 from dba_sql_plan_baselines
3 where sql_text like 'select /*LOAD_AUTO*/%';

no rows selected

SQL>

```

- 10) 接下来，查看如何从 SQL 优化集加载计划。仍然在连接的 SQL\*Plus 会话中，检查以下 SQL 语句的执行计划，然后执行该计划（使用 explain\_query2.sql 和 query2.sql 脚本）：
- ```

select /*LOAD_STG*/ * from sh.sales where quantity_sold >
40 order by prod_id;

```

```

SQL>
SQL> @explain_query2
SQL> set echo on
SQL>
SQL> explain plan for
2   select /*LOAD_STG*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT
-----
Plan hash value: 3803407550

-----
| Id | Operation | Name |
-----
| 0 | SELECT STATEMENT | |
| 1 | SORT ORDER BY |
| 2 | PARTITION RANGE ALL |
| 3 | TABLE ACCESS FULL | SALES |
-----

```

练习 10-1：使用 SQL 计划管理（续）

```
10 rows selected.

SQL>
SQL> @query2
SQL> set echo on
SQL>
SQL> select /*LOAD_STS*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>
```

11) 将优化程序模式改为使用 FIRST_ROWS 优化，检查执行计划，然后重新执行该查询：

```
SQL>
SQL> alter session set optimizer_mode = first_rows;

Session altered.

SQL> @explain_query2
SQL> set echo on
SQL>
SQL> explain plan for
  2 select /*LOAD_STS*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT
-----
Plan hash value: 899219946

-----
| Id  | Operation                                | Name                |
|-----|-----|-----|
|  0  | SELECT STATEMENT                        |                     |
|  1  |   SORT ORDER BY                        |                     |
|  2  |     PARTITION RANGE ALL                 |                     |
|  3  |       TABLE ACCESS BY LOCAL INDEX ROWID| SALES                |
|  4  |         BITMAP CONVERSION TO ROWIDS     |                     |
|  5  |           BITMAP INDEX FULL SCAN        | SALES_PROMO_BIX     |
```

练习 10-1：使用 SQL 计划管理（续）

```

PLAN_TABLE_OUTPUT
-----
-----

12 rows selected.

SQL>
SQL> @query2
SQL> set echo on
SQL>
SQL> select /*LOAD_STS*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>

```

12) 将优化程序模式重置为 ALL_ROWS 优化：

```

SQL>
SQL> alter session set optimizer_mode = all_rows;

Session altered.

SQL>

```

13) 检查语句当前是否没有基线计划（使用 check_baselines2.sql 脚本）：

```

SQL> @check_baselines2
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_STS*/%';

no rows selected

SQL>

```

14) 创建一个 SQL 优化集来捕获包含 LOAD_STS 提示的 SELECT 语句。这些语句都在游标高速缓存中。该 STS 的名称为 SPM_STS，并归 SPM 用户所有。使用 catchup_sts.sql 脚本捕获这些语句。

```

SQL>
SQL> @catchup_sts
SQL> set echo on
SQL>
SQL> exec sys.dbms_sqltune.create_sqlset(sqlset_name =>
'SPM_STS', sqlset_owner => 'SPM');

```

练习 10-1：使用 SQL 计划管理（续）

```

PL/SQL procedure successfully completed.

SQL>
SQL> DECLARE
  2      stscur      dbms_sqltune.sqlset_cursor;
  3  BEGIN
  4      OPEN stscur FOR
  5          SELECT VALUE(P)
  6          FROM      TABLE(dbms_sqltune.select_cursor_cache(
  7                      'sql_text like 'select /*LOAD_STS*/%',
  8                      null, null, null, null, null, null, 'ALL'))
P;
  9
 10      -- populate the sqlset
 11      dbms_sqltune.load_sqlset(sqlset_name      =>
'SPM_STS',
 12                                populate_cursor => stscur,
 13                                sqlset_owner    => 'SPM');
 14  END;
 15  /

PL/SQL procedure successfully completed.

SQL>

```

15) 确认 SPM_STS 中包含哪些 SQL 语句（使用 check_sts.sql 脚本）：

a) 因此，SPM_STS 中的 SQL 语句有两个不同的计划。

```

SQL>
SQL> @check_sts
SQL> set echo on
SQL>
SQL> select sql_text from dba_sqlset_statements where
sqlset_name='SPM_STS';

SQL_TEXT
-----
select /*LOAD_STS*/ * from sh.sales where quantity_sold > 40
order by prod_id
select /*LOAD_STS*/ * from sh.sales where quantity_sold > 40
order by prod_id

SQL>

```

练习 10-1：使用 SQL 计划管理（续）

- 16) 使用 SPM_STS 中的计划填充计划基线资料档案库。使用 populate_baseline.sql 脚本执行此操作：

```
SQL>
SQL> @populate_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> execute :cnt := dbms_spm.load_plans_from_sqlset( -
>               sqlset_name => 'SPM_STS', -
>               basic_filter => 'sql_text like ''select
/*LOAD_STS*/%'');

PL/SQL procedure successfully completed.

SQL>
```

- 17) 确认已加载计划基线，并记下起始列中的值。会看到什么结果？（使用 check_baselines2.sql。）
- a) 因为是以手动方式加载这些计划，所以应看到 MANUAL-LOAD。另请注意两个计划此次都被接受。

```
SQL>
SQL> @check_baselines2
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_STS*/%';

SIGNATURE SQL_HANDLE
-----
SQL_TEXT
-----
-----
PLAN_NAME                                ORIGIN                                ENA ACC FIX AUT
-----
1.2134E+19 SYS_SQL_a8632bd857a4a25e
select /*LOAD_STS*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_ahstbv1bu98ky11df68d0 MANUAL-LOAD      YES YES NO  YES

1.2134E+19 SYS_SQL_a8632bd857a4a25e
select /*LOAD_STS*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_ahstbv1bu98ky54bc8843 MANUAL-LOAD      YES YES NO  YES
```


练习 10-1：使用 SQL 计划管理（续）

SIGNATURE	SQL_HANDLE				
-----	-----				
SQL_TEXT	-----				
-----	-----				
PLAN_NAME	ORIGIN	ENA	ACC	FIX	AUT
-----	-----	---	---	---	---
SQL>					

- 18) 清除计划基线，然后删除 SPM_STS。使用 `purge_sts_baseline.sql` 脚本执行此操作：

```
SQL>
SQL> @purge_sts_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> exec :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_a8632bd857a4a25e');

PL/SQL procedure successfully completed.

SQL>
SQL> print cnt;

          CNT
-----
          2

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
   2  from dba_sql_plan_baselines
   3  where sql_text like 'select /*LOAD_STS*/%';

no rows selected

SQL>
SQL> exec sys.dbms_sqltune.drop_sqlset(sqlset_name =>
'SPM_STS', -
>                                     sqlset_owner => 'SPM');

PL/SQL procedure successfully completed.

SQL>
```

练习 10-1：使用 SQL 计划管理（续）

19) 现在，将看到如何从游标高速缓存直接加载计划基线。开始之前，需要一些 SQL 语句。仍然在连接的 SQL*Plus 会话中，检查以下 SQL 语句的执行计划，然后执行该计划（使用 explain_query3.sql 和 query3.sql 脚本）：

```
select /*LOAD_CC*/ * from sh.sales where quantity_sold >
40 order by prod_id;
```

```
SQL>
SQL> @explain_query3
SQL> set echo on
SQL>
SQL> explain plan for
  2  select /*LOAD_CC*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT
-----
Plan hash value: 3803407550

-----
| Id | Operation                                | Name |
-----|-----|-----|
|  0 | SELECT STATEMENT                        |      |
|  1 |   SORT ORDER BY                        |      |
|  2 |     PARTITION RANGE ALL                |      |
|  3 |       TABLE ACCESS FULL              | SALES |
-----

10 rows selected.

SQL>
SQL> @query3
SQL> set echo on
SQL>
SQL> select /*LOAD_CC*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>
```

练习 10-1：使用 SQL 计划管理（续）

20) 现在，将优化程序模式更改为使用 FIRST_ROWS 优化，然后重新执行上一步骤。会看到什么结果？

a) 应看到一个不同的执行计划：

```
SQL>
SQL> alter session set optimizer_mode = first_rows;

Session altered.

SQL> @explain_query3
SQL> set echo on
SQL>
SQL> explain plan for
  2  select /*LOAD_CC*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT
-----
-----
Plan hash value: 899219946

-----
-
| Id  | Operation                                | Name |
|-----|-----|-----|
|  0  | SELECT STATEMENT                        |      |
|  1  |   SORT ORDER BY                        |      |
|  2  |    PARTITION RANGE ALL                  |      |
|  3  |     TABLE ACCESS BY LOCAL INDEX ROWID | SALES |
|  4  |      BITMAP CONVERSION TO ROWIDS        |      |
|  5  |       BITMAP INDEX FULL SCAN            | SALES_PROMO_BIX |

PLAN_TABLE_OUTPUT
-----
-----

12 rows selected.

SQL>
SQL> @query3
SQL> set echo on
```

练习 10-1：使用 SQL 计划管理（续）

```
SQL>
SQL> select /*LOAD_CC*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>
```

21) 将优化程序模式重置为 ALL_ROWS:

```
SQL>
SQL> alter session set optimizer_mode = all_rows;

Session altered.

SQL>
```

22) 现在已填充好游标高速缓存，需要为 SQL 语句获取 SQL ID。使用 SQL ID 筛选游标高速缓存的内容，然后加载具有这两个计划的基线。使用 load_cc_baseline.sql 脚本执行此操作：

```
SQL> @load_cc_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> variable sqlid varchar2(20);
SQL>
SQL> begin
  2   select distinct sql_id into :sqlid from v$sql where
sql_text like 'select /*LOAD_CC*/%';
  3   end;
  4   /

PL/SQL procedure successfully completed.

SQL>
SQL> print sqlid;

SQLID
-----
9qx77nb12bh6z

SQL>
SQL> execute :cnt :=
dbms_spm.load_plans_from_cursor_cache(sql_id => :sqlid);

PL/SQL procedure successfully completed.

SQL>
```

练习 10-1：使用 SQL 计划管理（续）

23) 确认基线已加载（使用 check_baselines3.sql）：

```

SQL>
SQL> @check_baselines3
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge from
dba_sql_plan_baselines where sql_text like 'select
/*LOAD_CC*/%';

SIGNATURE SQL_HANDLE
-----
SQL_TEXT
-----
-----
PLAN_NAME ORIGIN ENA ACC FIX AUT
-----
1.7783E+19 SYS_SQL_f6cb7f742ef93547
select /*LOAD_CC*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_gdkvzfhrhkda711df68d0 MANUAL-LOAD YES YES NO YES

1.7783E+19 SYS_SQL_f6cb7f742ef93547
select /*LOAD_CC*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_gdkvzfhrhkda754bc8843 MANUAL-LOAD YES YES NO YES

SIGNATURE SQL_HANDLE
-----
SQL_TEXT
-----
-----
PLAN_NAME ORIGIN ENA ACC FIX AUT
-----
-----
SQL>

```

24) 清除计划基线：

```

SQL>
SQL> @purge_cc_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> exec :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_f6cb7f742ef93547');

```

练习 10-1：使用 SQL 计划管理（续）

```

PL/SQL procedure successfully completed.

SQL>
SQL> print cnt;

          CNT
-----
          2

SQL>
SQL> REM Check that plan baselines were purged:
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
   2   from dba_sql_plan_baselines
   3   where sql_text like 'select /*LOAD_CC*/%';

no rows selected

SQL>

```

- 25) 既然已了解了如何捕获计划，下面就来看一下优化程序是如何选择要使用的计划基线的。首先，为一个语句创建两个基线，并将其显示为正在使用中。然后禁用其中一个基线，并查看正在使用中的另一个基线。最后禁用这两个基线，现在会看到优化程序回到基于成本的计划的默认行为。首先使用两个不同的计划执行同一查询。确定以下语句的执行计划，然后执行该计划（使用 `explain_query4.sql` 和 `query4.sql` 脚本）：
- ```

select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id

```

```

SQL>
SQL> @explain_query4
SQL> set echo on
SQL>
SQL> explain plan for
 2 select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT

Plan hash value: 3803407550

```

## 练习 10-1: 使用 SQL 计划管理 (续)

```

| Id | Operation | Name |

0	SELECT STATEMENT	
1	SORT ORDER BY	
2	PARTITION RANGE ALL	
3	TABLE ACCESS FULL	SALES

10 rows selected.

SQL>
SQL> @query4
SQL> set echo on
SQL>
SQL> select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>

```

26) 现在，将优化程序模式更改为使用 FIRST\_ROWS 优化，然后重新执行上一步骤。会看到什么结果？

a) 应看到一个不同的执行计划：

```

SQL>
SQL> alter session set optimizer_mode = first_rows;

Session altered.

SQL> @explain_query4
SQL> set echo on
SQL>
SQL> explain plan for
 2 select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from
table(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT

Plan hash value: 899219946

```

# 练习 10-1：使用 SQL 计划管理（续）

| Id | Operation                         | Name            |
|----|-----------------------------------|-----------------|
| 0  | SELECT STATEMENT                  |                 |
| 1  | SORT ORDER BY                     |                 |
| 2  | PARTITION RANGE ALL               |                 |
| 3  | TABLE ACCESS BY LOCAL INDEX ROWID | SALES           |
| 4  | BITMAP CONVERSION TO ROWIDS       |                 |
| 5  | BITMAP INDEX FULL SCAN            | SALES_PROMO_BIX |

PLAN\_TABLE\_OUTPUT

12 rows selected.

```

SQL>
SQL> @query4
SQL> set echo on
SQL>
SQL> select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id;

no rows selected

SQL>

```

27) 将优化程序模式重置为 ALL\_ROWS:

|                                                          |
|----------------------------------------------------------|
| SQL>                                                     |
| SQL> <b>alter session set optimizer_mode = all_rows;</b> |
| Session altered.                                         |
| SQL>                                                     |

- 28) 直接从游标高速缓存用语句的两个计划填充基线。使用 load\_use\_baseline.sql 脚本执行此操作。完成后，确认基线已得以加载。会看到什么结果？
- a) 应看到两个计划基线均已加载。请注意，这两个计划已标记为可接受。这是因为这两个计划在加载时位于游标高速缓存中，并且已手动进行了加载，所以认为这两个计划的性能是可接受的。

|                                   |
|-----------------------------------|
| SQL> @load_use_baseline           |
| SQL> set echo on                  |
| SQL>                              |
| SQL> variable cnt number;         |
| SQL>                              |
| SQL> variable sqlid varchar2(20); |



## 练习 10-1：使用 SQL 计划管理（续）

```
SQL>
SQL> begin
 2 select distinct sql_id into :sqlid from v$sql where
sql_text like 'select /*SPM_USE*/%';
 3 end;
 4 /

PL/SQL procedure successfully completed.

SQL>
SQL> print sqlid;

SQLID

gudfs2kqb90sz

SQL>
SQL> execute :cnt :=
dbms_spm.load_plans_from_cursor_cache(sql_id => :sqlid);

PL/SQL procedure successfully completed.

SQL>
SQL> print cnt;

 CNT

 2

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
 2 from dba_sql_plan_baselines
 3 where sql_text like 'select /*SPM_USE*/%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p11df68d0 MANUAL-LOAD YES YES NO YES
```

# 练习 10-1：使用 SQL 计划管理（续）

```

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p54bc8843 MANUAL-LOAD YES YES NO YES

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

SQL>

```

29) 确定用于执行以下查询的基线和执行计划：

```

select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id。

```

会看到什么结果？

- 说明性输出内容结尾的注释表明系统正在使用基线。在执行计划中，可以看到正在使用第一个基线（全表扫描）。使用 `explain_query4_note.sql` 脚本。

```

SQL>
SQL> @explain_query4_note
SQL> set echo on
SQL>
SQL> explain plan for
 2 select /*SPM_USE*/ * from sh.sales where quantity_sold >
40 order by prod_id;

Explained.

SQL>
SQL> select * from table(dbms_xplan.display(null,null,'basic
+note'));

PLAN_TABLE_OUTPUT

Plan hash value: 3803407550

| Id | Operation | Name |

0	SELECT STATEMENT	
1	SORT ORDER BY	
2	PARTITION RANGE ALL	
3	TABLE ACCESS FULL	SALES

```

## 练习 10-1：使用 SQL 计划管理（续）

```

PLAN_TABLE_OUTPUT

附注

- SQL plan baseline "SQL_PLAN_0p7c74s0ftt2p54bc8843" used
for this statement

14 rows selected.

SQL>

```

- 30) 禁用该计划基线，并检查系统在重新执行该语句时是否使用了另一个计划基线。使用 `check_baseline_used.sql` 脚本执行此操作。会看到什么结果？
- a) 现在，从该执行计划中，可以看到正在使用索引扫描而不是全表扫描，因此这次使用的是第二个基线。

```

SQL>
SQL>
SQL> @check_baseline_used
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> select sql_handle,plan_name
2 from dba_sql_plan_baselines
3 where sql_text like 'select /*SPM_USE*/%';

SQL_HANDLE PLAN_NAME

SYS_SQL_0a9d872600ece455 SQL_PLAN_0p7c74s0ftt2p11df68d0
SYS_SQL_0a9d872600ece455 SQL_PLAN_0p7c74s0ftt2p54bc8843

SQL>
SQL>
SQL> exec :cnt := dbms_spm.alter_sql_plan_baseline(-
> sql_handle =>
'SYS_SQL_0a9d872600ece455', -
> plan_name =>
'SQL_PLAN_0p7c74s0ftt2p54bc8843', -
> attribute_name => 'ENABLED', -
> attribute_value => 'NO');

PL/SQL procedure successfully completed.

```

# 练习 10-1：使用 SQL 计划管理（续）

```

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
 2 from dba_sql_plan_baselines
 3 where sql_text like 'select /*SPM_USE*/%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p11df68d0 MANUAL-LOAD YES YES NO YES

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p54bc8843 MANUAL-LOAD NO YES NO YES

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

SQL>
SQL>
SQL> explain plan for select /*SPM_USE*/ * from sh.sales where
quantity_sold > 40 order by prod_id;

Explained.

SQL>
SQL> select * from table(dbms_xplan.display(null, null, 'basic
+note'));

PLAN_TABLE_OUTPUT

Plan hash value: 899219946

```

## 练习 10-1：使用 SQL 计划管理（续）

| Id | Operation                         | Name            |
|----|-----------------------------------|-----------------|
| 0  | SELECT STATEMENT                  |                 |
| 1  | SORT ORDER BY                     |                 |
| 2  | PARTITION RANGE ALL               |                 |
| 3  | TABLE ACCESS BY LOCAL INDEX ROWID | SALES           |
| 4  | BITMAP CONVERSION TO ROWIDS       |                 |
| 5  | BITMAP INDEX FULL SCAN            | SALES_PROMO_BIX |

PLAN\_TABLE\_OUTPUT

-----

-

附注

-----

- SQL plan baseline "SQL\_PLAN\_0p7c74s0ftt2p1ldf68d0" used for this statement

16 rows selected.

SQL>

- 31) 禁用另一计划基线，并检查重新执行该语句的解释计划时系统是否退回到基于成本的方法。使用 `check_baseline_used2.sql` 脚本执行此操作：
- a) 该计划的末尾部分没有关于使用了基线的注释，由此可知，优化程序已退回到基于成本的默认方法。

```
SQL>
SQL> @check_baseline_used2
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> exec :cnt := dbms_spm.alter_sql_plan_baseline(-
> sql_handle =>
'SYS_SQL_0a9d872600ece455', -
> plan_name =>
'SQL_PLAN_0p7c74s0ftt2p1ldf68d0', -
> attribute_name => 'ENABLED', -
> attribute_value => 'NO');

PL/SQL procedure successfully completed.

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
```

# 练习 10-1：使用 SQL 计划管理（续）

```

2 from dba_sql_plan_baselines
3 where sql_text like 'select /*SPM_USE*/%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p11df68d0 MANUAL-LOAD NO YES NO YES

7.6492E+17 SYS_SQL_0a9d872600ece455
select /*SPM_USE*/ * from sh.sales where quantity_sold > 40
order by prod_id
SQL_PLAN_0p7c74s0ftt2p54bc8843 MANUAL-LOAD NO YES NO YES

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

SQL>
SQL>
SQL> explain plan for select /*SPM_USE*/ * from sh.sales where
quantity_sold > 40 order by prod_id;

Explained.

SQL>
SQL> select * from table(dbms_xplan.display(null, null, 'basic
+note'));

PLAN_TABLE_OUTPUT

Plan hash value: 3803407550

```

### 练习 10-1：使用 SQL 计划管理（续）

| Id | Operation           | Name  |
|----|---------------------|-------|
| 0  | SELECT STATEMENT    |       |
| 1  | SORT ORDER BY       |       |
| 2  | PARTITION RANGE ALL |       |
| 3  | TABLE ACCESS FULL   | SALES |

10 rows selected.

SQL>

32) 删除计划基线，然后检查这些基线是否已得到清除。使用 `purge_use_baseline.sql` 脚本执行此操作。

```
SQL>
SQL> @purge_use_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> exec :cnt :=
dbms_spm.drop_sql_plan_baseline('SYS_SQL_0a9d872600ece455');

PL/SQL procedure successfully completed.

SQL>
SQL> print cnt;

 CNT

 2

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
 2 from dba_sql_plan_baselines
 3 where sql_text like 'select /*SPM_USE*/%';

no rows selected

SQL>
```





练习 10-1：使用 SQL 计划管理（续）

```
PLAN_TABLE_OUTPUT

| 0 | SELECT STATEMENT | | 1 | 13 |
898 (2) | 00:00:1
1 | | |
| 1 | HASH GROUP BY | | 1 | 13 |
898 (2) | 00:00:1
1 | | |
| 2 | NESTED LOOPS | | 1 | 13 |
897 (2) | 00:00:1
1 | | |
|* 3 | TABLE ACCESS FULL | CUSTOMERS | 1 | 5 |
405 (1) | 00:00:0
5 | | |

PLAN_TABLE_OUTPUT

| 4 | PARTITION RANGE ALL | | 1 | 8 |
492 (3) | 00:00:0
6 | 1 | 28 |
|* 5 | TABLE ACCESS FULL | SALES | 1 | 8 |
492 (3) | 00:00:0
6 | 1 | 28 |

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

 3 - filter("C"."CUST_ID"<2)
 5 - filter("S"."CUST_ID"<2 AND "S"."CUST_ID"="C"."CUST_ID")

18 rows selected.

SQL>
SQL>
```

## 练习 10-1: 使用 SQL 计划管理（续）

34) 立即执行该语句，以便获取游标高速缓存中的计划并加载相应的计划基线。  
使用 load\_evolve\_baseline.sql 脚本执行此操作。会看到什么结果？

a) 可以看到当前计划已启用并被接受，但是尚未修复。

```
SQL>
SQL> @load_evolve_baseline
SQL> set echo on
SQL>
SQL> variable cnt number;
SQL>
SQL> variable sqlid varchar2(20);
SQL>
SQL> select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
 2 from sh.sales s, sh.customers c
 3 where s.cust_id = c.cust_id and c.cust_id < 2
 4 group by c.cust_id;

no rows selected

SQL>
SQL> begin
 2 select sql_id into :sqlid from v$sql where sql_text like
'select /*+ USE_NL(s c) FULL(s) FULL(c) */%';
 3 end;
 4 /

PL/SQL procedure successfully completed.

SQL>
SQL> execute :cnt :=
dbms_spm.load_plans_from_cursor_cache(sql_id => :sqlid);

PL/SQL procedure successfully completed.

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
 2 from dba_sql_plan_baselines
 3 where sql_text like 'select /*+ USE_NL(s c) FULL(s)
FULL(c) */%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

1.7750E+18 SYS_SQL_18a1ef14c17f5b75
```

**练习 10-1：使用 SQL 计划管理（续）**

```

select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
from s
SQL_PLAN_1j8gg2m0ryqvpdc5f94e5 MANUAL-LOAD YES YES NO YES

SQL>

```

- 35) 手动创建并执行 SQL 优化任务以优化您的语句。使用 `tune_evolve_sql.sql` 脚本执行此操作。

```

SQL>
SQL> @tune_evolve_sql
SQL> set echo on
SQL>
SQL> variable sqltext varchar2(4000);
SQL>
SQL> BEGIN
 2 :sqltext := q'# select /*+ USE_NL(s c) FULL(s) FULL(c)
/* c.cust_id, sum(s.quantity_sold)
 3 from sh.sales s, sh.customers c
 4 where s.cust_id = c.cust_id
 5 and c.cust_id < 2
 6 group by c.cust_id
 7 #';
 8 END;
 9 /

PL/SQL procedure successfully completed.

SQL>
SQL> variable spmtune varchar2(30);
SQL>
SQL> exec :spmtune := dbms_sqltune.create_tuning_task(sql_text
=> :sqltext);

PL/SQL procedure successfully completed.

SQL>
SQL> exec dbms_sqltune.execute_tuning_task(:spmtune);

PL/SQL procedure successfully completed.

SQL>
SQL>

```

## 练习 10-1：使用 SQL 计划管理（续）

36) 现在，已完成了优化任务，请运行报告并查看针对您的语句提出了哪些建议。会看到什么结果？

- a) 有以下两种建议：创建 SQL 概要文件或创建新索引。使用 report\_evolve\_tuning.sql 脚本。

```
SQL>
SQL> @report_evolve_tuning
SQL> set echo on
SQL>
SQL> set long 10000
SQL>
SQL> select dbms_sqltune.report_tuning_task(:spmtune,'TEXT')
from dual;

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

GENERAL INFORMATION SECTION

Tuning Task Name : TASK_163
Tuning Task Owner : SPM
Workload Type : Single SQL Statement
Scope : COMPREHENSIVE
Time Limit(seconds) : 1800
Completion Status : COMPLETED
Started at : 06/19/2009 14:54:47
Completed at : 06/19/2009 14:54:53

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Schema Name: SPM
SQL ID : dgs41cdr4t7q2
SQL Text : select /*+ USE_NL(s c) FULL(s) FULL(c) */
c.cust_id,
 sum(s.quantity_sold)
 from sh.sales s, sh.customers c
 where s.cust_id = c.cust_id
 and c.cust_id < 2
 group by c.cust_id
```

练习 10-1：使用 SQL 计划管理（续）

```
DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

FINDINGS SECTION (2 findings)

1- SQL Profile Finding (see explain plans section below)

 A potentially better execution plan was found for this
 statement.

 Recommendation (estimated benefit: 98.47%)

 - Consider accepting the recommended SQL profile. A SQL plan
 baseline

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

 corresponding to the plan with the SQL profile will also
 be created.
 execute dbms_sqltune.accept_sql_profile(task_name =>
'TASK_163',
 task_owner => 'SPM', replace => TRUE);

 Validation results

 The SQL profile was tested by executing both its plan and
 the original plan
 and measuring their respective execution statistics. A plan
 may have been
 only partially executed if the other could be run to
 completion in less time.

 Original Plan With SQL Profile %
Improved

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Completion Status: COMPLETE COMPLETE
Elapsed Time(us): 158435 217
99.86 %
CPU Time(us): 157576 100
99.93 %
User I/O Time(us): 0 0
Buffer Gets: 2541 39
```

## 练习 10-1：使用 SQL 计划管理（续）

```
98.46 %
Physical Read Requests: 0 0
Physical Write Requests: 0 0
Physical Read Bytes: 0 0
Physical Write Bytes: 0 0
Rows Processed: 0 0

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Fetches: 0 0
Executions: 1 1

Notes

1. The original plan was first executed to warm the buffer
cache.
2. Statistics for original plan were averaged over next 4
executions.
3. The SQL profile plan was first executed to warm the
buffer cache.
4. Statistics for the SQL profile plan were averaged over
next 9 executions.

2- Index Finding (see explain plans section below)

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

The execution plan of this statement can be improved by
creating one or more
indices.

Recommendation (estimated benefit: 90.98%)

- Consider running the Access Advisor to improve the
physical schema design
 or creating the recommended index.
 create index SH.IDX$$_00A30001 on SH.SALES("CUST_ID");

Rationale

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Creating the recommended indices significantly improves
the execution plan
 of this statement.However, it might be preferable to run
"Access Advisor"
```

练习 10-1：使用 SQL 计划管理（续）

```
using a representative SQL workload as opposed to a single
statement. This
will allow to get comprehensive index recommendations
which takes into
account index maintenance overhead and additional space
consumption.

EXPLAIN PLANS SECTION

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

1- Original With Adjusted Cost

Plan hash value: 4005616876

| Id | Operation | Name | Rows | Bytes |
Cost (%CPU)| Time
| Pstart| Pstop |

| 0 | SELECT STATEMENT | | 1 | 13 |
898 (2) | 00:00:1
DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

1 | | |
| 1 | HASH GROUP BY | | 1 | 13 |
898 (2) | 00:00:1
1 | | |
| 2 | NESTED LOOPS | | 1 | 13 |
897 (2) | 00:00:1
1 | | |
|* 3 | TABLE ACCESS FULL | CUSTOMERS | 1 | 5 |
405 (1) | 00:00:0
5 | | |
| 4 | PARTITION RANGE ALL | | 1 | 8 |
492 (3) | 00:00:0
6 | 1 | 28 |
|* 5 | TABLE ACCESS FULL | SALES | 1 | 8 |
492 (3) | 00:00:0
6 | 1 | 28 |
```

练习 10-1：使用 SQL 计划管理（续）

```
DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Predicate Information (identified by operation id):

 3 - filter("C"."CUST_ID"<2)
 5 - filter("S"."CUST_ID"<2 AND "S"."CUST_ID"="C"."CUST_ID")

2- Using SQL Profile

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

Plan hash value: 3070788227

Id	Operation	Name			
Rows	Bytes	Cost (%CPU)	Time	Pstart	Pstop
-----	-----				
0	SELECT STATEMENT				
1	13	55 (2)	00:00:01		
1	HASH GROUP BY				
1	13				

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

 55 (2) | 00:00:01 | | |
2	NESTED LOOPS			
1	13			
54 (0)	00:00:01			
3	PARTITION RANGE ALL			
1	8			
54 (0)	00:00:01	1	28	
4	TABLE ACCESS BY LOCAL INDEX ROWID	SALES		
1	8			
54 (0)	00:00:01	1	28	
5	BITMAP CONVERSION TO ROWIDS			
```





## 练习 10-1: 使用 SQL 计划管理 (续)

```

3	NESTED LOOPS			
1	13			
5	(0)	00:00:01		
* 4	INDEX RANGE SCAN	CUSTOMERS_PK		
1	5			
2	(0)	00:00:01		

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

|* 5 | INDEX RANGE SCAN | IDX$$_00A30001
| 1 | |
| 2 | (0) | 00:00:01 | | |
| 6 | TABLE ACCESS BY GLOBAL INDEX ROWID | SALES
| 1 | 8 |
| 3 | (0) | 00:00:01 | ROWID | ROWID |

Predicate Information (identified by operation id):

4 - access("C"."CUST_ID"<2)

DBMS_SQLTUNE.REPORT_TUNING_TASK(:SPMTUNE,'TEXT')

5 - access("S"."CUST_ID"="C"."CUST_ID")
 filter("S"."CUST_ID"<2)

SQL>
SQL>

```

37) 接受 SQL 优化指导建议的 SQL 概要文件。使用

`accept_evolve_baseline.sql` 脚本执行此操作。会发生什么情况？

a) 接受概要文件会导致为语句创建新的 SQL 概要文件和计划基线。现在会看到语句的两个基线。这两个基线均已启用并被接受。

注：一个是 MANUAL-LOAD，另一个是 MANUAL-SQLTUNE。

```

SQL>
SQL> @accept_evolve_baseline
SQL> set echo on
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
2 from dba_sql_plan_baselines
3 where sql_text like 'select /*+ USE_NL(s c) FULL(s)
FULL(c) */%';

```

## 练习 10-1：使用 SQL 计划管理（续）

```

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

1.7750E+18 SYS_SQL_18a1ef14c17f5b75
select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2
group by c.cust_id
SQL_PLAN_1j8gg2m0ryqvpdc5f94e5 MANUAL-LOAD YES YES NO YES

SQL>
SQL> exec dbms_sqltune.accept_sql_profile(task_name =>
:spmtune, name => 'SPM_SQL_PROF');

PL/SQL procedure successfully completed.

SQL>
SQL> select signature, category, name, sql_text from
dba_sql_profiles where name like 'SPM%';

SIGNATURE CATEGORY NAME

SQL_TEXT

1.7750E+18 DEFAULT SPM_SQL_PROF
select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2
group by c.cust_id

SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
2 from dba_sql_plan_baselines
3 where sql_text like 'select /*+ USE_NL(s c) FULL(s)
FULL(c) */%';

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

```

## 练习 10-1：使用 SQL 计划管理（续）

```

1.7750E+18 SYS_SQL_18a1ef14c17f5b75
select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2
group by c.cust_id
SQL_PLAN_1j8gg2m0ryqvpa10c1dcf MANUAL-SQLTUNE YES YES NO YES

SIGNATURE SQL_HANDLE

SQL_TEXT

PLAN_NAME ORIGIN ENA ACC FIX AUT

1.7750E+18 SYS_SQL_18a1ef14c17f5b75
select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2
group by c.cust_id
SQL_PLAN_1j8gg2m0ryqvpdc5f94e5 MANUAL-LOAD YES YES NO YES

SQL>

```

38) 确定执行语句时使用的计划。会看到什么结果？

- a) 下次执行查询时，查询将使用此计划基线 and 此 SQL 概要文件。使用 `explain_query5.sql` 脚本。

```

SQL>
SQL> @explain_query5
SQL> set echo on
SQL>
SQL> explain plan for
2 select /*+ USE_NL(s c) FULL(s) FULL(c) */ c.cust_id,
sum(s.quantity_sold)
3 from sh.sales s, sh.customers c
4 where s.cust_id = c.cust_id and c.cust_id < 2
5 group by c.cust_id;

Explained.

SQL>
SQL> select * from table(dbms_xplan.display(null, null, 'basic
+note'));

PLAN_TABLE_OUTPUT

Plan hash value: 3070788227

```

### 练习 10-1：使用 SQL 计划管理（续）

| Id                                                        | Operation                         | Name           |
|-----------------------------------------------------------|-----------------------------------|----------------|
| 0                                                         | SELECT STATEMENT                  |                |
| 1                                                         | HASH GROUP BY                     |                |
| 2                                                         | NESTED LOOPS                      |                |
| 3                                                         | PARTITION RANGE ALL               |                |
| 4                                                         | TABLE ACCESS BY LOCAL INDEX ROWID | SALES          |
| 5                                                         | BITMAP CONVERSION TO ROWIDS       |                |
| PLAN_TABLE_OUTPUT                                         |                                   |                |
| 6                                                         | BITMAP INDEX RANGE SCAN           | SALES_CUST_BIX |
| 7                                                         | INDEX UNIQUE SCAN                 | CUSTOMERS_PK   |
| 附注                                                        |                                   |                |
| -----                                                     |                                   |                |
| - SQL profile "SPM_SQL_PROF" used for this statement      |                                   |                |
| - SQL plan baseline "SQL_PLAN_1j8gg2m0ryqvpal0c1dcf" used |                                   |                |
| for this statement                                        |                                   |                |
| 19 rows selected.                                         |                                   |                |
| SQL>                                                      |                                   |                |
| SQL>                                                      |                                   |                |

39) 执行 cleanup\_spm.sql 脚本清除本练习的环境。

|                                                              |
|--------------------------------------------------------------|
| SQL>                                                         |
| SQL> @cleanup_spm                                            |
| SQL> set echo on                                             |
| SQL>                                                         |
| SQL> exec dbms_sqltune.drop_sql_profile(name =>              |
| 'SPM_SQL_PROF');                                             |
| PL/SQL procedure successfully completed.                     |
| SQL>                                                         |
| SQL> exec :cnt :=                                            |
| dbms_spm.drop_sql_plan_baseline('SYS_SQL_18a1ef14c17f5b75'); |
| PL/SQL procedure successfully completed.                     |

## 练习 10-1：使用 SQL 计划管理（续）

```
SQL>
SQL> select signature, sql_handle, sql_text, plan_name,
origin, enabled, accepted, fixed, autopurge
 2 from dba_sql_plan_baselines
 3 where sql_text like 'select /*+ USE_NL(s c) FULL(s)
FULL(c) */%';

no rows selected

SQL>
SQL> select signature, category, name, sql_text from
dba_sql_profiles where name like 'SPM%';

no rows selected

SQL>
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
$
```

## 第 11 课的练习

在此练习中，您将使用一个小型应用程序更新一个表，将该应用程序作为数据库重放会话的基础，从而将运行该练习所需的资源量降到最少。

## 练习 11-1：使用数据库重放

在此练习中，首先使用数据库重放捕获小型应用程序的工作量。然后自动重放该工作量，以确保获得的结果与最初执行时的结果相同。然后，重放工作量，但是这次是在异步模式下进行。

- 1) 在开始本练习之前，需要设置其环境。执行 `wcr_init_grid.sh` 脚本。该脚本调用名为 `wcr_demo_init_grid.sql` 的 SQL 脚本，创建新用户 JFV 和表 `wcr_grid`，后者代表一个 10x10 网格，其中每个单元都有颜色。首先，该脚本对网格进行初始化，将所有单元都设置为黑色。

```
$ cd /home/oracle/solutions/DBReplay
$./wcr_init_grid.sh

SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 11:24:50
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> drop user
jfv cascade
 *
ERROR at line 1:
ORA-01918: user 'JFV' does not exist

SQL> SQL> 2 3
User created.

SQL> SQL>
Grant succeeded.

SQL> SQL> Connected.
SQL> SQL> drop table wcr_grid purge
 *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL> 2 3 4 5 6 7
Table created.

SQL> SQL> 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27
PL/SQL procedure successfully completed.
```



## 练习 11-1：使用数据库重放（续）

```
SQL> SQL>
Commit complete.
```

```
SQL> SQL> SQL> Disconnected from Oracle Database 11g
Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 2) 为确保练习的设置是正确的，请使用 `wcr_printgrid.sh` 脚本打印网格的当前内容。会看到什么结果？

a) 应该看到一个 10x10 网格，其中填满了 b。

```
$./wcr_printgrid.sh

SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 11:26:35
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SET SERVEROUTPUT ON
SQL>
SQL> declare
 2 col varchar2(10);
 3 begin
 4 for i in 0..9 loop
 5 for j in 0..9 loop
 6 select substr(color,1,1) into col from wcr_grid where
xcoor=i and ycoor=j;
 7 dbms_output.put(col||' ');
 8 end loop;
 9 dbms_output.put_line('');
 10 end loop;
 11 end;
 12 /
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
```

## 练习 11-1：使用数据库重放（续）

```
b b b b b b b b b b
```

```
PL/SQL procedure successfully completed.
```

```
SQL> exit;
```

```
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
```

```
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```

```
$
```

- 3) 要使用数据库重放，需要创建一个操作系统目录，用于保存重放会话生成的所有文件。在您的系统上创建 /home/oracle/solutions/dbreplay 目录（不是 **DBReplay!**）。

```
$ rm -rf /home/oracle/solutions/dbreplay
$ mkdir -p /home/oracle/solutions/dbreplay
$
```

- 4) 还需创建一个 Oracle 目录对象，用于指向前面创建的操作系统目录。

```
$. oraenv
```

```
ORACLE_SID = [db10g] ? orcl
```

```
$ sqlplus system
```

```
SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 11:29:04
2009
```

```
Copyright (c) 1982, 2009, Oracle. All rights reserved.
```

```
Enter password: *****
```

```
Connected to:
```

```
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
```

```
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```

```
SQL> drop directory dbreplay;
```

```
drop directory dbreplay
```

```
*
```

```
ERROR at line 1:
```

```
ORA-04043: object DBREPLAY does not exist
```

```
SQL> create or replace directory dbreplay as
'/home/oracle/solutions/dbreplay';
```

```
Directory created.
```

```
SQL> exit
```

## 练习 11-1：使用数据库重放（续）

```
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

5) 下面是使用数据库重放的分步过程：

- a) 以 SYS 用户身份登录 Enterprise Manager。单击 “Software and Support（软件和支持）” 选项卡。
- b) 在 “Software and Support（软件和支持）” 子页上，单击 “Real Application Testing（实际应用程序测试）” 部分中的 “Database Replay（数据库重放）” 链接。
- c) 在 “Database Replay（数据库重放）” 页上，在 “Capture Workload（捕获工作量）” 行的 “Go to Task（转到任务）” 列中单击 “Capture Workload（捕获工作量）” 图标。
- d) 在 “Capture Workload: Plan environment（捕获工作量：计划环境）” 页中，确认已满足两个先决条件，然后单击 “Next（下一步）”。
- e) 在 “Capture Workload: Options（捕获工作量：选项）” 页上，确保在 “Database Restart Options（数据库重新启动选项）” 部分中选定了 “Do not restart the database prior to capture（在捕获之前不重新启动数据库）”，并确保 “Filter Mode（过滤模式）” 设置为 “Exclusion（排除）”，同时确保 Excluded Sessions 表包含 Oracle Management Service 和 Oracle Management Agent。单击 “Next（下一步）”。
- f) 在 “Capture Workload: Parameters（捕获工作量：参数）” 页上，指定 “Capture Name（捕获名称）”（例如，使用 capturegrid1）并在 “Capture Directory（捕获目录）” 中选择先前创建的目录 (DBREPLAY)。单击 “Next（下一步）”。
- g) 在 “Capture Workload: Schedule（捕获工作量：调度）” 页上，指定 “Job Name（作业名）”（例如，使用 capturegrid1\_job），并从 “Job Schedule（作业调度）” 部分中选择 “Immediately（立即执行）”。
- h) 在 “Job Credentials（作业身份证明）” 部分，为 “Host Credentials（主机身份证明）” 输入 oracle/oracle，然后选择 “Save as Preferred Credentials（另存为首选身份证明）”。单击 “Next（下一步）”。在 “Capture Workload: Review（捕获工作量：复查）” 页上，单击 “Submit（提交）”。
- i) 在 “Processing: Workload Capture（正在处理：工作量捕获）” 页上等待。
- j) 此时将转到 “View Workload Capture（查看工作量捕获）” 页，在该页中可看到捕获会话处于 “In Progress（正在进行）” 状态。

## 练习 11-1：使用数据库重放（续）

- k) 回到终端会话，执行 `wcr_demo_workload.sh` 脚本。此脚本生成可将网格表随机更新为不同颜色（红色、绿色、蓝色、黑色、黄色、橙色和橙红色）的数据库会话。此脚本的运行时间应该不超过两分钟。

```
[oracle@edrsr40p1-orcl DBReplay]$./wcr_demo_workload.sh
Mon Oct 5 11:38:48 UTC 2009
Connecting to the database...Connecting...
Connecting to the database...Connecting...
Connecting to the database...Connecting...
Connecting to the database...Connecting...
Connecting to the database...Connecting...
Connecting to the database...Connecting...
Connecting to the database...Connecting...
connected.
connected.
connected.
connected.
connected.
connected.
connected.
connected.
Done.
Done.
Done.
Done.
Done.
Done.
Mon Oct 5 11:39:56 UTC 2009
$
```

- l) 在脚本执行过程中，从另一个终端窗口查看  
`/home/oracle/solutions/dbreplay` 目录。应看到生成的文件。

```
$ ls -l /home/oracle/solutions/dbreplay
total 8
drwxr----- 2 oracle dba 4096 Oct 5 11:36 cap
drwxr----- 3 oracle dba 4096 Oct 5 11:36 capfiles
-rw-r----- 1 oracle dba 0 Oct 5 11:36 wcr_cap_00001.start
$
$ ls -l /home/oracle/solutions/dbreplay/capfiles
total 4
drwxr----- 12 oracle dba 4096 Oct 5 11:36 inst1
$
$ ls -l /home/oracle/solutions/dbreplay/capfiles/inst1
total 44
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:41 aa
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ab
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ac
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ad
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ae
```

## 练习 11-1：使用数据库重放（续）

```
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 af
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ag
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ah
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 ai
drwxr-xr-x 2 oracle dba 4096 Oct 5 11:36 aj
-rw-r--r-- 1 oracle dba 1198 Oct 5 11:36
wcr_cmqkrh0000000.rec
$
$ ls -l /home/oracle/solutions/dbreplay/capfiles/inst1/aa
total 980
-rw-r--r-- 1 oracle dba 982 Oct 5 11:36
wcr_cmqkth0000002.rec
-rw-r----- 1 oracle dba 1302 Oct 5 11:36
wcr_cmqkth0000003.rec
-rw-r--r-- 1 oracle dba 1200 Oct 5 11:36
wcr_cmqkth0000004.rec
-rw-r--r-- 1 oracle dba 1198 Oct 5 11:36
wcr_cmqkth0000005.rec
-rw-r--r-- 1 oracle dba 1198 Oct 5 11:36
wcr_cmqkuh0000006.rec
-rw-r--r-- 1 oracle dba 1199 Oct 5 11:36
wcr_cmqkvh0000007.rec
-rw-r--r-- 1 oracle dba 1198 Oct 5 11:37
wcr_cmqkzh0000008.rec
-rw-r--r-- 1 oracle dba 988 Oct 5 11:37
wcr_cmqkzh0000009.rec
-rw-r--r-- 1 oracle dba 1210 Oct 5 11:37
wcr_cmqm5h000000c.rec
-rw-r--r-- 1 oracle dba 982 Oct 5 11:37
wcr_cmqm9h000000d.rec
-rw-r--r-- 1 oracle dba 1774 Oct 5 11:38
wcr_cmqpjh000000h.rec
-rw-r--r-- 1 oracle dba 1293 Oct 5 11:38
wcr_cmqpph000000j.rec
-rw-r--r-- 1 oracle dba 130023 Oct 5 11:39
wcr_cmqqah000000n.rec
-rw-r--r-- 1 oracle dba 129680 Oct 5 11:39
wcr_cmqqah000000p.rec
-rw-r--r-- 1 oracle dba 130284 Oct 5 11:39
wcr_cmqqbh000000q.rec
-rw-r--r-- 1 oracle dba 130319 Oct 5 11:39
wcr_cmqqch000000r.rec
-rw-r--r-- 1 oracle dba 130612 Oct 5 11:39
wcr_cmqqfh000000s.rec
-rw-r--r-- 1 oracle dba 130616 Oct 5 11:39
wcr_cmqqfh000000t.rec
-rw-r--r-- 1 oracle dba 130619 Oct 5 11:39
wcr_cmqqfh000000u.rec
-rw-r--r-- 1 oracle dba 2973 Oct 5 11:41
wcr_cmqubh000000x.rec
```

## 练习 11-1：使用数据库重放（续）

```
-rw-r--r-- 1 oracle dba 2790 Oct 5 11:41
wcr_cmquch000000y.rec
$
$ ls -l /home/oracle/solutions/dbreplay/cap
total 4
-rw-r----- 1 oracle dba 206 Oct 5 11:36 wcr_scapture.wmd
$
```

- m) 等待 K 步骤的工作量全部得以执行。
- n) 返回 Enterprise Manager 的“View Workload Capture（查看工作量捕获）”页，单击“Stop Capture（停止捕获）”。
- o) 在“Confirmation（确认）”页上，单击“Yes（是）”。
- p) 在“Processing: Workload Capture（正在处理：停止捕获）”页上等待。
- q) 出现“Export AWR Data（导出 AWR 数据）”页时，单击“No（否）”。
- r) 返回“View Workload Capture（查看工作量捕获）”页，此时应看到刷新后的统计信息，其中包括：Database Time（数据库时间）、Average Active Sessions（平均活动会话）、User Calls（用户调用）、Transactions（事务处理）、Session Logins（会话登录）及 Application Errors（应用程序错误）。记下这些统计信息（或单击“View Workload Capture Report（查看工作量捕获报表）”。由于 Firefox 会阻止弹出窗口，因此请单击“Preferences（首选项）”并允许弹出窗口，然后单击“Save to File（保存到文件）”生成 HTML 报表），然后单击“OK（确定）”。此时将返回“Database Replay（数据库重放）”页。
- s) 在终端会话中，通过执行 wcr\_printgrid.sh 脚本查看工作量结果。记下网格中生成的颜色。稍后将用这些颜色确认重放会话是否正常工作。确保保留会话输出，或执行 CREATE TABLE modified\_grid AS SELECT \* FROM jfv.wcr\_grid;
- t) 返回“Database Replay（数据库重放）”页，在“Process Workload task（处理工作量任务）”的“Go to Task（转到任务）”列中单击“Preprocess Workload（预处理的工作量）”图标。
- u) 在“Preprocess Captured Workload（预处理捕获的工作量）”页上，选择捕获目录 (DBREPLAY)。现在，在该页上应能看到处于“Completed（完成）”状态的“Capture Summary（捕获概要）”部分。单击“Preprocess Workload（预处理工作量）”。
- v) 在“Preprocess Captured Workload: Database Version（预处理捕获的工作量：数据库版本）”页上，单击“Next（下一步）”。

## 练习 11-1：使用数据库重放（续）

- w) 在 “Preprocess Captured Workload: Schedule（预处理捕获的工作量：调度）” 页上，输入作业名称（例如，`process_capturegrid1`），确保选中了 “Immediately（立即执行）”，如有必要，还要指定 “Host Credentials（主机身份证明）” (`oracle/oracle`)。完成后，单击 “Next（下一步）”。
- x) 在 “Preprocess Captured Workload: Review（预处理捕获的工作量：复查）” 页上，单击 “Submit（提交）”。
- y) 此时将返回 “Database Replay（数据库重放）” 页，该页包含一些确认信息。
- z) 等待一两分钟，然后单击 “Confirmation（确认）” 部分中的 “View Job（查看作业）”。
- aa) 此时将转到相应的 “Execution（执行）” 页，该页中应显示作业已成功。在浏览器上单击 “Back（返回）” 箭头以返回到 “Database Replay（数据库重放）” 页。
- bb) 返回到第二个终端会话，在  
`/home/oracle/solutions/dbreplay/pp11.2.0.1.0` 目录中查看生成的文件。应看到与您刚刚执行的过程阶段相对应的其它文件（其他 `wmd`、`.extb` 和 `.pp` 文件）。
- cc) 需要将环境还原为开始捕获之前的状态，才能重放以前捕获的工作量。在本次练习中，只需再次将网格表初始化为黑色。因此，返回到第一个终端会话，执行 `wcr_init_grid_table.sh` 脚本。
- dd) 完成后，确保使用 `wcr_printgrid.sh` 脚本以黑色正确地对网格表进行了初始化。
- ee) 返回 Enterprise Manager 的 “Database Replay（数据库重放）” 页，在 “Replay Workload Task（重放工作量任务）” 的 “Go to Task（转到任务）” 列中选择 “Replay Workload（重放工作量）” 图标。
- ff) 在 “Replay Workload（重放工作量）” 页上，选择重放目录 (`DBREPLAY`)。该页上应显示状态为 “Completed（完成）” 的 “Capture Summary（捕获概要）” 部分。请注意，此时 Replay History 表应该为空。单击 “Set Up Replay（设置重放）”。
- gg) 在 “Replay Workload: Prerequisites（重放工作量：先决条件）” 页上，完成此部分后便可安全地单击 “Continue（继续）”。
- hh) 在 “Replay Workload: References to External Systems（重放工作量：对外部系统的引用）” 页上，完成此部分后便可安全地单击 “Continue（继续）”。

## 练习 11-1：使用数据库重放（续）

- ii) 在 “Replay Workload: Choose Initial Options（重放工作量：选择初始选项）” 页上，输入重放名称（例如，`replay_capturegrid1`），然后确保选中了 “Use the default replay options（使用默认重放选项）”。单击 “Next（下一步）”。
- jj) 在 “Replay Workload: Customize Options（重放工作量：定制选项）” 页上，单击 “Replay Parameters（重放参数）” 选项卡。
- kk) 在 “Replay Parameters（重放参数）” 子页上，确保同步值已设置为 SCN。单击 “Next（下一步）”。
- ll) 在 “Replay Workload: Prepare Replay Clients（重放工作量：准备重放客户机）” 页上，单击 “Next（下一步）”。
- mm) 此时显示 “Replay Workload: Wait for Client Connections（重放工作量：等待客户机连接）” 页。系统正在等待您启动重放客户机。
- nn) 在第二个终端会话中，在操作系统提示符下输入以下命令：

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ wrc REPLAYDIR=/home/oracle/solutions/dbreplay USERID=system
PASSWORD=oracle_4U

Workload Replay Client: Release 11.2.0.1.0 - Production on Mon
Oct 5 12:07:09 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

Wait for the replay to start (12:07:09)
```

- oo) 在第三个终端会话中，在操作系统提示符下输入以下命令：

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ wrc REPLAYDIR=/home/oracle/solutions/dbreplay USERID=system
PASSWORD=oracle_4U

Workload Replay Client: Release 11.2.0.1.0 - Production on Mon
Oct 5 12:08:24 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

Wait for the replay to start (12:08:24)
```



## 练习 11-1：使用数据库重放（续）

- pp) wrc 可执行文件现在正等待您从数据库启动重放操作。
- qq) 返回 “Replay Workload: Wait for Client Connections（重放工作量：等待客户机连接）” 页，在此页中应看到 “Client Connections（客户机连接）” 表有两行内容。这些行对应于您刚启动的客户机会话。单击 “Next（下一步）”。
- rr) 在 “Replay Workload: Review（重放工作量：复查）” 页上，单击 “Submit（提交）” 为 capturegrid1 启动重放会话。此时将转到 “View Workload Replay（查看工作量重放）” 页。
- ss) 返回到第二个和第三个终端会话，您应看到重放客户机已启动了重放。
- tt) 返回 Enterprise Manger 的 “View Workload Replay（查看工作量重放）” 页，等待重放完成。
- uu) 完成后，可以在 “View Workload Replay（查看工作量重放）” 页上查看生成的统计信息。请查看 “Detailed Comparison（详细比较）” 部分。（如果要查看此信息，请单击 “+” 图标。）单击 “OK（确定）”，返回到 “Database Replay（数据库重放）” 页。
- vv) 在第二个和第三个终端窗口中，客户机会话也应已完成。
- ww) 在第一个终端会话中，执行 wcr\_printgrid.sh 脚本以在重放完成后查看网格表。应该确保这些值与首次应用工作量时看到的值完全相同。这是因为重放会话是使用同步模式完成的。

```
$./wcr_printgrid.sh

SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 12:23:26
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SET SERVEROUTPUT ON
SQL>
SQL> declare
 2 col varchar2(10);
 3 begin
 4 for i in 0..9 loop
 5 for j in 0..9 loop
 6 select substr(color,1,1) into col from wcr_grid where
xcoor=i and ycoor=j;
 7 dbms_output.put(col||' ');
```

## 练习 11-1：使用数据库重放（续）

```

 8 end loop;
 9 dbms_output.put_line('');
 10 end loop;
 11 end;
 12 /
y r g r s B B r b g
o B b s r g B g s B
b o y B g g y b o g
o y y s o r g y r o
b r g r g g s o s o
s s o r B y y s o r
o o B g g s g y b b
B y y g b B B B B y
s o r r B s B s B o
y y r g g s g o r r

PL/SQL procedure successfully completed.

SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

## 练习 11-2: 使用数据库回放 — 异步模式

在本练习中，您将在异步模式下重放工作量。应看到结果网格被设置成了不同的颜色。

- 1) 需要将环境还原为开始捕获之前的状态，才能重放以前捕获的工作量。在本次练习中，只需再次将网格表初始化为黑色。在第一个终端会话中，执行 `wcr_init_grid_table.sh` 脚本。

```
$./wcr_init_grid_table.sh
SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 15:41:08
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> Connected.
SQL> SQL>
100 rows deleted.

SQL> SQL>
Commit complete.

SQL> SQL> 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22 23 24
25 26 27
PL/SQL procedure successfully completed.

SQL> SQL>
Commit complete.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 11-2：使用数据库回放 — 异步模式（续）

- 2) 完成后，确保使用 wcr\_printgrid.sh 脚本以黑色正确地对网格表进行了初始化。

```
$./wcr_printgrid.sh
SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 15:42:35
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> SET SERVEROUTPUT ON
SQL>
SQL> declare
 2 col varchar2(10);
 3 begin
 4 for i in 0..9 loop
 5 for j in 0..9 loop
 6 select substr(color,1,1) into col from wcr_grid where
xcoor=i and ycoor=j;
 7 dbms_output.put(col||' ');
 8 end loop;
 9 dbms_output.put_line('');
 10 end loop;
 11 end;
 12 /
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b
b b b b b b b b b b

PL/SQL procedure successfully completed.

SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 11-2: 使用数据库回放 — 异步模式 (续)

- 3) 在 Enterprise Manager 的 “Database Replay (数据库重放)” 页上, 在 “Replay Workload (重放工作量)” 行的 “Go to Task (转到任务)” 列中单击 “Replay Workload (重放工作量)” 图标。
- 4) 在 “Replay Workload (重放工作量)” 页上, 选择重放目录 (DBREPLAY)。该页上应显示状态为 “Completed (完成)” 的 “Capture Summary (捕获概要)” 部分。请注意, Replay History 表应包含两个条目。
- 5) 单击 “Set Up Replay (设置重放)”。
- 6) 在 “Replay Workload: Prerequisites (重放工作量: 先决条件)” 页上, 完成此部分后便可安全地单击 “Continue (继续)”。
- 7) 在 “Replay Workload: References to External Systems (重放工作量: 对外部系统的引用)” 页上, 在完成此部分后, 同样可以安全地单击 “Continue (继续)”。
- 8) 在 “Replay Workload: Choose Initialize Options (重放工作量: 选择初始选项)” 页上, 输入重放名称 (例如, replay\_capturegrid3), 然后确保选中了 “Use the default replay options (使用默认重放选项)”。单击 “Next (下一步)”。
- 9) 在 “Replay Workload: Customize Options (重放工作量: 定制选项)” 页上, 单击 “Replay Parameters (重放参数)” 选项卡。
- 10) 在 “Replay Parameters (重放参数)” 子页上, 确保同步值已设置为 **OFF**。单击 “Next (下一步)”。
- 11) 在 “Replay Workload: Prepare Replay Clients (重放工作量: 准备重放客户机)” 页上, 单击 “Next (下一步)”。
- 12) 系统正在等待您启动重放客户机。
- 13) 在第二个终端会话中, 在操作系统提示符下输入以下命令:

```
$ wrc REPLAYDIR=/home/oracle/solutions/dbreplay USERID=system
PASSWORD=oracle_4U
Workload Replay Client: Release 11.2.0.1.0 - Production on Mon
Oct 5 15:47:33 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

Wait for the replay to start (15:47:33)
```

## 练习 11-2：使用数据库回放 — 异步模式（续）

14) 在第三个终端会话中，在操作系统提示符下输入以下命令：

```
$ wrc REPLAYDIR=/home/oracle/solutions/dbreplay USERID=system
PASSWORD=oracle_4U
Workload Replay Client: Release 11.2.0.1.0 - Production on Mon
Oct 5 15:48:20 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

Wait for the replay to start (15:48:20)
```

15) 该命令现在正等待您从数据库启动重放操作。

16) 返回 “Replay Workload: Wait for Client Connections（重放工作量：等待客户机连接）” 页，应看到 “Client Connections（客户机连接）” 表中有两行。这些行对应于刚启动的客户机会话。单击 “Next（下一步）”。

17) 在 “Replay Workload: Review（重放工作量：复查）” 页上，单击 “Submit（提交）” 为 capturegrid1 启动重放会话。

18) 此时将转到 “View Workload Replay（查看工作量重放）” 页。

19) 返回到第二个和第三个终端会话，您应看到重放客户机已启动了重放。

20) 返回到 “View Workload Replay（查看工作量重放）” 页，等待重放完成。

21) 在第二个和第三个终端窗口中，客户机会话也应已完成。

22) 在第一个终端会话中，执行 wcr\_printgrid.sh 脚本以在重放完成后查看网格表。您会看到与原始工作量结果存在一些差别。这是因为重放会话是以异步模式完成的。

```
$./wcr_printgrid.sh
SQL*Plus: Release 11.2.0.1.0 Production on Mon Oct 5 15:59:28
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> SET SERVEROUTPUT ON
SQL>
SQL> declare
 2 col varchar2(10);
 3 begin
 4 for i in 0..9 loop
```

## 练习 11-2：使用数据库回放 — 异步模式（续）

```

5 for j in 0..9 loop
6 select substr(color,1,1) into col from wcr_grid where
xcoor=i and ycoor=j;
7 dbms_output.put(col||' ');
8 end loop;
9 dbms_output.put_line('');
10 end loop;
11 end;
12 /
r g g y B o b y o o
b o o o s y g B y s
g s r b o y r s r o
o y o s s s r y y y
s B y o s s y o r o
s o o y b o y s g y
o o b b y s y g r r
s o o s r o y s g B
g y y b s b y y g o
r b y g g s o B r s

PL/SQL procedure successfully completed.

SQL> exit;
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
$

```

## 第 12 课的练习

默认情况下，会在每个夜间维护窗口期间自动执行自动 SQL 优化。在本练习中，您将模拟自动 SQL 优化的执行过程，并研究其结果。



## 练习 12-1：使用自动 SQL 优化

在本练习中，您需手动启动自动 SQL 优化来自动优化小型应用程序工作量。然后，调查结果和配置可能性。

- 1) 导航至 Enterprise Manager 的“Server（服务器）”页。单击“Oracle Scheduler（Oracle 调度程序）”部分中的“Automated Maintenance Tasks（自动维护任务）”。确认“Status（状态）”设置为“Enabled（启用）”单击“Configure（配置）”。单击“Automatic SQL Tuning（自动 SQL 优化）”旁边的“Configure（配置）”按钮。对于“Automatic Implementation of SQL Profiles（自动实施 SQL 概要文件）”，选择“Yes（是）”。然后，单击“Apply（应用）”。
- 2) 在以 oracle 用户身份连接的终端窗口中执行 ast\_setup.sh 脚本。此脚本用于创建本练习中所用的 AST 用户，关闭自动维护任务，并删除由 AST 用户执行的查询上的任何现有概要文件。

```
$ cd /home/oracle/solutions/AST
$. oraenv
ORACLE_SID = [db10g] ? orcl
$
$./ast_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:16:32
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> drop user ast cascade
*
ERROR at line 1:
ORA-01918: user 'AST' does not exist

SQL> SQL>
User created.

SQL> SQL>
Grant succeeded.

SQL> SQL>
System altered.
```

## 练习 12-1：使用自动 SQL 优化（续）

```

SQL> SQL> SQL> SQL> SQL> SQL>
System altered.

SQL> SQL> SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL> SQL> SQL> 2 3 4 5 6 7
8 9
PL/SQL procedure successfully completed.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 3) 您应执行一定的工作量，为本练习做准备。执行 `run_workload_stream.sh` 脚本。此脚本将多次执行未正确优化的查询。存在问题的查询使用提示来强制优化程序选择非最佳的执行计划。此脚本的执行时间大约为 30 到 60 秒。

```

$./run_workload_stream.sh
Tue Oct 6 07:18:33 UTC 2009

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:18:33
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
no rows selected

SQL>
no rows selected

SQL>
no rows selected
...

SQL>
no rows selected

```

## 练习 12-1：使用自动 SQL 优化（续）

```
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Tue Oct 6 07:19:01 UTC 2009
$
```

- 4) 自动 SQL 优化是使用在维护窗口期间运行的自动任务实现的。但是，您不打算等待下一个维护窗口打开，因为这可能会花费很长时间。您改为立即强制打开下一个维护窗口。这将自动触发自动 SQL 优化任务。执行 run\_ast.sh 脚本来完成该操作。执行此脚本会花费几分钟的时间。

```
$./run_ast.sh
Tue Oct 6 07:21:08 UTC 2009

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:21:08
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL> 2 3 4
PL/SQL procedure successfully completed.

SQL> SQL>
WINDOW

TUESDAY_WINDOW

SQL> SQL> SQL> SQL> SQL> SQL> 2
System altered.

SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL>
PL/SQL procedure successfully completed.
```

## 练习 12-1：使用自动 SQL 优化（续）

```

SQL> SQL> SQL> SQL> SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> 2 3 4 5 6 7 8 9 10 11
12 13 14 15 16 17 18 19 20 21 22
PL/SQL procedure successfully completed.

SQL> SQL> 2
System altered.

SQL> SQL> SQL> SQL> SQL> SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL> >
PL/SQL procedure successfully completed.

SQL> SQL> SQL> Disconnected from Oracle Database 11g
Enterprise Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Tue Oct 6 07:32:16 UTC 2009
$

```

- 5) 再次执行 run\_workload\_stream.sh 脚本。会看到什么结果？
- a) 您可以看出，run\_workload\_stream.sh 的执行时间比原始执行要快得多。这可能是由于自动 SQL 优化已自动为语句实施了概要文件所造成的。

```

$./run_workload_stream.sh
Tue Oct 6 07:35:15 UTC 2009

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:35:15
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
no rows selected

SQL>
no rows selected
...
SQL>
no rows selected

```

## 练习 12-1：使用自动 SQL 优化（续）

```
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Tue Oct 6 07:35:15 UTC 2009
$
```

### 6) 强制创建 AWR 快照。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
$ sqlplus ast/ast

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:37:45
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> set echo on

SQL>
SQL> exec dbms_workload_repository.create_snapshot;

PL/SQL procedure successfully completed.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 12-1：使用自动 SQL 优化（续）

- 7) 如何确认已自动实施了 SQL 概要文件？
  - a) 在 Oracle Enterprise Manager 中，在 “Server > Automated Maintenance Tasks > Automatic SQL Tuning（服务器 > 自动维护任务 > 自动 SQL 优化）” 下找到 “Automatic SQL Tuning（自动 SQL 优化）” 概要页。此任务已在某个维护窗口中运行，并且已生成了可供查看的结果。
  - b) 查看优化结果。
  - c) 查看 “Automatic SQL Tuning Result Summary（自动 SQL 优化结果概要）” 页上的图形。
  - d) 重点了解饼图及其旁边的条形图。您应能够对一般查找结果包含的项及该任务实施的 SQL 概要文件数量有个感性的认识。
  - e) 单击 “View Report（查看报告）” 查看 SQL 级报告的详细内容。查找并选择在 AST 方案中运行的 SQL。注意，绿色的复选标记表明该概要文件已实施。
  - f) 单击 “View Recommendations（查看建议）” 按钮。
  - g) 单击 SQL 概要文件条目的 “Compare Explain Plans（比较解释计划）” 眼镜图标。
  - h) 查看查询的旧解释计划和新解释计划。
  - i) 然后单击 “Recommendations for SQL ID（以下 SQL ID 的建议）” 定位器链接返回上一页。
  - j) 查看 SQL 概要文件。当仍处在 “Recommendations for SQL\_ID（以下 SQL ID 的建议）” 页时，单击 SQL 文本转至该 SQL 的 “SQL Details（SQL 详细资料）” 页。
  - k) 此时将显示 “Tuning History（优化历史记录）” 选项卡页。请注意，显示的 SYS\_AUTO\_SQL\_TUNING\_TASK 链接表明此优化任务已优化了该 SQL 语句。
  - l) 查看 “Plan Control（计划控制）” 子页，并注意已自动为此 SQL 创建了概要文件。AUTO 类型表明概要文件是自动创建的。
  - m) 单击 “Statistics（统计信息）” 选项卡查看该 SQL 的执行历史记录。
  - n) 由于计算机速度的原因，您也许无法看到两个散列值。如果出现这种情况，请忽略此步骤和下一步骤。选择 “Real Time: Manual Refresh（实时：手动刷新）”，然后从相应的下拉列表中依次选择可能出现的两个 “Plan Hash Values（计划散列值）” 值。逐个选择散列值，每次选择后都要等待页面进行刷新。

# 练习 12-1：使用自动 SQL 优化（续）

- o) 根据环境的速度，应会看到一个每次执行的占用时间相对较高的语句和一个每次执行的占用时间非常低的语句。此时将显示改进的计划。如果从“Plan Hash Values（计划散列值）”下拉列表中选择“All（所有）”，则在“Summary（概要）”图上优化语句后，可能无法看到相应语句的执行。这可能是由于工作量的执行所用时间太短的缘故。
- 8) 生成文本报表以获得更详细的信息。在命令行中执行 `get_task_report.sh` 脚本。会看到什么结果？
  - a) 请注意，首先是从指导方案中提取执行名称和对象编号的第一批查询，然后是获得文本报表的最终查询。在文本报表中，查看关于 SQL 概要文件查找结果的部分并认真阅读“Validation Results（验证结果）”部分。这样您可以看到执行测试期间的执行统计信息，因此可对概要文件的质量有更深入的了解。还可使用 `report_auto_tuning_task` API 获得多次执行该任务所生成的多个报表。

```
$./get_task_report.sh

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 07:50:20
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL>
Session altered.

SQL> SQL> 2 3 4
EXECUTION_NAME STATUS EXECUTION_START

EXEC_298 COMPLETED 10/06/2009 07:21:18

SQL> SQL> SQL> SQL> 2 3 4 5 6 7
PL/SQL procedure successfully completed.

SQL> SQL>
LAST_EXEC

EXEC_298

SQL> SQL> SQL> SQL> SQL> SQL> SQL> SQL> 2 3 4 5
6 7 8 9 10
PL/SQL procedure successfully completed.
```

## 练习 12-1：使用自动 SQL 优化（续）

```

SQL> SQL>
 OBJ_ID

 11

SQL> SQL> SQL> SQL> SQL> SQL> 2 3 GENERAL INFORMATION
SECTION

Tuning Task Name :
SYS_AUTO_SQL_TUNING_TASK
Tuning Task Owner : SYS
Workload Type : Automatic High-Load
SQL Workload
Scope : COMPREHENSIVE
Global Time Limit(seconds) : 3600
Per-SQL Time Limit(seconds) : 1200
Completion Status : COMPLETED
Started at : 10/06/2009 07:21:18
Completed at : 10/06/2009 07:31:36
Number of Candidate SQLs : 42
Cumulative Elapsed Time of SQL (s) : 369

Object ID : 11
Schema Name : AST
SQL ID : by9m5m597zh19
SQL Text : select /*+ USE_NL(s c) FULL(s) FULL(c) AST */
c.cust_id,sum(s.quantity_sold) from sh.sales s, sh.customers c
where s.cust_id = c.cust_id and c.cust_id < 2 group by
c.cust_id

FINDINGS SECTION (2 findings)

1- SQL Profile Finding (see explain plans section below)

A potentially better execution plan was found for this
statement.
SQL profile "SYS_SQLPROF_012428bb9c6e0000" was created
automatically for this statement.

Recommendation (estimated benefit: 98.47%)

- An automatically-created SQL profile is present on the
system.
 Name: SYS_SQLPROF_012428bb9c6e0000
 Status: ENABLED

```



## 练习 12-1：使用自动 SQL 优化（续）

|                                                                                                                                                                                                                                     |               |                  |            |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------|------------------|------------|
| Validation results                                                                                                                                                                                                                  |               |                  |            |
| -----                                                                                                                                                                                                                               |               |                  |            |
| The SQL profile was tested by executing both its plan and the original plan and measuring their respective execution statistics. A plan may have been only partially executed if the other could be run to completion in less time. |               |                  |            |
|                                                                                                                                                                                                                                     | Original Plan | With SQL Profile | % Improved |
|                                                                                                                                                                                                                                     | -----         | -----            | -----      |
| Completion Status:                                                                                                                                                                                                                  | COMPLETE      | COMPLETE         |            |
| Elapsed Time(us):                                                                                                                                                                                                                   | 184526        | 194              | 99.89 %    |
| CPU Time(us):                                                                                                                                                                                                                       | 183172        | 200              | 99.89 %    |
| User I/O Time(us):                                                                                                                                                                                                                  | 0             | 0                |            |
| Buffer Gets:                                                                                                                                                                                                                        | 2541          | 39               | 98.46 %    |
| Physical Read Requests:                                                                                                                                                                                                             | 0             | 0                |            |
| Physical Write Requests:                                                                                                                                                                                                            | 0             | 0                |            |
| Physical Read Bytes:                                                                                                                                                                                                                | 0             | 0                |            |
| Physical Write Bytes:                                                                                                                                                                                                               | 0             | 0                |            |
| Rows Processed:                                                                                                                                                                                                                     | 0             | 0                |            |
| Fetches:                                                                                                                                                                                                                            | 0             | 0                |            |
| Executions:                                                                                                                                                                                                                         | 1             | 1                |            |
| Notes                                                                                                                                                                                                                               |               |                  |            |
| -----                                                                                                                                                                                                                               |               |                  |            |
| 1. The original plan was first executed to warm the buffer cache.                                                                                                                                                                   |               |                  |            |
| 2. Statistics for original plan were averaged over next 4 executions.                                                                                                                                                               |               |                  |            |
| 3. The SQL profile plan was first executed to warm the buffer cache.                                                                                                                                                                |               |                  |            |
| 4. Statistics for the SQL profile plan were averaged over next 9 executions.                                                                                                                                                        |               |                  |            |
| 2- Index Finding (see explain plans section below)                                                                                                                                                                                  |               |                  |            |
| -----                                                                                                                                                                                                                               |               |                  |            |
| The execution plan of this statement can be improved by creating one or more indices.                                                                                                                                               |               |                  |            |
| Recommendation (estimated benefit: 90.98%)                                                                                                                                                                                          |               |                  |            |
| -----                                                                                                                                                                                                                               |               |                  |            |
| - Consider running the Access Advisor to improve the physical schema design or creating the recommended index.                                                                                                                      |               |                  |            |
| create index SH. IDX\$\$_00010003 on SH.SALES("CUST_ID");                                                                                                                                                                           |               |                  |            |

练习 12-1：使用自动 SQL 优化（续）

|                                                                                                                                                                                                                                                                                                                                                                              |                       |           |      |       |  |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------|-----------|------|-------|--|
| Rationale                                                                                                                                                                                                                                                                                                                                                                    |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| Creating the recommended indices significantly improves the execution plan of this statement. However, it might be preferable to run "Access Advisor" using a representative SQL workload as opposed to a single statement. This will allow to get comprehensive index recommendations which takes into account index maintenance overhead and additional space consumption. |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| EXPLAIN PLANS SECTION                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| 1- Original With Adjusted Cost                                                                                                                                                                                                                                                                                                                                               |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| Plan hash value: 4005616876                                                                                                                                                                                                                                                                                                                                                  |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| Id                                                                                                                                                                                                                                                                                                                                                                           | Operation             | Name      | Rows | Bytes |  |
| Cost (%CPU)                                                                                                                                                                                                                                                                                                                                                                  | Time                  |           |      |       |  |
| Pstart                                                                                                                                                                                                                                                                                                                                                                       | Pstop                 |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| 0                                                                                                                                                                                                                                                                                                                                                                            | SELECT STATEMENT      |           | 1    | 13    |  |
| 898                                                                                                                                                                                                                                                                                                                                                                          | (2)   00:00:1         |           |      |       |  |
| 1                                                                                                                                                                                                                                                                                                                                                                            |                       |           |      |       |  |
| 1                                                                                                                                                                                                                                                                                                                                                                            | HASH GROUP BY         |           | 1    | 13    |  |
| 898                                                                                                                                                                                                                                                                                                                                                                          | (2)   00:00:1         |           |      |       |  |
| 1                                                                                                                                                                                                                                                                                                                                                                            |                       |           |      |       |  |
| 2                                                                                                                                                                                                                                                                                                                                                                            | NESTED LOOPS          |           | 1    | 13    |  |
| 897                                                                                                                                                                                                                                                                                                                                                                          | (2)   00:00:1         |           |      |       |  |
| 1                                                                                                                                                                                                                                                                                                                                                                            |                       |           |      |       |  |
| *                                                                                                                                                                                                                                                                                                                                                                            | 3   TABLE ACCESS FULL | CUSTOMERS | 1    | 5     |  |
| 405                                                                                                                                                                                                                                                                                                                                                                          | (1)   00:00:0         |           |      |       |  |
| 5                                                                                                                                                                                                                                                                                                                                                                            |                       |           |      |       |  |
| 4                                                                                                                                                                                                                                                                                                                                                                            | PARTITION RANGE ALL   |           | 1    | 8     |  |
| 492                                                                                                                                                                                                                                                                                                                                                                          | (3)   00:00:0         |           |      |       |  |
| 6                                                                                                                                                                                                                                                                                                                                                                            | 1   28                |           |      |       |  |
| *                                                                                                                                                                                                                                                                                                                                                                            | 5   TABLE ACCESS FULL | SALES     | 1    | 8     |  |
| 492                                                                                                                                                                                                                                                                                                                                                                          | (3)   00:00:0         |           |      |       |  |
| 6                                                                                                                                                                                                                                                                                                                                                                            | 1   28                |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |
| Predicate Information (identified by operation id):                                                                                                                                                                                                                                                                                                                          |                       |           |      |       |  |
| -----                                                                                                                                                                                                                                                                                                                                                                        |                       |           |      |       |  |

练习 12-1：使用自动 SQL 优化（续）

```
3 - filter("C"."CUST_ID"<2)
5 - filter("S"."CUST_ID"<2 AND "S"."CUST_ID"="C"."CUST_ID")

2- Using SQL Profile

Plan hash value: 3070788227

| Id | Operation | Name |
| Rows | Bytes |
Cost (%CPU) | Time | Pstart | Pstop |

| 0 | SELECT STATEMENT |
| 1 | 13 |
55 (2) | 00:00:01 |
| 1 | HASH GROUP BY |
| 1 | 13 |
55 (2) | 00:00:01 |
| 2 | NESTED LOOPS |
| 1 | 13 |
54 (0) | 00:00:01 |
| 3 | PARTITION RANGE ALL |
| 1 | 8 |
54 (0) | 00:00:01 | 1 | 28 |
| 4 | TABLE ACCESS BY LOCAL INDEX ROWID | SALES |
| 1 | 8 |
54 (0) | 00:00:01 | 1 | 28 |
5	BITMAP CONVERSION TO ROWIDS		
* 6	BITMAP INDEX RANGE SCAN	SALES_CUST_BIX	
		1	28
* 7	INDEX UNIQUE SCAN	CUSTOMERS_PK	
1	5		
0 (0) | 00:00:01 |

Predicate Information (identified by operation id):

6 - access("S"."CUST_ID"<2)
 filter("S"."CUST_ID"<2)
7 - access("S"."CUST_ID"="C"."CUST_ID")
 filter("C"."CUST_ID"<2)
```

练习 12-1：使用自动 SQL 优化（续）

```
3- Using New Indices

Plan hash value: 2481939942

| Id | Operation | Name |
| Rows | Bytes |
Cost (%CPU) | Time | Pstart | Pstop |

| 0 | SELECT STATEMENT |
| 1 | 13 |
5 (0) | 00:00:01 |
| 1 | SORT GROUP BY NOSORT |
| 1 | 13 |
5 (0) | 00:00:01 |
2	NESTED LOOPS
3	NESTED LOOPS
1	13
5 (0)	00:00:01
* 4	INDEX RANGE SCAN
1	5
2 (0)	00:00:01
* 5	INDEX RANGE SCAN
1	
2 (0)	00:00:01
6	TABLE ACCESS BY GLOBAL INDEX ROWID
1	8
3 (0) | 00:00:01 | ROWID | ROWID |

Predicate Information (identified by operation id):

4 - access("C"."CUST_ID"<2)
5 - access("S"."CUST_ID"="C"."CUST_ID")
 filter("S"."CUST_ID"<2)

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 12-1：使用自动 SQL 优化（续）

- 9) 研究如何使用 Enterprise Manager 配置自动 SQL 优化。
  - a) 返回至 Enterprise Manager，转至 “Automated Maintenance Tasks（自动维护任务）” 页。
  - b) 此处的图表显示过去执行每个客户机的时间，以及计划将来再次执行这些客户机的时间。
  - c) 使用图表下方的箭头修改图形的起始点和结束点。
  - d) 单击 “Configure（配置）” 按钮。
  - e) 此时将显示 “Automated Maintenance Tasks Configuration（自动维护任务配置）” 页。
  - f) 在该页上，可禁用单个客户机和更改运行这些客户机的窗口。
  - g) 选中 “Disabled（禁用）”，禁用自动 SQL 优化客户机。单击 “Apply（应用）”，然后单击定位器链接返回 “Automated Maintenance Tasks（自动维护任务）” 页。
  - h) 请注意任务名称旁边的禁止标志（带斜杠的红圈）。
  - i) 单击 “Configure（配置）” 返回配置页。对自动 SQL 优化任务单击 “Enabled（启用）”，然后单击 “Apply（应用）”。单击定位器链接返回 “Automated Maintenance Tasks（自动维护任务）” 页。
  - j) 在 “Automated Maintenance Tasks Configuration（自动维护任务配置）” 页上单击 “Automatic SQL Tuning（自动 SQL 优化）” 链接。
  - k) 此时将转到可以配置任务本身的页。
  - l) 在 “Automatic SQL Tuning Result Summary（自动 SQL 优化结果概要）” 页上，单击 “Automatic Implementation of SQL Profiles is currently Enabled（自动实施 SQL 概要文件当前处于启用状态）” 字段右侧的 “Configure（配置）” 按钮。
  - m) 在 “Automatic SQL Tuning Settings（自动 SQL 优化设置）” 页上，在 “Automatic Implementation of SQL Profiles（自动实施 SQL 概要文件）” 的前面选择 “No（否）”，然后单击 “Apply（应用）”。单击 “Database Instance（数据库实例）” 链接，返回 “Server（服务器）” 页。
- 10) 可选：您可以通过查看 `manual_config.sh` 脚本了解如何使用 PL/SQL 配置自动 SQL 优化。

**注：**在您的练习中，由于所评估的工作量很小，所以该任务执行得很快。但是，如果任务持续的时间过长，您可使用来自另一个会话的 `interrupt_task.sh` 脚本停止该任务。

## 第 13 课的练习

在本课的练习中，您将创建 AWR 基线。您将监视 AUTOTASK 作业的性能并调整 AUTOTASK 作业的资源窗口。最后，您将校准 I/O 资源。

## 练习 13-1：创建基线

使用 `orcl` 数据库，创建基线。（此练习要求自数据库启动后默认的快照收集就一直处于运行状态。）

- 1) 对过去的快照创建基线，并根据此静态基线计算统计信息。
  - a) 以 `SYS` 用户身份，在 Database Control 的主页上单击“Server（服务器）”选项卡。在“Server（服务器）”页上，单击“Statistics Management（统计信息管理）”部分的“AWR Baselines（AWR 基线）”链接。
  - b) 在“AWR Baselines（AWR 基线）”页上，单击“Create（创建）”按钮。
  - c) 在“Create Baseline: Baseline Interval Type（创建基线：基线间隔类型）”页上，选择“Single（单一）”，然后单击“Continue（继续）”。
  - d) 在“Create Baseline: Single Baseline（创建基线：单一基线）”页上，将“Baseline Name（基线名）”设置为“Monday（星期一）”。展开“Change Chart Time Period（更改图表时段）”部分。请选择星期天的日期作为“Chart Start Date（图表的开始日期）”，星期二的日期作为“Chart End Date（图表的结束日期）”，然后单击“Go（执行）”。然后单击图表下方的第一个快照图标（代表星期一的最早时刻）。（此时可能显示“Period Start Time（时段开始时间）”下拉框。如果是这样，请选择最早的时刻。）单击“Period End Time（时段结束时间）”单选按钮，然后单击图表下方最接近星期二午夜时间的快照图标。单击“Finish（完成）”。
  - e) 在“AWR Baselines（AWR 基线）”页上，选择“Monday（星期一）”基线，然后从“Actions（操作）”下拉列表中选择“Schedule Statistics Computation（调度统计信息计算）”。单击“Go（执行）”。
  - f) 在“Compute Threshold Statistics: Monday（计算阈值统计信息：星期一）”页上，确认选中了“Immediately（立即执行）”，然后单击“Submit（提交）”启动计算作业。
  - g) 此时将显示“AWR Baselines（AWR 基线）”页。请稍等片刻，等待页面刷新。作业完成时，“Statistics Computed（计算的统计信息）”列会显示“Yes（是）”。可能必须多次刷新该页。
- 2) 定制“Performance（性能）”页。
  - a) 在“AWR Baselines（AWR 基线）”页上，从“Actions（操作）”下拉列表中选择“Customize Performance（定制性能）”，然后单击“Go（执行）”。
  - b) 接受默认设置。单击“OK（确定）”。
  - c) 导航至“Performance（性能）”页，观察不同之处。

## 练习 13-1：创建基线（续）

- 3) 稍后创建重复基线。
  - a) 导航至“AWR Baseline（AWR 基线）”页，然后单击“Create（创建）”。
  - b) 选择“Repeating（重复）”作为基线间隔类型，然后单击“Continue（继续）”。
  - c) 在“Create Baseline: Repeating Baseline Template（创建基线：重复基线模板）”中，设置以下内容：  
Baseline Name（基线名称）：afterwork  
Start Time（开始时间）：6:00 PM  
Duration（持续时间）：12 小时  
Frequency（频率）：每天  
Start Time（开始时间）：设为本课之前的星期日  
End Time（结束时间）：设为本课之后的星期六  
Retention Time（保留时间）：28 天
  - d) 单击“Finish（完成）”。
  - e) 在“AWR Baselines（AWR 基线）”页上，单击“Related Links（相关链接）”部分的“AWR Baseline Templates（AWR 基线模板）”。
  - f) 在“AWR Baseline Templates（AWR 基线模板）”页上，注意已为将来要创建的基线创建了一个模板。
- 4) 查看“Data Dictionary（数据字典）”视图。
  - a) 以 SYS 用户身份连接到 orcl 数据库实例。
  - b) 从 DBA\_HIST\_BASELINE 中选择 BASELINE\_NAME 和 BASELINE\_TYPE 列。

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 11:24:33
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select baseline_name, baseline_type from
dba_hist_baseline;
```



练习 13-1：创建基线（续）

| BASELINE_NAME        | BASELINE_TYPE |
|----------------------|---------------|
| -----                | -----         |
| Monday               | STATIC        |
| SYSTEM_MOVING_WINDOW | MOVING_WINDOW |
| SQL>                 |               |

c) 描述 DBA\_HIST\_BASELINE\_DETAILS。

| SQL> desc dba_hist_baseline_details                        |       |              |
|------------------------------------------------------------|-------|--------------|
| Name                                                       | Null? | Type         |
| -----                                                      | ----- | -----        |
| DBID                                                       |       | NUMBER       |
| INSTANCE_NUMBER                                            |       | NUMBER       |
| BASELINE_ID                                                |       | NUMBER       |
| BASELINE_NAME                                              |       | VARCHAR2(64) |
| BASELINE_TYPE                                              |       | VARCHAR2(13) |
| START_SNAP_ID                                              |       | NUMBER       |
| START_SNAP_TIME                                            |       | TIMESTAMP(3) |
| END_SNAP_ID                                                |       | NUMBER       |
| END_SNAP_TIME                                              |       | TIMESTAMP(3) |
| SHUTDOWN                                                   |       | VARCHAR2(3)  |
| ERROR_COUNT                                                |       | NUMBER       |
| PCT_TOTAL_TIME                                             |       | NUMBER       |
| LAST_TIME_COMPUTED                                         |       | DATE         |
| MOVING_WINDOW_SIZE                                         |       | NUMBER       |
| CREATION_TIME                                              |       | DATE         |
| EXPIRATION                                                 |       | NUMBER       |
| TEMPLATE_NAME                                              |       | VARCHAR2(64) |
| SQL> exit                                                  |       |              |
| Disconnected from Oracle Database 11g Enterprise Edition   |       |              |
| Release 11.2.0.1.0 - Production                            |       |              |
| With the Partitioning, Automatic Storage Management, OLAP, |       |              |
| Data Mining and Real Application Testing options           |       |              |
| \$                                                         |       |              |

## 练习 13-2：控制自动任务作业

自动维护任务在维护窗口中运行。这些任务是根据作业类分配到使用者组的。作业和窗口都在分配的时间内完成，而不使用过多资源。

- 1) 查看自动维护任务的窗口设置。“Tuesday（星期二）”窗口何时打开以及打开时间有多长？
  - a) 导航至“Server（服务器）”页，单击“Oracle Scheduler（Oracle 调度程序）”部分中的“Window Groups（窗口组）”。
  - b) 在“Scheduler Window Groups（调度程序窗口组）”页中，单击 MAINTENANCE\_WINDOW\_GROUP。
  - c) 在“View Window Group: MAINTENANCE\_WINDOW\_GROUP（查看窗口组：MAINTENANCE\_WINDOW\_GROUP）”页上，单击 TUESDAY\_WINDOW。
  - d) “Tuesday（星期二）”窗口在每个星期二的下午 10:00 打开，共开启四个小时。
  - e) 单击“Database（数据库）”链接，返回到“Database（数据库）”主页。
- 2) 完成自动维护任务需要多长时间？为这些作业分配了多少资源？关于自动维护任务的统计信息将保存在 DBA\_AUTOTASK\_CLIENT 视图中。
  - a) 在终端窗口中，调用 SQL\*Plus，然后以 SYS 用户身份进行连接。用 DESCRIBE 语句描述 DBA\_AUTOTASK\_CLIENT 视图。

```
$ cd /home/oracle/solutions/INFRA
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 11:33:43
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```

## 练习 13-2：控制自动任务作业（续）

```
SQL> desc dba_autotask_client
Name Null? Type

CLIENT_NAME VARCHAR2(64)
STATUS VARCHAR2(8)
CONSUMER_GROUP VARCHAR2(30)
CLIENT_TAG VARCHAR2(2)
PRIORITY_OVERRIDE VARCHAR2(7)
ATTRIBUTES VARCHAR2(4000)
WINDOW_GROUP VARCHAR2(64)
SERVICE_NAME VARCHAR2(64)
RESOURCE_PERCENTAGE NUMBER
USE_RESOURCE_ESTIMATES VARCHAR2(5)
MEAN_JOB_DURATION INTERVAL DAY(9) TO SECOND(9)
MEAN_JOB_CPU INTERVAL DAY(9) TO SECOND(9)
MEAN_JOB_ATTEMPTS NUMBER
MEAN_INCOMING_TASKS_7_DAYS NUMBER
MEAN_INCOMING_TASKS_30_DAYS NUMBER
TOTAL_CPU_LAST_7_DAYS INTERVAL DAY(9) TO SECOND(9)
TOTAL_CPU_LAST_30_DAYS INTERVAL DAY(9) TO SECOND(9)
MAX_DURATION_LAST_7_DAYS INTERVAL DAY(3) TO SECOND(0)
MAX_DURATION_LAST_30_DAYS INTERVAL DAY(3) TO SECOND(0)
WINDOW_DURATION_LAST_7_DAYS INTERVAL DAY(9) TO SECOND(9)
WINDOW_DURATION_LAST_30_DAYS INTERVAL DAY(9) TO SECOND(9)

SQL>
```

- b) 选择 CLIENT\_NAME、MEAN\_JOB\_DURATION 和 RESOURCE\_PERCENTAGE 列。在该示例中，作业平均持续时间最长的作业是 SQL 优化指导。如果增加后该作业的 MEAN\_JOB\_DURATION 与该窗口的持续时间接近，则应增加窗口的持续时间或者增加 RESOURCE\_PERCENTAGE，以便允许需要资源的自动任务作业在分配的时间内完成。该查询作为 solutions/INFRA 目录中的脚本提供。使用 get\_autotask\_time.sql 脚本。

```
SQL> @get_autotask_time
SQL>
SQL> SELECT client_name,
2 mean_job_duration,
3 resource_percentage
4 FROM DBA_AUTOTASK_CLIENT
5 /
```

练习 13-2：控制自动任务作业（续）

| CLIENT_NAME                     | MEAN_JOB_DURATION              |
|---------------------------------|--------------------------------|
| -----                           | -----                          |
| RESOURCE_PERCENTAGE             |                                |
| -----                           |                                |
| auto optimizer stats collection | +0000000000 00:02:51.285714286 |
| 25                              |                                |
| auto space advisor              | +0000000000 00:00:40.142857143 |
| 25                              |                                |
| sql tuning advisor              | +0000000000 00:02:55.857142857 |
| 25                              |                                |
| SQL> exit                       |                                |

### 练习 13-3: 校准 I/O 资源

了解服务器上 I/O 子系统的容量很重要。使用 I/O 校准功能，可监视 I/O 负载相对于 I/O 容量的情况。

- 1) 查看当前 I/O 配置。导航至 “I/O Calibration (I/O 校准)” 页。
  - a) 在 Database Control 主页上，单击 “Performance (性能)” 选项卡。
  - b) 单击 “Performance (性能)” 页中心附近的 I/O 选项卡。
  - c) 在 I/O 图表设置中选择 “I/O Type (I/O 类型)”。
- 2) 使用 Enterprise Manger 校准 I/O 资源度量。输入新校准的输入。
  - a) 在 “Performance (性能)” 页上，单击 “I/O Calibration (I/O 校准)” 按钮。
  - b) 在 “I/O Calibration (I/O 校准)” 页上，滚动至 “Inputs for a New Calibration (新校准的输入)” 部分。在 “Database’s Storage System (数据库的存储系统)” 字段中的 “Approximate Number of Physical Disks (物理磁盘的大致数量)” 中，输入 1。  
在 “Maximum Tolerable Latency in Milliseconds for Single-Block I/O Requests (单个块 I/O 请求的最大可容忍等待时间 (毫秒))” 中，输入 50。
  - c) 单击 “Submit (提交)”。此操作最长可能需要 15 分钟。
  - d) 可在 “Scheduler Central (中心调度程序)” 页上查看已提交作业的信息。  
单击数据库实例链接，返回 “Performance (性能)” 选项卡。单击 “Server (服务器)” 选项卡。在 “Server (服务器)” 页上，单击 “Oracle Scheduler (Oracle 调度程序)” 部分中的 “Jobs (作业)” 链接。
  - e) 在 “Scheduler Jobs (调度程序作业)” 页上，应看到 “I/O Calibration (I/O 校准)” 链接。您应该看到它在运行。
  - f) 可在 “Performance (性能)” 页上查看 I/O 活动。单击 “Database (数据库)” 选项卡。然后单击 “Performance (性能)” 选项卡。
  - g) 完成校准后，可以单击 “I/O Calibration (I/O 校准)” 按钮复查 “Calibration (校准)” 页。
  - h) 在 “I/O Calibration (I/O 校准)” 页上，查看 “Existing Calibration Results (现有校准结果)” 部分。看到的结果会因计算机而异。
  - i) 查看结果后，单击 “Database (数据库)” 链接返回到 “Database (数据库)” 主页。

## 第 14 课的练习

在本练习中，您将练习使用多种新增的诊断功能。

## 练习 14-1：使用支持工作台分析损坏

在本练习中，您将先损坏一个数据字典表，然后使用支持工作台和 ADRCI 分析损坏程度。

- 1) 执行 /home/oracle/solutions/HM 目录中的 hm\_setup.sh 脚本。此脚本会损坏数据字典。

```
$ cd /home/oracle/solutions/HM
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
$./hm_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 12:06:49
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
Table created.

SQL> SQL> SQL> SQL> 2 3 4
PL/SQL procedure successfully completed.

SQL> SQL>
 OBJ

 76538

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 310380928 bytes
Database Buffers 104857600 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
```

## 练习 14-1：使用支持工作台分析损坏（续）

```
SQL> SQL>
1 row updated.

SQL> SQL>
Commit complete.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 310380928 bytes
Database Buffers 104857600 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 2) 使用 Enterprise Manger 触发对该数据字典的健康状况检查。
  - a) 以 SYS 用户身份登录 Enterprise Manager。在主页底部单击“Related Links（相关链接）”部分的“Advisor Central（指导中心）”链接。
  - b) 在“Advisor Central（指导中心）”页中，单击“Checkers（检查器）”选项卡。
  - c) 在“Checker（检查器）”子页上，单击“Dictionary Integrity Check（字典完整性检查）”。
  - d) 在“Run Dictionary Integrity Check（运行字典完整性检查）”页上，指定 DicoCheck 的运行名称。
  - e) 单击“OK（确定）”。
  - f) 在“Advisor Central Checkers（指导中心检查器）”页上，应看到一条确认消息，指示检查已成功运行。
  - g) 在“Result（结果）”选项卡中，选择 DicoCheck 运行并单击“Details（详细资料）”。

此时将转到“Run Details Findings（运行详细资料查找结果）”子页，从中可以看到一些数据损坏。尤其会看到以下查找结果：“SQL dictionary health check: invalid column number 8 on object TAB\$ failed.”这对应于 TAB\$ 中引用 SCOTT.TABJFV 的一行。



## 练习 14-1：使用支持工作台分析损坏（续）

3) 如何使用 SQL\*Plus 获取关于先前查找结果的报告？

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 12:18:04
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> set long 100000
SQL> select dbms_hm.get_run_report('DicoCheck') from dual;

DBMS_HM.GET_RUN_REPORT('DICOCHECK')

Basic Run Information
Run Name : DicoCheck
Run Id : 41
Check Name : Dictionary Integrity Check
Mode : MANUAL
Status : COMPLETED
Start Time : 2009-10-06 12:11:34.153957 +00:00
End Time : 2009-10-06 12:11:38.938352 +00:00
Error Encountered : 0
Source Incident Id : 0
Number of Incidents Created: 0

DBMS_HM.GET_RUN_REPORT('DICOCHECK')

Input Paramters for the Run
TABLE_NAME=ALL_CORE_TABLES
CHECK_MASK=ALL

Run Findings And Recommendations
Finding
Finding Name : Dictionary Inconsistency
Finding ID : 42
Type : FAILURE
Status : OPEN
```

练习 14-1：使用支持工作台分析损坏（续）

```
DBMS_HM.GET_RUN_REPORT('DICOCHECK')

Priority : CRITICAL
Message : SQL dictionary health check: invalid column
 number 8 on object TAB$ failed
Message : Damaged rowid is AAAAAACAABAAAWFaAAB -
 description: Object SCOTT.TABJFV is referenced
Finding
Finding Name : Dictionary Inconsistency
Finding ID : 45
Type : FAILURE
Status : OPEN
Priority : CRITICAL

DBMS_HM.GET_RUN_REPORT('DICOCHECK')

Message : SQL dictionary health check: file$ pk 42 on
 object FILE$ failed
Message : Damaged rowid is AAAAAARAABAAAADpAAG -
 description: No further damage description
 available
Finding
Finding Name : Dictionary Inconsistency
Finding ID : 48
Type : FAILURE
Status : OPEN
Priority : CRITICAL
Message : SQL dictionary health check: file$ pk 42 on
 object FILE$

DBMS_HM.GET_RUN_REPORT('DICOCHECK')

 failed
Message : Damaged rowid is AAAAAARAABAAAADpAAH -
 description: No further damage description
 available

SQL>
SQL>
```

- 4) 返回 Enterprise Manager 主页。会看到什么结果？
- a) 单击 “Database（数据库）” 选项卡，返回到 “Database（数据库）” 主页。
  - b) 应该会看到一个先前在主页的 “Alerts（预警）” 部分中检测到的损坏的新严重预警（数据故障）。

## 练习 14-1：使用支持工作台分析损坏（续）

- 5) 返回终端会话。首先刷新共享池和缓冲区高速缓存，然后退出并以 sys 用户身份重新连接。接下来执行以下命令：select \* from scott.tabjfv; 会看到什么结果？

```
SQL> alter system flush shared_pool;

System altered.

SQL> alter system flush buffer_cache;

System altered.

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 12:39:27
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select * from scott.tabjfv;

select * from scott.tabjfv
*
ERROR at line 1:
ORA-03113: end-of-file on communication channel
Process ID: 6404
Session ID: 37 Serial number: 778

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 14-1：使用支持工作台分析损坏（续）

- 6) 使用 Enterprise Manager 调查前一个问题。
- a) 在“Database（数据库）”主页上，“Diagnostic Summary（诊断概要）”区域中应显示一个活动的意外事件。单击“Active Incident（活动意外事件）”链接（非法访问）。如果没有看到意外事件，则单击“Software and Support（软件和支持）”。在“Software and Support（软件和支持）”子页上，单击“Support（支持）”部分中的“Support Workbench（支持工作台）”。
  - b) 在“Support Workbench Problems（支持工作台问题）”子页上，应显示一个带有类似于 ORA 7445 [qcstda()+720] 说明的新活动意外事件。
  - c) 展开“Details（详细资料）”列中的“Show（显示）”链接。
  - d) 单击“incident（意外事件）”链接。
  - e) 此时将转到相应的“Incident Details（事件详细资料）”页。记录意外事件编号：\_\_\_\_\_
  - f) 在该页的“Application Information（应用程序信息）”部分中，可以清楚地看到导致该意外事件的原因。“select \* from scott.tabjfv”导致了该事件。
  - g) 在“Dump Files（转储文件）”子页上，还可以看到两个转储文件。第一个转储文件对应于意外事件转储文件，位于 ADR 目录下。（例如：  
/u01/app/oracle/diag/rdbms/orcl/orcl/incident/incdir\_18273）第二个转储文件是相应的标准跟踪文件，位于  
/u01/app/oracle/diag/rdbms/orcl/orcl/trace。
  - h) 单击“View Contents（查看内容）”列中的眼镜图标，以跟踪意外事件（第一行）。
  - i) 如果先前没有保存主机身份证明，则显示“Host Credentials（主机身份证明）”页。输入 oracle/oracle 并单击“Continue（继续）”。
  - j) 此时将转到“Contents（内容）”页，在该页中会看到结构化跟踪信息。
  - k) 例如，单击“Expand All（全部展开）”链接，然后在 custom\_assert\_dump 部分中单击“incident dump（意外事件转储）”链接下面的“current sql statement（当前 sql 语句）”链接。此时将打印窗口中“Trace Map（跟踪映射）”下有问题的 SQL 语句。
  - l) 单击“OK（确定）”，返回到“Incident Details（意外事件详细资料）”页。
  - m) 在“Incident Details（意外事件详细资料）”页上，单击顶部的“Problem Details（问题详细资料）”定位器链接。
  - n) 在“Problem Details（问题详细资料）”页上，单击 SR 字段旁边的“Edit（编辑）”。

## 练习 14-1：使用支持工作台分析损坏（续）

- o) 在出现的“SR Number（SR 编号）”弹出式窗口中输入 1234，然后单击“OK（确定）”。假定您已使用 My Oracle Support 打开了 SR (1234)。
  - p) 现在，“Problem Details（问题详细资料）”页上应显示 SR 编号。
  - q) 单击“Activity Log（活动日志）”选项卡。
  - r) 此时将转到“Activity Log（活动日志）”子页，现在该页上会显示针对该问题的最后操作。可在此处输入注释。
- 7) 返回终端窗口。在 ADR 中查找事件和健康状况检查报告。

```
$ cd $ORACLE_BASE/diag/rdbms/orcl/orcl
$ ls
alert cdump hm incident incpkg ir lck metadata stage
sweep trace
$ cd incident
$ ls
inccdir_18273
$ cd inccdir_18273
$ ls
orcl_ora_12211_i18273.trc orcl_ora_12211_i18273.trm
$ cd ../..
$ cd hm
$ ls
HMREPORT_DicoCheck.hm
$ cd ..
```

- 8) 使用 ADRCI 查找意外事件和问题信息。

```
$ adrci

ADRCI: Release 11.2.0.1.0 - Production on Tue Oct 6 12:58:51
2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> set homopath diag/rdbms/orcl/orcl
adrci> show homes
ADR Homes:
diag/rdbms/orcl/orcl
adrci> show incidents

ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:

INCIDENT_ID PROBLEM_KEY
CREATE_TIME

```

## 练习 14-1：使用支持工作台分析损坏（续）

```
6233 ORA 600 [HAH!]
2009-09-29 13:15:37.877000 +00:00
6193 ORA 600 [HAH!]
2009-09-29 13:35:00.123000 +00:00
18235 ORA 7445 [qcstda()+720]
2009-10-06 12:40:19.014000 +00:00
3 rows fetched

adrci> show problem

ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:

PROBLEM_ID PROBLEM_KEY
LAST_INCIDENT LASTINC_TIME

2 ORA 7445 [qcstda()+720]
18235 2009-10-06 12:40:19.014000 +00:00
1 ORA 600 [HAH!]
6193 2009-09-29 13:35:00.123000 +00:00
2 rows fetched
adrci> exit
$
```

- 9) 使用 Enterprise Manager 快速打包问题。
- a) 在“Problem Details（问题详细资料）”页上，单击“Investigate and Resolve（调查并解决）”部分中的“Quick Package（快速打包）”按钮。
  - b) 在“Quick Packaging: Create New Package（快速打包：新建程序包）”页上，接受默认程序包名称（与 ORA7445qc\_20091006132246 类似）并在“Send to Oracle Support（发送到 Oracle 技术支持）”条目旁边选中“**No（否）**”。
  - c) 刷新此页后，单击“Next（下一步）”。
  - d) 在“Quick Packaging: View Contents（快速打包：查看内容）”页上，将显示此程序包所包含的意外事件的列表。在本例中，只有一个意外事件。单击“Next（下一步）”。
  - e) 在“Quick Packaging: View Manifest（快速打包：查看声明）”页上，查找声明名称。单击“Next（下一步）”。
  - f) “Quick Packaging: Schedule（快速打包：调度）”页上，确保选中了“**Immediately（立即执行）**”，然后单击“Submit（提交）”。
  - g) 此时将显示“Processing: Generating Upload File for Package（正在处理：为程序包生成上载文件）”页。请耐心等待，直到完成。

## 练习 14-1：使用支持工作台分析损坏（续）

- h) 在“Confirmation（确认）”页上，应看到类似下面的消息：“Generating an upload file for package: ORA7445qc\_2009 1006132246 has been successfully generated. The upload file is located at /u01/app/oracle/product/11.1.0/db\_1/edrsr14p1.us.oracle.com\_orcl/sysman/emd/state/ORA7445qc\_20090622162930\_COM\_1.zip. Please send it manually to Oracle.”注：如果收到一条有关文件创建的错误提示，请继续本练习的剩余步骤。您将看到程序包已创建。
  - i) 单击“OK（确定）”。
  - j) 在“Problem Details Activity Log（问题详细资料活动日志）”子页上，现在可以看到日志中有两个新条目，报告有关此问题的程序包创建情况。
  - k) 在“Summary（概要）”部分中，单击“Packaged（打包）”字段的“Yes（是）”链接。
  - l) 此时将转到“Packages（程序包）”页。
  - m) 选择程序包，然后单击“View（查看）”。  
此时将转到“Package Details（程序包详细资料）”页，在该页中可看到程序包详细资料。
  - n) 单击“Files（文件）”选项卡。
  - o) 在“Files（文件）”子页上，现在可以看到“Full Package Content（完整程序包内容）”视图。
- 10) 返回终端会话，并使用 ADRCI 确定现有程序包的列表。退出 ADRCI。

```
$ adrci

ADRCI: Release 11.2.0.1.0 - Production on Tue Oct 6 13:32:35
2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

ADR base = "/u01/app/oracle"
adrci> set homopath diag/rdbms/orcl/orcl
adrci> query ips_package

ADR Home = /u01/app/oracle/diag/rdbms/orcl/orcl:

PACKAGE_ID FLAGS STATUS
CREATION_TIME LAST_SEQUENCE
LAST_COMPLETE_SEQUEN PROBLEM_ID NAME
DESCRIPTION
CORRELATION_LEVEL DRIVING_INCIDENT BEGIN_TIME
END_TIME UPDATE_TIME
LAST_BASE_SEQUENCE
```

# 练习 14-1：使用支持工作台分析损坏（续）

```


1 0 4
2009-10-06 13:23:38.219283 +00:00 1
1 1
ORA7445qc_20091006132246
2 18235
2009-10-06 13:23:38.455133 +00:00 0
1 rows fetched

adrci> exit
$

```

11) 分析此问题后，您知道 TAB\$ 中引用 TABJFV 的列号是错误的。通过更新 TAB\$ 解决问题，并关闭该问题。为此，可执行 hm\_fix.sh。

```

$ cd /home/oracle/solutions/HM
$./hm_fix.sh

SQL*Plus: Release 11.2.0.1.0 Production on Tue Oct 6 13:36:13
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> SQL> SQL> 2 3 4
PL/SQL procedure successfully completed.

SQL> SQL>
 OBJ

 76538

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> ORACLE instance started.

```



**练习 14-1：使用支持工作台分析损坏（续）**

```

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 310380928 bytes
Database Buffers 104857600 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> SQL>
1 row updated.

SQL> SQL>
Commit complete.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 310380928 bytes
Database Buffers 104857600 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> SQL> SQL>
DBMS_HM.GET_RUN_REPORT('DICOCHECKJFV')

Basic Run Information
Run Name : DicoCheckjfv
Run Id : 61
Check Name : Dictionary Integrity Check
Mode : MANUAL
Status : COMPLETED
Start Time : 2009-10-06 13:36:57.581510 +00:00
End Time : 2009-10-06 13:36:58.798228 +00:00
Error Encountered : 0
Source Incident Id : 0
Number of Incidents Created: 0

DBMS_HM.GET_RUN_REPORT('DICOCHECKJFV')

Input Paramters for the Run
TABLE_NAME=tab$
CHECK_MASK=ALL

```

## 练习 14-1：使用支持工作台分析损坏（续）

```
Run Findings And Recommendations

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

Recovery Manager: Release 11.2.0.1.0 - Production on Tue Oct 6
13:37:04 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1226549444)
using target database control file instead of recovery catalog

RMAN>
RMAN>
List of Database Failures
=====

Failure ID Priority Status Time Detected Summary

48 CRITICAL OPEN 06-OCT-09 SQL dictionary
health check: file$ pk 42 on object FILE$ failed
45 CRITICAL OPEN 06-OCT-09 SQL dictionary
health check: file$ pk 42 on object FILE$ failed

RMAN>

Recovery Manager complete.
$
```

12) 使用 Enterprise Manager 关闭意外事件。

- 在“Package Details（程序包详细资料）”页上，单击“Summary（概要）”部分中“Package（程序包）”字段的“Problems（问题）”旁边的问题链接。
- 在“Problem Details Incidents（问题详细资料意外事件）”子页上，选择意外事件并单击“Close（关闭）”。
- 在“Confirmation（确认）”页上，单击“Yes（是）”。
- 单击“Database（数据库）”，返回到“Database（数据库）”主页。

## 练习 14-2: 使用 SQL 修复指导

在本练习中，您将模拟 SQL 崩溃并使用 SQL 修复指导进行分析。

- 1) 在终端会话中，转到 /home/oracle/solutions/SQL\_Repair 目录。以 sys 用户身份在 SQL\*plus 会话中进行连接，执行 sql\_repair1.sql 脚本。

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$
$ cd /home/oracle/solutions/SQL_Repair
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Mon Jun 29 00:33:12
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> @sql_repair1
SQL>
SQL> -- This example generates a workaround for a crash. This
bug has already
SQL> -- been fixed but we toggle the bug fix using an
underscore parameter
SQL> -- which uses the (internal) feature called bug fix
control.
SQL> -- This script will pause periodically to allow you to
read the comments
SQL> -- and see the output of the previous command on the
screen. Just press
SQL> -- return to make the demo resume.
SQL>
SQL> pause
```

## 练习 14-2: 使用 SQL 修复指导 (续)

- 2) 按 [Enter] 继续执行操作。现在脚本会为用户 SCOTT 授予演示所需的权限。

```
SQL>
SQL> -- To begin the demo we will create the user scott and
grant
SQL> -- advisor privileges to him.
SQL>
SQL> pause

SQL>
SQL> grant connect, resource, dba, query rewrite, unlimited
tablespace to scott identified by tiger;

Grant succeeded.

SQL>
SQL> alter user scott account unlock;

User altered.

SQL>
SQL> -- Next we need to create and populate the table used by
the demo.
SQL> -- We will also create an index on the table;
SQL>
SQL> pause
```

- 3) 按 [Enter] 后，将创建一个表并填充此表。

```
SQL> connect scott/tiger;
Connected.
SQL>
SQL> drop table simple_table;
drop table simple_table
 *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create table simple_table(a varchar(40), b number, c
varchar(240), d varchar(240));

Table created.

SQL>
SQL> create index tc on simple_table(b, d, a);

Index created.
```

**练习 14-2: 使用 SQL 修复指导 (续)**

```

SQL>
SQL> insert into simple_table values('a', 1, 'b', 'c');

1 row created.

SQL> insert into simple_table values('a', 1, 'x', 'c');

1 row created.

SQL> insert into simple_table values('e', 2, 'f', 'g');

1 row created.

SQL>
SQL> -- In order to crash the system we need to switch off the
code line that
SQL> -- normal protects against this type of crash. We can
switch off the code
SQL> -- using the (internal) feature called bug fix control.
SQL>
SQL> pause

```

- 4) 按 [Enter] 继续执行操作。需要关闭系统中的一些代码，故意在执行特定 SQL 语句时导致崩溃。

```

SQL>
SQL> -- switch the code
SQL> alter system set "_fix_control"="5868490:OFF";

System altered.

SQL> -- alter session set optimizer_dynamic_sampling = 0;
SQL>
SQL> -- Now that the code line has been switched off lets get
the execution
SQL> -- plan for a simple delete statement.
SQL>
SQL> pause

```

- 5) 按 [Enter] 继续执行操作。对本练习中使用的 DELETE 语句进行解释。

```

SQL>
SQL> --- explain the plan
SQL> explain plan for delete
 2 /*+
 3 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
 4 USE_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")
 5 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1")
 6 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")
 7 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")
 8 OUTLINE(@"DEL$1")

```

练习 14-2：使用 SQL 修复指导（续）

```

 9 OUTLINE(@"SEL$1")
10 OUTLINE(@"SEL$AD0B6B07")
11 OUTLINE(@"SEL$7D4DB4AA")
12 UNNEST(@"SEL$1")
13 OUTLINE(@"SEL$75B5BFA2")
14 MERGE(@"SEL$7D4DB4AA")
15 OUTLINE_LEAF(@"SEL$80F8B8C6")
16 ALL_ROWS
17 OPT_PARAM('_optimizer_cost_model' 'fixed')
18 DB_VERSION('11.1.0.7')
19 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
20 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
21 */
22 from simple_table t1 where t1.a = 'a' and rowid <>
(select max(rowid) from simple_table t2 where t1.a= t2.a and
t1.b = t2.b and t1.d=t2.d);

Explained.

SQL>
SQL> --- display the plan
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null));

PLAN_TABLE_OUTPUT

Plan hash value: 1481897562

| Id | Operation | Name | Rows | Bytes |
Cost (%CPU) | Time
|

PLAN_TABLE_OUTPUT

| 0 | DELETE STATEMENT | | 1 | 338 |
6 (34) | 00:00:01 |
0:01 |
```

练习 14-2：使用 SQL 修复指导（续）

```
| 1 | DELETE | SIMPLE_TABLE | | |
|
|
|* 2 | FILTER | | | |
|
|
| 3 | HASH GROUP BY | | 1 | 338 |
6 (34) | 00:0
0:01 |

PLAN_TABLE_OUTPUT

|* 4 | HASH JOIN | | 1 | 338 |
5 (20) | 00:0
0:01 |

|* 5 | TABLE ACCESS FULL | SIMPLE_TABLE | 1 | 169 |
2 (0) | 00:0
0:01 |

|* 6 | TABLE ACCESS FULL | SIMPLE_TABLE | 1 | 169 |
2 (0) | 00:0
0:01 |

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

2 - filter(ROWID<>MAX(ROWID))
4 - access("T1"."A"="T2"."A" AND "T1"."B"="T2"."B" AND
"T1"."D"="T2"."D")
5 - filter("T2"."A"='a')
6 - filter("T1"."A"='a')

21 rows selected.
```

**练习 14-2: 使用 SQL 修复指导 (续)**

```

SQL>
SQL> -- The plan shows that we we will do a full table scan
oun r.
SQL> -- If we execut this simple system it will crash the
system.
SQL>
SQL> Pause

```

- 6) 按 [Enter] 继续执行操作。现在，执行以下语句后就会看到它崩溃了。

```

SQL> --- This statement caused the system to crash.
SQL> delete
2 /*+
3 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
4 USE_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")
5 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1")
6 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")
7 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")
8 OUTLINE(@"DEL$1")
9 OUTLINE(@"SEL$1")
10 OUTLINE(@"SEL$AD0B6B07")
11 OUTLINE(@"SEL$7D4DB4AA")
12 UNNEST(@"SEL$1")
13 OUTLINE(@"SEL$75B5BFA2")
14 MERGE(@"SEL$7D4DB4AA")
15 OUTLINE_LEAF(@"SEL$80F8B8C6")
16 ALL_ROWS
17 OPT_PARAM('_optimizer_cost_model' 'fixed')
18 DB_VERSION('11.1.0.7')
19 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
20 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
21 */
22 from simple_table t1 where t1.a = 'a' and rowid <>
(select max(rowid) from simple_table t2 where t1.a= t2.a and
t1.b = t2.b and t1.d=t2.d);
from simple_table t1 where t1.a = 'a' and rowid <> (select
max(rowid) from simple_table t2 where t1.a= t2.a and t1.b =
t2.b and t1.d=t2.d)
*
ERROR at line 22:
ORA-00600: internal error code, arguments: [13011], [74582],
[16777740], [0],
[16777740], [17], [], [], [], [], [], [], []
SQL>

```



## 练习 14-2: 使用 SQL 修复指导 (续)

- 7) 必须使用 SQL 修复指导分析情况并查找更好的替代项。仍然在连接的 SQL\*Plus 会话中, 执行 sql\_repair2.sql 脚本。

```
SQL> @sql_repair2
SQL>
SQL>
SQL> connect scott/tiger
Connected.
SQL> set lines 10000;
SQL> set pages 10000;
SQL> set long 20000;
SQL> set serveroutput on;
SQL> SET TRIMSPOOL ON;
SQL> SET TAB OFF;
SQL> SET FEEDBACK 1;
SQL>
SQL> alter system flush shared_pool;

System altered.

SQL>
SQL> set echo on;
SQL>
SQL> -- At this point your previous session has crashed due to
a problem
SQL> -- with our delete statement. We now want to use the SQL
Repair Advisor
SQL> -- to fit that problem. Before we begin to fix this
problem lets ensure
SQL> -- that the actual code that prevents this issue in real
life is switched off.
SQL>
SQL> pause
```

- 8) 按 [Enter] 继续执行操作。确保仍禁用前一个问题的代码。

```
SQL>
SQL> alter system set "_fix_control"="5868490:OFF";

System altered.

SQL>
SQL> -- Before we can run the SQL Repair Advisor we need to
create a
SQL> -- diagnosis task using dbms_sqldiag. We need to specify
the SQL text
SQL> -- for the statement that causes the crash and a task
name. We also need
SQL> -- to specify the problem type. There are three possible
problem types
SQL>
```

**练习 14-2: 使用 SQL 修复指导 (续)**

```

SQL> -- PROBLEM_TYPE_COMPILATION_ERROR
SQL> -- PROBLEM_TYPE_EXECUTION_ERROR
SQL> -- WRONG_RESULTS
SQL>
SQL> -- In this cause since the statement failed during
execution we will use
SQL> -- PROBLEM_TYPE_EXECUTION_ERROR. When the SQL Repair
Advisor completes we will run
SQL> -- dbms_sqldiag.report_diagnosis_task to output the
results report to the screen.
SQL>
SQL> pause

```

- 9) 按 [Enter] 继续执行操作。必须使用 SQL 修复指导分析此情况。您可以看到，在分析过程中发现了 SQL 补丁程序。

```

SQL>
SQL> exec
dbms_sqldiag.drop_diagnosis_task('sqldiag_bug_5869490');
BEGIN dbms_sqldiag.drop_diagnosis_task('sqldiag_bug_5869490');
END;

*
ERROR at line 1:
ORA-13605: The specified task or object sqldiag_bug_5869490
does not exist for the current user.
ORA-06512: at "SYS.PRVT_ADVISOR", line 2323
ORA-06512: at "SYS.DBMS_ADVISOR", line 172
ORA-06512: at "SYS.DBMS_SQLTUNE", line 812
ORA-06512: at "SYS.DBMS_SQLDIAG", line 977
ORA-06512: at line 1

SQL>
SQL> declare
 2 rep_out clob;
 3 t_id varchar2(50);
 4 begin
 5 t_id := dbms_sqldiag.create_diagnosis_task(
 6 sql_text => 'delete /*+
 7 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
 8 USE_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")
 9 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1")
10 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")
11 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")
12 OUTLINE(@"DEL$1")
13 OUTLINE(@"SEL$1")
14 OUTLINE(@"SEL$AD0B6B07")
15 OUTLINE(@"SEL$7D4DB4AA")
16 UNNEST(@"SEL$1")
17 OUTLINE(@"SEL$75B5BFA2")
18 MERGE(@"SEL$7D4DB4AA")

```

**练习 14-2: 使用 SQL 修复指导 (续)**

```

19 OUTLINE_LEAF(@"SEL$80F8B8C6")
20 ALL_ROWS
21 OPT_PARAM('_optimizer_cost_model' 'fixed')
22 DB_VERSION('11.1.0.7')
23 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
24 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
25 */
26 from simple_table t1 where t1.a = 'a' and rowid <>
(select max(rowid) from simple_table t2 where t1.a= t2.a and
t1.b = t2.b and t1.d=t2.d)',
27 task_name => 'sqldiag_bug_5869490',
28 problem_type
=>DBMS_SQLDIAG.PROBLEM_TYPE_COMPILATION_ERROR);
29
30 dbms_sqltune.set_tuning_task_parameter(
31 t_id,
32 '_SQLDIAG_FINDING_MODE',
33 dbms_sqldiag.SQLDIAG_FINDINGS_FILTER_PLANS);
34
35 dbms_output.put_line ('t_id = ' || t_id);
36
37 dbms_sqldiag.execute_diagnosis_task (t_id);
38
39 dbms_output.put_line ('executed t_id = ' || t_id);
40
41 rep_out := dbms_sqldiag.report_diagnosis_task (t_id,
DBMS_SQLDIAG.TYPE_TEXT);
42
43 dbms_output.put_line ('Report : ' || rep_out);
44
45 end;
46 /
t_id = sqldiag_bug_5869490
executed t_id = sqldiag_bug_5869490
Report : GENERAL INFORMATION SECTION

Tuning Task Name : sqldiag_bug_5869490
Tuning Task Owner : SCOTT
Workload Type : Single SQL Statement
Scope : COMPREHENSIVE
Time Limit(seconds) : 1800
Completion Status : COMPLETED
Started at : 06/29/2009 00:37:08
Completed at : 06/29/2009 00:37:17

```

## 练习 14-2: 使用 SQL 修复指导 (续)

```

Schema Name: SCOTT
SQL ID : apnt3dq2mq1ds
SQL Text : delete /*+
 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
 USE_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")
 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1"
"T1"@"DEL$1")
 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")
 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")
 OUTLINE(@"DEL$1")
 OUTLINE(@"SEL$1")
 OUTLINE(@"SEL$AD0B6B07")
 OUTLINE(@"SEL$7D4DB4AA")
 UNNEST(@"SEL$1")
 OUTLINE(@"SEL$75B5BFA2")
 MERGE(@"SEL$7D4DB4AA")
 OUTLINE_LEAF(@"SEL$80F8B8C6")
 ALL_ROWS
 OPT_PARAM('_optimizer_cost_model' 'fixed')
 DB_VERSION('11.1.0.7')
 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
 */
 from simple_table t1 where t1.a = 'a' and rowid
<> (select
 max(rowid) from simple_table t2 where t1.a= t2.a
and t1.b = t2.b
 and t1.d=t2.d)

FINDINGS SECTION (1 finding)

1- SQL Patch Finding (see explain plans section below)

 A potentially better execution plan was found for this
statement.

 建议

 - Consider accepting the recommended SQL patch.
 execute dbms_sqldiag.accept_sql_patch(task_name =>
'sqldiag_bug_5869490',task_owner => 'SCOTT', replace => TRUE);

 Rationale

 Recommended plan with hash value 3259336479 has number of
rows 0, check sum 0, execution time 0 and 12 buffer gets

```

**练习 14-2：使用 SQL 修复指导（续）**

```

PL/SQL procedure successfully completed.

SQL>
SQL> -- The above report tells us that the statement fails
during execute
SQL> -- and it has found a SQL patch for this statement that
will prevent
SQL> -- it from crashing in the future. However, unning the
repair advisor
SQL> -- wont fix the problem automatically. If we run our
problematic SQL
SQL> -- statement now the session will crash.
SQL>
SQL> pause

```

- 10) 按 [Enter] 两次继续执行操作。如果您再次执行该语句，会出现相同的错误。这是因为您尚未接受建议的 SQL 补丁程序。

```

SQL> connect scott/tiger
Connected.
SQL>
SQL> -- We need to accept the SQL Patch that the SQL Repair
Advisor recommended before
SQL> -- our SQL statement will be able to run without
crashing. To accept the SQL Patch
SQL> -- we must execute dbms_sqldiag.accept_sql_patch and pass
it task name
SQL>
SQL> pause

SQL> execute dbms_sqldiag.accept_sql_patch(task_name =>
'sqldiag_bug_5869490', name => 'delete_patch', task_owner =>
'SCOTT', replace => TRUE);

PL/SQL procedure successfully completed.

SQL>
SQL> -- Now that we have successfully accepted the SQL patch
we should be able to
SQL> -- run our statement without it crashing. Lets first look
at the execution
SQL> -- plan to see if it has changed.
SQL>
SQL> pause

```

练习 14-2：使用 SQL 修复指导（续）

11) 按 [Enter] 继续执行操作。查看 culprit SQL 语句的备用执行计划。

```
SQL>
SQL> explain plan for
 2 delete
 3 /*+
 4 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
 5 USE_HASH(@"SEL$80F8B8C6" "T1"@"DEL$1")
 6 LEADING(@"SEL$80F8B8C6" "T2"@"SEL$1" "T1"@"DEL$1")
 7 FULL(@"SEL$80F8B8C6" "T1"@"DEL$1")
 8 FULL(@"SEL$80F8B8C6" "T2"@"SEL$1")
 9 OUTLINE(@"DEL$1")
 10 OUTLINE(@"SEL$1")
 11 OUTLINE(@"SEL$AD0B6B07")
 12 OUTLINE(@"SEL$7D4DB4AA")
 13 UNNEST(@"SEL$1")
 14 OUTLINE(@"SEL$75B5BFA2")
 15 MERGE(@"SEL$7D4DB4AA")
 16 OUTLINE_LEAF(@"SEL$80F8B8C6")
 17 ALL_ROWS
 18 OPT_PARAM('_optimizer_cost_model' 'fixed')
 19 DB_VERSION('11.1.0.7')
 20 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
 21 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
 22 */
 23 from simple_table t1 where t1.a = 'a' and rowid <>
(select max(rowid) from simple_table t2 where t1.a= t2.a and
t1.b = t2.b and t1.d=t2.d);

Explained.

SQL>
SQL> -- display the plan
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'advanced'));

PLAN_TABLE_OUTPUT

...
Plan hash value: 3259336479

| Id | Operation | Name | Rows | Bytes |
Cost (%CPU) | Time |

```

练习 14-2: 使用 SQL 修复指导 (续)

|                                                               |     |  |                   |  |              |
|---------------------------------------------------------------|-----|--|-------------------|--|--------------|
| -----                                                         |     |  |                   |  |              |
| -----                                                         |     |  |                   |  |              |
|                                                               | 0   |  | DELETE STATEMENT  |  | 1   169      |
| 6                                                             | (0) |  | 00:00:01          |  |              |
|                                                               | 1   |  | DELETE            |  | SIMPLE_TABLE |
|                                                               |     |  |                   |  |              |
| *                                                             | 2   |  | FILTER            |  |              |
|                                                               |     |  |                   |  |              |
| *                                                             | 3   |  | INDEX FULL SCAN   |  | 4   676      |
| 0                                                             | (0) |  | 00:00:01          |  |              |
|                                                               | 4   |  | SORT AGGREGATE    |  | 1   169      |
|                                                               |     |  |                   |  |              |
| *                                                             | 5   |  | TABLE ACCESS FULL |  | 1   169      |
| 3                                                             | (0) |  | 00:00:01          |  |              |
| -----                                                         |     |  |                   |  |              |
| -----                                                         |     |  |                   |  |              |
| Query Block Name / Object Alias (identified by operation id): |     |  |                   |  |              |
| -----                                                         |     |  |                   |  |              |
| 1 - DEL\$1                                                    |     |  |                   |  |              |
| 3 - DEL\$1 / T1@DEL\$1                                        |     |  |                   |  |              |
| 4 - SEL\$1                                                    |     |  |                   |  |              |
| 5 - SEL\$1 / T2@SEL\$1                                        |     |  |                   |  |              |
| Outline Data                                                  |     |  |                   |  |              |
| -----                                                         |     |  |                   |  |              |
| /*+                                                           |     |  |                   |  |              |
| BEGIN_OUTLINE_DATA                                            |     |  |                   |  |              |
| FULL(@"SEL\$1" "T2"@"SEL\$1")                                 |     |  |                   |  |              |
| INDEX(@"DEL\$1" "T1"@"DEL\$1" ("SIMPLE_TABLE"."B"             |     |  |                   |  |              |
| "SIMPLE_TABLE"."D"                                            |     |  |                   |  |              |
| "SIMPLE_TABLE"."A"))                                          |     |  |                   |  |              |
| OUTLINE_LEAF(@"DEL\$1")                                       |     |  |                   |  |              |
| OUTLINE_LEAF(@"SEL\$1")                                       |     |  |                   |  |              |
| ALL_ROWS                                                      |     |  |                   |  |              |
| OPT_PARAM('_fix_control' '5868490:0')                         |     |  |                   |  |              |
| OPT_PARAM('_optimizer_cost_model' 'fixed')                    |     |  |                   |  |              |
| DB_VERSION('11.2.0.1')                                        |     |  |                   |  |              |
| OPTIMIZER_FEATURES_ENABLE('11.1.0.7')                         |     |  |                   |  |              |
| IGNORE_OPTIM_EMBEDDED_HINTS                                   |     |  |                   |  |              |
| END_OUTLINE_DATA                                              |     |  |                   |  |              |
| */                                                            |     |  |                   |  |              |
| Predicate Information (identified by operation id):           |     |  |                   |  |              |
| -----                                                         |     |  |                   |  |              |

**练习 14-2: 使用 SQL 修复指导 (续)**

```

2 - filter(ROWID<> (SELECT /*+ UNNEST UNNEST NO_INDEX
("T2") NO_INDEX
 ("T2") */ MAX(ROWID) FROM "SIMPLE_TABLE" "T2"
WHERE "T2"."A"=:B1 AND
 "T2"."B"=:B2 AND "T2"."D"=:B3))
3 - access("T1"."A"='a')
 filter("T1"."A"='a')
5 - filter("T2"."A"=:B1 AND "T2"."B"=:B2 AND "T2"."D"=:B3)

```

Column Projection Information (identified by operation id):

```

2 - (cmp=2,3,4) ROWID[ROWID,10], "T1"."A"[VARCHAR2,40],
 "T1"."B"[NUMBER,22], "T1"."D"[VARCHAR2,240]
3 - ROWID[ROWID,10], "T1"."A"[VARCHAR2,40],
 "T1"."B"[NUMBER,22],
 "T1"."D"[VARCHAR2,240]
4 - (#keys=0) MAX(ROWID)[10]
5 - ROWID[ROWID,10], "T2"."A"[VARCHAR2,40],
 "T2"."B"[NUMBER,22],
 "T2"."D"[VARCHAR2,240]

```

附注

```

- SQL patch "delete_patch" used for this statement

```

64 rows selected.

```

SQL>
SQL> -- Now lets execute it.
SQL>
SQL> pause

```

- 12) 按 [Enter] 继续执行操作。再次尝试执行该语句。应不会再看到有错误产生了。脚本末尾会重新构建原始代码路径。之后可以退出 SQL\*Plus 会话。

```

SQL>
SQL> delete
2 /*+
3 USE_HASH_AGGREGATION(@"SEL$80F8B8C6")
4 USE_HASH(@"SEL$80F8B8C6" "T1"@ "DEL$1")
5 LEADING(@"SEL$80F8B8C6" "T2"@ "SEL$1" "T1"@ "DEL$1")
6 FULL(@"SEL$80F8B8C6" "T1"@ "DEL$1")
7 FULL(@"SEL$80F8B8C6" "T2"@ "SEL$1")
8 OUTLINE(@"DEL$1")
9 OUTLINE(@"SEL$1")
10 OUTLINE(@"SEL$AD0B6B07")
11 OUTLINE(@"SEL$7D4DB4AA")
12 UNNEST(@"SEL$1")

```



## 练习 14-2: 使用 SQL 修复指导 (续)

```

13 OUTLINE(@"SEL$75B5BFA2")
14 MERGE(@"SEL$7D4DB4AA")
15 OUTLINE_LEAF(@"SEL$80F8B8C6")
16 ALL_ROWS
17 OPT_PARAM('_optimizer_cost_model' 'fixed')
18 DB_VERSION('11.1.0.7')
19 OPTIMIZER_FEATURES_ENABLE('11.1.0.7')
20 NO_INDEX(@"SEL$1" "T2"@"SEL$1")
21 */
22 from simple_table t1 where t1.a = 'a' and rowid <>
(select max(rowid) from simple_table t2 where t1.a= t2.a and
t1.b = t2.b and t1.d=t2.d);

1 row deleted.

SQL>
SQL> -- Finally we need to restore the system back the way we
found it.
SQL> -- We must drop the SQL Patch we have accepted and turn
back on the
SQL> -- original code path that prevented this problem in the
first place.
SQL>
SQL> execute dbms_sqldiag.drop_sql_patch('delete_patch');

PL/SQL procedure successfully completed.

SQL>
SQL> alter system set "_fix_control"="5868490:ON";

System altered.

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options
$

```

## 第 15 课的练习

在本课的练习中，您将使用 Enterprise Manager 监视 SQL 命令。

## 练习 15-1：使用 Enterprise Manager 监视 SQL 命令

在本练习中，您将使用 Enterprise Manager 监视 SQL 命令。

- 1) 转至 /home/oracle/solutions/SQLMONITOR 目录。以 SYS 用户身份登录 SQL\*Plus，然后执行 `moni_setup.sql` 脚本。在最后一行 SELECT 语句仍在运行时，继续执行本练习的下一个步骤。

```
$ cd /home/oracle/solutions/SQLMONITOR
$. oraenv
ORACLE_SID = [db10g] ? orcl
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 06:59:34
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @moni_setup
SQL> set long 10000000
SQL> set longchunksize 10000000
SQL> set linesize 200
SQL> set pagesize 1000
SQL>
SQL>
SQL> /*== Drop and create a test tablespace and user ==*/
SQL>
SQL>
SQL> DROP USER moni CASCADE;
DROP USER moni CASCADE
 *
ERROR at line 1:
ORA-01918: user 'MONI' does not exist

SQL>
SQL> DROP TABLESPACE moni_tbs INCLUDING CONTENTS;
DROP TABLESPACE moni_tbs INCLUDING CONTENTS
 *
ERROR at line 1:
ORA-00959: tablespace 'MONI_TBS' does not exist
```

## 练习 15-1：使用 Enterprise Manager 监视 SQL 命令（续）

```
SQL>
SQL> CREATE SMALLFILE TABLESPACE MONI_TBS
 2 DATAFILE '+DATA'
 3 SIZE 100M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M
NOLOGGING
 4 EXTENT MANAGEMENT LOCAL SEGMENT SPACE MANAGEMENT AUTO;

Tablespace created.

SQL>
SQL> CREATE USER moni IDENTIFIED BY moni
 2 DEFAULT TABLESPACE moni_tbs TEMPORARY TABLESPACE temp;

User created.

SQL>
SQL> GRANT connect, resource, dba to moni;

Grant succeeded.

SQL>
SQL> /*== Drop and create a MONI_TEST table as MONI user ==*/
SQL>
SQL> CONNECT moni/moni
Connected.
SQL>
SQL> DROP TABLE moni_test purge;
DROP TABLE moni_test purge
 *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> CREATE TABLE moni_test(c number, d varchar2(500));

Table created.

SQL>
SQL> begin
 2 for i in 1..200000 loop
 3 insert into moni_test
values(1,'aaa
aaa
aaa
aaa');
 4 end loop;
 5 commit;
 6 end;
 7 /
```

## 练习 15-1：使用 Enterprise Manager 监视 SQL 命令（续）

```
PL/SQL procedure successfully completed.

SQL>
SQL> CREATE INDEX moni_test_c_indx on moni_test(c);

Index created.

SQL>
SQL> /*== Start long running query ==*/
SQL>
SQL> SELECT count(*) FROM moni_test t1, moni_test t2
 2 WHERE t1.c=t2.c AND t1.c=1;
```

- 2) 要使用 Enterprise Manager Database Control 界面查看并监视长时间运行的 SQL 语句，请执行以下操作：
  - a) 在浏览器窗口中打开 Enterprise Manager Database Control。
  - b) 在“Database Login（数据库登录）”页上，在“User Name（用户名）”中输入 SYS，在“Password（口令）”中输入 oracle\_4U。从“Connect As（连接身份）”下拉列表中选择 SYSDBA，然后单击“Login（登录）”。
  - c) 单击“Performance（性能）”选项卡。可在“Performance（性能）”页上查看系统活动。
  - d) 滚动至该页底部，然后在“Additional Monitoring Links（其他监视链接）”部分中单击“SQL Monitoring（SQL 监视）”。
  - e) 此时将显示“Monitored SQL Executions（监视的 SQL 执行）”页。
  - f) 在“Monitored SQL Execution（监视的 SQL 执行）”页上，单击“SQL ID”链接。
  - g) 此时将显示“Monitored SQL Execution Details（监视的 SQL 执行详细资料）”页。
  - h) 在该页中的上半部分中，可以看到“Overview（概览）”信息。在右上方视图中，可以看到“Time and Wait Statistics（时间和等待统计信息）”以及“I/O Statistics（I/O 统计信息）”。
  - i) “Details（详细资料）”部分至少有两个选项卡页：“Plan Statistics（计划统计信息）”和“Activity（活动）”。如果查询有并行执行，还会有一个“Parallel（并行）”选项卡页。
  - j) 在“Plan Statistics（计划统计信息）”页的右侧显示了更多详细信息。
  - k) 单击“Activity（活动）”选项卡。
  - l) 从该图中可以看到 SELECT 语句在使用所有的 CPU 资源。

## 练习 15-1：使用 Enterprise Manager 监视 SQL 命令（续）

- m) 在“Monitored SQL Execution Details（监视的 SQL 执行详细资料）”页上，单击右上方的“View Report（查看报告）”。
  - n) 该 SQL 监视报表显示的信息与之前在图表页看到的信息相同。如果需要，请向下滚动查看。
  - o) 在该页底部，您将看到“SQL Plan Monitoring Details（SQL 计划监视详细资料）”。单击“OK（确定）”。
  - p) 在“Monitored SQL Execution Details（监视的 SQL 执行详细资料）”页上，单击“User MONI（用户 MONI）”链接以显示该用户的会话信息。
  - q) 在“Session Details（会话详细资料）”页上查看会话信息。
  - r) 该页中包含了几个选项卡页。查看会话信息后，单击右侧的“SQL Monitoring（SQL 监视）”选项卡。
  - s) 单击“SQL ID”可返回“Monitored SQL Execution Details（监视的 SQL 执行详细资料）”页。
  - t) 要查看 SQL 详细资料，请在“Monitored SQL Execution Details（监视的 SQL 执行详细资料）”页上单击“SQL ID”。
  - u) 查看 SQL 详细资料。
  - v) 然后单击该页右上部分中的“Logout（注销）”。
- 3) 要清理环境，请执行下列操作：
- a) 完成对查询的监视后，返回 SQL\*Plus 会话，然后输入 Ctrl + C 停止语句执行。以 SYSDBA 身份连接，然后执行 `moni_cleanup.sql` 脚本。

```
SELECT count(*) FROM moni_test t1, moni_test t2
 *
ERROR at line 1:
ORA-01013: user requested cancel of current operation
SQL> connect / as sysdba
Connected.
SQL> @moni_cleanup
SQL>
SQL> /*== Drop test tablespace and user ==*/
SQL>
SQL> DROP USER moni CASCADE;

User dropped.

SQL>
SQL> DROP TABLESPACE moni_tbs INCLUDING CONTENTS AND
DATAFILES;

Tablespace dropped.
```

## 练习 15-1：使用 Enterprise Manager 监视 SQL 命令（续）

```
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 第 16 课的练习

在本课的练习中，您将使用自动内存管理和优化程序延迟统计信息。



## 练习 16-1：使用自动内存管理

在本练习中，您将使用自动内存管理，这样无需再手动修改 `SGA_TARGET` 和 `PGA_AGGREGATE_TARGET`。在运行会消耗大量 SGA 内存的开销很大的并行查询，或者在运行会消耗大量 PGA 内存的开销很大的 PL/SQL 块时，可以观察内存的分配情况。

- 1) 在开始本练习之前，停止 `orcl` 数据库的 Enterprise Manager Database Control。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.
$
```

- 2) 请制作一个 SPFILE 文件副本。

```
$ cd /home/oracle/solutions/AMM
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 07:42:19
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> CREATE
PFILE='/u01/app/oracle/acfsmounts/acfs_db1/dbs/initorcl.ora.ba
k' FROM SPFILE;

File created.

SQL>
```

## 练习 16-1：使用自动内存管理（续）

- 3) 仍然在以 sys 身份连接的 SQL\*Plus 中，仅在 SPFILE 中将以下参数设置为指定值。使用位于 /home/oracle/solutions/AMM 目录下的

```
change_parameters.sql 文件。
_PX_use_large_pool = TRUE
_memory_broker_stat_interval = 5
_memory_management_tracing = 31
parallel_execution_message_size = 36864
parallel_max_servers = 200
parallel_adaptive_multi_user = FALSE
processes = 200
sga_target = 0
pga_aggregate_target = 0
memory_target = 280M
```

```
SQL> @change_parameters
SQL>
SQL> alter system set "_PX_use_large_pool" = TRUE
SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "_memory_broker_stat_interval" = 5
SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "_memory_management_tracing" = 31
SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "parallel_execution_message_size" =
36864 SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "parallel_max_servers" = 200
SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "parallel_adaptive_multi_user" = FALSE
SCOPE=SPFILE;
```

## 练习 16-1：使用自动内存管理（续）

```
System altered.

SQL>
SQL> alter system set "processes" = 200 SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "pga_aggregate_target" = 0 SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "sga_target" = 0 SCOPE=SPFILE;

System altered.

SQL>
SQL> alter system set "memory_target" = 280M SCOPE=SPFILE;

System altered.

SQL>
```

- 4) 仍然在以 SYS 身份连接的 SQL\*Plus 会话中，执行 amm\_setup.sql 脚本。

```
SQL>
SQL> @amm_setup
SQL>
SQL> set echo on
SQL>
SQL> drop tablespace tbssga including contents and datafiles;
drop tablespace tbssga including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'TBSSGA' does not exist

SQL>
SQL> create tablespace tbssga datafile '+DATA' size 20m;

Tablespace created.

SQL>
SQL> drop tablespace mytemp including contents and datafiles;
drop tablespace mytemp including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'MYTEMP' does not exist
```

# 练习 16-1：使用自动内存管理（续）

```

SQL>
SQL> create temporary tablespace mytemp tempfile '+DATA' size
40m reuse;

Tablespace created.

SQL>
SQL> drop user amm cascade;
drop user amm cascade
 *
ERROR at line 1:
ORA-01918: user 'AMM' does not exist

SQL>
SQL> create user amm identified by amm default tablespace
tbssga temporary tablespace mytemp;

User created.

SQL>
SQL> grant connect,resource,dba to amm;

Grant succeeded.

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;

COMP CS US

shared poo 260046848 0
large pool 4194304 0
java pool 4194304 0
streams po 8388608 0
SGA Target 423624704 423624704
DEFAULT bu 104857600 0
Shared IO 33554432 33554432
PGA Target 159383552 159383552

8 rows selected.

SQL>
SQL> connect amm/amm
Connected.
SQL>
SQL> drop table tabsga purge;
drop table tabsga purge
 *
ERROR at line 1:
ORA-00942: table or view does not exist

```

## 练习 16-1：使用自动内存管理（续）

```
SQL>
SQL> create table tabsga(a number, b number) tablespace
tbssga;

Table created.

SQL>
SQL> begin
 2 for i in 1..100000 loop
 3 insert into tabsga values (i, i);
 4 end loop;
 5 end;
 6 /

PL/SQL procedure successfully completed.

SQL>
SQL> commit;

Commit complete.

SQL>
SQL> alter table tabsga parallel 64;

Table altered.

SQL>
SQL> create or replace procedure testpga(psize number) as
 2 begin
 3 declare
 4 TYPE nAllotment_tabtyp IS TABLE OF char(2048) INDEX
BY BINARY_INTEGER;
 5 myarray nAllotment_tabtyp;
 6 begin
 7 for i in 1..psize loop
 8 myarray(i) := to_char(i);
 9 end loop;
 10 end;
 11 end;
 12 /

Procedure created.
```

# 练习 16-1：使用自动内存管理（续）

```

SQL>
SQL> show errors
No errors.
SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;

COMP CS US

shared poo 260046848 0
large pool 4194304 0
java pool 4194304 0
streams po 8388608 0
SGA Target 423624704 423624704
DEFAULT bu 104857600 0
Shared IO 33554432 33554432
PGA Target 159383552 159383552

8 rows selected.

SQL>
SQL> connect / as sysdba
Connected.
SQL>

```

5) 仍然在以 sys 用户身份连接的 SQL\*Plus 会话中，关闭数据库实例然后再启动：

```

SQL> shutdown immediate;
Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> startup;
ORACLE instance started.

Total System Global Area 292933632 bytes
Fixed Size 1336092 bytes
Variable Size 201329892 bytes
Database Buffers 83886080 bytes
Redo Buffers 6381568 bytes
Database mounted.
Database opened.
SQL>

```

## 练习 16-1：使用自动内存管理（续）

- 6) 在以 AMM 用户身份连接的 SQL\*Plus 会话中，确定各内存缓冲区的当前设置以及自实例启动以来完成的调整大小操作。使用 components.sql 脚本执行此操作。

```
SQL> connect amm/amm
Connected.
SQL> @components
SQL> set echo on
SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;
```

| COMP       | CS        | US |
|------------|-----------|----|
| shared poo | 83886080  | 0  |
| large pool | 4194304   | 0  |
| java pool  | 4194304   | 0  |
| SGA Target | 176160768 | 0  |
| DEFAULT bu | 75497472  | 0  |
| PGA Target | 117440512 | 0  |

6 rows selected.

```
SQL>
SQL> select substr(COMPONENT, 0, 10), FINAL_SIZE, OPER_TYPE,
OPER_MODE, status from v$memory_resize_ops order by
START_TIME;
```

| SUBSTR(COMPONENT,0,10) | FINAL_SIZE | OPER_TYPE    | OPER_MODE | STATUS   |
|------------------------|------------|--------------|-----------|----------|
| DEFAULT 8K             | 0          | STATIC       |           | COMPLETE |
| DEFAULT bu             | 92274688   | STATIC       |           | COMPLETE |
| DEFAULT 2K             | 0          | STATIC       |           | COMPLETE |
| DEFAULT bu             | 92274688   | STATIC       | IMMEDIATE | COMPLETE |
| shared poo             | 67108864   | STATIC       |           | COMPLETE |
| large pool             | 4194304    | STATIC       |           | COMPLETE |
| large pool             | 4194304    | GROW         | IMMEDIATE | COMPLETE |
| java pool              | 4194304    | STATIC       |           | COMPLETE |
| streams po             | 0          | STATIC       |           | COMPLETE |
| SGA Target             | 176160768  | STATIC       |           | COMPLETE |
| DEFAULT 32             | 0          | STATIC       |           | COMPLETE |
| DEFAULT 16             | 0          | STATIC       |           | COMPLETE |
| PGA Target             | 117440512  | STATIC       |           | COMPLETE |
| RECYCLE bu             | 0          | STATIC       |           | COMPLETE |
| KEEP buffe             | 0          | STATIC       |           | COMPLETE |
| ASM Buffer             | 0          | STATIC       |           | COMPLETE |
| DEFAULT 4K             | 0          | STATIC       |           | COMPLETE |
| DEFAULT bu             | 88080384   | STATIC       | IMMEDIATE | COMPLETE |
| DEFAULT bu             | 88080384   | INITIALIZING | IMMEDIATE | COMPLETE |
| shared poo             | 71303168   | GROW         | IMMEDIATE | COMPLETE |

# 练习 16-1：使用自动内存管理（续）

```

shared poo 75497472 GROW IMMEDIATE COMPLETE
DEFAULT bu 83886080 SHRINK IMMEDIATE COMPLETE
shared poo 79691776 GROW IMMEDIATE COMPLETE
DEFAULT bu 79691776 SHRINK IMMEDIATE COMPLETE
DEFAULT bu 75497472 SHRINK IMMEDIATE COMPLETE
shared poo 83886080 GROW IMMEDIATE COMPLETE

26 rows selected.
SQL>

```

- 7) 仍然在以 AMM 用户身份连接的 SQL\*Plus 会话中，执行下面的查询，之后立即再次确定组件大小和调整大小操作。使用 query1.sql 脚本执行此操作。会看到什么结果？

```

select /*+ PARALLEL(s 24) */ count(*) from (select /*+
parallel(s 24) */ * from tabsga s group by a);

```

- a) 您可以看到大型池的大小很大，而缓冲区高速缓存比较小。系统会自动完成内存转移。

```

SQL> @query1
SQL> set echo on
SQL>
SQL> select /*+ PARALLEL(s 24) */ count(*) from (select /*+
parallel(s 24) */ * from tabsga s group by a);

 COUNT(*)

 100000

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;

COMP CS US

shared poo 83886080 0
large pool 83886080 0
java pool 4194304 0
SGA Target 184549376 0
DEFAULT bu 4194304 0
PGA Target 109051904 0

6 rows selected.

SQL>
SQL> select substr(COMPONENT, 0, 10), FINAL_SIZE, OPER_TYPE,
OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

```



# 练习 16-1：使用自动内存管理（续）

| SUBSTR(COMPONENT,0,10) | FINAL_SIZE | OPER_TYPE | OPER_MODE | STATUS   |
|------------------------|------------|-----------|-----------|----------|
| shared poo             | 67108864   | STATIC    |           | COMPLETE |
| large pool             | 4194304    | GROW      | IMMEDIATE | COMPLETE |
| large pool             | 4194304    | STATIC    |           | COMPLETE |
| java pool              | 4194304    | STATIC    |           | COMPLETE |
| streams po             | 0          | STATIC    |           | COMPLETE |
| SGA Target             | 176160768  | STATIC    |           | COMPLETE |
| ASM Buffer             | 0          | STATIC    |           | COMPLETE |
| ...                    |            |           |           |          |
| large pool             | 41943040   | GROW      | IMMEDIATE | COMPLETE |
| large pool             | 37748736   | GROW      | IMMEDIATE | COMPLETE |
| large pool             | 33554432   | GROW      | IMMEDIATE | COMPLETE |
| large pool             | 29360128   | GROW      | IMMEDIATE | COMPLETE |
| SGA Target             | 184549376  | GROW      | IMMEDIATE | COMPLETE |

68 rows selected.

SQL>

- 8) 重复上一个步骤，但这次使用 query2.sql 脚本执行此操作。会看到什么结果？
- a) 会再次出现相同的趋势。

SQL> @query2

SQL> set echo on

SQL>

SQL> select /\*+ PARALLEL(s 25) \*/ count(\*) from (select /\*+ parallel(s 25) \*/ \* from tabsga s group by a);

| COUNT(*) |
|----------|
| 100000   |

SQL>

SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT\_SIZE CS, USER\_SPECIFIED\_SIZE US from v\$memory\_dynamic\_components where CURRENT\_SIZE!=0;

| COMP       | CS        | US |
|------------|-----------|----|
| shared poo | 83886080  | 0  |
| large pool | 83886080  | 0  |
| java pool  | 4194304   | 0  |
| SGA Target | 188743680 | 0  |
| DEFAULT bu | 8388608   | 0  |
| PGA Target | 104857600 | 0  |

6 rows selected.

# 练习 16-1：使用自动内存管理（续）

```

SQL>
SQL> select substr(COMPONENT, 0, 10), FINAL_SIZE, OPER_TYPE,
OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

SUBSTR(COMPONENT,0,10) FINAL_SIZE OPER_TYPE OPER_MODE STATUS

RECYCLE bu 0 STATIC
PGA Target 117440512 STATIC
shared poo 67108864 STATIC
large pool 4194304 STATIC
large pool 4194304 GROW IMMEDIATE COMPLETE
java pool 4194304 STATIC
...
DEFAULT bu 12582912 GROW DEFERRED ERROR
DEFAULT bu 8388608 SHRINK IMMEDIATE COMPLETE
shared poo 83886080 GROW IMMEDIATE COMPLETE
shared poo 79691776 SHRINK DEFERRED ERROR
DEFAULT bu 12582912 GROW DEFERRED ERROR
DEFAULT bu 8388608 SHRINK IMMEDIATE COMPLETE
shared poo 83886080 GROW IMMEDIATE COMPLETE

91 rows selected.

SQL>

```

- 9) 仍然在以 AMM 用户身份连接的 SQL\*Plus 会话中，执行 query3.sql 脚本，之后立即确定内存组件大小及调整大小操作。此脚本的执行时间大约为 10 到 15 分钟，所以您可以休息一下。会看到什么结果？
  - a) 现在您可以看到当 PGA 自动增长时，SGA 会自动收缩。这是因为编号为 3 的查询需要大量不可优化的 PGA 内存。退出 SQL\*Plus。

```

SQL> @query3
SQL> set echo on
SQL>
SQL> exec testpga(800000);

PL/SQL procedure successfully completed.

SQL>
SQL> select substr(COMPONENT, 0, 10) COMP, CURRENT_SIZE CS,
USER_SPECIFIED_SIZE US from v$memory_dynamic_components where
CURRENT_SIZE!=0;

COMP CS US

shared poo 88080384 0
large pool 4194304 0
java pool 4194304 0

```

# 练习 16-1：使用自动内存管理（续）

```

SGA Target 184549376 0
DEFAULT bu 79691776 0
PGA Target 109051904 0

6 rows selected.

SQL>
SQL> select substr(COMPONENT, 0, 10), FINAL_SIZE, OPER_TYPE,
OPER_MODE, status from v$memory_resize_ops order by
START_TIME;

SUBSTR(COMPONENT,0,10)FINAL_SIZE OPER_TYPE OPER_MODE STATUS

RECYCLE bu 0 STATIC COMPLETE
PGA Target 117440512 STATIC COMPLETE
shared poo 67108864 STATIC COMPLETE
...
shared poo 83886080 GROW IMMEDIATE COMPLETE
DEFAULT bu 88080384 SHRINK IMMEDIATE COMPLETE
SGA Target 184549376 SHRINK DEFERRED COMPLETE
DEFAULT bu 79691776 SHRINK DEFERRED COMPLETE
PGA Target 109051904 GROW MANUAL COMPLETE

128 rows selected.

SQL>
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 10) 在终端窗口中重新启动 Enterprise Manager Database Control，观察在此练习期间内存发生的变化。会看到什么结果？

```

$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database
Control started.

Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/ed.us.oracle.com_orcl/sym
an/log
$

```

## 练习 16-1：使用自动内存管理（续）

- a) 转至 “Enterprise Manager ” 主页。
  - b) 单击 “Server（服务器）” 选项卡。
  - c) 在 “Server（服务器）” 子页上，单击 “Database Configuration（数据库配置）” 部分中的 “Memory Advisors（内存指导）”。
  - d) 在 “Memory Advisors（内存指导）” 页上，查看前两个图形。
  - e) 您会看到大型池增长了，随后总 SGA 也增长了，之后 SGA 由于 PGA 的调整而出现了收缩。
  - f) 从 Enterprise Manager Database Console 中注销。
- 11) 要清理环境，请关闭数据库实例，还原原始的 SPFILE，并重新启动数据库实例。要执行上述所有操作，请执行 amm\_cleanup.sh 脚本以撤消您的更改。

```
$ cd /home/oracle/solutions/AMM
$./amm_cleanup.sh
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 08:54:59
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
User dropped.

SQL> SQL>
Tablespace dropped.

SQL> SQL>
Tablespace dropped.
```

## 练习 16-1：使用自动内存管理（续）

```
SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> SQL>
File created.

SQL> SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1337116 bytes
Variable Size 310380772 bytes
Database Buffers 104857600 bytes
Redo Buffers 6094848 bytes
Database mounted.
Database opened.
SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control
..... started.

Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/ed.us.oracle.com_orcl/sysm
an/log
$
```

## 练习 16-2：使用优化程序延迟统计信息

在此练习中，您将处理延迟统计信息发布及统计信息扩展。此练习的基本目的是对特定表测试各种统计信息收集功能，以向生产环境发布最佳的统计信息收集功能。

- 1) 执行 stats\_setup.sh 脚本。此脚本将创建新用户 STATS，同时创建并填充新表 STATS.TABJFV。

```
$ cd /home/oracle/solutions/stats
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$./stats_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 10:24:58
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> drop user stats cascade
*
ERROR at line 1:
ORA-01918: user 'STATS' does not exist

SQL> SQL>
User created.

SQL> SQL>
Grant succeeded.

SQL> SQL> Connected.
SQL> SQL> drop table tabjfv purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL> SQL> 2
Table created.

SQL> SQL> SQL> SQL> 2 3 4 5 6 7 8
PL/SQL procedure successfully completed.
```

## 练习 16-2：使用优化程序延迟统计信息（续）

```
SQL>
Commit complete.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 2) 启动以 STATS 用户身份连接的 SQL\*Plus 会话，并且不要在此会话中断开连接。确保已删除了 STATS.TABJFV 的所有现有统计信息，并进行检查，以确定既不存在公共统计信息也不存在待定统计信息。使用 delete\_stats.sql 脚本执行此操作。

```
$ sqlplus stats/stats

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 10:26:44
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @delete_stats

Session altered.

PL/SQL procedure successfully completed.

old 3: where table_name = '&1'
new 3: where table_name = 'TABJFV'

TABLE_NAME ANALYZE_TIME NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

no rows selected

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'
```

练习 16-2：使用优化程序延迟统计信息（续）

| COLUMN_NAME                                                   | ANALYZE_TIME | NUM_DISTINCT | NUM_NULLS | DENSITY |
|---------------------------------------------------------------|--------------|--------------|-----------|---------|
| C1                                                            |              |              |           |         |
| C2                                                            |              |              |           |         |
| C3                                                            |              |              |           |         |
| C4                                                            |              |              |           |         |
| C5                                                            |              |              |           |         |
| C6                                                            |              |              |           |         |
| C7                                                            |              |              |           |         |
| C8                                                            |              |              |           |         |
| C9                                                            |              |              |           |         |
| 9 rows selected.                                              |              |              |           |         |
| SQL>                                                          |              |              |           |         |
| SQL>                                                          |              |              |           |         |
| SQL> -- private statistics: none                              |              |              |           |         |
| SQL> @ show_pending_stats TABJFV                              |              |              |           |         |
| SQL>                                                          |              |              |           |         |
| SQL> set echo off                                             |              |              |           |         |
| old 3: where table_name = '&1' and partition_name is null     |              |              |           |         |
| new 3: where table_name = 'TABJFV' and partition_name is null |              |              |           |         |
| no rows selected                                              |              |              |           |         |
| old 4: where table_name = '&1' and partition_name is null     |              |              |           |         |
| new 4: where table_name = 'TABJFV' and partition_name is null |              |              |           |         |
| no rows selected                                              |              |              |           |         |
| old 4: where table_name = '&1' and partition_name is null     |              |              |           |         |
| new 4: where table_name = 'TABJFV' and partition_name is null |              |              |           |         |
| no rows selected                                              |              |              |           |         |
| SQL>                                                          |              |              |           |         |
| SQL>                                                          |              |              |           |         |



## 练习 16-2：使用优化程序延迟统计信息（续）

- 3) 确定 STATS.TABJFV 统计信息的发布模式，将其设置为 PENDING 模式。使用 pending\_mode.sql 脚本执行此操作。

```
SQL>
SQL> @pending_mode
SQL> -- table publish mode should be TRUE, for now
SQL> select dbms_stats.get_prefs('PUBLISH', 'stats', 'tabjfv')
publish from dual;

PUBLISH

TRUE

SQL>
SQL> -- set table value to false, i.e, do not publish
statistics
SQL> exec dbms_stats.set_table_prefs('stats', 'tabjfv',
'PUBLISH', 'FALSE');

PL/SQL procedure successfully completed.

SQL>
SQL> -- table publish mode should be FALSE now
SQL> select dbms_stats.get_prefs('PUBLISH', 'stats', 'tabjfv')
publish from dual;

PUBLISH

FALSE

SQL>
SQL>
```

- 4) 收集 STATS.TABJFV 的统计信息并调查结果。会看到什么结果？使用 collect\_pending.sql 脚本执行此操作。
- a) 只能在特定区查看统计信息。还要注意，默认情况下，会话仅使用公共统计信息。

```
SQL> @collect_pending
SQL> -- Collect stats
SQL> begin
2 dbms_stats.gather_table_stats(null, 'tabjfv',
3 method_opt => 'for all columns size 1 for columns c1
size 254 for columns c2 size 254');
4 end;
5 /

PL/SQL procedure successfully completed.

SQL>
```

练习 16-2：使用优化程序延迟统计信息（续）

```
SQL> -- public statistics: still nothing
SQL> @ show_public_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1'
new 3: where table_name = 'TABJFV'

TABLE_NAME ANALYZE_TIME NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

no rows selected

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

COLUMN_NAME ANALYZE_TIME NUM_DISTINCT NUM_NULLS DENSITY

C1
C2
C3
C4
C5
C6
C7
C8
C9

9 rows selected.

SQL>
SQL>
SQL> -- private statistics: should see values (0.001 for
density for both C1 and C2, 0.2 for others)
SQL> @ show_pending_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1' and partition_name is null
new 3: where table_name = 'TABJFV' and partition_name is
null

TABLE_NAME analyze time NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV 10/07 10:34:00 500 5 27
```

# 练习 16-2：使用优化程序延迟统计信息（续）

```
old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null

no rows selected

old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null
```

| COLUMN_NAME | analyze time   | NUM_DISTINCT | NUM_NULLS | DENSITY |
|-------------|----------------|--------------|-----------|---------|
| C1          | 10/07 10:33:59 | 5            | 0         | .001    |
| C2          | 10/07 10:33:59 | 5            | 0         | .001    |
| C3          | 10/07 10:33:59 | 5            | 0         | .2      |
| C4          | 10/07 10:33:59 | 5            | 0         | .2      |
| C5          | 10/07 10:33:59 | 5            | 0         | .2      |
| C6          | 10/07 10:33:59 | 5            | 0         | .2      |
| C7          | 10/07 10:33:59 | 5            | 0         | .2      |
| C8          | 10/07 10:33:59 | 5            | 0         | .2      |
| C9          | 10/07 10:33:59 | 5            | 0         | .2      |

```
9 rows selected.

SQL>
SQL>
SQL> -- Should be false: use public statistics
SQL> show parameter optimizer_use_pending_statistics
```

| NAME                             | TYPE    | VALUE |
|----------------------------------|---------|-------|
| optimizer_use_pending_statistics | boolean | FALSE |

```
SQL>
SQL>
```

- 在终端窗口中，打开一个以 STATS 用户身份连接的 SQL\*Plus 会话（完成此步骤后不要退出此会话！），在此会话中禁用动态采样，然后确定优化程序当前会对以下查询的行数得出什么估计值：  

```
select count(*) from tabjfv where c1 = 1 and c2 = 1;
```

 您可得出什么结论？使用 `determine1.sql` 脚本执行此步骤。
  - 由于未使用动态采样，并且表目前不存在公共统计信息，所以优化程序对于查询返回的行数的猜测是错误的：它使用的是 1，而不是 100。

```
SQL> @determine1
SQL> set echo on
SQL>
SQL> alter session set optimizer_dynamic_sampling = 0;

Session altered.
```

# 练习 16-2：使用优化程序延迟统计信息（续）

```

SQL>
SQL> select count(*) from tabjfv where c1 = 1 and c2 = 1;

 COUNT(*)

 100

SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
 2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

| Id | Operation | Name | Rows |

0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
2	TABLE ACCESS FULL	TABJFV	1

9 rows selected.

SQL>

```

6) 现在，将会话切换为使用先前收集的待定统计信息。

```

SQL> alter session set optimizer_use_pending_statistics =
true;

Session altered.

SQL>

```

# 练习 16-2：使用优化程序延迟统计信息（续）

- 7) 再次确定优化程序对查询返回的行数的估计值。会看到什么结果？使用 `determine2.sql` 脚本执行此步骤。
  - a) 优化程序再次提供了一个错误的估计值。这是由于优化程序无法猜出 `c1` 和 `c2` 与值 1 是高度相关的。

```
SQL> @determine2
SQL> set echo on
SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
 2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

| Id | Operation | Name | Rows |

0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
2	TABLE ACCESS FULL	TABJFV	20

9 rows selected.

SQL>
```

- 8) 对组 C1 和 C2 创建统计信息扩展，指明这两列在 `STATS.TABJFV` 中都是相关的。完成后，再次为扩展收集表的最高精度统计信息。使用 `create_stats_extension.sql` 脚本执行上述操作。

```
SQL> @create_stats_extension
SQL> col table_name format a10
SQL> col extension format a12
SQL>
SQL> select * from user_stat_extensions;

no rows selected
```

## 练习 16-2：使用优化程序延迟统计信息（续）

```

SQL>
SQL> -- Create extention on the 2 columns
SQL> select
dbms_stats.create_extended_stats(null,'tabjfv','(c1,c2)') from
dual;

DBMS_STATS.CREATE_EXTENDED_STATS(NULL,'TABJFV','(C1,C2)')

SYS_STUF3GLKIOP5F4B0BTTCTMX0W

SQL>
SQL> select * from user_stat_extensions;

TABLE_NAME EXTENSION_NAME EXTENSION CREATOR DRO

TABJFV SYS_STUF3GLKIOP5F4B0BTTCTMX0W ("C1","C2") USER YES

SQL>
SQL> -- Collect stats
SQL> begin
 2 dbms_stats.gather_table_stats(null, 'tabjfv',
 3 method_opt => 'for all columns size 1 for columns
(c1,c2) size 254');
 4 end;
 5 /

PL/SQL procedure successfully completed.

SQL>
SQL> -- public statistics: still nothing
SQL> @ show_public_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1'
new 3: where table_name = 'TABJFV'

TABLE_NAME ANALYZE_TIME NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

no rows selected

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

```

练习 16-2：使用优化程序延迟统计信息（续）

| COLUMN_NAME                                                                                                                  | ANALYZE_TIME   | NUM_DISTINCT   | NUM_NULLS    | DENSITY     |
|------------------------------------------------------------------------------------------------------------------------------|----------------|----------------|--------------|-------------|
| C1                                                                                                                           |                |                |              |             |
| C2                                                                                                                           |                |                |              |             |
| C3                                                                                                                           |                |                |              |             |
| C4                                                                                                                           |                |                |              |             |
| C5                                                                                                                           |                |                |              |             |
| C6                                                                                                                           |                |                |              |             |
| C7                                                                                                                           |                |                |              |             |
| C8                                                                                                                           |                |                |              |             |
| C9                                                                                                                           |                |                |              |             |
| 9 rows selected.                                                                                                             |                |                |              |             |
| SQL>                                                                                                                         |                |                |              |             |
| SQL>                                                                                                                         |                |                |              |             |
| SQL> -- private statistics: should see values and also the new virtual column! Density=0.2 for all except virtual one: 0.001 |                |                |              |             |
| SQL> @ show_pending_stats TABJFV                                                                                             |                |                |              |             |
| SQL>                                                                                                                         |                |                |              |             |
| SQL> set echo off                                                                                                            |                |                |              |             |
| old 3: where table_name = '&1' and partition_name is null                                                                    |                |                |              |             |
| new 3: where table_name = 'TABJFV' and partition_name is null                                                                |                |                |              |             |
|                                                                                                                              |                |                |              |             |
| TABLE_NAME                                                                                                                   | analyze time   | NUM_ROWS       | BLOCKS       | AVG_ROW_LEN |
| TABJFV                                                                                                                       | 10/07 10:44:59 | 500            | 5            | 39          |
| old 4: where table_name = '&1' and partition_name is null                                                                    |                |                |              |             |
| new 4: where table_name = 'TABJFV' and partition_name is null                                                                |                |                |              |             |
| no rows selected                                                                                                             |                |                |              |             |
| old 4: where table_name = '&1' and partition_name is null                                                                    |                |                |              |             |
| new 4: where table_name = 'TABJFV' and partition_name is null                                                                |                |                |              |             |
|                                                                                                                              |                |                |              |             |
| COLUMN_NAME                                                                                                                  |                | analyze time   | NUM_DISTINCT |             |
| NUM_NULLS                                                                                                                    | DENSITY        |                |              |             |
| C1                                                                                                                           |                | 10/07 10:44:59 | 5            |             |
| 0                                                                                                                            | .2             |                |              |             |
| C2                                                                                                                           |                | 10/07 10:44:59 | 5            |             |
| 0                                                                                                                            | .2             |                |              |             |
| C3                                                                                                                           |                | 10/07 10:44:59 | 5            |             |
| 0                                                                                                                            | .2             |                |              |             |

练习 16-2：使用优化程序延迟统计信息（续）

```
C4 10/07 10:44:59 5
0 .2
C5 10/07 10:44:59 5
0 .2
C6 10/07 10:44:59 5
0 .2
C7 10/07 10:44:59 5
0 .2
C8 10/07 10:44:59 5
0 .2
C9 10/07 10:44:59 5
0 .2
SYS_STUF3GLKIOP5F4B0BTTTCFTMX0W 10/07 10:44:59 5
0 .001

10 rows selected.

SQL>
```

- 9) 再次确定优化程序对查询返回的行数的估计值。会看到什么结果？（使用 `determine2.sql` 脚本。）
- a) 这一次，应看到正确的估计值。

```
SQL> @determine2
SQL> set echo on
SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
 2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

| Id | Operation | Name | Rows |

0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
2	TABLE ACCESS FULL	TABJFV	100

```



## 练习 16-2：使用优化程序延迟统计信息（续）

```
9 rows selected.
```

```
SQL>
```

- 10) 在 SQL\*Plus 会话中执行 add\_rows.sql 脚本。完成后，表将包含五个不同的值，每个值重复 200 次（相同）。所有列都具有相同的值。该表还有 200 个空行。

```
SQL> @add_rows
SQL> -- Insert some more rows. Now the table has 5 values each
repeating 200
SQL> -- times (uniform). All columns have same value.
SQL> begin
 2 for i in 1..5 loop
 3 for j in 1..100 loop
 4 insert into tabjfv(c1,c2,c3,c4,c5,c6,c7,c8,c9)
values (i,i,i,i,i,i,i,i,i,i);
 5 end loop;
 6 end loop;
 7 end;
 8 /
```

```
PL/SQL procedure successfully completed.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL>
```

```
SQL>
```

```
SQL> -- Insert 200 null rows
```

```
SQL> begin
 2 for j in 1..200 loop
 3 insert into tabjfv(c1) values (null);
 4 end loop;
 5 end;
 6 /
```

```
PL/SQL procedure successfully completed.
```

```
SQL> commit;
```

```
Commit complete.
```

```
SQL>
```

## 练习 16-2：使用优化程序延迟统计信息（续）

- 11) 针对以下三种情况，再次确定优化程序对查询返回的行数的估计值：不使用动态采样、使用动态采样级别 3 和使用动态采样级别 4。分别会看到什么结果？使用 `show_dynamic_stats.sql` 脚本执行此操作。
- a) 不使用动态采样时，由于统计信息尚未刷新，所以现在的估计值是不正确的。动态采样级别 3 不足以得出更好的估计值。使用动态采样级别 4 时，可得出更好的估计值。

```
SQL> @show_dynamic_stats
SQL> -- Try without dynamic sampling
SQL> --
SQL> explain plan for select * from tabjfv where c1 = 1 and c2
= 1;

Explained.

SQL>
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 349803950

| Id | Operation | Name | Rows |

| 0 | SELECT STATEMENT | | 100 |
| 1 | TABLE ACCESS FULL | TABJFV | 100 |

8 rows selected.

SQL>
SQL>
SQL> -- Try the queries with dynamic sampling level 3. Dynamic
sampling will not
SQL> -- kick in. It will use multi column stats and produce
the same estimate
SQL> -- (100 rows) as before.
SQL> explain plan for select /*+ dynamic_sampling(3) */ * from
tabjfv where c1 = 1 and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));
```

## 练习 16-2：使用优化程序延迟统计信息（续）

```
PLAN_TABLE_OUTPUT

Plan hash value: 349803950

| Id | Operation | Name | Rows |

| 0 | SELECT STATEMENT | | 100 |
| 1 | TABLE ACCESS FULL| TABJFV | 100 |

8 rows selected.

SQL>
SQL>
SQL> -- Try with level 4. Dynamic sampling kicks in. we should
see the improved
SQL> -- estimate (200 rows)
SQL> explain plan for select /*+ dynamic_sampling(4) */ * from
tabjfv where c1 = 1 and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'BASIC ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 349803950

| Id | Operation | Name | Rows |

| 0 | SELECT STATEMENT | | 200 |
| 1 | TABLE ACCESS FULL| TABJFV | 200 |

8 rows selected.

SQL>
```

# 练习 16-2：使用优化程序延迟统计信息（续）

12) 再次为表收集特定统计信息。使用 collect\_pending2.sql 脚本执行此操作。

```
SQL> @collect_pending2
SQL> -- Collect stats
SQL> begin
 2 dbms_stats.gather_table_stats(null, 'tabjfv',
 3 method_opt => 'for all columns size 1 for columns
(c1,c2) size 254');
 4 end;
 5 /

PL/SQL procedure successfully completed.

SQL>
SQL> -- public statistics: still nothing
SQL> @ show_public_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1'
new 3: where table_name = 'TABJFV'

TABLE_NAME ANALYZE_TIME NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

no rows selected

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

COLUMN_NAME ANALYZE_TIME NUM_DISTINCT NUM_NULLS DENSITY

C1
C2
C3
C4
C5
C6
C7
C8
C9

9 rows selected.
```

练习 16-2：使用优化程序延迟统计信息（续）

```
SQL>
SQL>
SQL> -- private statistics: should see values (0.001 for
density for both C1 and C2, 0.2 for others)
SQL> @ show_pending_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1' and partition_name is null
new 3: where table_name = 'TABJFV' and partition_name is
null

TABLE_NAME analyze time NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV 10/07 10:54:28 1200 5 37

old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null

no rows selected

old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null

COLUMN_NAME analyze time NUM_DISTINCT
NUM_NULLS DENSITY

C1 10/07 10:54:28 5
200 .2
C2 10/07 10:54:28 5
200 .2
C3 10/07 10:54:28 5
200 .2
C4 10/07 10:54:28 5
200 .2
C5 10/07 10:54:28 5
200 .2
C6 10/07 10:54:28 5
200 .2
C7 10/07 10:54:28 5
200 .2
C8 10/07 10:54:28 5
200 .2
C9 10/07 10:54:28 5
200 .2
SYS_STUF3GLKIOP5F4B0BTTCFTMXOW 10/07 10:54:28 5
200 .0005
```

# 练习 16-2：使用优化程序延迟统计信息（续）

```

10 rows selected.

SQL>
SQL>
SQL> -- Should be false: use public statistics
SQL> show parameter optimizer_use_pending_statistics

NAME TYPE VALUE
----- -
optimizer_use_pending_statistics boolean TRUE
SQL>

```

- 13) 再次确定优化程序对查询返回的行数的估计值。会看到什么结果？（使用 determine2.sql。）
- a) 此次估计是正确的。

```

SQL> @determine2
SQL> set echo on
SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
 2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

| Id | Operation | Name | Rows |

0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
2	TABLE ACCESS FULL	TABJFV	200

9 rows selected.

SQL>
SQL>

```

练习 16-2：使用优化程序延迟统计信息（续）

14) 将会话切换为公共统计信息模式。

```
SQL> alter session set optimizer_use_pending_statistics =
false;

Session altered.

SQL>
```

15) 再次确定优化程序对查询返回的行数的估计值。会看到什么结果？（使用 determine2.sql。）

a) 由于尚无公共统计信息，并且动态采样尚未开启，所以此估计是错误的。

```
SQL> @determine2
SQL> set echo on
SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

| Id | Operation | Name | Rows |

0	SELECT STATEMENT		1
1	SORT AGGREGATE		1
2	TABLE ACCESS FULL	TABJFV	1

9 rows selected.

SQL>
SQL>
```

# 练习 16-2：使用优化程序延迟统计信息（续）

16) 现在您对这些统计信息感到满意，将它们发布到公共级别。使用 publish\_stats.sql 脚本执行此步骤。

```

SQL> @publish_stats
SQL> set echo on
SQL>
SQL> exec dbms_stats.publish_pending_stats('stats','tabjfv');

PL/SQL procedure successfully completed.

SQL>
SQL> @ show_public_stats TABJFV
SQL>
SQL> set echo off
old 3: where table_name = '&1'
new 3: where table_name = 'TABJFV'

TABLE_NAME ANALYZE_TIME NUM_ROWS BLOCKS AVG_ROW_LEN

TABJFV 10/07 10:54:28 1200 5 37

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

no rows selected

old 4: where table_name = '&1'
new 4: where table_name = 'TABJFV'

COLUMN_NAME ANALYZE_TIME NUM_DISTINCT NUM_NULLS DENSITY

C1 10/07 10:54:28 5 200 .2
C2 10/07 10:54:28 5 200 .2
C3 10/07 10:54:28 5 200 .2
C4 10/07 10:54:28 5 200 .2
C5 10/07 10:54:28 5 200 .2
C6 10/07 10:54:28 5 200 .2
C7 10/07 10:54:28 5 200 .2
C8 10/07 10:54:28 5 200 .2
C9 10/07 10:54:28 5 200 .2

9 rows selected.

SQL>
SQL>
SQL> @ show_pending_stats TABJFV
SQL>
SQL> set echo off

```



**练习 16-2：使用优化程序延迟统计信息（续）**

```

old 3: where table_name = '&1' and partition_name is null
new 3: where table_name = 'TABJFV' and partition_name is
null

no rows selected

old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null

no rows selected

old 4: where table_name = '&1' and partition_name is null
new 4: where table_name = 'TABJFV' and partition_name is
null

no rows selected

SQL>
SQL>

```

17) 再次确定优化程序对查询返回的行数的估计值。会看到什么结果？（使用 `determine2.sql`。）

a) 这次的统计信息估计是正确的！

```

SQL> @determine2
SQL> set echo on
SQL>
SQL> explain plan for select count(*) from tabjfv where c1 = 1
and c2 = 1;

Explained.

SQL>
SQL> select plan_table_output
 2 from table(dbms_xplan.display('plan_table',null,'BASIC
ROWS'));

PLAN_TABLE_OUTPUT

Plan hash value: 3122694931

```

# 练习 16-2：使用优化程序延迟统计信息（续）

| Id | Operation         | Name   | Rows |
|----|-------------------|--------|------|
| 0  | SELECT STATEMENT  |        | 1    |
| 1  | SORT AGGREGATE    |        | 1    |
| 2  | TABLE ACCESS FULL | TABJFV | 200  |

9 rows selected.

SQL>

18) 确定 STATS.TABJFV 统计信息的发布模式，将其设置回 PUBLIC 模式（使用 `determine_publish.sql`）。退出 SQL\*Plus。

```

SQL>
SQL> @determine_publish
SQL> select dbms_stats.get_prefs('PUBLISH', 'stats', 'tabjfv')
publish
 2 from dual;

PUBLISH

FALSE

SQL>
SQL> exec
dbms_stats.set_table_prefs('stats','tabjfv','PUBLISH','TRUE');

PL/SQL procedure successfully completed.

SQL>
SQL> select dbms_stats.get_prefs('PUBLISH', 'stats', 'tabjfv')
publish
 2 from dual;

PUBLISH

TRUE
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

## 练习 16-2：使用优化程序延迟统计信息（续）

19) 执行 stats\_cleanup.sh 脚本清理您的环境。

```
$./stats_cleanup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 11:07:20
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL>
User dropped.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 第 17 课的练习

在本课的练习中，您将使用结果高速缓存。

## 练习 17-1：使用结果高速缓存

在本练习中，您将了解 SGA 中高速缓存查询结果的多种可能性。执行以下步骤了解如何使用查询结果高速缓存。

- 1) 转到 \$HOME/solutions/qrc 目录，然后执行 result\_cache\_setup.sh 脚本。

```
$ cd /home/oracle/solutions/qrc
$
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$./result_cache_setup.sh

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 11:16:29
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
SQL> SQL> SQL> SQL> drop user qrc cascade
*
ERROR at line 1:
ORA-01918: user 'QRC' does not exist

SQL> SQL> 2 3
User created.

SQL> SQL>
Grant succeeded.

SQL> SQL> Connected.
SQL> SQL>
PL/SQL procedure successfully completed.

SQL> SQL> drop table cachejfv purge
*
ERROR at line 1:
ORA-00942: table or view does not exist

SQL> SQL>
Table created.
```

## 练习 17-1：使用结果高速缓存（续）

```

SQL> SQL>
1 row created.

SQL>
1 row created.

SQL>
2 rows created.

SQL>
4 rows created.

...

SQL>
524288 rows created.

SQL> SQL>
1 row created.

SQL> SQL>
Commit complete.

SQL> SQL>
System altered.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 2) 在终端窗口中，以 QRC 用户身份登录 SQL\*Plus。从现在开始，不要断开与此会话的连接。使用以下语句确定查询高速缓存的当前内容：

```

select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

```

可以使用 check\_result\_cache.sql 脚本。会看到什么结果？现在，查询高速缓存应为空。

```

$ sqlplus qrc/qrc

SQL*Plus: Release 11.2.0.1.0 Production on Wed Oct 7 11:21:43
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production

```

## 练习 17-1：使用结果高速缓存（续）

```
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @check_result_cache
SQL>
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

no rows selected

SQL>
```

- 3) 设置时间，然后执行以下查询。可以使用 query1.sql 脚本。记下执行所用的时间。

```
select /*+ result_cache q_name(Q1) */ count(*)
from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv c4,
cachejfv c5
where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b'
and c5.c='b';
```

```
SQL> set timing on
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
c5.c='b';

COUNT(*)

1

Elapsed: 00:00:01.64
SQL>
```

- 4) 使用 explain\_query1.sql 脚本确定上次查询的执行计划。会看到什么结果？由于 result\_cache 提示，将使用结果高速缓存计算查询结果。

```
SQL> @explain_query1
SQL> set echo on
SQL>
SQL> explain plan for select /*+ result_cache q_name(Q1) */
count(*) from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5 where c1.c='b' and c2.c='b' and c3.c='b' and
c4.c='b' and c5.c='b';

Explained.
```

# 练习 17-1：使用结果高速缓存（续）

```

Elapsed: 00:00:00.20
SQL>
SQL> set linesize 180
SQL>
SQL> @/u01/app/oracle/acfsmounts/acfs_db1/rdbms/admin/utlxpls
SQL> Rem
SQL> Rem $Header: utlxpls.sql 26-feb-2002.19:49:37 bdagevil
..
SQL> Rem
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'serial'));

PLAN_TABLE_OUTPUT

Plan hash value: 2522916280

| Id | Operation | Name |
Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		
1	1260	2295G (1)	999:59:59
1	RESULT CACHE	c334km80wg4dq0418y5p1frmnz	
2	SORT AGGREGATE		
1	1260		
3	MERGE JOIN CARTESIAN		
92G	105T	2295G (1)	999:59:59
4	MERGE JOIN CARTESIAN		
591M	555G	14G (1)	999:59:59
5	MERGE JOIN CARTESIAN		
3794K	2735M	94M (1)	314:36:35

PLAN_TABLE_OUTPUT

6	MERGE JOIN CARTESIAN		
24326	11M	605K (1)	02:01:03
* 7	TABLE ACCESS FULL	CACHEJFV	
156	39312	3856 (1)	00:00:47
8	BUFFER SORT		
156	39312	601K (1)	02:00:17
* 9	TABLE ACCESS FULL	CACHEJFV	
156	39312	3855 (1)	00:00:47
10	BUFFER SORT		
156	39312	94M (1)	314:35:49
* 11	TABLE ACCESS FULL	CACHEJFV	
156	39312	3855 (1)	00:00:47

```



练习 17-1：使用结果高速缓存（续）

```
| 12 | BUFFER SORT |
| 156 | 39312 | 14G (1)|999:59:59 |
|* 13 | TABLE ACCESS FULL | CACHEJFV
156	39312	3855 (1)	00:00:47
14	BUFFER SORT		
156	39312	2295G (1)	999:59:59
* 15	TABLE ACCESS FULL	CACHEJFV	
156	39312	3855 (1)	00:00:47

PLAN_TABLE_OUTPUT

Predicate Information (identified by operation id):

 7 - filter("C1"."C"='b')
 9 - filter("C2"."C"='b')
 11 - filter("C3"."C"='b')
 13 - filter("C4"."C"='b')
 15 - filter("C5"."C"='b')

Result Cache Information (identified by operation id):

PLAN_TABLE_OUTPUT

 1 - column-count=1; dependencies=(QRC.CACHEJFV);
 attributes=(single-row); parameters=(nls); name="select /*+
 result_cache q_name(Q1) */ count(*) from cachejfv c1,cachejfv
 c2,cac
 hejfv c3,cachejfv c4,cachejfv c5 where c1.c='b' a"

附注

 - dynamic sampling used for this statement (level=2)

40 rows selected.

Elapsed: 00:00:00.63
SQL>
```

## 练习 17-1：使用结果高速缓存（续）

- 5) 使用 `check_result_cache.sql` 脚本确定查询高速缓存的当前内容。会看到什么结果？现在可以看到高速缓存的查询结果。

```
SQL> @check_result_cache
SQL> set echo on
SQL>
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

TYPE STATUS NAME
OBJECT_NO ROW_COUNT

ROW_SIZE_AVG

Dependency Published QRC.CACHEJFV
76960 0
 0

Result Published select /*+ result_cache q_name(Q1) */
count(*)
0 1
 from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
 where c1.c='b' a
 5

Elapsed: 00:00:00.00
SQL>
```

- 6) 刷新实例的缓冲区高速缓存，然后重新运行在步骤 3 中执行的查询。会看到什么结果？现在，查询的执行几乎在瞬间就会完成。

```
SQL> alter system flush buffer_cache;

System altered.

Elapsed: 00:00:00.01
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
c5.c='b';
 COUNT(*)

 1

Elapsed: 00:00:00.00
SQL>
```

# 练习 17-1：使用结果高速缓存（续）

- 使用以下语句向 CACHEJFV 表中插入新行：insert into cachejfv values('c');。会看到什么结果？相应的结果高速缓存条目会自动失效。

```
SQL> insert into cachejfv values('c');

1 row created.

Elapsed: 00:00:00.00
SQL> commit;

Commit complete.

Elapsed: 00:00:00.01
SQL> @check_result_cache
SQL> set echo on
SQL>
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;
```

| TYPE         | STATUS    | NAME                                  |
|--------------|-----------|---------------------------------------|
| OBJECT_NO    | ROW_COUNT |                                       |
| -----        | -----     | -----                                 |
| -----        | -----     | -----                                 |
| ROW_SIZE_AVG |           |                                       |
| -----        |           |                                       |
| Dependency   | Published | QRC.CACHEJFV                          |
| 76960        | 0         |                                       |
|              | 0         |                                       |
| Result       | Invalid   | select /*+ result_cache q_name(Q1) */ |
| count(*)     |           |                                       |
| 0            | 1         |                                       |
|              |           | from cachejfv c1,cachejfv c2,cachejfv |
|              |           | c3,cachejfv c4,cachejfv c5            |
|              |           | where c1.c='b' a                      |
|              | 5         |                                       |

```
Elapsed: 00:00:00.00
SQL>
```

练习 17-1：使用结果高速缓存（续）

- 8) 再次执行第一个查询，然后检查结果高速缓存。会看到什么结果？执行查询仍需要花费一些时间。查看结果高速缓存，会看到新的结果中已添加了新条目。

```
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
c5.c='b';

COUNT(*)

 1

Elapsed: 00:00:01.36
SQL>
SQL> @check_result_cache
SQL> set echo on
SQL>
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

TYPE STATUS NAME
OBJECT_NO ROW_COUNT

ROW_SIZE_AVG

Dependency Published QRC.CACHEJFV
76960 0
 0

Result Invalid select /*+ result_cache q_name(Q1) */
count(*)
0 1
 from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
 where c1.c='b' a
 5

Result Published select /*+ result_cache q_name(Q1) */
count(*)
0 1

TYPE STATUS NAME
OBJECT_NO ROW_COUNT


```

## 练习 17-1：使用结果高速缓存（续）

```

ROW_SIZE_AVG

 from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
 where c1.c='b' a

 5

Elapsed: 00:00:00.01
SQL>

```

- 9) 系统将生成详细的结果高速缓存内存报表。

```

SQL> set serveroutput on
SQL> EXEC DBMS_RESULT_CACHE.MEMORY_REPORT(detailed=>true);
Result Cache Memory Report
[Parameters]
Block Size = 1K bytes
Maximum Cache Size = 2080K bytes (2080 blocks)
Maximum Result Size = 104K bytes (104 blocks)
[Memory]
Total Memory = 107836 bytes [0.041% of the Shared Pool]
... Fixed Memory = 9440 bytes [0.004% of the Shared Pool]
..... Memory Mgr = 124 bytes
..... Bloom Fltr = 2K bytes
..... Cache Mgr = 4416 bytes
..... State Objs = 2852 bytes
... Dynamic Memory = 98396 bytes [0.038% of the Shared Pool]
..... Overhead = 65628 bytes
..... Hash Table = 32K bytes (4K buckets)
..... Chunk Ptrs = 12K bytes (3K slots)
..... Chunk Maps = 12K bytes
..... Miscellaneous = 8284 bytes
..... Cache Memory = 32K bytes (32 blocks)
..... Unused Memory = 29 blocks
..... Used Memory = 3 blocks
..... Dependencies = 1 blocks (1 count)
..... Results = 2 blocks
..... SQL = 1 blocks (1 count)
..... Invalid = 1 blocks (1 count)

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.33
SQL>

```

## 练习 17-1：使用结果高速缓存（续）

10) 再次执行查询。会看到什么结果？查询会再次使用以前高速缓存的结果。

```
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
 3 c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
 3 c5.c='b';

 COUNT(*)

 1

Elapsed: 00:00:00.01
SQL>
```

11) 在执行下一个步骤前，确保绕过结果高速缓存。

```
SQL> exec DBMS_RESULT_CACHE.BYPASS(bypass_mode=>true);

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.00
SQL>
```

12) 再次执行查询。会看到什么结果？因为这次查询不再使用结果高速缓存，所以执行查询又会花费较长的时间。

```
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
 3 c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
 3 c5.c='b';

 COUNT(*)

 1

Elapsed: 00:00:01.36
SQL>
```

## 练习 17-1：使用结果高速缓存（续）

13) 确保不再绕过结果高速缓存，然后验证查询是否重新使用结果高速缓存。

```
SQL> exec DBMS_RESULT_CACHE.BYPASS(bypass_mode=>false);

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.00
SQL> @query1
SQL> select /*+ result_cache q_name(Q1) */ count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
c5.c='b';

 COUNT(*)

 1

Elapsed: 00:00:00.01
SQL>
```

14) 使用 query2.sql 脚本执行以下查询：

```
select count(*) from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b'
and c5.c='b';
```

会看到什么结果？虽然此查询与步骤 3 中使用的查询相同，但是因为不包含提示，所以不会将其识别为高速缓存的查询。因此，该查询所用的执行时间依然很长。

```
SQL> @query2
SQL> select count(*)
 2 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
 3 where c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and
c5.c='b';

 COUNT(*)

 1

Elapsed: 00:00:01.38
SQL>
```

### 练习 17-1：使用结果高速缓存（续）

- 15) 如何强制前一个查询使用高速缓存结果而不使用提示？使用  
force\_query2.sql 脚本执行前一个查询，然后确认已成功使用了高速缓存  
的结果。最后，撤消更改。

```
SQL> @force_query2
SQL> set echo on
SQL>
SQL> show parameter result_cache_mode
```

| NAME              | TYPE   | VALUE  |
|-------------------|--------|--------|
| result_cache_mode | string | MANUAL |

```
SQL>
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;
```

| TYPE         | STATUS    | NAME                                  |
|--------------|-----------|---------------------------------------|
| OBJECT_NO    | ROW_COUNT |                                       |
| -----        |           |                                       |
| -----        |           |                                       |
| -----        |           |                                       |
| ROW_SIZE_AVG |           |                                       |
| -----        |           |                                       |
| Dependency   | Published | QRC.CACHEJFV                          |
| 76960        | 0         |                                       |
|              | 0         |                                       |
| Result       | Invalid   | select /*+ result_cache q_name(Q1) */ |
| count(*)     |           |                                       |
| 0            | 1         |                                       |
|              |           | from cachejfv c1,cachejfv c2,cachejfv |
|              |           | c3,cachejfv c4,cachejfv c5            |
|              |           | where c1.c='b' a                      |
|              | 5         |                                       |
| Result       | Published | select /*+ result_cache q_name(Q1) */ |
| count(*)     |           |                                       |
| 0            | 1         |                                       |
| TYPE         | STATUS    | NAME                                  |
| OBJECT_NO    | ROW_COUNT |                                       |
| -----        |           |                                       |
| -----        |           |                                       |
| -----        |           |                                       |
| ROW_SIZE_AVG |           |                                       |
| -----        |           |                                       |



### 练习 17-1：使用结果高速缓存（续）

```

 from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
 where c1.c='b' a
 5

Elapsed: 00:00:00.00
SQL>
SQL> alter session set result_cache_mode=force;

Session altered.

Elapsed: 00:00:00.01
SQL>
SQL> explain plan for select count(*) from cachejfv
c1,cachejfv c2,cachejfv c3,cachejfv c4,cachejfv c5 where
c1.c='b' and c2.c='b' and c3.c='b' and c4.c='b' and c5.c='b';

Explained.

Elapsed: 00:00:00.03
SQL>
SQL> @/u01/app/oracle/acfsmounts/acfs_db1/rdbms/admin/utlxpls
..
SQL> Rem
SQL> select plan_table_output from
table(dbms_xplan.display('plan_table',null,'serial'));

PLAN_TABLE_OUTPUT

Plan hash value: 2522916280

| Id | Operation | Name
Rows	Bytes	Cost (%CPU)	Time
0	SELECT STATEMENT		
1	1260	2295G (1)	999:59:59
1	RESULT CACHE	c334km80wg4dq0418y5p1frmnz	
2	SORT AGGREGATE		
1	1260		
3	MERGE JOIN CARTESIAN		
92G	105T	2295G (1)	999:59:59
4	MERGE JOIN CARTESIAN		
591M	555G	14G (1)	999:59:59
5	MERGE JOIN CARTESIAN		
3794K	2735M	94M (1)	314:36:35
```

练习 17-1：使用结果高速缓存（续）

|                                                               |                              |
|---------------------------------------------------------------|------------------------------|
| PLAN_TABLE_OUTPUT                                             |                              |
| -----                                                         |                              |
| 6                                                             | MERGE JOIN CARTESIAN         |
| 24326                                                         | 11M  605K (1)  02:01:03      |
| * 7                                                           | TABLE ACCESS FULL   CACHEJFV |
| 156   39312                                                   | 3856 (1)  00:00:47           |
| 8                                                             | BUFFER SORT                  |
| 156   39312                                                   | 601K (1)  02:00:17           |
| * 9                                                           | TABLE ACCESS FULL   CACHEJFV |
| 156   39312                                                   | 3855 (1)  00:00:47           |
| 10                                                            | BUFFER SORT                  |
| 156   39312                                                   | 94M (1)  314:35:49           |
| * 11                                                          | TABLE ACCESS FULL   CACHEJFV |
| 156   39312                                                   | 3855 (1)  00:00:47           |
| 12                                                            | BUFFER SORT                  |
| 156   39312                                                   | 14G (1)  999:59:59           |
| * 13                                                          | TABLE ACCESS FULL   CACHEJFV |
| 156   39312                                                   | 3855 (1)  00:00:47           |
| 14                                                            | BUFFER SORT                  |
| 156   39312                                                   | 2295G (1)  999:59:59         |
| * 15                                                          | TABLE ACCESS FULL   CACHEJFV |
| 156   39312                                                   | 3855 (1)  00:00:47           |
| -----                                                         |                              |
| PLAN_TABLE_OUTPUT                                             |                              |
| -----                                                         |                              |
| Predicate Information (identified by operation id):           |                              |
| -----                                                         |                              |
| 7 - filter("C1"."C"='b')                                      |                              |
| 9 - filter("C2"."C"='b')                                      |                              |
| 11 - filter("C3"."C"='b')                                     |                              |
| 13 - filter("C4"."C"='b')                                     |                              |
| 15 - filter("C5"."C"='b')                                     |                              |
| Result Cache Information (identified by operation id):        |                              |
| -----                                                         |                              |
| PLAN_TABLE_OUTPUT                                             |                              |
| -----                                                         |                              |
| 1 - column-count=1; dependencies=(QRC.CACHEJFV);              |                              |
| attributes=(single-row); parameters=(nls); name="select /*+   |                              |
| result_cache q_name(Q1) */ count(*) from cachejfv c1,cachejfv |                              |
| c2,cac                                                        |                              |
| hejfv c3,cachejfv c4,cachejfv c5 where c1.c='b' a"            |                              |



**练习 17-1：使用结果高速缓存（续）**

```

ROW_SIZE_AVG

 from cachejfv c1,cachejfv c2,cachejfv
c3,cachejfv c4,cachejfv c5
 where c1.c='b' a
 5

Elapsed: 00:00:00.01
SQL>
SQL> alter session set result_cache_mode=manual;

Session altered.

Elapsed: 00:00:00.00
SQL>

```

16) 清除结果高速缓存。查询 V\$RESULT\_CACHE\_OBJECTS 以确认清除操作。

```

SQL> exec dbms_result_cache.flush;

PL/SQL procedure successfully completed.

Elapsed: 00:00:00.00
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

no rows selected

Elapsed: 00:00:00.00
SQL>

```

17) 运行 cre\_func.sql 脚本创建使用结果高速缓存的 PL/SQL 函数。

```

SQL> @cre_func
SQL> create or replace function CACHEJFV_COUNT(v varchar2)
2 return number
3 result_cache relies_on (cachejfv)
4 is
5 cnt number;
6 begin
7 select count(*) into cnt
8 from cachejfv c1,cachejfv c2,cachejfv c3,cachejfv
c4,cachejfv c5
9 where c1.c=v and c2.c=v and c3.c=v and c4.c=v and
c5.c=v;
10
11 return cnt;
12 end;
13 /

```

练习 17-1：使用结果高速缓存（续）

```
Function created.

Elapsed: 00:00:01.26
SQL>
```

18) 通过查询 V\$RESULT\_CACHE\_OBJECTS 确定结果高速缓存中的内容。

```
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

no rows selected

Elapsed: 00:00:00.00
SQL>
```

19) 调用新函数，使用 'b' 作为其参数。会看到什么结果？因为尚未高速缓存结果，所以执行此次查询又一次花费了较长的时间。执行完此函数后，将高速缓存函数针对参数 'b' 的结果。

```
SQL> select cachejfv_count('b') from dual;

CACHEJFV_COUNT('B')

1

Elapsed: 00:00:02.84
SQL> select type,status,name,object_no,row_count,row_size_avg
from v$result_cache_objects order by 1;

TYPE STATUS NAME
OBJECT_NO ROW_COUNT

ROW_SIZE_AVG

Dependency Published QRC.CACHEJFV
76960 0
0

Dependency Published QRC.CACHEJFV_COUNT
76961 0
0

Result Published
"QRC"."CACHEJFV_COUNT"::8."CACHEJFV_COUNT"#8440831613f0f5d3 #1
0 1
4

Elapsed: 00:00:00.00
SQL>
```



练习 17-1：使用结果高速缓存（续）

```
TYPE STATUS NAME
OBJECT_NO ROW_COUNT

ROW_SIZE_AVG

Result Published
"QRC"."CACHEJFV_COUNT":8."CACHEJFV_COUNT"#8440831613f0f5d3 #1
0 1
 4

Elapsed: 00:00:00.01
SQL> select cachejfv_count('c') from dual;

CACHEJFV_COUNT('C')

 1

Elapsed: 00:00:00.00
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 第 18 课的练习

在本课的练习中，您将复制活动数据库并合并恢复目录。



## 练习 18-1：复制活动数据库

您负责一个无法关闭的活动数据库。它正在 ARCHIVELOG 模式下运行。为进行测试，现在您需要将此数据库复制到另一个目录。

为模拟此环境，假设活动数据库是 orcl（该数据库存储在 ASM 中）。

- 1) 开始克隆数据库前，请先将最后两个 ASM 磁盘添加到 DATA 磁盘组中。
  - a) 在以 oracle 用户身份连接的图形终端窗口中，将环境设置为使用 +ASM 实例，然后调用 ASMCA。

```
$. oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$
$ asmca
```

- b) 在 “Configure ASM: Disk Groups（配置 ASM：磁盘组）” 子页上，右键单击 DATA 磁盘组。
  - c) 从显示的菜单中选择 “Add disks（添加磁盘）”。
  - d) 在 “Add Disks（添加磁盘）” 页上，选择 ORCL:ASMDISK12 和 ORCL:ASMDISK13。
  - e) 单击 “OK（确定）”。
  - f) 在显示的信息窗口中单击 “OK（确定）”。
  - g) 返回 “Configure ASM: Disk Groups（配置 ASM：磁盘组）” 子页，单击 “Exit（退出）”。
  - h) 在 “ASM Configuration Assistant” 窗口中，单击 “Yes（是）”。
- 2) 等待重新平衡操作完成。
  - a) 以 SYS 用户身份登录 Enterprise Manager。
  - b) 在主页上，单击 “General（一般信息）” 部分中的 +ASM 链接。
  - c) 在 ASM 主页上，单击 “Disk Groups（磁盘组）” 子选项卡。
  - d) 如果尚未登录 ASM 实例，则需要先登录：在 “Automatic Storage Management Login（自动存储管理登录）” 页上，在 “Username（用户名）” 字段中输入 SYS，在 “Password（口令）” 字段中输入 oracle\_4U，在 “Connect As（连接身份）” 字段中输入 SYSASM。  
选择 “Save as Preferred Credential（另存为首选身份证明）”。  
单击 “Login（登录）”。

## 练习 18-1：复制活动数据库（续）

- e) 在 “Disks Groups（磁盘组）” 页上，单击 DATA 链接。
  - f) 在 “Disk Group: DATA（磁盘组：DATA）” 页上，查看 “General（一般信息）” 部分中的 “Pending Operations（暂挂操作）” 字段。
  - g) 刷新浏览器页，直至不再有暂挂操作。
- 3) 为确保指向 orcl 数据库，同时此数据库处于 ARCHIVELOG 模式且有 7 GB 的快速恢复区，请执行 /home/oracle/solutions/RMAN 目录下的 rman\_archivelog.sh 脚本。

```
$ cd /home/oracle/solutions/RMAN
$./rman_archivelog.sh

For demo purposes ONLY:
 * Enable ARCHIVELOG mode for database

The script may appear to hang at the SQL prompt
when the database is shutting down and being
opened.Wait a few minutes and it should progress.

orcl

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 09:27:02
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 314575232 bytes
Database Buffers 100663296 bytes
Redo Buffers 6094848 bytes
Database mounted.
SQL>
Database altered.
```

## 练习 18-1：复制活动数据库（续）

```
SQL>
Database altered.

SQL> Database log mode Archive Mode
Automatic archival Enabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 99
Next log sequence to archive 101
Current log sequence 101
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 4) 将 dbtest 设置为 dbtest 计划数据库的网络服务名。

- a) 在以 oracle 用户身份连接的图形终端窗口中，将环境设置为指向 orcl 实例，然后调用 NETCA。

```
$. oraenv
ORACLE_SID = [+ASM] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ netca
```

- b) Oracle Net Configuration Assistant (NETCA) 将打开一个窗口。在 “Welcome（欢迎使用）” 页上，选择 “Local Net Service Name configuration（本地 Net 服务名配置）”，然后单击 “Next（下一步）”。
- c) 在 “Net Service Name Configuration（网络服务名配置）” 页上，选择 “Add（添加）”，然后单击 “Next（下一步）”。
- d) 在 “Service Name（服务名）” 字段中，输入 dbtest，然后单击 “Next（下一步）”。
- e) 在 “Net Service Name Configuration, Select Protocol（网络服务名配置，请选择协议）” 页上，选择 TCP，然后单击 “Next（下一步）”。
- f) 在 “Net Service Name Configuration, TCP/IP Protocol（网络服务名配置，TCP/IP 协议）” 页上输入主机名（例如 edt3r10p10.us.oracle.com），选择 “Use the standard port number of 1521（使用标准端口号 1521）”，然后单击 “Next（下一步）”。
- g) 在 “Net Service Name Configuration, Test（网络服务名配置，测试）” 页上，选择 “No, do not test（不，不进行测试）”（因为 dbtest 数据库尚不存在），然后单击 “Next（下一步）”。

## 练习 18-1：复制活动数据库（续）

- h) 在 “Net Service Name Configuration, Net Service Name（网络服务名配置，网络服务名）” 页上，输入 dbtest 作为网络服务名，然后单击 “Next（下一步）”。
  - i) 单击 “No（否）” 回答问题 “Would you like to configure another net service name?（是否配置另一个网络服务名？）”，然后单击 “Next（下一步）”。
  - j) 看到完成消息后，再次单击 “Next（下一步）”。
  - k) 最后，单击 “Finish（完成）”。
- 5) 使用 Oracle Net Manager 配置具有 dbtest 和 orcl 数据库服务的 LISTENER。
- a) 在以 oracle 用户身份连接的图形终端窗口中，将环境设置为指向 +ASM 实例，然后调用 NETMGR。

```
$. oraenv
ORACLE_SID = [orcl] ? +ASM
The Oracle base for
ORACLE_HOME=/u01/app/oracle/product/11.2.0/grid is
/u01/app/oracle
$ netmgr
```

- b) Oracle Net Manager 将打开一个窗口。单击 “Local（本地）” 前方的 “+” 图标。然后单击 “Listeners（监听程序）” 前方的 “+” 图标，展开导航树中的节点，直至看到名为 LISTENER 的监听程序。
- c) 首先，在窗口左侧选择 LISTENER。然后从窗口右侧的下拉列表中选择 “Database Services（数据库服务）”。
- d) 单击 “Add Database（添加数据库）” 按钮。
- e) 输入以下值：  
Global Database Name（全局数据库名）：dbtest  
Oracle Home Directory（Oracle 主目录）：  
/u01/app/oracle/acfsmounts/acfs\_db1/  
SID: dbtest
- f) 再次单击 “Add Database（添加数据库）” 按钮。
- g) 输入以下值：  
Global Database Name（全局数据库名）：orcl  
Oracle Home Directory（Oracle 主目录）：  
/u01/app/oracle/acfsmounts/acfs\_db1/  
SID: orcl
- h) 从 Oracle Net Manager 菜单栏中，选择 “File > Save Network Configuration（文件 > 保存网络配置）”，然后选择 “File > Exit（文件 > 退出）”。

## 练习 18-1：复制活动数据库（续）

- 6) 使用 RMAN 克隆数据库。在 Enterprise Manager 中启动复制进程。
- 以 SYS 用户身份和 `oracle_4U` 口令登录 Enterprise Manager，然后以 SYSDBA 身份进行连接。
  - 首先单击“Data Movement（数据移动）”选项卡，然后单击“Move Database Files（移动数据库文件）”部分中的“Clone Database（克隆数据库）”。
  - 在“Clone Database: Source Type（克隆数据库：源类型）”页上，选择“A running database（正在运行的数据库）”和“Use Recovery Manager (RMAN) to copy database files（使用 Recovery Manager (RMAN) 复制数据库文件）”，然后单击“Continue（继续）”。
  - 在“Clone Database: Source Options（克隆数据库：源选项）”页上，输入 `oracle` 作为用户名和口令，单击“Save as Preferred Credential（另存为首选身份证明）”，然后单击“Next（下一步）”。
  - 在“Clone Database: Select Destination（克隆数据库：选择目标）”页上，在“Global Database Name（全局数据库名）”和“Instance Name（实例名称）”中输入 `dbtest`，然后从“Database Storage（数据库存储）”下拉列表中选择“Automatic Storage Management (ASM)（自动存储管理 (ASM)）”。然后，单击“Next（下一步）”。
  - 如果显示“Clone Database: ASM Instance Login（克隆数据库：ASM 实例登录）”页，则请输入 `oracle_4U` 作为 SYS 的口令，然后单击“Login（登录）”。
  - 在“Clone Database: Destination Options（克隆数据库：目标选项）”页的“Flash Recovery Area（快速恢复区）”字段中输入 `FRA`。然后，单击“Next（下一步）”。
  - 在“Clone Database: Database Configuration（克隆数据库：数据库配置）”页上，
    - 选择“Configure Enterprise Manager Database Control for this database（为此数据库配置 Enterprise Manager Database Control）”。
    - 确保在“Configuration File Location（配置文件位置）”字段中输入 `/u01/app/oracle/acfsmounts/acfs_db1/network/admin`。
    - 在“Listener Oracle Home（监听程序 Oracle 主目录）”字段中输入 `/u01/app/oracle/product/11.2.0/grid`。
    - 在所有口令字段中输入 `oracle_4U` 六次，并输入 5505 作为 HTTP 端口。
    - 单击“Next（下一步）”。

## 练习 18-1：复制活动数据库（续）

- i) 如果出现警告，指示 sqlnet.ora 文件或 listener.ora 文件不存在，请单击 “Yes（是）” 接受该警告。
  - j) 在 “Clone Database: Schedule（克隆数据库：调度）” 页上，确保该作业立即启动。单击 “Next（下一步）”。
  - k) 在 “Clone Database: Review（克隆数据库：复查）” 页上，复查 “Details（详细资料）”（其中包括 “Database Storage（数据库存储）”），然后单击 “Submit Job（提交作业）”。
  - l) 在 “Clone Database job is being submitted（正在提交克隆数据库作业）” 页上稍等片刻。
  - m) “Clone Database: Confirmation（克隆数据库：确认）” 页应显示一条成功消息。单击 “View Status（查看状态）” 转到 “Job Activity（作业活动）” 页。
  - n) 此时将转到 “Execution（执行）” 页。有时，需要一直单击浏览器的重新加载按钮显示作业进度，直到出现成功消息。（作业的执行时间取决于硬件和可用的系统资源。）
- 7) 在 SQL\*Plus 中测试对克隆数据库的访问。
- a) 如果以 oracle 用户身份在图形终端会话中连接，请确保您指向 orcl 数据库。

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$
```

- b) 以 SYS 用户身份连接到 orcl 数据库并执行以下查询：

```
select dbid, name, created, open_mode
from v$database;
```

```
$ sqlplus sys/oracle_4U@orcl as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 10:49:40
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select dbid, name, created, open_mode from v$database;
```

# 练习 18-1：复制活动数据库（续）

| DBID       | NAME | CREATED   | OPEN_MODE  |
|------------|------|-----------|------------|
| 1226549444 | ORCL | 28-SEP-09 | READ WRITE |

SQL>

c) 现在以 SYSTEM 用户身份和 oracle\_4U 口令连接 dbtest 数据库并执行上述查询。退出 SQL\*Plus。

SQL> **connect system/oracle\_4U@dbtest**  
Connected.  
SQL> **select dbid, name, created, open\_mode from v\$database;**

| DBID       | NAME   | CREATED   | OPEN_MODE  |
|------------|--------|-----------|------------|
| 1092566820 | DBTEST | 08-OCT-09 | READ WRITE |

SQL> **exit**  
Disconnected from Oracle Database 11g Enterprise Edition  
Release 11.2.0.1.0 - Production  
With the Partitioning, Automatic Storage Management, OLAP,  
Data Mining and Real Application Testing options  
\$

## 练习 18-2：管理目录

在本练习中，您会将多个 RMAN 目录合并到一个数据库中。

- 1) 通过执行工作目录 (/home/oracle/solutions/RMAN) 中的 setup\_cat.sh 脚本，在 orcl 实例和 dbtest 实例中创建一个目录，为此场景做准备。

```
$ cd /home/oracle/solutions/RMAN
$
$./setup_cat.sh

For training purposes ONLY, not for production use
 * Setup RMAN catalogs in ORCL and DBTEST instances
 Execute as oracle OS user orcl

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 11:20:59
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL>
SQL> /*== Confirm database connection ==*/
SQL> SELECT NAME FROM V$DATABASE;

NAME

ORCL

SQL>
SQL> /*== Create RCAT tablespace of 10 MB ==*/
SQL> DROP TABLESPACE RCAT including contents and datafiles;
DROP TABLESPACE RCAT including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'RCAT' does not exist

SQL>
SQL> CREATE TABLESPACE RCAT
 2 DATAFILE '+DATA' SIZE 10M;

Tablespace created.
```



## 练习 18-2: 管理目录 (续)

```
SQL>
SQL> /*== Create CATOWNER user with "oracle" password ==*/
SQL> DROP USER catowner CASCADE;
DROP USER catowner CASCADE
 *
ERROR at line 1:
ORA-01918: user 'CATOWNER' does not exist

SQL>
SQL> create user catowner identified by oracle_4U
 2 default tablespace rcat
 3 quota unlimited on rcat;

User created.

SQL>
SQL> /*== Grant privileges to CATOWNER ==*/
SQL> GRANT CREATE SESSION to catowner;

Grant succeeded.

SQL>
SQL> GRANT RECOVERY_CATALOG_OWNER to catowner;

Grant succeeded.

SQL> PROMPT "Setup of CATOWNER user with "oracle_4U" password
and RCAT tablespace complete."
"Setup of CATOWNER user with "oracle_4U" password and RCAT
tablespace complete."
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

Recovery Manager: Release 11.2.0.1.0 - Production on Thu Oct 8
11:21:14 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1226549444)
connected to recovery catalog database

RMAN>
RMAN> CREATE CATALOG TABLESPACE RCAT;
2>
3> REGISTER DATABASE;
```

## 练习 18-2: 管理目录 (续)

```
4>
5> exit;
recovery catalog created

database registered in recovery catalog
starting full resync of recovery catalog
full resync complete

Recovery Manager complete.
dbtest

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 11:22:28
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL>
SQL> /*== Confirm database connection ==*/
SQL> SELECT NAME FROM V$DATABASE;

NAME

DBTEST

SQL>
SQL> /*== Create RCAT tablespace of 10 MB ==*/
SQL> DROP TABLESPACE RCAT including contents and datafiles;
DROP TABLESPACE RCAT including contents and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'RCAT' does not exist

SQL>
SQL> CREATE TABLESPACE RCAT
 2 DATAFILE '+DATA' SIZE 10M;

Tablespace created.
```

## 练习 18-2: 管理目录 (续)

```
SQL>
SQL> /*== Create CATOWNER user with "oracle_4U" password ==*/
SQL> DROP USER catowner CASCADE;
DROP USER catowner CASCADE
 *
ERROR at line 1:
ORA-01918: user 'CATOWNER' does not exist

SQL>
SQL> create user catowner identified by oracle_4U
 2 default tablespace rcat
 3 quota unlimited on rcat;

User created.

SQL>
SQL> /*== Grant privileges to CATOWNER ==*/
SQL> GRANT CREATE SESSION to catowner;

Grant succeeded.

SQL>
SQL> GRANT RECOVERY_CATALOG_OWNER to catowner;

Grant succeeded.

SQL> PROMPT "Setup for RMAN catalog in DBTEST complete."
"Setup for RMAN catalog in DBTEST complete."
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

Recovery Manager: Release 11.2.0.1.0 - Production on Thu Oct 8
11:22:35 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: DBTEST (DBID=1092566820)
connected to recovery catalog database

RMAN>
RMAN> CREATE CATALOG TABLESPACE RCAT;
2>
3> REGISTER DATABASE;
4>
5> exit;
recovery catalog created
```

## 练习 18-2: 管理目录 (续)

```
database registered in recovery catalog
starting full resync of recovery catalog
full resync complete
```

```
Recovery Manager complete.
$
```

- 2) 从 orcl 数据库备份 EXAMPLE 表空间, 并将其记录在 RMAN 目录中。

- a) 确保指向 orcl 数据库。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
```

- b) 使用 RMAN 和 RMAN 目录从 orcl 数据库备份 EXAMPLE 表空间。执行以下命令:

```
rman target sys/oracle_4U catalog catowner/oracle_4U
backup tablespace example;
exit
```

```
$ rman target sys/oracle_4U catalog catowner/oracle_4U

Recovery Manager: Release 11.2.0.1.0 - Production on Thu Oct 8
11:34:05 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID= 1226549444)
connected to recovery catalog database

RMAN> backup tablespace example;

Starting backup at 08-OCT-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=53 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00005
name=+DATA/orcl/datafile/example.265.698768805
channel ORA_DISK_1: starting piece 1 at 08-OCT-09
channel ORA_DISK_1: finished piece 1 at 08-OCT-09
piece
handle=+FRA/orcl/backupset/2009_10_08/nnndf0_tag20091008t11350
4_0.287.699708907 tag=TAG20091008T113504 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:16
```

## 练习 18-2：管理目录（续）

```
Finished backup at 08-OCT-09
```

```
RMAN> exit
```

```
Recovery Manager complete.
```

```
$
```

### 3) 从 dbtest 数据库备份 EXAMPLE 表空间。

#### a) 确保您指向 dbtest 数据库。

```
$. oraenv
```

```
ORACLE_SID = [orcl] ? dbtest
```

```
The Oracle base for
```

```
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
```

```
/u01/app/oracle
```

```
$
```

#### b) 使用 RMAN 和 RMAN 目录从 dbtest 数据库备份 EXAMPLE 表空间。执行以下命令：

```
rman target sys/oracle_4U@dbtest catalog
```

```
catowner/oracle_4U@dbtest
```

```
backup tablespace example;
```

```
exit
```

```
$ rman target sys/oracle_4U@dbtest catalog
```

```
catowner/oracle_4U@dbtest
```

```
Recovery Manager: Release 11.2.0.1.0 - Production on Thu Oct 8
11:37:19 2009
```

```
Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.
```

```
connected to target database: DBTEST (DBID=1092566820)
```

```
connected to recovery catalog database
```

```
RMAN> backup tablespace example;
```

```
Starting backup at 08-OCT-09
```

```
allocated channel: ORA_DISK_1
```

```
channel ORA_DISK_1: SID=54 device type=DISK
```

```
channel ORA_DISK_1: starting full datafile backup set
```

```
channel ORA_DISK_1: specifying datafile(s) in backup set
```

```
input datafile file number=00005
```

```
name=+DATA/dbtest/datafile/example.274.699703705
```

```
channel ORA_DISK_1: starting piece 1 at 08-OCT-09
```

```
channel ORA_DISK_1: finished piece 1 at 08-OCT-09
```

```
piece
```

```
handle=+FRA/dbtest/backupset/2009_10_08/nnndf0_tag20091008t113
```

```
801_0.288.699709083 tag=TAG20091008T113801 comment=NONE
```

## 练习 18-2: 管理目录 (续)

```
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:15
Finished backup at 08-OCT-09

RMAN> exit

Recovery Manager complete.
$
```

- 4) 将 dbtest 目录合并到 orcl 数据库目录中。

- a) 确保指向 orcl 数据库。

```
$. oraenv
ORACLE_SID = [dbtest] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$
```

- b) 执行以下命令:

```
rman catalog catowner/oracle_4U
IMPORT CATALOG catowner/oracle_4U@dbtest;
```

```
$ rman catalog catowner/oracle_4U

Recovery Manager: Release 11.2.0.1.0 - Production on Thu Oct 8
11:39:36 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to recovery catalog database

RMAN> IMPORT CATALOG catowner/oracle_4U@dbtest;

Starting import catalog at 08-OCT-09
connected to source recovery catalog database
import validation complete
database unregistered from the source recovery catalog
Finished import catalog at 08-OCT-09

RMAN>
```

# 练习 18-2：管理目录（续）

5) 从 orcl 目录报告 dbtest 备份。确定在 orcl RMAN 资料档案库中注册了哪些数据库。列出为 dbtest 数据库记录的备份。

a) 在 RMAN 提示符下，列出在 orcl 资料档案库中注册的原型。

```

RMAN> LIST INCARNATION;

List of Database Incarnations
DB Key Inc Key DB Name DB ID STATUS Reset SCN Reset Time

167 189 DBTEST 1092566820 PARENT 754488 28-SEP-09
167 169 DBTEST 1092566820 CURRENT 2678383 08-OCT-09
2 23 ORCL 1226549444 PARENT 1 13-AUG-09
2 4 ORCL 1226549444 CURRENT 754488 28-SEP-09
RMAN>

```

b) 在 RMAN 提示符下，列出记录的 dbtest 数据库备份。然后退出 RMAN。

```

RMAN> connect target sys/oracle_4U@dbtest

connected to target database: DBTEST (DBID= 1092566820)

RMAN> list backup;

List of Backup Sets
=====

BS Key Type LV Size Device Type Elapsed Time Completion
Time

234 Full 69.57M DISK 00:00:08 08-OCT-09
 BP Key: 236 Status: AVAILABLE Compressed: NO Tag:
TAG20091008T113801
 Piece Name:
+FRA/dbtest/backupset/2009_10_08/nnndf0_tag20091008t113801_0.2
88.699709083
 List of Datafiles in backup set 234
 File LV Type Ckp SCN Ckp Time Name

 5 Full 2834274 08-OCT-09
+DATA/dbtest/datafile/example.274.699703705

RMAN> exit

Recovery Manager complete.
$

```

## 练习 18-2: 管理目录 (续)

- 6) 执行 `cleanup_rcat.sh` 脚本关闭 `dbtest` 实例，删除操作系统文件，并对 `orcl` 数据库禁用 `ARCHIVELOG` 模式。

```
$./cleanup_rcat.sh
For training purposes ONLY, not for production use
 Shutdown DBTEST database and delete recovery files
 Execute as oracle OS user
dbtest
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5505/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
... Stopped.

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 11:48:21
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining
and Real Application Testing options

SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
orcl

SQL*Plus: Release 11.2.0.1.0 Production on Thu Oct 8 11:48:40
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
```



## 练习 18-2：管理目录（续）

```
SQL> SQL>
User dropped.

SQL>
Tablespace dropped.

SQL> SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Database log mode No Archive Mode
Automatic archival Disabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 102
Current log sequence 104
SQL> SQL> "Cleanup finished."
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 第 19 课的练习

在本练习中，您将使用闪回数据归档和闪回事务处理功能。使用闪回数据归档功能时，您将以三个不同的角色执行任务：

- 以具有 SYSDBA 权限的 SYS 用户身份，处理表空间、用户、权限和还原区。
- 以 ARCHIVE\_ADMIN 用户身份，创建和维护闪回数据归档（在给定的表空间中）并向 HR 用户授予 FLASHBACK ARCHIVE 对象权限。
- 以 HR 用户身份，对 HR.EMPLOYEES 表启用历史记录跟踪，创建事务处理，在切换还原表空间后（证明您从闪回数据归档而不是从还原区检索信息）对以前的内容执行查询，并使用闪回数据归档中的一行更新当前的“错误”行。

## 练习 19-1：使用闪回数据归档

在本练习中，您将使用闪回数据归档。

- 1) 确保指向 orcl 数据库。使用 SQL\*Plus，以 SYS 用户身份连接到数据库并执行 /home/oracle/solutions/FDA 目录中的 flada\_setup.sql 脚本。安装脚本会创建一个小型 FLA\_TBS1 表空间、口令为 ARCHIVE\_ADMIN 的 ARCHIVE\_ADMIN 用户，并对口令为 HR 的 HR 用户取消锁定。默认情况下，口令区分大小写。该脚本还会创建一个名为 HR.EMPLOYEES2 的表，本练习中将使用该表。

```
$. oraenv
ORACLE_SID = [db10g] ? orcl
$ cd /home/oracle/solutions/FDA
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 07:21:07
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @flada_setup
Connected.
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on
SQL>
SQL> /*== Create a tablespace for your flashback data archive
==*/
SQL> DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS
2 /
DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS
*
ERROR at line 1:
ORA-00959: tablespace 'FLA_TBS1' does not exist

SQL> CREATE SMALLFILE TABLESPACE fla_tbs1
2 DATAFILE '+DATA'
3 SIZE 10M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M
```

## 练习 19-1：使用闪回数据归档（续）

```
4 NOLOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
5 /

Tablespace created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> /*== Set up the HR database account for this lesson ==*/
SQL> /*== Note: The HR user has the UNLIMITED TABLESPACE
system privilege. ==*/
SQL>
SQL> ALTER USER hr IDENTIFIED BY "HR" ACCOUNT UNLOCK
2 /

User altered.

SQL> /*== Create an ARCHIVE_ADMIN user like the HR user ==*/
SQL> /*== with FLA_TBS1 default tablespace ==*/
SQL>
SQL> CREATE USER ARCHIVE_ADMIN PROFILE DEFAULT IDENTIFIED BY
ARCHIVE_ADMIN
2 DEFAULT TABLESPACE FLA_TBS1 TEMPORARY TABLESPACE TEMP
3 ACCOUNT UNLOCK
4 /

User created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> GRANT ALTER SESSION TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT CREATE DATABASE LINK TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT CREATE SEQUENCE TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT CREATE SESSION TO ARCHIVE_ADMIN;

Grant succeeded.
```

## 练习 19-1：使用闪回数据归档（续）

```
SQL> GRANT CREATE SYNONYM TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT CREATE VIEW TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT UNLIMITED TABLESPACE TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT EXECUTE ON SYS.DBMS_STATS TO ARCHIVE_ADMIN;

Grant succeeded.

SQL> GRANT CONNECT, RESOURCE TO ARCHIVE_ADMIN;

Grant succeeded.

SQL>
SQL> /*== copy employees table for demo purposes ==*/
SQL>
SQL> connect hr/HR
Connected.
SQL>
SQL> drop table employees2 purge;
drop table employees2 purge
 *
ERROR at line 1:
ORA-00942: table or view does not exist

SQL>
SQL> create table employees2 as select * from employees;

Table created.

SQL>
SQL> CONNECT / AS SYSDBA
Connected.
SQL>
SQL> /*== Setup for Flashback Data Archive completed ==*/
SQL> /*== The ARCHIVE_ADMIN user has the password:
ARCHIVE_ADMIN ==*/
SQL> /*== The HR user has the password: HR ==*/
SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

## 练习 19-1：使用闪回数据归档（续）

```
SQL>
SQL> set pause off
SQL> set feedback on
SQL>
```

- 2) 要授予 ARCHIVE\_ADMIN 用户管理权限以创建、维护和删除闪回数据归档，请以 SYS 用户身份执行以下命令：

```
GRANT FLASHBACK ARCHIVE ADMINISTER TO archive_admin;
```

```
SQL> GRANT FLASHBACK ARCHIVE ADMINISTER TO archive_admin;

Grant succeeded.

SQL>
```

- 3) 在 SQL\*Plus 中，以 ARCHIVE\_ADMIN 用户身份使用 ARCHIVE\_ADMIN 口令进行连接。

**注：**口令必须为大写；用户名不区分大小写。

```
SQL> connect archive_admin/ARCHIVE_ADMIN
Connected.
SQL>
```

- 4) 要创建闪回数据归档，请执行 **create\_flal.sql** 脚本或执行以下命令：

```
CREATE FLASHBACK ARCHIVE flal
TABLESPACE fla_tbs1
QUOTA 10M
RETENTION 1 YEAR;
```

```
SQL> @create_flal
SQL> set echo on
SQL> DROP FLASHBACK ARCHIVE flal
2 /
DROP FLASHBACK ARCHIVE flal
*
ERROR at line 1:
ORA-55605: Incorrect Flashback Archive is specified

SQL> CREATE FLASHBACK ARCHIVE flal
2 TABLESPACE fla_tbs1
3 QUOTA 10M
4 RETENTION 1 YEAR
5
SQL> /

Flashback archive created.

SQL>
```

## 练习 19-1：使用闪回数据归档（续）

- 5) 通过执行以下命令，授予 HR 用户使用 FLA1 归档的权限：

```
GRANT FLASHBACK ARCHIVE on FLA1 to HR;
```

```
SQL> GRANT FLASHBACK ARCHIVE on FLA1 to HR;

Grant succeeded.

SQL>
```

- 6) 现在，切换到闪回归档用户角色。以 HR 用户身份使用 HR 口令进行连接。要为 EMPLOYEES 表启用闪回数据归档，请执行以下命令：

```
ALTER TABLE hr.employees FLASHBACK ARCHIVE fla1;
```

```
SQL> connect HR/HR
Connected.
SQL> ALTER TABLE hr.employees2 FLASHBACK ARCHIVE fla1;

Table altered.

SQL>
```

- 7) 要查看 Fox 先生的薪金并为其加薪三次，每次增加 1000，请执行 flada\_dml.sql 脚本。此时会在闪回数据归档中生成活动。

```
SQL> @flada_dml
SQL> REM "***** "
SQL> REM "For demo purposes ONLY: Flashback Data Archive "
SQL>
SQL> connect hr/HR
Connected.
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue..."
SQL>
SQL> /*== Query the current salary for Mr. Fox ==*/
SQL>
SQL> SELECT employee_id, last_name, salary
2 FROM hr.employees2
3 WHERE last_name = 'Fox'
4 /
Press [Enter] to continue...
```

## 练习 19-1：使用闪回数据归档（续）

```

EMPLOYEE_ID LAST_NAME SALARY

 170 Fox 9600

1 row selected.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
SQL>
SQL> /*== Increase the salary three times by 1000 ==*/
SQL>
SQL> UPDATE hr.employees2
 2 SET salary = salary + 1000
 3 WHERE last_name = 'Fox'
 4 /

1 row updated.

SQL> COMMIT
 2 /

Commit complete.

SQL> UPDATE hr.employees2
 2 SET salary = salary + 1000
 3 WHERE last_name = 'Fox'
 4 /

1 row updated.

SQL> COMMIT
 2 /

Commit complete.

SQL> UPDATE hr.employees2
 2 SET salary = salary + 1000
 3 WHERE last_name = 'Fox'
 4 /

1 row updated.

SQL> COMMIT
 2 /

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```



# 练习 19-1：使用闪回数据归档（续）

```

SQL>
SQL> /*== Query the up-to-date value for Mr. Fox ==*/
SQL>
SQL> SELECT employee_id, last_name, salary
 2 FROM hr.employees2
 3 WHERE last_name = 'Fox'
 4 /
Press [Enter] to continue...

EMPLOYEE_ID LAST_NAME SALARY

 170 Fox 12600

1 row selected.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL>
SQL> set pause off
SQL> set feedback on
SQL>

```

8) 要查询归档表的内部名称，请执行以下命令：

```

SELECT * FROM USER_FLASHBACK_ARCHIVE_TABLES;

```

```

SQL> SELECT * FROM USER_FLASHBACK_ARCHIVE_TABLES;

TABLE_NAME OWNER_NAME FLASHBACK_ARCHIVE_NAME

ARCHIVE_TABLE_NAME STATUS

EMPLOYEES2 HR FLA1
SYS_FBA_HIST_75036 ENABLED

1 row selected.

SQL>

```

## 练习 19-1：使用闪回数据归档（续）

- 9) 以 HR 用户身份，选择在创建闪回数据归档之后到执行错误 DML 之前这段时间内的一个时间。要查看 Fox 先生在该时间的雇员记录，请执行以下查询（用您所选的历史日期替换 '10' MINUTE，格式示例如下：'50' SECOND、'10' DAY、'5' MONTH）：

**注：**如果指定了启动闪回数据归档之前的时间，将出现 ORA-1466 错误。缩短时间间隔，然后重试。如果仍然看到薪金为 12600，请增加时间间隔。

```
SELECT employee_id, last_name, salary
FROM hr.employees2 AS OF TIMESTAMP
 (SYSTIMESTAMP - INTERVAL '10' MINUTE)
WHERE last_name = 'Fox';
```

```
SQL> connect hr/HR
Connected.
SQL> SELECT employee_id, last_name, salary
FROM hr.employees2 AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL
'10' MINUTE)
WHERE last_name = 'Fox'; 2 3

EMPLOYEE_ID LAST_NAME SALARY

 170 Fox 9600

1 row selected.

SQL>
```

- 10) 作为 HR 用户，您认识到最近的更新是错误的。要还原到所选历史日期的原始值（例如，十分钟前的），请执行以下命令（使用您所选的历史日期替换 '10' MINUTE）：

```
UPDATE hr.employees2
SET salary = (SELECT salary FROM hr.employees2
 AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '10' MINUTE)
 WHERE last_name = 'Fox')
WHERE last_name = 'Fox';
```

```
SQL> UPDATE hr.employees2
SET salary = (SELECT salary FROM hr.employees2
 AS OF TIMESTAMP (SYSTIMESTAMP - INTERVAL '10' MINUTE)
 WHERE last_name = 'Fox')
WHERE last_name = 'Fox';
2 3 4 5
1 row updated.

SQL> commit;

Commit complete.

SQL>
```

# 练习 19-1：使用闪回数据归档（续）

- 11) 继续在 SQL\*Plus 中执行以下查询。请注意，电子邮箱地址为每个人名字的首字母和姓连接后的字符串。

```
SELECT email, last_name, salary
FROM hr.employees2
WHERE email LIKE 'T%';
```

```
SQL> SELECT email, last_name, salary
FROM hr.employees2
WHERE email LIKE 'T%'; 2 3

EMAIL LAST_NAME SALARY

TJOLSON Olson 2100
TRAJS Rajs 3500
TFOX Fox 9600
TGATES Gates 2900

4 rows selected.

SQL>
```

- 12) 执行以下 UPDATE 和 COMMIT 命令将 @mycompany.com 附加到所有电子邮箱地址中：

```
UPDATE hr.employees2 SET email = email|| '@mycompany.com';
```

```
SQL> UPDATE hr.employees2 SET email = email|| '@mycompany.com';

107 rows updated.

SQL> COMMIT;

Commit complete.

SQL>
```

- 13) 重新查询该表，确认这些更改已执行。

```
SQL> SELECT email, last_name, salary
FROM hr.employees2
WHERE email LIKE 'T%'; 2 3

EMAIL LAST_NAME SALARY

TJOLSON@mycompany.com Olson 2100
TRAJS@mycompany.com Rajs 3500
TFOX@mycompany.com Fox 9600
TGATES@mycompany.com Gates 2900

4 rows selected.

SQL>
```

## 练习 19-1：使用闪回数据归档（续）

- 14) 假设将实施一种新的电子邮件标准。当前的电子邮件地址将暂存在 OLD\_EMAIL 列中，而原有的电子邮件列会进行更新，以包含新的地址。请使用以下语句添加所需的列：

```
ALTER TABLE hr.employees2 ADD (old_email varchar2(30));
```

```
SQL> ALTER TABLE hr.employees2 ADD (old_email varchar2(30));
```

```
Table altered.
```

```
SQL>
```

- 15) 使用当前电子邮件地址填充新添加的列：

```
SQL> UPDATE hr.employees2 SET old_email = email;
```

```
107 rows updated.
```

```
SQL> COMMIT;
```

```
Commit complete.
```

```
SQL>
```

- 16) 另外，还需要实施另一个新列来存储雇员总薪酬的计算值。请使用以下语句添加新列：

```
ALTER TABLE hr.employees2 ADD (total_comp number(8,2));
```

```
SQL> ALTER TABLE hr.employees2 ADD (total_comp number(8,2));
```

```
Table altered.
```

```
SQL>
```

- 17) 使用 TOTAL\_COMP 的计算值填充新列：

```
SQL> UPDATE hr.employees2 SET total_comp =
(1+NVL(commission_pct,0))*salary;
```

```
107 rows updated.
```

```
SQL> COMMIT;
```

```
Commit complete.
```

```
SQL>
```

# 练习 19-1：使用闪回数据归档（续）

18) 执行以下查询，查看自本练习开始以来 Fox 先生的所有记录版本。请注意，查询使用了 VERSIONS\_ENDTIME 和 VERSIONS\_ENDSCN 伪列。VERSIONS\_ENDTIME 显示相应记录版本的过期时间。由于最后一条记录是最新的，所以其 VERSIONS\_ENDTIME 为 NULL。VERSIONS\_ENDSCN 提供正确的事务处理顺序，在无法用时间区分的情况下，可以使用该值区分。可以使用 **the query\_fox\_versions.sql** 脚本。

```
SELECT last_name, email, old_email, salary, total_comp,
 versions_endtime
FROM hr.employees2
VERSIONS BETWEEN TIMESTAMP (SYSTIMESTAMP - INTERVAL '60'
MINUTE) AND SYSTIMESTAMP
WHERE last_name = 'Fox'
ORDER BY versions_endscn;
```

```
SQL> @query_fox_versions
SQL> SELECT last_name, email, old_email, salary, total_comp,
 versions_endtime
 2 FROM hr.employees2
 3 VERSIONS BETWEEN TIMESTAMP (SYSTIMESTAMP - INTERVAL '60'
MINUTE) AND SYSTIMESTAMP
 4 WHERE last_name = 'Fox'
 5 ORDER BY versions_endscn
 6
SQL> /
LAST_NAME EMAIL OLD_EMAIL
SALARY TOTAL_COMP VERSIONS_ENDTIME

Fox TFOX
9600 09-OCT-09 08.10.40.000000000 AM
Fox TFOX
10600 09-OCT-09 08.10.43.000000000 AM
Fox TFOX
11600 09-OCT-09 08.10.43.000000000 AM
Fox TFOX
12600 09-OCT-09 08.18.08.000000000 AM
Fox TFOX@mycompany.com
9600 09-OCT-09 08.19.34.000000000 AM
Fox TFOX@mycompany.com
TFOX@mycompany.com 9600

6 rows selected.
SQL>
```

# 练习 19-1：使用闪回数据归档（续）

19) 此时，不再需要 OLD\_EMAIL 列。可以使用以下语句删除它：

```
SQL> ALTER TABLE hr.employees2 DROP COLUMN old_email;

Table altered.

SQL>
```

20) 前一个查询的结果显示了 Fox 先生一段时间内的记录变化情况，而了解 EMPLOYEES2 表的变化情况可能也同样有用。请执行以下语句收集此类信息。记下 SYS\_FBA\_DDL\_COLMAP\_nnnnn 表的名称（您的表名中的数字标识符可能有所不同）。

```
SQL> SELECT * FROM tab;

TNAME TABTYPE CLUSTERID

COUNTRIES TABLE
DEPARTMENTS TABLE
EMPLOYEES TABLE
EMPLOYEES2 TABLE
EMP_DETAILS_VIEW VIEW
JOBS TABLE
JOB_HISTORY TABLE
LOCATIONS TABLE
REGIONS TABLE
SYS_FBA_DDL_COLMAP_77042 TABLE
SYS_FBA_HIST_77042 TABLE
SYS_FBA_TCRV_77042 TABLE

12 rows selected.

SQL>
```

21) 通过查询前一步骤中找到的表，确定系统更改号 (SCN) 范围（该范围说明了表中的每列是否存在）：

```
SQL> SELECT column_name, startscn, endscn
FROM SYS_FBA_DDL_COLMAP_77042; 2

COLUMN_NAME STARTSCN ENDSCN

EMPLOYEE_ID 2931863
FIRST_NAME 2931863
LAST_NAME 2931863
EMAIL 2931863
```

# 练习 19-1：使用闪回数据归档（续）

|                     |         |         |
|---------------------|---------|---------|
| PHONE_NUMBER        | 2931863 |         |
| HIRE_DATE           | 2931863 |         |
| JOB_ID              | 2931863 |         |
| SALARY              | 2931863 |         |
| COMMISSION_PCT      | 2931863 |         |
| MANAGER_ID          | 2931863 |         |
| DEPARTMENT_ID       | 2931863 |         |
| D_2937150_OLD_EMAIL | 2936436 | 2937150 |
| TOTAL_COMP          | 2936564 |         |
| 13 rows selected.   |         |         |
| SQL>                |         |         |

- 22) 从前面的结果可以看出，ENDSCN 值为 NULL，表明该列仍然存在于表中。可以使用 SCN\_TO\_TIMESTAMP 函数将 SCN 转换为时间戳。  
**注：**现在您已经拥有重建数据的完整历史记录所需的全部信息了。

|                                                  |
|--------------------------------------------------|
| SQL> SELECT SCN_TO_TIMESTAMP(2931863) FROM DUAL; |
| SCN_TO_TIMESTAMP(2931863)                        |
| -----                                            |
| 09-OCT-09 07.22.08.0000000000 AM                 |
| 1 row selected.                                  |
| SQL>                                             |

- 23) 在 SQL\*Plus 会话中，以 sys 用户身份连接到数据库，并执行 flada\_list1.sql 文件来列出可用的数据字典视图：

|                                    |
|------------------------------------|
| SQL> connect / as sysdba           |
| Connected.                         |
| SQL> @flada_list1                  |
| SQL> REM "*****"                   |
| SQL> REM "For demo purposes ONLY:" |
| SQL>                               |
| SQL> connect / as sysdba           |
| Connected.                         |
| SQL>                               |
| SQL> set echo on                   |
| SQL> set serveroutput on           |
| SQL> -- set verify on              |
| SQL> set term on                   |

## 练习 19-1：使用闪回数据归档（续）

```

SQL> set lines 200
SQL> set pages 44
SQL> set pause on pause "Press [Enter] to continue ..."
SQL>
SQL> /*== To list the available data dictioary views ==*/
SQL>
SQL> SELECT table_name
 2 FROM dict
 3 WHERE table_name LIKE '%FLASHBACK_ARCHIVE%'
 4 /
Press [Enter] to continue ...

TABLE_NAME

DBA_FLASHBACK_ARCHIVE
DBA_FLASHBACK_ARCHIVE_TABLES
DBA_FLASHBACK_ARCHIVE_TS
USER_FLASHBACK_ARCHIVE
USER_FLASHBACK_ARCHIVE_TABLES

5 rows selected.

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> col FLASHBACK_ARCHIVE_NAME format A25
SQL> col ARCHIVE_TABLE_NAME format A20
SQL> col TABLE_NAME format A12
SQL> col OWNER_NAME format A10
SQL>
SQL> DESC dba_flashback_archive
Name Null? Type

OWNER_NAME VARCHA2(30)
FLASHBACK_ARCHIVE_NAME NOT NULL VARCHA2(255)
FLASHBACK_ARCHIVE# NOT NULL NUMBER
RETENTION_IN_DAYS NOT NULL NUMBER
CREATE_TIME TIMESTAMP(9)
LAST_PURGE_TIME TIMESTAMP(9)
STATUS VARCHA2(7)

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

```



# 练习 19-1：使用闪回数据归档（续）

```

SQL>
SQL> /*== To query the time when the flashback data archive(s)
have been created ==*/
SQL>
SQL> SELECT flashback_archive_name, create_time, status
2 FROM dba_flashback_archive
3 /
Press [Enter] to continue ...

FLASHBACK_ARCHIVE_NAME CREATE_TIME
STATUS

FLA1 09-OCT-09 07.28.09.0000000000 AM

1 row selected.

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> DESC dba_flashback_archive_ts
Name Null? Type

FLASHBACK_ARCHIVE_NAME NOT NULL VARCHAR2(255)
FLASHBACK_ARCHIVE# NOT NULL NUMBER
TABLESPACE_NAME NOT NULL VARCHAR2(30)
QUOTA_IN_MB VARCHAR2(40)

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> /*== To list the tablespace(s), which are used for
flashback data archives ==*/
SQL>
SQL> SELECT *
2 FROM dba_flashback_archive_ts
3 /
Press [Enter] to continue ...

FLASHBACK_ARCHIVE_NAME FLASHBACK_ARCHIVE# TABLESPACE_NAME
QUOTA_IN_MB

FLA1 1 FLA_TBS1
10

1 row selected.

```

## 练习 19-1：使用闪回数据归档（续）

```
SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> DESC dba_flashback_archive_tables
Name Null? Type

TABLE_NAME NOT NULL VARCHAR2(30)
OWNER_NAME NOT NULL VARCHAR2(30)
FLASHBACK_ARCHIVE_NAME NOT NULL VARCHAR2(255)
ARCHIVE_TABLE_NAME VARCHAR2(53)

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL>
SQL> /*== Query the table name(s), the owner name(s), and ==*/
SQL> /*== the internal "history" table name(s) of the
flashback data archive ==*/
SQL>
SQL> SELECT *
 2 FROM dba_flashback_archive_tables
 3 /
Press [Enter] to continue ...

TABLE_NAME OWNER_NAME FLASHBACK_ARCHIVE_NAME
ARCHIVE_TABLE_NAME STATUS

EMPLOYEES2 HR FLA1
SYS_FBA_HIST_77042 ENABLED

1 row selected.

SQL>
SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL> clear columns
SQL>
SQL> connect hr/HR
Connected.
SQL>
SQL> SELECT table_name
 2 FROM dict
 3 WHERE table_name LIKE '%FLASHBACK_ARCHIVE%'
 4 /
Press [Enter] to continue ...
```

## 练习 19-1：使用闪回数据归档（续）

```

TABLE_NAME

USER_FLASHBACK_ARCHIVE
USER_FLASHBACK_ARCHIVE_TABLES

2 rows selected.

SQL> pause Press [Enter] to continue ...
Press [Enter] to continue ...

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 24) 要练习其它闪回数据归档维护任务，请使用 SQL\*Plus 以 ARCHIVE\_ADMIN 用户身份连接到数据库。

```

$ sqlplus ARCHIVE_ADMIN/ARCHIVE_ADMIN

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 08:53:09
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL>

```

- 25) 超过保留时间后，将自动清除闪回数据归档中的数据。不过，也可以显式清除数据。请使用以下命令清除两分钟以前的数据：

```

ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE TIMESTAMP
(SYSTIMESTAMP - INTERVAL '2' MINUTE);

```

```

SQL> ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE
TIMESTAMP(SYSTIMESTAMP - INTERVAL '2' MINUTE);

Flashback archive altered.

SQL>

```

## 练习 19-1：使用闪回数据归档（续）

26) 执行 flada\_tbs2.sql 脚本，以 SYS 用户身份创建另一个 10 MB 的表空间。

```
SQL> @flada_tbs2
Connected.
SQL> set serveroutput on
SQL> -- set verify on
SQL> set term on
SQL> set lines 200
SQL> set pages 44
SQL> set pause on
SQL>
SQL> /*== Create another tablespace ==*/
SQL>
SQL> DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS and DATAFILES
 2 /
DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS and DATAFILES
*
ERROR at line 1:
ORA-00959: tablespace 'FLA_TBS2' does not exist

SQL>
SQL> CREATE SMALLFILE TABLESPACE fla_tbs2
 2 DATAFILE '+DATA'
 3 SIZE 10M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE 32767M
 4 NOLOGGING EXTENT MANAGEMENT LOCAL SEGMENT SPACE
MANAGEMENT AUTO
 5 /

Tablespace created.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

27) 以 ARCHIVE\_ADMIN 用户身份，将 5 MB 的 FLA\_TBS2 表空间添加到 FLA1 闪回数据归档。

```
SQL> connect ARCHIVE_ADMIN/ARCHIVE_ADMIN
Connected.
SQL> ALTER FLASHBACK ARCHIVE fla1 ADD TABLESPACE fla_tbs2
QUOTA 5M;
Flashback archive altered.

SQL>
```

## 练习 19-1：使用闪回数据归档（续）

- 28) 一个特定闪回数据归档中的所有表都具有相同的保留时间。将 FLA1 闪回数据归档的保留时间更改为两年。

```
SQL> ALTER FLASHBACK ARCHIVE fla1 MODIFY RETENTION 2 YEAR;

Flashback archive altered.

SQL>
```

- 29) 以 ARCHIVE\_ADMIN 用户身份，删除 FLA1 闪回数据归档。

**注：**删除闪回数据归档也会删除内部防篡改历史记录表。由于审计和安全要求，您无法直接删除此表。由于表空间可能包含其它数据，因此删除闪回数据归档不会删除它所在的表空间。

```
SQL> DROP FLASHBACK ARCHIVE fla1;

Flashback archive dropped.

SQL>
```

- 30) 以 SYS 用户身份进行连接，通过执行 flada\_cleanup.sql 脚本清理环境。

```
SQL> @flada_cleanup
SQL> REM Execute this script script for training purposes only
SQL> REM Undo Cleanup for Flashback Data Archive
SQL> REM Execute script as SYSDBA
SQL>
SQL> set echo on
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pause on
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> DROP TABLESPACE fla_tbs1 INCLUDING CONTENTS and DATAFILES
2 /

Tablespace dropped.

SQL> DROP TABLESPACE fla_tbs2 INCLUDING CONTENTS and DATAFILES
2 /

Tablespace dropped.

SQL> ALTER USER hr PASSWORD EXPIRE ACCOUNT LOCK
2 /
```

## 练习 19-1：使用闪回数据归档（续）

```
User altered.

SQL> DROP USER archive_admin CASCADE
2 /

User dropped.

SQL> drop table hr.employees2 purge;

Table dropped.

SQL>
SQL> prompt Flashback Data Archive cleanup complete.
Flashback Data Archive cleanup complete.
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 19-2：执行闪回事务处理回退

闪回事务处理回退是一个逻辑恢复选项，用于回退特定事务处理及相关事务处理，同时保持数据库联机。相关事务处理通过写后写 (WAW) 关系或主键约束关系相关联，在前者中事务处理对目标事务处理更改的同一数据进行修改，而在后者中事务处理重新插入由目标事务处理删除的同一主键值。闪回事务处理利用为还原块生成的还原和重做来创建并执行校正事务处理，以将受影响的数据还原到其原始状态。

- 1) 闪回事务处理回退要求数据库处于 ARCHIVELOG 模式。在以 oracle 用户身份连接的终端会话中，转到 /home/oracle/solutions/FTX 目录。确保指向 orcl 数据库。执行 flatxn\_archivelog.sh 脚本。

```
$ cd /home/oracle/solutions/FTX
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$./flatxn_archivelog.sh

For demo purposes ONLY:
 * Unlock HR account
 * Enable ARCHIVELOG mode for database

The script may appear to hang at the SQL prompt
when the database is shutting down and being
opened. Wait a few minutes and it should progress.

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 09:10:34
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL>
User altered.

SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.
```

**练习 19-2：执行闪回事务处理回退（续）**

```

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Database log mode Archive Mode
Automatic archival Enabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 108
Next log sequence to archive 110
Current log sequence 110
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

- 2) 使用 SQL\*Plus, 以 SYS 用户身份连接到数据库并运行 flatxn\_setup1.sql 脚本。此设置脚本设置了数据库和 HR 用户, 以便使用闪回回退事务处理。请注意 ALTER DATABASE 和 GRANT 命令。

```

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 09:12:54
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @flatxn_setup1
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pause on
SQL>
SQL> /*== Set up the HR database account for this lab ==*/
SQL>
SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA;

```



## 练习 19-2：执行闪回事务处理回退（续）

```
Database altered.

SQL> ALTER DATABASE ADD SUPPLEMENTAL LOG DATA (PRIMARY KEY)
COLUMNS;

Database altered.

SQL> GRANT EXECUTE ON dbms_flashback TO hr;

Grant succeeded.

SQL> GRANT select any transaction TO hr;

Grant succeeded.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 3) 按 [Enter] 继续执行脚本的下一步骤。脚本以 HR 用户身份连接到数据库并创建第一个测试事务处理，该事务处理要插入五行。

```
SQL>
SQL> /*== Create test data for flashback transaction ==*/
SQL> connect hr/hr
Connected.
SQL>
SQL> /*== Test transaction 1 ==*/
SQL> INSERT INTO hr.regions VALUES (10,'Pole');

1 row created.

SQL> INSERT INTO hr.regions VALUES (20,'Moon');

1 row created.

SQL> INSERT INTO hr.regions VALUES (30,'Venus');

1 row created.

SQL> INSERT INTO hr.regions VALUES (40,'Mars');

1 row created.

SQL> INSERT INTO hr.regions VALUES (50,'Saturn');

1 row created.
```

## 练习 19-2：执行闪回事务处理回退（续）

```
SQL> COMMIT;

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 4) 按 [Enter] 继续执行脚本的下一步骤。脚本继续创建第二个测试事务处理，该事务处理要更新您刚插入的两行。此时将创建写后写 (WAW) 相关性。当事务处理更新或删除由从属事务处理插入或更新的行时，就存在 WAW 相关性。

```
SQL>
SQL> /*== Test transaction 2 ==*/
SQL> /*== Region 10 and 20 has a WAW dependency on transaction 1 ==*/
SQL> UPDATE hr.regions SET region_name='Two Poles' WHERE
region_id = 10;

1 row updated.

SQL> UPDATE hr.regions SET region_name='Many Moons' WHERE
region_id = 20;

1 row updated.

SQL> COMMIT;

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 5) 按 [Enter] 继续执行脚本的下一步骤。脚本继续创建第三个事务处理，该事务处理要更新三行，其中的一行也在前面的事务处理中更新过。此操作将创建另一个 WAW 相关性。

```
SQL>
SQL> /*== Test transaction 3 ==*/
SQL> /*== Region 10 has a WAW dependency on transaction 1 and 2 ==*/
SQL> /*== Region 40 and 50 has a WAW dependency on transaction 1 ==*/
SQL> UPDATE hr.regions SET region_name='No star' WHERE
region_id = 10;

1 row updated.

SQL> UPDATE hr.regions SET region_name='Red star' WHERE
region_id = 40;
```

## 练习 19-2：执行闪回事务处理回退（续）

```

1 row updated.

SQL> UPDATE hr.regions SET region_name='Big star' WHERE
region_id = 50;

1 row updated.

SQL> COMMIT;

Commit complete.

SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```

- 6) 按 [Enter] 继续执行脚本的下一步骤。脚本继续创建第四个事务处理，该事务处理要更新在事务处理 1 中插入的行。此操作将创建另一个 WAW 相关性。

```

SQL>
SQL> /*== Test transaction 4 ==*/
SQL> /*== Region 30 has a WAW dependency on transaction 1 ==*/
SQL> UPDATE hr.regions SET region_name='Still called Venus'
WHERE region_id = 30;

1 row updated.

SQL> COMMIT;

Commit complete.

SQL>
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...

```

- 7) 按 [Enter] 继续执行脚本的下一步骤。最后，脚本以 SYS 用户身份连接到数据库并切换归档日志。闪回事务处理至少需要一份归档日志才能启动重做分析。

```

SQL>
SQL> connect / as sysdba
Connected.
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;

System altered.

SQL>
SQL> ALTER SYSTEM ARCHIVE LOG CURRENT;

System altered.

```

## 练习 19-2：执行闪回事务处理回退（续）

```
SQL>
SQL> prompt "Setup for Flashback Transaction completed"
"Setup for Flashback Transaction completed"
SQL> pause Press [Enter] to continue...
Press [Enter] to continue...
```

- 8) 按 [Enter] 退出脚本。

```
SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

- 9) 事务处理可具有写后写 (WAW) 相关性，这意味着事务处理会更新或删除由以前的事务处理插入或更新的行数据。在此部分中，您希望回退错误事务处理及其 WAW 相关事务处理，这些相关事务处理也造成了不正确的更改。执行下列步骤：
- 以 SYS 用户身份登录 Enterprise Manager Database Control。
  - 您怀疑 HR. REGIONS 表包含错误事务处理所导致的更改，希望查看完整的表数据。单击“Schema（方案）”选项卡，然后单击“Database Objects（数据库对象）”下的“Tables（表）”。
  - 要查看属于 HR 方案用户的所有表，请在“Schema（方案）”字段中输入 HR 并单击“Go（搜索）”按钮。
  - 在“Tables（表）”页上，选择 REGIONS 表，然后选择“Actions（操作）”下拉列表中的“View Data（查看数据）”。单击“Go（执行）”按钮查看 HR. REGIONS 表中的数据。
  - 查看表内容时，您发现其不应包含以下行：“Many moons”和“Still called Venus”。单击“OK（确定）”。
  - 返回表的列表，选择 REGIONS 表，然后从“Actions（操作）”下拉列表中选择“Flashback Transaction（闪回事务处理）”，再单击“Go（执行）”按钮。
  - 复查查询条件。指定您认为发生错误更改的时间段。对于本示例，接受默认的时间段并单击“Next（下一步）”。
  - 在“Flashback Transaction: Select Transaction（闪回事务处理：选择事务处理）”页上，单击已更新了两个区域的事务处理的“Transaction ID（事务处理 ID）”链接。
  - 在“Select Transaction: Show Details（选择事务处理：显示详细资料）”页上，查看错误的 UPDATE 命令，并单击“OK（确定）”。

## 练习 19-2：执行闪回事务处理回退（续）

- j) 单击“Next（下一步）”标识所有相关事务处理并回退所有所需更改。此时将显示“Processing: Flashback Transaction（正在处理：闪回事务处理）”窗口。在“Flashback Transaction: Show Dependencies（闪回事务处理：显示相关性）”页上，单击第一个“Transaction ID（事务处理 ID）”链接。
- k) 在“Show Dependencies: Details（显示相关性：详细资料）”页上，查看校正事务处理。请注意，其中一个校正事务处理已被执行，它被标记为 [exec=yes]。另一个未执行，被标记为 [exec=no]。单击“OK（确定）”。
- l) 在“Flashback Transaction: Show Dependencies（闪回事务处理：显示相关性）”页上，单击“Change Recovery Option（更改恢复选项）”按钮。
- m) 在“Show Dependencies: Change Recovery Option（显示相关性：更改恢复选项）”页上，选择“Cascade（级联）”，这也会回退错误的相关事务处理更改，然后单击“OK（确定）”。
- n) 在“Flashback Transaction: Show Dependencies（闪回事务处理：显示相关性）”页上，单击“Next（下一步）”。
- o) 在“Flashback Transaction: Review（闪回事务处理：复查）”页上，单击“Show Undo SQL Script（显示还原 SQL 脚本）”按钮。
- p) 在“Review: Show Undo SQL（复查：显示还原 SQL）”页上，复查所有校正 UPDATE 命令，然后单击“OK（确定）”。
- q) 在“Execute SQL（执行 SQL）”区域中输入以下命令，然后单击“Execute SQL（执行 SQL）”按钮，预览校正还原语句生成的结果：

```
SELECT * FROM HR.REGIONS;
```

- r) 请注意，“Many Moons”行已从结果列表中消失。单击“OK（确定）”。
- s) 在“Flashback Transaction: Review（闪回事务处理：复查）”页上，单击“Finish（完成）”。
- t) 闪回事务处理执行成功。单击“OK（确定）”。
- u) 返回到 HR.REGIONS 表并查看其内容。返回到“Tables（表）”页。指定 HR 作为“Schema（方案）”，单击“Go（执行）”，选择 REGIONS 表，从“Actions（操作）”下拉列表中选择“View Data（查看数据）”，然后单击“Go（执行）”。
- v) 表的内容与上述四个步骤显示的内容相同。由于事务处理 2 和 4 之间不存在 WAW 相关性，所以回退事务处理 2 和 3，而仍保留事务处理 4（区域名设置为“Still called Venus”）。
- w) 从 Enterprise Manager 中注销。

## 练习 19-2：执行闪回事务处理回退（续）

- 10) 通过执行以下步骤来清理环境：确保指向 orcl 数据库。在以 SYS 用户身份连接的 SQL\*Plus 会话中，执行 flatxn\_cleanup1.sql 脚本。

```
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 09:31:33
2009
Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> @flatxn_cleanup1
SQL> set serveroutput on
SQL> set term on
SQL> set lines 200
SQL> set pause on
SQL>
SQL> connect / as sysdba
Connected.
SQL>
SQL> DELETE FROM hr.regions WHERE region_id=10;

1 row deleted.

SQL> DELETE FROM hr.regions WHERE region_id=20;

1 row deleted.

SQL> DELETE FROM hr.regions WHERE region_id=30;

1 row deleted.

SQL> DELETE FROM hr.regions WHERE region_id=40;

1 row deleted.

SQL> DELETE FROM hr.regions WHERE region_id=50;

1 row deleted.

SQL> COMMIT;

Commit complete.
```

**练习 19-2：执行闪回事务处理回退（续）**

```

SQL>
SQL> ALTER DATABASE DROP SUPPLEMENTAL LOG DATA (PRIMARY KEY)
COLUMNS;

Database altered.

SQL> ALTER DATABASE DROP SUPPLEMENTAL LOG DATA;

Database altered.

SQL> REVOKE EXECUTE ON dbms_flashback FROM hr;

Revoke succeeded.

SQL> REVOKE select any transaction FROM hr;

Revoke succeeded.

SQL>
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$

```

11) 现在执行 flatxn\_cleanup.sh 脚本以禁用 ARCHIVELOG 模式。

```

$./flatxn_cleanup.sh

For cleanup of demo:
 * Change HR password back to HR
 * Disable ARCHIVELOG mode

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 09:32:52
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL>
System altered.

```

## 练习 19-2：执行闪回事务处理回退（续）

```
SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Database log mode No Archive Mode
Automatic archival Disabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 110
Current log sequence 112
SQL> SQL> "Cleanup finished."
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```



## 第 20 课的练习

数据恢复指导是一种可帮助您诊断并修复数据故障和损坏的工具。数据恢复指导根据故障现象分析故障，并智能地确定最佳修复策略。该工具还可以自动修复已诊断的故障。

在本练习中，您将在三个不同的场景中使用数据恢复指导：

- 使用 Enterprise Manager 修复“停机”的数据库
- 使用 Enterprise Manager 修复块损坏
- 使用 RMAN 命令修复“停机”的数据库

## 练习 20-1：使用 Enterprise Manager 修复数据库

第一个场景假定数据库处于“停机”状态。在生产环境中，这将是一种非常严重的情况，必须尽快加以解决以“挽救业务”。您尝试重新打开数据库，但是失败了。然后，您使用 Enterprise Manager Database Control 提供的数据库恢复指导诊断故障。诊断结果显示有两个数据文件无法访问。对于第一个数据文件问题，使用手动修复选项（重命名数据文件）。此问题可能是由于 DBA 执行 ALTER DATABASE ... RENAME FILE ... 命令和系统管理员忽略了正确重命名操作系统中的数据文件而导致的。

通过重命名文件解决第一个数据文件问题后，再次向数据库恢复指导请求修复建议。对于第二个数据文件问题，使用自动修复选项恢复丢失的数据文件。最后，再次检查数据库是否已启动并且正在运行。

- 1) 在终端窗口中，转到 /home/oracle/solutions/DRA 目录，执行 ./dra\_setup.sh 命令。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ cd /home/oracle/solutions/DRA
$./dra_setup.sh

Recovery Manager: Release 11.2.0.1.0 - Production on Fri Oct 9
10:34:57 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1226549444)
using target database control file instead of recovery catalog

RMAN>
RMAN>
no failures found that match specification

RMAN>
no failures found that match specification

RMAN>
no failures found that match specification

RMAN>
Recovery Manager complete.

For demo purposes ONLY
```

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

```
Run as the OS user who installed the Oracle software, e.g.
oracle

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 10:35:02
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL> SQL> SQL> drop tablespace users2 including contents
and datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'USERS2' does not exist

SQL> SQL> drop tablespace example2 including contents and
datafiles
*
ERROR at line 1:
ORA-00959: tablespace 'EXAMPLE2' does not exist

SQL> SQL> SQL>
Tablespace created.

SQL> SQL>
Tablespace created.

SQL> SQL>
Table created.

SQL>
Table created.

SQL>
Table created.

SQL>
Table created.

SQL> SQL>
Table created.

SQL>
Table created.
```

**练习 20-1：使用 Enterprise Manager 修复数据库（续）**

```

SQL>
Table created.

SQL> SQL> Disconnected from Oracle Database 11g Enterprise
Edition Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

For demo purposes ONLY:
 * Unlock HR account with the HR password
 * Enable ARCHIVELOG mode for database

The script may appear to hang at the SQL prompt
when the database is shutting down and being
opened.Wait a few minutes and it should progress.

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 10:35:05
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL>
User altered.

SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
Database mounted.
SQL>
Database altered.

```

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

```
SQL>
Database altered.

SQL> Database log mode Archive Mode
Automatic archival Enabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 110
Next log sequence to archive 112
Current log sequence 112
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

Recovery Manager: Release 11.2.0.1.0 - Production on Fri Oct 9
10:35:28 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1226549444)

RMAN> configure controlfile autobackup on;
2>
3> sql 'alter system archive log current';
4>
5> sql 'alter system archive log current';
6> sql 'alter system archive log current';
7> backup database;
8> sql 'alter system archive log current';
9> sql 'alter system archive log current';
10> sql 'alter system archive log current';
11> backup archivelog all delete input;
12> sql 'alter system archive log current';
13> sql 'alter system archive log current';
14> sql 'alter system archive log current';
15> list backup summary;
16>
17>
using target database control file instead of recovery catalog
old RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters:
CONFIGURE CONTROLFILE AUTOBACKUP ON;
new RMAN configuration parameters are successfully stored

sql statement: alter system archive log current

sql statement: alter system archive log current
```

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

```
sql statement: alter system archive log current

Starting backup at 09-OCT-09
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=51 device type=DISK
channel ORA_DISK_1: starting full datafile backup set
channel ORA_DISK_1: specifying datafile(s) in backup set
input datafile file number=00002
name=+DATA/orcl/datafile/sysaux.257.698768527
channel ORA_DISK_1: starting piece 1 at 09-OCT-09
channel ORA_DISK_1: finished piece 1 at 09-OCT-09
piece
handle=+FRA/orcl/backupset/2009_10_09/nnndf0_tag20091009t10354
7_0.294.699791749 tag=TAG20091009T103547 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:01:06
...
input datafile file number=00004
name=+DATA/orcl/datafile/users.259.698768531
channel ORA_DISK_1: starting piece 1 at 09-OCT-09
channel ORA_DISK_1: finished piece 1 at 09-OCT-09
piece
handle=+FRA/orcl/backupset/2009_10_09/nnndf0_tag20091009t10354
7_0.297.699791905 tag=TAG20091009T103547 comment=NONE
channel ORA_DISK_1: backup set complete, elapsed time:
00:00:45
Finished backup at 09-OCT-09

Starting Control File and SPFILE Autobackup at 09-OCT-09
piece
handle=+FRA/orcl/autobackup/2009_10_09/s_699791951.298.6997919
55 comment=NONE
Finished Control File and SPFILE Autobackup at 09-OCT-09

sql statement: alter system archive log current

sql statement: alter system archive log current

sql statement: alter system archive log current

Starting backup at 09-OCT-09
current log archived
using channel ORA_DISK_1
channel ORA_DISK_1: starting archived log backup set
channel ORA_DISK_1: specifying archived log(s) in backup set
input archived log thread=1 sequence=101 RECID=1
STAMP=699702518
input archived log thread=1 sequence=102 RECID=2
STAMP=699703795
...
```

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

```
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_118.302.6997
91969 RECID=12 STAMP=699791969
Finished backup at 09-OCT-09

Starting Control File and SPFILE Autobackup at 09-OCT-09
piece
handle=+FRA/orcl/autobackup/2009_10_09/s_699791997.302.6997919
99 comment=NONE
Finished Control File and SPFILE Autobackup at 09-OCT-09

sql statement: alter system archive log current

sql statement: alter system archive log current

sql statement: alter system archive log current

List of Backups
=====
Key TY LV S Device Type Completion Time #Pieces #Copies
Compressed Tag

1 B F A DISK 08-OCT-09 1 1 NO
TAG20091008T113504
2 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103547
3 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103547
4 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103547
5 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103547
6 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103911
7 B A A DISK 09-OCT-09 1 1 NO
TAG20091009T103929
8 B F A DISK 09-OCT-09 1 1 NO
TAG20091009T103957

Recovery Manager complete.
Basic setup done.
Now, run one of the setup scripts for a specific scenario.
$
```

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

- 2) 看到以下行时，请继续执行操作：“Basic setup done. Now, run one of the setup scripts for a specific scenario”。为了设置数据故障并关闭数据库，请在终端窗口中执行以下命令：

```
./dra1_corruption.sh。
```

users2.dbf 数据文件将被删除，example2.dbf 文件将被重命名。您将使用手动修复和自动修复选项诊断并修复这些故障。

```
$./dra1_corruption.sh

For demo purposes ONLY:
* Shutdown database

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 10:49:07
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

For demo purposes ONLY:
* Corrupt datafiles to produce failures
$
```

- 3) 使用 Enterprise Manager 尽快修复这种情况。
- 以 SYS 用户身份和 oracle\_4U 口令登录 Enterprise Manager，然后以 SYSDBA 身份进行连接。
  - 如果无法访问 Enterprise Manager Database Control，请使用以下命令停止该组件，然后再启动：

```
$ emctl stop dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Stopping Oracle Enterprise Manager 11g Database Control ...
```



## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

```
all attempts to stop oc4j failed... now trying to kill 9
--- Failed to shutdown DBConsole Gracefully ---
... Stopped.
$ emctl start dbconsole
Oracle Enterprise Manager 11g Database Control Release
11.2.0.1.0
Copyright (c) 1996, 2009, Oracle Corporation. All rights
reserved.
https://ed.us.oracle.com:5500/em/console/aboutApplication
Starting Oracle Enterprise Manager 11g Database Control
..... started.

Logs are generated in directory
/u01/app/oracle/acfsmounts/acfs_db1/ed.us.oracle.com_orcl/sysm
an/log
$
```

- c) 遇到数据库“停机”。单击“Startup（启动）”。
- d) 在“Startup/Shutdown（启动/关闭）”页上指定所有需要的身份证明。将 oracle/oracle 作为主机身份证明，将 SYS/oracle\_4U as SYSDBA 作为数据库身份证明。单击“OK（确定）”。
- e) 在“Select Startup Type（选择启动类型）”页上，确保已选中“Start database along with dependent resources（启动数据库同时启动从属资源）”（该选项应为默认选项），然后单击“OK（确定）”。
- f) 在“Startup/Shutdown Confirmation（启动/关闭确认）”页上，单击“Yes（是）”。
- g) “Startup（启动）”页很快就会显示出来。请等待。
- h) 此时将显示“Startup/Shutdown: Select Operation（启动/关闭：选择操作）”页，该页有一条错误消息，表明有一个文件无法找到。
- i) 确保选中了 orcl，然后单击“Startup（启动）”。
- j) 在“Confirmation（确认）”页上，单击“Advanced Options（高级选项）”按钮。
- k) 在“Advanced Options（高级选项）”页上，选择“Mount the Database（装载数据库）”，然后单击“OK（确定）”。
- l) 在“Confirmation（确认）”页上，单击“Yes（是）”。
- m) 在“Startup（启动）”页上稍等片刻。
- n) 随后会在“Startup/Shutdown: Select Operation（启动/关闭：选择操作）”页上看到一条“Startup Successful（启动成功）”消息。单击“Return（返回）”。

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

- o) 稍后您将转到 “Database Instance（数据库实例）” 页。在 “Database Instance（数据库实例）” 页上，单击 “Perform Recovery（执行恢复）” 按钮诊断故障并执行修复。
- p) 在 “Perform Recovery: Credentials（执行恢复：身份证明）” 页上输入 oracle 作为用户名和口令，然后单击 “Continue（继续）”。（如果先前选择了 “Save Credentials（保存身份证明）”，则不显示此页。）
- q) 在 “Database Login（数据库登录）” 页上，输入 SYS/oracle\_4U as SYSDBA 作为身份证明，然后单击 “Login（登录）”。
- r) 如果显示 “Perform Recovery: Credentials（执行恢复：身份证明）” 页，请输入 oracle/oracle 作为主机身份证明。单击 “Continue（继续）”。
- s) 在 “Perform Recovery（执行恢复）” 页上，单击 “Oracle Advised Recovery（Oracle 建议的恢复）” 部分中的 “Advise and Recover（建议和恢复）” 按钮。
- t) 此时将转到 “View and Manage Failure（查看和管理故障）” 页。打开数据库失败时，数据恢复指导可自动诊断问题。查看此诊断的结果。单击 “Failure Description（故障说明）” 下的 “+” 图标查看详细信息。可以看到有两个文件缺失。
- u) 要查看建议的修复策略，请选中所有故障并单击 “Advise（建议）” 按钮。
- v) 此时将显示 “Manual Actions（手动操作）” 页，其中数据恢复指导会提示您考虑手动修复。**此时不要单击任何按钮。**执行下一步。
- w) 本练习假定错误地重命名了 example01.dbf 数据文件。在终端窗口中，输入以下命令以手动修复故障：

```
$
$ mv /u01/app/oracle/oradata/example2.dbf.old
/u01/app/oracle/oradata/example2.dbf
$
```

- x) 返回 Enterprise Manager 中的 “Manual Actions（手动操作）” 页。由于已完成前面的步骤，单击 “Re-assess Failures（重新评估故障）”。
- y) 数据恢复指导重新验证所有故障，并关闭您手动修复的故障。在 “View and Manage Failures（查看和管理故障）” 页上，如有必要，单击 “Failure Description（故障说明）” 下的 “+” 图标展开节点以查看其它任何未处理的故障。
- z) 请注意，您手动修复的故障已关闭，但是仍存在另一个打开的故障。选中它，然后单击 “Advise（建议）”。
- aa) 在 “Manual Actions（手动操作）” 页上，单击 “Continue with Advise（继续使用建议）” 启动自动修复。

## 练习 20-1：使用 Enterprise Manager 修复数据库（续）

- bb) 在 “Recovery Advise（恢复建议）” 页上，数据恢复指导将生成并显示 RMAN 修复脚本。单击 “Continue（继续）”。
- cc) 在 “Review（复查）” 页上，可以复查故障和用于解决该故障的修复脚本。单击 “Execute Recovery（执行恢复）”。
- dd) 等待此修复执行过程完成。数据恢复指导完成后，会显示 “Recovery Result（恢复结果）” 页并会在该页顶部显示 “Recovery Succeeded（已成功恢复）” 消息。查看所有修复结果。使用垂直滚动条向下滚动页面。
- ee) 单击 “Open Database（打开数据库）”。
- ff) 成功修复后，数据库就会打开。在 “Recovery Results（恢复结果）” 页上，单击 “OK（确定）”。
- gg) 此时 Enterprise Manager 显示 “Login（登录）” 页。使用 SYS/oracle\_4U 以 SYSDBA 身份证明登录。
- hh) 在 “Database Instance（数据库实例）” 主页上，确认数据库实例已启动并且正在运行。可能要经过几分钟（或者几秒钟，取决于可用的系统资源），并要刷新几次，您才能看到它处于打开状态。

## 练习 20-2：使用 Enterprise Manager 修复块损坏

要诊断并修复块损坏，请执行以下步骤：

- 1) 要在 example01.dbf 数据文件中设置块损坏，请使用终端窗口转到 /home/oracle/solutions/DRA 目录，并执行 dra2\_corruption.sh 脚本。  
应用程序用户通常以这种形式造成块损坏错误。DBA 可在 Enterprise Manager Database Control 主页上看到该错误。（请参阅下一步骤）。不过，由于在发现错误到 Enterprise Manager 显示错误之间存在一些内部处理，因此会稍微有些延迟。

```
$ cd /home/oracle/solutions/DRA
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$./dra2_corruption.sh

For demo purposes ONLY:
Create block corruption for scenario 2
Please wait setup in process

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:12:53
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> Connected.
SQL>
dd of=/u01/app/oracle/oradata/example2.dbf bs=8192
conv=notrunc seek=131 << EOF
CORRUPT corrupt CORRUPT corrupt CORRUPT corrupt CORRUPT
corrupt CORRUPT corrupt
EOF

dd of=/u01/app/oracle/oradata/example2.dbf bs=8192
conv=notrunc seek=147 << EOF
CORRUPT corrupt CORRUPT corrupt CORRUPT corrupt CORRUPT
corrupt CORRUPT corrupt
EOF

dd of=/u01/app/oracle/oradata/example2.dbf bs=8192
conv=notrunc seek=139 << EOF
CORRUPT corrupt CORRUPT corrupt CORRUPT corrupt CORRUPT
corrupt CORRUPT corrupt
EOF
```

## 练习 20-2: 使用 Enterprise Manager 修复块损坏 (续)

```
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
0+1 records in
0+1 records out
246 bytes (246 B) copied, 0.0158417 seconds, 15.5 kB/s
0+1 records in
0+1 records out
246 bytes (246 B) copied, 0.0026572 seconds, 92.6 kB/s
0+1 records in
0+1 records out
246 bytes (246 B) copied, 0.0326892 seconds, 7.5 kB/s

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:12:54
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> Connected.
SQL>
System altered.

select * from hr.regions2
 *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 8, block # 131)
ORA-01110: data file 8: '/u01/app/oracle/oradata/example2.dbf'

select * from hr.locations2
 *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 8, block # 139)
ORA-01110: data file 8: '/u01/app/oracle/oradata/example2.dbf'

select * from hr.employees2
 *
ERROR at line 1:
ORA-01578: ORACLE data block corrupted (file # 8, block # 147)
ORA-01110: data file 8: '/u01/app/oracle/oradata/example2.dbf'

Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
Setup 2 completed.
$
```

## 练习 20-2：使用 Enterprise Manager 修复块损坏（续）

- 2) 使用 Enterprise Manager 快速解决这些错误。
  - a) 使用 SYS/oracle\_4U 以 SYSDBA 身份证明登录 Enterprise Manager。
  - b) 在“Database Instance（数据库实例）”主页上，单击“Advisor Central（指导中心）”。
  - c) 单击“Data Recovery Advisor（数据恢复指导）”。
  - d) 单击“Failure Description（故障说明）”下的“+”图标展开节点以查看详细信息。
  - e) 在“View and Manage Failures（查看和管理故障）”页上，查看块损坏的详细信息。在“Host（主机）”和“Password（口令）”中输入 oracle 作为身份证明（如果先前未保存这些信息）。选中所有故障，并单击“Advise（建议）”。
  - f) 在“Recovery Advice（恢复建议）”页上，查看修复脚本并单击“Continue（继续）”。
  - g) 在“Review（复查）”页上，可复查故障和建议的修复。单击“Submit Recovery Job（提交恢复作业）”。
  - h) 在“Job Activity（作业活动）”页上，您应收到一条表示作业已成功创建的确认消息。
  - i) 在“Job Activity（作业活动）”页上，是否能看到处于“正在运行”状态的作业取决于此作业的完成速度。定期单击浏览器的“Reload（重新加载）”或“Refresh（刷新）”按钮。不再显示“正在运行”状态的作业时，即表明该作业已完成。单击确认消息下的作业名称链接。
  - j) 在“Job Run（作业运行）”页上，确认作业已成功执行，并单击该页右上角的“Database（数据库）”选项卡。
  - k) 在“Database Instance（数据库实例）”主页上查看块损坏预警。DBA 定期监视此页，因此能看到块损坏预警（显示稍有一点延迟）。

## 练习 20-2: 使用 Enterprise Manager 修复块损坏 (续)

- 3) 为了确认块损坏已修复, 请在终端窗口中执行以下命令:

```
SELECT * FROM HR.REGIONS2;
```

```
$
$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:35:11
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select * from hr.regions2;

 REGION_ID REGION_NAME

 1 Europe
 2 Americas
 3 Asia
 4 Middle East and Africa

SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

## 练习 20-3：使用 RMAN 修复数据库

要诊断和修复停机数据库，请执行以下步骤：

- 1) 为了设置数据故障并关闭数据库，请使用终端窗口转到  
/home/oracle/solutions/DRA 目录，执行 dra1\_corruption.sh 脚本。  
users01.dbf 数据文件将被删除，example01.dbf 文件将被重命名。您将  
使用手动修复和自动修复选项诊断并修复这些故障。

```
$. oraenv
ORACLE_SID = [orcl] ? orcl
The Oracle base for
ORACLE_HOME=/u01/app/oracle/acfsmounts/acfs_db1 is
/u01/app/oracle
$ cd /home/oracle/solutions/DRA
$
$./dra1_corruption.sh

For demo purposes ONLY:
* Shutdown database

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:39:01
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

For demo purposes ONLY:
* Corrupt datafiles to produce failures
$
```



## 练习 20-3: 使用 RMAN 修复数据库 (续)

- 2) 如果发现数据库处于停机状态, 请尝试启动它:
  - a) 请注意, 此时数据库已装载, 其中包含数据文件错误。

```
$ sqlplus /nolog

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:40:08
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

SQL> connect / as sysdba
Connected to an idle instance.
SQL> startup
ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
Database mounted.
ORA-01157: cannot identify/lock data file 7 - see DBWR trace
file
ORA-01110: data file 7: '/u01/app/oracle/oradata/users2.dbf'

SQL>
```

- 3) 退出 SQL\*Plus 并启动 RMAN。

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$ rman target / nocatalog

Recovery Manager: Release 11.2.0.1.0 - Production on Fri Oct 9
11:42:14 2009

Copyright (c) 1982, 2009, Oracle and/or its affiliates. All
rights reserved.

connected to target database: ORCL (DBID=1226549444, not open)
using target database control file instead of recovery catalog

RMAN>
```

### 练习 20-3：使用 RMAN 修复数据库（续）

4) 列出故障和故障详细信息。

```

RMAN> LIST FAILURE;

List of Database Failures
=====

Failure ID Priority Status Time Detected Summary

422 HIGH OPEN 09-OCT-09 One or more non-
system datafiles are missing

RMAN> LIST FAILURE DETAIL;

List of Database Failures
=====

Failure ID Priority Status Time Detected Summary

422 HIGH OPEN 09-OCT-09 One or more non-
system datafiles are missing
 Impact: See impact for individual child failures
 List of child failures for parent failure ID 422
 Failure ID Priority Status Time Detected Summary

 526 HIGH OPEN 09-OCT-09 Datafile 8:
'/u01/app/oracle/oradata/example2.dbf' is missing
 Impact: Some objects in tablespace EXAMPLE2 might be
unavailable
 520 HIGH OPEN 09-OCT-09 Datafile 7:
'/u01/app/oracle/oradata/users2.dbf' is missing
 Impact: Some objects in tablespace USERS2 might be
unavailable
RMAN>

```

5) 诊断这些故障。

a) 您应该看到两个故障。执行下一步骤后，就会返回此窗口。

```

RMAN> ADVISE FAILURE;

List of Database Failures
=====

Failure ID Priority Status Time Detected Summary

422 HIGH OPEN 09-OCT-09 One or more non-
system datafiles are missing
 Impact: See impact for individual child failures
 List of child failures for parent failure ID 422
 Failure ID Priority Status Time Detected Summary

```

**练习 20-3：使用 RMAN 修复数据库（续）**

```

526 HIGH OPEN 09-OCT-09 Datafile 8:
'/u01/app/oracle/oradata/example2.dbf' is missing
Impact: Some objects in tablespace EXAMPLE2 might be
unavailable
520 HIGH OPEN 09-OCT-09 Datafile 7:
'/u01/app/oracle/oradata/users2.dbf' is missing
Impact: Some objects in tablespace USERS2 might be
unavailable

analyzing automatic repair options; this may take some time
allocated channel: ORA_DISK_1
channel ORA_DISK_1: SID=24 device type=DISK
analyzing automatic repair options complete

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
1. If file /u01/app/oracle/oradata/users2.dbf was
unintentionally renamed or moved, restore it
2. If file /u01/app/oracle/oradata/example2.dbf was
unintentionally renamed or moved, restore it

Automated Repair Options
=====
Option Repair Description

1 Restore and recover datafile 7; Restore and recover
datafile 8
Strategy: The repair includes complete media recovery with
no data loss
Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_4235069421.hm

RMAN>

```

- 6) 本练习假定错误地重命名了 example2.dbf 数据文件。打开另一个终端窗口并正确地重新命名该文件，从而手动修复故障。

```

$ mv /u01/app/oracle/oradata/example2.dbf.old
/u01/app/oracle/oradata/example2.dbf
$

```

## 练习 20-3：使用 RMAN 修复数据库（续）

7) 返回 RMAN 窗口，检查手动修复的结果。

a) 您应该看到一个故障。

```

RMAN> ADVISE FAILURE;

List of Database Failures
=====

Failure ID Priority Status Time Detected Summary

422 HIGH OPEN 09-OCT-09 One or more non-
system datafiles are missing
 Impact: See impact for individual child failures
 List of child failures for parent failure ID 422
 Failure ID Priority Status Time Detected Summary

 520 HIGH OPEN 09-OCT-09 Datafile 7:
'/u01/app/oracle/oradata/users2.dbf' is missing
 Impact: Some objects in tablespace USERS2 might be
unavailable

analyzing automatic repair options; this may take some time
using channel ORA_DISK_1
analyzing automatic repair options complete

Mandatory Manual Actions
=====
no manual actions available

Optional Manual Actions
=====
1. If file /u01/app/oracle/oradata/users2.dbf was
unintentionally renamed or moved, restore it

Automated Repair Options
=====
Option Repair Description

1 Restore and recover datafile 7
 Strategy: The repair includes complete media recovery with
no data loss
 Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_3823931837.hm

RMAN>

```

## 练习 20-3：使用 RMAN 修复数据库（续）

- 8) 查看修复脚本。

```

RMAN> REPAIR FAILURE PREVIEW;

Strategy: The repair includes complete media recovery with no
data loss
Repair script: Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_3823931837.hm

contents of repair script:
 # restore and recover datafile
 restore datafile 7;
 recover datafile 7;

RMAN>

```

- 9) 启动自动修复，然后重新启动数据库并退出 RMAN。

```

RMAN> REPAIR FAILURE;

Strategy: The repair includes complete media recovery with no
data loss
Repair script: Repair script:
/u01/app/oracle/diag/rdbms/orcl/orcl/hm/reco_3823931837.hm

contents of repair script:
 # restore and recover datafile
 restore datafile 7;
 recover datafile 7;

Do you really want to execute the above repair (enter YES or
NO)? YES
executing repair script

Starting restore at 09-OCT-09
using channel ORA_DISK_1

channel ORA_DISK_1: starting datafile backup set restore
channel ORA_DISK_1: specifying datafile(s) to restore from
backup set
channel ORA_DISK_1: restoring datafile 00007 to
/u01/app/oracle/oradata/users2.dbf
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_10_09/nnndf0_tag20091009t103547_0.296
.699791891
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_10_09/nnndf0_tag20091009t10354
7_0.296.699791891 tag=TAG20091009T103547
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:01
Finished restore at 09-OCT-09

```

## 练习 20-3：使用 RMAN 修复数据库（续）

```
Starting recover at 09-OCT-09
using channel ORA_DISK_1

starting media recovery

archived log for thread 1 with sequence 119 is already on disk
as file
+FRA/orcl/archivelog/2009_10_09/thread_1_seq_119.301.699792001
archived log for thread 1 with sequence 120 is already on disk
as file
+FRA/orcl/archivelog/2009_10_09/thread_1_seq_120.300.699792003
archived log for thread 1 with sequence 121 is already on disk
as file
+FRA/orcl/archivelog/2009_10_09/thread_1_seq_121.299.699792005
channel ORA_DISK_1: starting archived log restore to default
destination
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=115
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=116
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=117
channel ORA_DISK_1: restoring archived log
archived log thread=1 sequence=118
channel ORA_DISK_1: reading from backup piece
+FRA/orcl/backupset/2009_10_09/annnf0_tag20091009t103929_0.303
.699791971
channel ORA_DISK_1: piece
handle=+FRA/orcl/backupset/2009_10_09/annnf0_tag20091009t10392
9_0.303.699791971 tag=TAG20091009T103929
channel ORA_DISK_1: restored backup piece 1
channel ORA_DISK_1: restore complete, elapsed time: 00:00:07
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_115.293.6997
96223 thread=1 sequence=115
channel default: deleting archived log(s)
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_115.293.6997
96223 RECID=24 STAMP=699796225
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_116.291.6997
96225 thread=1 sequence=116
channel default: deleting archived log(s)
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_116.291.6997
96225 RECID=26 STAMP=699796226
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_117.292.6997
96225 thread=1 sequence=117
channel default: deleting archived log(s)
```

## 练习 20-3：使用 RMAN 修复数据库（续）

```

archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_117.292.6997
96225 RECID=25 STAMP=699796225
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_118.290.6997
96227 thread=1 sequence=118
channel default: deleting archived log(s)
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_118.290.6997
96227 RECID=27 STAMP=699796227
archived log file
name=+FRA/orcl/archivelog/2009_10_09/thread_1_seq_119.301.6997
92001 thread=1 sequence=119
media recovery complete, elapsed time: 00:00:04
Finished recover at 09-OCT-09
repair failure complete

Do you want to open the database (enter YES or NO)? YES
database opened

RMAN> exit

Recovery Manager complete.
$

```

10) 使用 SQL\*Plus 检查数据库是否已启动并正在运行。

```

$ sqlplus / as sysdba

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:56:06
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> select open_mode from v$database;

OPEN_MODE

READ WRITE

```

## 练习 20-3：使用 RMAN 修复数据库（续）

```
SQL> exit
Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

11) 要清理环境，请执行 dra\_cleanup.sh 脚本：

```
$./dra_cleanup.sh

For cleanup of demo:
 * Lock HR account
 * Disable ARCHIVELOG mode

SQL*Plus: Release 11.2.0.1.0 Production on Fri Oct 9 11:57:14
2009

Copyright (c) 1982, 2009, Oracle. All rights reserved.

Connected to:
Oracle Database 11g Enterprise Edition Release 11.2.0.1.0 -
Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options

SQL> SQL>
Tablespace dropped.

SQL>
Tablespace dropped.

SQL> SQL>
User altered.

SQL>
System altered.

SQL> SQL> Database closed.
Database dismounted.
ORACLE instance shut down.
SQL> ORACLE instance started.

Total System Global Area 422670336 bytes
Fixed Size 1336960 bytes
Variable Size 318769536 bytes
Database Buffers 96468992 bytes
Redo Buffers 6094848 bytes
```



## 练习 20-3：使用 RMAN 修复数据库（续）

```
Database mounted.
SQL>
Database altered.

SQL>
Database altered.

SQL> Database log mode No Archive Mode
Automatic archival Disabled
Archive destination USE_DB_RECOVERY_FILE_DEST
Oldest online log sequence 120
Current log sequence 122
SQL> SQL> "Cleanup finished."
SQL> Disconnected from Oracle Database 11g Enterprise Edition
Release 11.2.0.1.0 - Production
With the Partitioning, Automatic Storage Management, OLAP,
Data Mining and Real Application Testing options
$
```

These eKit materials are to be used ONLY by you for the express purpose SELF STUDY. SHARING THE FILE IS STRICTLY PROHIBITED.

# 调度程序增强功能



ORACLE®

版权所有 © 2010, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 使用轻量作业
- 使用调度程序电子邮件通知
- 创建文件监视器
- 创建远程数据库作业
- 创建多个目标作业

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

## 轻量作业

### 持久轻量作业：

- 是使用作业模板创建的
- 是可恢复的
- `JOB_STYLE => LIGHTWEIGHT`

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 轻量作业

有些客户需要每秒创建数百个作业。对于常规作业，每个作业都要创建一个数据库对象，用于描述作业、修改多个表以及在进程中创建重做。在 Oracle Database 11g 调度程序中，有一个“持久轻量作业”。轻量作业的目的是减少启动作业所需的开销和时间。将为作业创建最少的元数据。这可以减少启动作业时所需的时间和创建的重做。

要达到这些目的，轻量作业在磁盘上占用了很小的空间用于作业元数据和存储运行时数据。磁盘上的占用空间还支持在 RAC 环境中进行恢复和负载平衡。轻量作业始终是使用作业模板创建的，作业模板可以是存储过程或程序。存储过程可以保存作业所需的所有信息。可能会设置一些作业属性（如作业参数）。

作业模板是使用 `DBMS_SCHEDULER.CREATE_PROGRAM` 过程创建的。

Oracle Database 11g 继续支持基于数据库对象的作业，自 Oracle 10g 中首次引入 Oracle 调度程序以来，这些作业就一直存在。轻量作业的目的不是取代这些作业，因为每个作业类型都有其自身的优势，用户可以根据自己的需求灵活地选择作业。

## 选择正确的作业

- **常规作业**
  - 开销最高
  - 最佳恢复
  - 最灵活
- **持久轻量作业**
  - 开销较低
  - 部分恢复
  - 属性更改受限

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 选择正确的作业

这两种作业类型的优点和缺点如下：

- **常规作业**的灵活性最高，但在执行创建/删除时会产生大量开销。可以用单个命令创建作业。用户可以细粒度地控制对作业的权限，还可以使用属于其它用户的程序或存储过程。常规作业需要创建和删除作业数据库对象。此操作将更新多个表及关联的重做。如果用户创建的作业数量相对较少，并且这些作业的运行频率相对较低，则应选择常规作业。
- **持久轻量作业**在创建/删除时间方面有很大改进，因为它没有创建数据库对象的开销。每个轻量作业都是使用作业模板创建的，并作为程序进行存储。因为持久轻量作业在运行时要向磁盘写入状态信息，所以在执行时应该只有少量改进。持久轻量作业有多种限制：
  - 用户无法设置对这些作业的权限，它们将从父作业模板中继承其权限。
  - 必须使用模板，无法创建完全自包含的持久轻量作业。
  - 只有一些特定的作业属性是可设置的，如 `JOB_ARGUMENTS`。

如果用户需要在非常短的时间内创建大量作业（每秒 10 - 100 个作业），并且有一组程序（作业模板）可供使用，则最适合使用轻量作业。

## 调度程序电子邮件通知：概览

- 可对作业进行配置，以便在作业状态更改时发送电子邮件通知。
- 电子邮件消息可由指定作业状态事件列表中的任一事件（如作业失败）触发。
- 可为一个作业配置多个电子邮件通知。
- 一个电子邮件消息可发送至多个收件人。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 调度程序电子邮件通知：概览

使用作业电子邮件通知功能，可以向现有作业添加电子邮件通知，以便作业中发生受关注事件后向指定的电子邮件地址发送通知。对于每个作业，可以针对不同的事件添加通知。可以将一个电子邮件通知发送至多个收件人。

## 使用调度程序电子邮件通知

- 要使用调度程序电子邮件通知，请在各个数据库上执行以下配置步骤：
  - 将调度程序 `email_server` 属性设置为用于发送电子邮件消息的 **SMTP** 服务器的地址。
  - （可选）将调度程序 `email_sender` 属性设置为默认的发件人电子邮件地址。
- 针对希望系统为其发送电子邮件消息的每一个作业，执行以下操作：
  - 使用 `DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION` 过程为指定的作业添加电子邮件通知。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用调度程序电子邮件通知

要启用电子邮件通知功能，请在每个数据库上执行下列一次性配置步骤：

- 设置 `email_server` 调度程序属性。
- （可选）您可以使用 `email_sender` 调度程序属性为电子邮件通知指定默认的发件人电子邮件地址。

创建作业后，执行 `DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION` 过程为作业添加一个或多个通知。

后续页面中提供了更多信息。



## 设置 email\_server 属性

指定 **SMTP** 服务器地址，调度程序将使用该地址来发送关于作业状态事件的电子邮件通知：

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE
('email_server', 'host[:port]');
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 设置 email\_server 属性

调度程序 email\_server 属性指定 SMTP 服务器地址，调度程序将使用该地址发送关于作业状态事件的电子邮件通知。该属性的格式为 `host[:port]`，其中 `host` 是 SMTP 服务器的主机名或 IP 地址，而 `port` 是 SMTP 服务器监听的 TCP 端口。如果未指定 `port`，则使用默认端口 25。

如果未设置 email\_server 属性，将其设置为 NULL，或设置为无效的 SMTP 服务器地址，则调度程序将无法发送作业状态电子邮件通知。不支持要求安全套接字层 (SSL) 连接或要求用户验证的 SMTP 服务器。

## 设置 email\_sender 属性

为作业状态电子邮件通知指定默认的发件人电子邮件地址：

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE
('email_sender','valid_email_address');
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 设置 email\_sender 属性

可以使用 email\_sender 属性为作业状态电子邮件通知指定默认的发件人电子邮件地址。指定的电子邮件地址必须有效。如果未设置该属性或将其设置为 NULL，则未指定发件人地址的作业状态电子邮件通知在电子邮件标头中没有“发件人”地址。

## 使用 ADD\_JOB\_EMAIL\_NOTIFICATION

```
DBMS_SCHEDULER.ADD_JOB_EMAIL_NOTIFICATION (
 job_name IN VARCHAR2,
 recipients IN VARCHAR2,
 sender IN VARCHAR2 DEFAULT NULL,
 subject IN VARCHAR2
 DEFAULT dbms_scheduler.default_notification_subject,
 body IN VARCHAR2
 DEFAULT dbms_scheduler.default_notification_body,
 events IN VARCHAR2
 DEFAULT 'JOB_FAILED,JOB_BROKEN,JOB_SCH_LIM_REACHED,
 JOB_CHAIN_STALLED,JOB_OVER_MAX_DUR',
 filter_condition IN VARCHAR2 DEFAULT NULL);
```

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 ADD\_JOB\_EMAIL\_NOTIFICATION

DBMS\_SCHEDULER.ADD\_JOB\_EMAIL\_NOTIFICATION 过程可添加一个或多个作业电子邮件通知。只要作业生成了列出的任一事件，系统就会将电子邮件消息发送至指定的收件人地址。作业将自动修改以便引发这些事件。如果指定了过滤条件，则只有与 FILTER\_CONDITION 中指定的条件相匹配的事件才会生成电子邮件消息。

如果未设置 EMAIL\_SERVER 调度程序属性或指定的作业不存在，则该过程将失败。

**注：**调用该过程的用户必须是该作业的所有者，并具有 CREATE ANY JOB 系统权限，或对该作业有 ALTER 权限。

过程参数如下：

- **JOB\_NAME：** 将其发送电子邮件通知的作业的名称。该值不能为 NULL。
- **RECIPIENTS：** 接收通知的电子邮件地址（以逗号分隔）的列表。电子邮件通知将发送至所有指定的电子邮件地址。该值不能为 NULL。
- **SENDER：** 电子邮件通知发件人的电子邮件地址。如果为 NULL，则将使用 DEFAULT\_EMAIL\_SENDER 调度程序属性的值（如果其为有效的电子邮件地址）。

## 使用 **ADD\_JOB\_EMAIL\_NOTIFICATION** (续)

- **SUBJECT:** 通知电子邮件的主题可以包含以下将用值替换的变量: %job\_owner%、%job\_name%、%event\_type%、%event\_timestamp%、%log\_id%、%error\_code%、%error\_message%、%run\_count%、%failure\_count%、%retry\_count%、%job\_subname% 和 %job\_class\_name%。
- **BODY:** 用作通知电子邮件消息正文; 可以包含任何在 SUBJECT 中有效的变量。
- **EVENTS:** 将为其发送电子邮件通知的事件 (以逗号分隔) 的列表; 不能为 NULL。请参阅 JOBS 的 RAISE\_EVENTS 属性的事件列表以了解有效的事件。
- **FILTER\_CONDITION:** 用于过滤要为其发送电子邮件通知的事件。如果为 NULL (默认值), 则只要指定事件出现, 系统就会向所有指定的收件人地址发送通知。

## 使用 REMOVE\_JOB\_EMAIL\_NOTIFICATION

```
DBMS_SCHEDULER.REMOVE_JOB_EMAIL_NOTIFICATION (
 job_name IN VARCHAR2,
 recipients IN VARCHAR2 DEFAULT NULL,
 events IN VARCHAR2 DEFAULT NULL);
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 使用 REMOVE\_JOB\_EMAIL\_NOTIFICATION

DBMS\_SCHEDULER.REMOVE\_JOB\_EMAIL\_NOTIFICATION 过程用于删除指定作业的一个或多个电子邮件通知。

**注：**调用该过程的用户必须是该作业的所有者，并具有 CREATE ANY JOB 系统权限，或对该作业有 ALTER 权限。

过程参数如下：

- **JOB\_NAME：** 将为其删除电子邮件通知的作业的名称；不能为 NULL。
- **RECIPIENTS：** 要为其删除通知的电子邮件地址（以逗号分隔）的列表。如果该值为 NULL，则将针对给定作业和指定事件删除所有通知。
- **EVENTS：** 要为其删除电子邮件通知的事件（以逗号分隔）的列表。如果该值为 NULL，则将针对给定作业和指定电子邮件地址删除所有通知。

## 电子邮件通知：新增的字典视图

查询以下视图可确定已设置了哪些电子邮件通知：

- **DBA\_SCHEDULER\_NOTIFICATIONS**
- **ALL\_SCHEDULER\_FILE\_NOTIFICATIONS**
- **USER\_SCHEDULER\_FILE\_NOTIFICATIONS**

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 电子邮件通知：新增的字典视图

新增了三个包含电子邮件通知信息的视图：DBA\_SCHEDULER\_NOTIFICATIONS、ALL\_SCHEDULER\_NOTIFICATIONS 和 USER\_SCHEDULER\_NOTIFICATIONS。

DBA\_SCHEDULER\_NOTIFICATIONS 和 ALL\_SCHEDULER\_NOTIFICATIONS 视图具有以下列（USER\_ 视图没有 OWNER 列）：

- OWNER：通知所针对的作业的所有者
- JOB\_NAME：通知所针对的作业的名称
- JOB\_SUBNAME：通知所针对的作业的子名称
- RECIPIENT：电子邮件通知要发送到的电子邮件地址
- SENDER：发送电子邮件通知的电子邮件地址
- SUBJECT：电子邮件通知的主题
- BODY：电子邮件通知的正文
- FILTER\_CONDITION：指定要针对哪些作业事件发送电子邮件的过滤器
- EVENT：要为其发送电子邮件的作业事件
- EVENT\_FLAG：要为其发送电子邮件的作业事件的事件编号

## 调度程序文件监视器：概览

- 调度程序具有一项内置功能，用于“监视”本地系统或远程系统上的文件。
- 调度程序检测到“受监视”的文件到达后，将引发可由基于事件的作业提取的事件。
- 基于事件的作业或调度可引用文件监视器。
- 对于远程计算机，文件监视功能将通过调度程序代理处理。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 调度程序文件监视器：概览

“文件监视器”是一个数据库对象，用于定义文件的位置、名称和其它属性。当文件到达系统时，调度程序将启动一个作业。可以通过使用 `DBMS_SCHEDULER.CREATE_FILE_WATCHER` 过程创建文件监视器对象，然后创建任意数量的引用该文件监视器的基于事件的作业或事件调度。当文件监视器检测到指定文件到达后，它将引发文件到达事件。由文件到达事件启动的作业可以检索该事件消息，以获取有关新到达文件的信息。

文件监视器可以监视本地系统（运行 Oracle DB 的服务器）或运行调度程序代理的远程系统上的文件。

## 创建文件监视器和基于事件的作业

要创建文件监视器和在指定文件到达后启动的基于事件的作业，请执行以下任务：

1. 创建调度程序身份证明对象（身份证明），该对象用于通过主机操作系统验证，以获取文件的访问权限。
2. 创建文件监视器。
3. 创建其元数据参数引用事件消息的调度程序程序对象。
4. 创建引用文件监视器的基于事件的作业。
5. 启用文件监视器、程序和作业。

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 创建文件监视器和基于事件的作业

您可以执行幻灯片中列出的任务，创建文件监视器和在指定文件到达后启动的基于事件的作业。后续页面将对这些步骤进行详细说明。



## 步骤 1：创建调度程序身份证明对象

- 为需要访问受监视文件的操作系统用户创建身份证明：

```
BEGIN
DBMS_SCHEDULER.CREATE_CREDENTIAL(' ', ' ', ' ');
END;
```

- 向文件监视器将启动的基于事件的作业所属的方案授予对该身份证明的 EXECUTE 权限：

```
GRANT EXECUTE ON credential_name to user_name;
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 1：创建调度程序身份证明对象

文件监视器需要有一个调度程序身份证明对象（身份证明），该对象用于通过主机操作系统验证，以获取文件的访问权限。

**注：**基于事件的作业和身份证明驻留在同一方案中时，无需授予方案对身份证明的 EXECUTE 权限。

## 步骤 2：创建文件监视器

- 创建文件监视器：

```
BEGIN
DBMS_SCHEDULER.CREATE_FILE_WATCHER(
FILE_WATCHER_NAME => '',
DESTINATION =>,
DIRECTORY_PATH => '',
FILE_NAME => '',
CREDENTIAL_NAME => '',
ENABLED =>);
END;
```

- 将对文件监视器的 EXECUTE 权限授予引用该文件监视器的基于事件的作业所属的方案：

```
GRANT EXECUTE ON file_watcher_name to user_name;
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 2：创建文件监视器

通过使用 DBMS\_SCHEDULER.CREATE\_FILE\_WATCHER 过程创建文件监视器。下一页中提供了详细信息。

**注：**基于事件的作业和文件监视器驻留在同一方案中时，无需授予方案对文件监视器的 EXECUTE 权限。

# 使用 DBMS\_SCHEDULER.CREATE\_FILE\_WATCHER

| 参数                    | 数据类型                   | 默认值  |
|-----------------------|------------------------|------|
| FILE_WATCHER_NAME     | VARCHAR2               | 无默认值 |
| DESTINATION           | VARCHAR2               | NULL |
| DIRECTORY_PATH        | VARCHAR2               | 无默认值 |
| FILE_NAME             | VARCHAR2               | 无默认值 |
| CREDENTIAL_NAME       | VARCHAR2               | 无默认值 |
| MIN_FILE_SIZE         | PLS_INTEGER            | 0    |
| STEADY_STATE_DURATION | INTERVAL DAY TO SECOND | NULL |
| COMMENTS              | VARCHAR2               | NULL |
| ENABLED               | BOOLEAN                | TRUE |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

## 使用 DBMS\_SCHEDULER.CREATE\_FILE\_WATCHER

通过使用 DBMS\_SCHEDULER.CREATE\_FILE\_WATCHER 过程创建称为文件监视器的数据库对象。该过程具有以下参数：

- FILE\_WATCHER\_NAME：文件监视器的名称。
- DESTINATION：文件将要到达的远程目标。如果为 NULL，则目标为本地计算机。
- DIRECTORY\_PATH：文件将要位于的目录。允许在路径名的开头使用单个通配符“？”，这表示 ORACLE\_HOME 的值。该值不能为 NULL。
- FILE\_NAME：被“监视”的文件的名称。允许使用“？”（任意单个字符）和“\*”（任意 0 个或多个字符的序列）这两种通配符。该值不能为 NULL。
- CREDENTIAL\_NAME：有效身份证明对象的名称。如果该创建调用成功执行，则文件监视器所有者将具有对该对象的执行权限。该值不能为 NULL。
- MIN\_FILE\_SIZE：文件的最小大小，只有文件达到该大小后文件监视器才会报告找到了文件；不能为 NULL。
- STEADY\_STATE\_DURATION：文件保持不变的最小时间间隔，只有达到该时间间隔后文件监视器才会报告找到了文件。如果为 NULL，系统将使用默认间隔值。
- COMMENTS：有关此文件监视器的注释。
- ENABLED：此文件监视器是否已启用。

## 步骤 3：创建调度程序程序对象

- 创建程序：

```
BEGIN
DBMS_SCHEDULER.CREATE_PROGRAM(
PROGRAM_NAME => '',
PROGRAM_TYPE => 'STORED_PROCEDURE',
PROGRAM_ACTION => '',
NUMBER_OF_ARGUMENTS => 1,
ENABLED => FALSE);
END;
```

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 3：创建调度程序程序对象

创建其元数据参数引用事件消息的调度程序程序对象。

## 步骤 3：创建调度程序程序对象

- 使用 `EVENT_MESSAGE` 属性定义元数据参数：

```
BEGIN
DBMS_SCHEDULER.DEFINE_METADATA_ARGUMENT (
PROGRAM_NAME => '',
METADATA_ATTRIBUTE => 'event_message',
ARGUMENT_POSITION => 1);
END;
```

- 创建一个参数类型为  
`SYS.SCHEDULER_FILEWATCHER_RESULT` 的存储过程，  
供该程序调用。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 3：创建调度程序程序对象（续）

然后，创建一个供该程序调用的存储过程。该存储过程必须具有一个类型为 `SYS.SCHEDULER_FILEWATCHER_RESULT` 的参数，这是事件消息的数据类型。该参数的位置必须与所定义的元数据参数的位置相匹配。该过程可以访问此抽象数据类型的属性，以了解有关已到达文件的信息。下一页中提供了有关 `SYS.SCHEDULER_FILEWATCHER_RESULT` 的更多信息。

# 使用 SCHEDULER\_FILEWATCHER\_RESULT 对象类型

| 参数                | 数据类型                               | 定义                                      |
|-------------------|------------------------------------|-----------------------------------------|
| DESTINATION       | VARCHAR2(4000)                     | 文件所在的目标位置                               |
| DIRECTORY_PATH    | VARCHAR2(4000)                     | 文件所在的目录                                 |
| ACTUAL_FILE_NAME  | VARCHAR2(4000)                     | 实际的文件名                                  |
| FILE_SIZE         | NUMBER                             | 文件大小                                    |
| FILE_TIMESTAMP    | TIMESTAMP WITH TIME<br>ZONE        | 文件的时间戳                                  |
| TS_MS_FROM_EPOCH  | NUMBER                             | 仅供内部使用                                  |
| MATCHING_REQUESTS | SCHEDULER_FILEWAT<br>CHER_REQ_LIST | 定义为 SCHEDULER_<br>FILEWATCHER_REQUEST 表 |

ORACLE

版权所有 © 2010，Oracle。保留所有权利。

## 使用 SCHEDULER\_FILEWATCHER\_RESULT 对象类型

SCHEDULER\_FILEWATCHER\_RESULT 是文件到达事件消息的数据类型。可以将该事件消息作为基于事件的作业的参数（或基于事件的作业引用的程序的参数）进行访问。该消息包含定位和处理到达本地或远程系统的文件所需的信息。

## 使用 SCHEDULER\_FILEWATCHER\_REQUEST 对象类型

| 参数                  | 数据类型           | 定义         |
|---------------------|----------------|------------|
| OWNER               | VARCHAR2(4000) | 文件监视器所有者   |
| NAME                | VARCHAR2(4000) | 文件监视器的名称   |
| REQUESTED_PATH_NAME | VARCHAR2(4000) | 请求的路径名     |
| REQUESTED_FILE_NAME | VARCHAR2(4000) | 所需文件名      |
| CREDENTIAL_OWNER    | VARCHAR2(4000) | 所用身份证明的所有者 |
| CREDENTIAL_NAME     | VARCHAR2(4000) | 身份证明名称     |
| MIN_FILE_SIZE       | NUMBER         | 最小文件大小     |
| STEADY_STATE_DUR    | NUMBER         | 稳定状态持续时间   |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 SCHEDULER\_FILEWATCHER\_REQUEST 对象类型

SCHEDULER\_FILEWATCHER\_REQUEST 对象是在 SCHEDULER\_FILEWATCHER\_RESULT 对象类型的 MATCHING\_REQUESTS 属性中返回的。

## 步骤 4：创建引用文件监视器的基于事件的作业

- 创建作业：

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
JOB_NAME => '',
PROGRAM_NAME => '',
EVENT_CONDITION => NULL,
QUEUE_SPEC => '<name_of_file_watcher>',
AUTO_DROP => FALSE,
ENABLED => FALSE);
END;
```

- （可选）使作业处理文件到达事件的每个实例：

```
BEGIN
DBMS_SCHEDULER.SET_ATTRIBUTE('', 'PARALLEL_INSTANCES',
 TRUE); END;
```

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 4：创建引用文件监视器的基于事件的作业

可以使用 DBMS\_SCHEDULER.SET\_ATTRIBUTE 过程使作业处理文件到达事件的每个实例，即使该作业已经在处理之前的一个事件。将 PARALLEL\_INSTANCES 属性设置为 TRUE。这将使作业作为轻量作业运行，以便快速启动该作业的多个实例。

如果将 PARALLEL\_INSTANCES 设置为默认值 FALSE，则当基于事件的作业已经在处理一个事件时，此时出现的文件监视器事件将被丢弃。



## 步骤 5：启用对象

启用文件监视器、程序和作业：

```
BEGIN
DBMS_SCHEDULER.ENABLE('MY_FILE_WATCHER,
 DSSUSER.MY_PROGRAM,
 DSSUSER.MY_JOB');
END;
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 步骤 5：启用对象

使用 `DBMS_SCHEDULER.ENABLE` 过程启用文件监视器、程序和作业。

## 启用来自远程系统的文件到达事件

要在远程系统上启用文件到达事件触发功能，请执行以下任务：

1. 对数据库进行设置，使其运行远程外部作业。
2. 在第一个远程系统上安装、配置、注册并启动调度程序代理。
3. 在其它远程系统上重复步骤 2。

**ORACLE**

版权所有 © 2010，Oracle。保留所有权利。

### 启用来自远程系统的文件到达事件

要从某个远程系统接收文件到达事件，必须在该系统上安装调度程序代理，且必须在数据库中注册该代理。该远程系统在生成文件到达事件时不需要 Oracle DB 实例。

有关详细信息，请参阅《Oracle Database Administrator's Guide 11g Release 2》。

# 使用 DBMS\_SCHEDULER.DROP\_FILE\_WATCHER

| 参数                | 数据类型     | 默认值   |
|-------------------|----------|-------|
| FILE_WATCHER_NAME | VARCHAR2 | 无默认值  |
| FORCE             | BOOLEAN  | FALSE |



版权所有 © 2010, Oracle。保留所有权利。

## 使用 DBMS\_SCHEDULER.DROP\_FILE\_WATCHER

可以通过使用 DBMS\_SCHEDULER.DROP\_FILE\_WATCHER 过程删除文件监视器数据库对象。该过程具有以下参数：

- FILE\_WATCHER\_NAME：文件监视器的名称。
- FORCE：强制执行标记。设置为 FALSE 时，如果存在任何依赖于该文件监视器的作业，则不会将该文件监视器删除。设置为 TRUE 时，引用被删除文件监视器的作业将被禁用。

如果在删除文件监视器时存在引用该文件监视器且正在执行的作业，则这些作业将继续执行，不会受此过程的影响。

## 文件监视器：对现有 DBMS\_SCHEDULER 过程的更改

- **CREATE\_JOB** 和 **CREATE\_JOBS**: **QUEUE\_SPEC** 参数可以是文件监视器名称。
- **CREATE\_EVENT\_SCHEDULE**: **QUEUE\_SPEC** 参数可以是文件监视器名称。
- **SET\_ATTRIBUTE** 和 **GET\_ATTRIBUTE**: 这些调用可用于设置和获取文件监视器的各个属性。
- **ENABLE** 和 **DISABLE**: 这些调用可接受文件监视器名称作为参数。

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 文件监视器：对现有 DBMS\_SCHEDULER 过程的更改

**CREATE\_JOB** 和 **CREATE\_JOBS**: 这些调用的 **QUEUE\_SPEC** 参数可以是文件监视器名称，不接受队列名称。创建的作业将是基于事件的作业，该作业的源队列是文件监视器排列其结果的队列。请勿用文件监视器指定代理名称。

**CREATE\_EVENT\_SCHEDULE**: 该调用的 **QUEUE\_SPEC** 参数可以是文件监视器名称，不接受队列名称。创建的调度将是基于事件的调度，该调度的源队列是文件监视器排列其结果的队列。请勿用文件监视器指定代理名称。

**SET\_ATTRIBUTE** 和 **GET\_ATTRIBUTE**: 这两个调用可以接受文件监视器名称作为参数。对于文件监视器，以下属性是合法的: **DESTINATION**、**DIRECTORY\_PATH**、**FILE\_NAME**、**CREDENTIAL\_NAME**、**MIN\_FILE\_SIZE**、**STEADY\_STATE\_DURATION** 和 **COMMENTS**。

**ENABLE** 和 **DISABLE**: 这两个调用可以接受文件监视器名称作为参数。只有当文件监视器的所有属性均已设置为合法值，且文件监视器所有者对当前所用的身份证明具有执行权限时，该文件监视器才会启用。如果文件监视器被禁用且强制执行标记设置为 **FALSE**，则不应存在依赖于该文件监视器的作业。如果存在，则该调用将失败。

## 文件监视器：对现有属性的更改

作业和调度的 `EVENT_SPEC` 属性可以接受文件监视器名称。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 文件监视器：对现有属性的更改

作业和调度的 `EVENT_SPEC` 属性可以接受文件监视器名称，不接受队列名称。生成的作业/调度将是基于事件的，以文件监视器队列作为其源队列。

## 文件监视器：新增的字典视图

可查询以下视图查看文件监视器配置信息：

- **DBA\_SCHEDULER\_FILE\_WATCHERS**
- **ALL\_SCHEDULER\_FILE\_WATCHERS**
- **USER\_SCHEDULER\_FILE\_WATCHERS**

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 文件监视器：新增的字典视图

新增了三个提供文件监视器信息的视图：DBA\_SCHEDULER\_FILE\_WATCHERS、ALL\_SCHEDULER\_FILE\_WATCHERS 和 USER\_SCHEDULER\_FILE\_WATCHERS。

DBA\_SCHEDULER\_FILE\_WATCHERS 和 ALL\_SCHEDULER\_FILE\_WATCHERS 视图具有以下列（USER\_ 视图没有 OWNER 列）：

- OWNER：文件监视器所有者的名称
- FILE\_WATCHER\_NAME：文件监视器的名称
- ENABLED：可以是 TRUE 或 FALSE，具体取决于是否启用了文件监视器
- DESTINATION：文件将位于的远程目标位置；如果目标位置为本地计算机，则设置为 NULL
- DIRECTORY\_PATH：文件所在目录的路径名
- FILE\_NAME：要查找的文件的名称
- CREDENTIAL\_OWNER：身份证明所有者的名称
- CREDENTIAL\_NAME：身份证明的名称
- MIN\_FILE\_SIZE：文件的最小大小，只有文件达到该大小后文件监视器才会报告文件已到达

### 文件监视器：新增的字典视图（续）

- `STEADY_STATE_DURATION`: 文件保持不变至少应达到的最小时间量，只有达到该时间量后文件监视器才会报告文件已到达
- `LAST_MODIFIED_TIME`: 该文件监视器的上次修改时间
- `COMMENTS`: 注释

## 文件监视器：现有视图中新增的列

**\*\_SCHEDULER\_JOBS 和 \*\_SCHEDULER\_SCHEDULES 视图  
中新增的列：**

| 列名                 | 数据类型         | 说明                                   |
|--------------------|--------------|--------------------------------------|
| FILE_WATCHER_OWNER | VARCHAR2(30) | 文件监视器所有者的名称。如果不是基于文件监视器的，则该列将为 NULL。 |
| FILE_WATCHER_NAME  | VARCHAR2(30) | 文件监视器的名称。如果不是基于文件监视器的，则该列将为 NULL。    |

ORACLE

版权所有 © 2010，Oracle。保留所有权利。

### 文件监视器：现有视图中新增的列

\*\_SCHEDULER\_JOBS 和 \*\_SCHEDULER\_SCHEDULES 视图中新增了用于提供文件监视器信息的列：

- FILE\_WATCHER\_NAME：文件监视器的名称。
- FORCE：强制执行标记。设置为 FALSE 时，如果存在任何依赖于该文件监视器的作业，则不会将该文件监视器删除。



## 调度程序远程数据库作业：概览

- 创建一个作业在同一主机或远程主机上的另一个数据库实例上运行存储过程和匿名 PL/SQL 块。
- 目标数据库可以是任何版本的 Oracle DB。
- 必须使用 `DBMS_SCHEDULER.CREATE_DATABASE_DESTINATION` 来标识远程数据库。
- 必须使用 `DBMS_SCHEDULER.CREATE_CREDENTIAL` 来标识用户。
- 作业类型为 `PLSQL_BLOCK` 和 `STORED_PROCEDURE` 的作业可以是 `SET_ATTRIBUTE` 调用的 `DESTINATION` 和 `CREDENTIAL` 属性的目标。

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 调度程序远程数据库作业：概览

现在，可以创建一个作业在同一主机或远程主机上的另一个数据库实例上运行存储过程和匿名 PL/SQL 块。目标数据库可以是任何版本的 Oracle DB。

现在没有用于支持远程数据库作业的新过程，但是对现有 `DBMS_SCHEDULER` 过程进行了更改以支持该功能。

必须使用 `DBMS_SCHEDULER.CREATE_DATABASE_DESTINATION` 来标识要在其上运行远程数据库作业的远程数据库。必须首先调用

`DBMS_SCHEDULER.CREATE_DB_DESTINATION` 来注册您要在其上运行远程数据库作业的每个远程数据库。

必须使用 `DBMS_SCHEDULER.CREATE_CREDENTIAL` 来标识在连接远程数据库时要使用的用户身份。

后续页面中提供了更多信息。

# 使用 CREATE\_DATABASE\_DESTINATION

| 参数               | 数据类型     | 默认值  |
|------------------|----------|------|
| DESTINATION_NAME | VARCHAR2 | 无默认值 |
| AGENT            | VARCHAR2 | 无默认值 |
| TNS_NAME         | VARCHAR2 | 无默认值 |
| COMMENTS         | VARCHAR2 | Null |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

## 使用 DBMS\_SCHEDULER.CREATE\_DATABASE\_DESTINATION

要创建远程数据库作业，必须首先使用 DBMS\_SCHEDULER.CREATE\_DATABASE\_DESTINATION 过程创建代表远程数据库的数据库目标。该过程具有以下参数：

- DESTINATION\_NAME：表示要连接的数据库的目标名称
- AGENT：表示用于连接远程数据库的代理的外部目标名称
- TNS\_NAME：任何有效的 Oracle Net 连接标识符（本地数据库 tnsnames.ora 文件或实际 Oracle Net 连接字符串中的网络服务名）。如果未指定该参数，则必须对调度程序代理进行配置，以使其具有用于连接的默认数据库
- COMMENTS：注释

## 使用 DROP\_DATABASE\_DESTINATION

| 参数               | 数据类型     | 默认值  |
|------------------|----------|------|
| DESTINATION_NAME | VARCHAR2 | 无默认值 |

- 目标组中所有引用此目标的成员都将被删除。
- 所有指向该目标的作业实例都将被删除。
- 针对此目标运行的作业将停止。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 DBMS\_SCHEDULER.DROP\_DATABASE\_DESTINATION

可使用 DBMS\_SCHEDULER.DROP\_DATABASE\_DESTINATION 过程删除目标。该过程具有以下参数：

- DESTINATION\_NAME：表示要连接的数据库的目标名称

数据库目标被删除后，目标组中所有引用该目标的成员也将被删除。

数据库目标被删除后，所有指向该目标的作业实例也将被删除。针对此目标运行的作业将停止。

## 创建远程数据库作业：配置

执行以下配置步骤：

1. 设置远程作业的发起数据库。
2. 使用 `DBMS_SCHEDULER.CREATE_DB_DESTINATION` 指定要在其上运行远程数据库作业的每个远程数据库。
3. 使用 `DBMS_SCHEDULER.CREATE_CREDENTIAL` 为所有要连接到远程数据库的用户创建身份证明。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 创建远程数据库作业：配置

可以执行幻灯片中列出的任务为远程数据库作业配置环境。

有关详细示例，请参阅《Oracle Database Administrator's Guide 11g Release 2》。

## 为远程作业设置数据库

- 验证是否已安装 XML DB。
- 启用与数据库的 HTTP 连接。

```
BEGIN
DBMS_XDB.SETHTTPPORT(port);
END;
```

- 执行 `prvtrsch.plb` 脚本。
- 设置调度程序代理的注册口令。

```
BEGIN
DBMS_SCHEDULER.SET_AGENT_REGISTRATION_PASS('password');
END;
```

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 为远程作业设置数据库

在发起数据库上执行幻灯片中所列步骤。

## 创建远程数据库作业

使用 `DBMS_SCHEDULER.CREATE_JOB` 创建作业。

- 使用 `CREDENTIAL_NAME` 参数设置默认身份证明。
- 使用 `DESTINATION_NAME` 参数设置目标。
- 通过将 `ENABLED` 参数设置为 `TRUE` 来启用作业。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 创建远程数据库作业

使用 `DBMS_SCHEDULER.CREATE_JOB` 过程创建各个远程作业。`CREATE_JOB` 过程包括以下适用于远程数据库作业的参数：

- `CREDENTIAL_NAME` 用于指定在作业中使用的默认身份证明。该身份证明必须存在。
- `DESTINATION_NAME` 用于指定作业的数据库目标或外部目标。

## 调度程序多目标作业：概览

- 该功能可用于指定多个要在其上执行作业的目标。
- 使用户能够从创建作业的数据库监视和控制这些作业。
- 多目标作业运行时，可将其视为一个作业集合，其中的各个作业是彼此近似相同的副本。
- 所有作业都将基于在作业开始日期中指定的时区运行，或将使用源数据库的时区。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 调度程序多目标作业：概览

多目标作业功能允许您指定多个要在其上执行作业的目标。您可以从创建作业的数据库监视和控制这些作业。其中包括以下功能：

- 指定必须在其上执行作业的数个数据库或计算机
- 将在多个目标上调度的同一作业改变为单个实体
- 停止或删除在一个或多个远程目标上运行的作业
- 查看作业实例在所有作业目标上的状态

请注意，在该功能的初始版本中，所有目标都基于在作业开始日期中指定的时区运行，或默认使用源数据库的时区。

## 调度程序多目标作业：术语

| 术语      | 定义                       |
|---------|--------------------------|
| 父作业     | 在源上创建的作业                 |
| 子作业     | 在不同目标上执行的副本              |
| 作业目标 ID | 系统生成的编号，用于唯一标识给定目标上的作业实例 |

ORACLE

版权所有 © 2010，Oracle。保留所有权利。

### 调度程序多目标作业：术语

“作业目标 ID”是系统生成的编号，用于唯一地标识给定目标上的作业实例。这是一个唯一标识元组（作业名、目标、身份证明）的编号。



## 调度程序多目标作业状态

| 父作业状态              |                                      |
|--------------------|--------------------------------------|
| <b>SUCCEEDED</b>   | 在上次调度的开始时间运行的所有子作业都成功完成              |
| <b>FAILED</b>      | 所有子作业都失败                             |
| <b>SOME FAILED</b> | 一部分失败，一部分成功                          |
| <b>INCOMPLETE</b>  | 父作业可以启动下一个调度运行时，仍有一个或多个子作业在处理之前的调度运行 |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 调度程序多目标作业状态

父作业状态将设置如下：

- **SUCCEEDED**，如果在上次调度的开始时间运行的所有子作业都成功完成
- **FAILED**，如果所有子作业都失败
- **SOME FAILED**，如果一部分失败，一部分成功
- **INCOMPLETE**，如果父作业可以启动下一个调度运行时，仍有一个或多个子作业在处理之前的调度运行

## 创建多目标作业

要创建多目标作业，请执行以下步骤：

1. 使用 `DBMS_SCHEDULER.CREATE_GROUP` 过程创建一个包含所有远程目标的目标组。
2. 使用 `DBMS_SCHEDULER.CREATE_JOB` 创建作业，并将目标设置为之前创建的目标组的名称。
3. 使用 `DBMS_SCHEDULER.ADD_GROUP_MEMBER` 添加其它目标。

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 创建多目标作业

可以按如下方式创建多目标作业：

1. 创建一个包含所有远程目标的目标组：

```
CREATE_GROUP('group_name', 'group_type', 'member, member');
```
2. 创建作业并设置目标：

```
CREATE_JOB(JOB_NAME=>'job_name', ...,
 CREDENTIAL_NAME=>'credential_name', DESTINATION_NAME=>
 'destination_name');
```

**注：**JOB\_TYPE 必须设置为 PLSQL\_BLOCK、STORED\_PROCEDURE 或 EXECUTABLE。

3. 添加其它目标：

```
ADD_GROUP_MEMBER('group_name', 'member');
```

## 在多目标作业中使用目标组

- 目标组由一组目标组成。
- 必须将目标组分配给作业的 `DESTINATION_NAME` 属性。
- 类型为 `EXECUTABLE` 的多目标作业要求将 `DESTINATION_NAME` 属性设置为组类型为 `EXTERNAL_DEST` 的目标组。
- 类型为 `PLSQL_BLOCK` 或 `STORED_PROCEDURE` 的多目标作业要求将 `DESTINATION_NAME` 属性设置为类型为 `DB_DEST` 的目标组。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 在多目标作业中使用目标组

创建作业时，可引用由一组目标组成的目标组。该作业将在组内的所有目标上运行。

要使某个作业在多个目标上运行，您必须创建一个数据库目标组或外部目标组，并将其分配给该作业的 `DESTINATION_NAME` 属性。

类型为 `EXECUTABLE` 的多目标作业必须将其 `DESTINATION_NAME` 属性设置为组类型为 `EXTERNAL_DEST` 的目标组。外部目标是调度程序代理在数据库中注册时自动创建的。`ALL_SCHEDULER_EXTERNAL_DESTS` 视图可列出所有已注册的外部目标。代理取消注册后，外部目标将被删除；当运行代理的主机不再可访问时，可使用 `DROP_AGENT_DESTINATION` 删除外部目标。

类型为 `PLSQL_BLOCK` 或 `STORED_PROCEDURE` 的多目标作业必须将其 `DESTINATION_NAME` 属性设置为类型为 `DB_DEST` 的目标组。数据库目标使用 `CREATE_DATABASE_DESTINATION` 过程创建的。`ALL_SCHEDULER_DB_DESTS` 视图可列出所有已创建的数据库目标。使用 `DROP_DATABASE_DESTINATION` 可将数据库目标删除。

# 使用 CREATE\_GROUP

| 参数         | 数据类型     | 默认值  |
|------------|----------|------|
| GROUP_NAME | VARCHAR2 | 无默认值 |
| GROUP_TYPE | VARCHAR2 | 无默认值 |
| MEMBER     | VARCHAR2 | NULL |
| COMMENTS   | VARCHAR2 | NULL |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

## 使用 DBMS\_SCHEDULER.CREATE\_GROUP

要创建远程数据库作业，首先必须使用 DBMS\_SCHEDULER.CREATE\_GROUP 过程创建一个目标组。该过程具有以下参数：

- GROUP\_NAME：组的名称
- GROUP\_TYPE：组类型，可为 WINDOW、EXTERNAL\_DEST 或 DB\_DEST
- MEMBER：组成员名称列表（可选），其格式为  
[[schema.]credential@][schema.]destination
- COMMENTS：注释

## 使用 ADD\_GROUP\_MEMBER

| 参数          | 数据类型     | 默认值  |
|-------------|----------|------|
| GROUP_NAME  | VARCHAR2 | 无默认值 |
| MEMBER_NAME | VARCHAR2 | 无默认值 |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 DBMS\_SCHEDULER.ADD\_GROUP\_MEMBER

要向组中添加成员，请使用 DBMS\_SCHEDULER.ADD\_GROUP\_MEMBER 过程。该过程具有以下参数：

- GROUP\_NAME：组的名称
- MEMBER\_NAME：要添加到组的一个或多个成员的名称

成员必须与组的类型相匹配。

## 使用 DROP\_GROUP

| 参数         | 数据类型     | 默认值  |
|------------|----------|------|
| GROUP_NAME | VARCHAR2 | 无默认值 |
| FORCE      | BOOLEAN  | 无默认值 |

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 DBMS\_SCHEDULER.DROP\_GROUP

要删除组，请使用 DBMS\_SCHEDULER.DROP\_GROUP 过程。该过程具有以下参数：

- GROUP\_NAME：组的名称
- FORCE：将来使用

类型为 DESTINATION 的组被删除后，其 DESTINATION 属性设置为该组的作业将被禁用。

某个组被删除时，该组的所有作业实例都将从 \*\_SCHEDULER\_JOB\_DESTS 视图中删除。

# 使用 REMOVE\_GROUP\_MEMBER

| 参数          | 数据类型     | 默认值  |
|-------------|----------|------|
| GROUP_NAME  | VARCHAR2 | 无默认值 |
| MEMBER_NAME | VARCHAR2 | 无默认值 |



版权所有 © 2010, Oracle。保留所有权利。

## 使用 DBMS\_SCHEDULER.REMOVE\_GROUP\_MEMBER

要从组中删除成员，请使用 DBMS\_SCHEDULER.REMOVE\_GROUP\_MEMBER 过程。该过程具有以下参数：

- GROUP\_NAME：组的名称
- MEMBER\_NAME：要从组中删除的一个或多个成员的名称

如果成员是 DESTINATION，则代表该目标的所有作业实例都将从 ALL\_SCHEDULER\_JOB\_DESTINATIONS 视图中删除。

## 对现有 DBMS\_SCHEDULER 过程的修改

- **CREATE\_JOB、SET\_ATTRIBUTE 和 GET\_ATTRIBUTE** 新增了两个参数：
  - **CREDENTIAL\_NAME**: 身份证明的全限定名称
  - **DESTINATION\_NAME**: 目标或目标组的全限定名称
- **STOP\_JOB**: 用于停止在特定目标上运行的单个子作业，需要提供该子作业的 **JOB\_DEST\_ID**。
- **ENABLE 和 DISABLE**: 用于启用/禁用外部目标、数据库目标和组。
- **PUT\_FILE 和 GET\_FILE** 的 **SOURCE\_HOST** 和 **DESTINATION\_HOST** 参数必须指定有效的外部目标名称。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 对现有 DBMS\_SCHEDULER 过程的修改

所有版本的 **CREATE\_JOB** 过程都在参数列表末尾新增了以下两个参数：

- **CREDENTIAL\_NAME**: 身份证明的全限定名称
- **DESTINATION\_NAME**: 目标或目标组的全限定名称

所有作业和链步骤都新增了这两个属性，可分别使用 **SET\_ATTRIBUTE** 和 **GET\_ATTRIBUTE** 对其进行设置和检索。

- **DESTINATION\_NAME**: 用于为特定作业或链步骤设置目标。指定单个给定目标或某种类型的组的名称。
- **CREDENTIAL\_NAME**: 用于为该作业或作业步骤设置身份证明。可以为单个目标设置身份证明，在目标属性为目标组名称时，可为所有目标设置默认身份证明。

禁用某个外部目标时，所有通过该代理的数据库目标也将自动禁用。禁用某个目标时，所有指向该目标的作业实例也将禁用。

**STOP\_JOB** 可用于停止在特定目标上运行的单个子作业，需要提供该子作业的 **JOB\_DEST\_ID** 而不是作业名。



## 多目标作业：新增的字典视图

可查询以下视图获取详细信息：

- **\*\_SCHEDULER\_DESTS**：可在其上调度远程作业的所有目标
- **\*\_SCHEDULER\_EXTERNAL\_DESTS**：所有已在数据库中注册的代理
- **\*\_SCHEDULER\_DB\_DESTS**：用户可在其上调度远程数据库作业的所有数据库
- **\*\_SCHEDULER\_GROUPS**：组信息
- **\*\_SCHEDULER\_GROUP\_MEMBERS**：成员信息
- **\*\_SCHEDULER\_JOB\_DESTS**：远程数据库上某个作业的状态

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 多目标作业：新增的字典视图

以下新视图可显示有关多目标作业的信息：

- **\*\_SCHEDULER\_DESTS**：显示可在其上调度远程作业的所有目标。该视图包含外部目标（对于远程外部作业）和数据库目标（对于远程数据库作业）
- **\*\_SCHEDULER\_EXTERNAL\_DESTS**：显示所有已在数据库中注册的并可用作远程外部作业目标的代理
- **\*\_SCHEDULER\_DB\_DESTS**：显示用户可在其上调度远程数据库作业的所有数据库
- **\*\_SCHEDULER\_GROUPS**：显示用户方案中的组或数据库中的所有组
- **\*\_SCHEDULER\_GROUP\_MEMBERS**：显示用户方案中的组成员或数据库中的所有组成员
- **\*\_SCHEDULER\_JOB\_DESTS**：显示远程数据库上某个作业的状态

## 多目标作业：现有字典视图中新增的列

以下视图中新增了一些列，以便为多目标作业提供附加信息：

- **\*\_SCHEDULER\_JOBS**
- **\*\_SCHEDULER\_RUNNING\_JOBS**
- **\*\_SCHEDULER\_JOB\_RUN\_DETAILS**
- **\*\_SCHEDULER\_JOB\_LOG**

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 多目标作业：现有字典视图中新增的列

以下字典视图新增了用于提供多目标作业相关信息的列：

- **\*\_SCHEDULER\_JOBS:**
  - **CREDENTIAL\_OWNER:** 身份证明所有者的名称
  - **CREDENTIAL\_NAME:** 源数据库的名称
  - **DESTINATION\_OWNER:** 目标所有者的名称
  - **NUMBER\_OF\_DESTINATIONS:** 与该作业相关的目标数量
- **\*\_SCHEDULER\_RUNNING\_JOBS**、**\*\_SCHEDULER\_JOB\_RUN\_DETAILS** 和 **\*\_SCHEDULER\_JOB\_LOG:**
  - **CREDENTIAL\_OWNER:** 身份证明所有者的名称
  - **CREDENTIAL\_NAME:** 身份证明的名称
  - **DESTINATION\_OWNER:** 目标所有者的名称
  - **DESTINATION:** 目标的名称

## 小结

在本课中，您应该已经学会：

- 使用轻量作业
- 使用调度程序电子邮件通知
- 创建文件监视器
- 创建远程数据库作业
- 创建多个目标作业

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

These eKit materials are to be used ONLY by you for the express purpose SELF STUDY. SHARING THE FILE IS STRICTLY PROHIBITED.

# Oracle Secure Backup 云模块

ORACLE®

版权所有 © 2010, Oracle。保留所有权利。

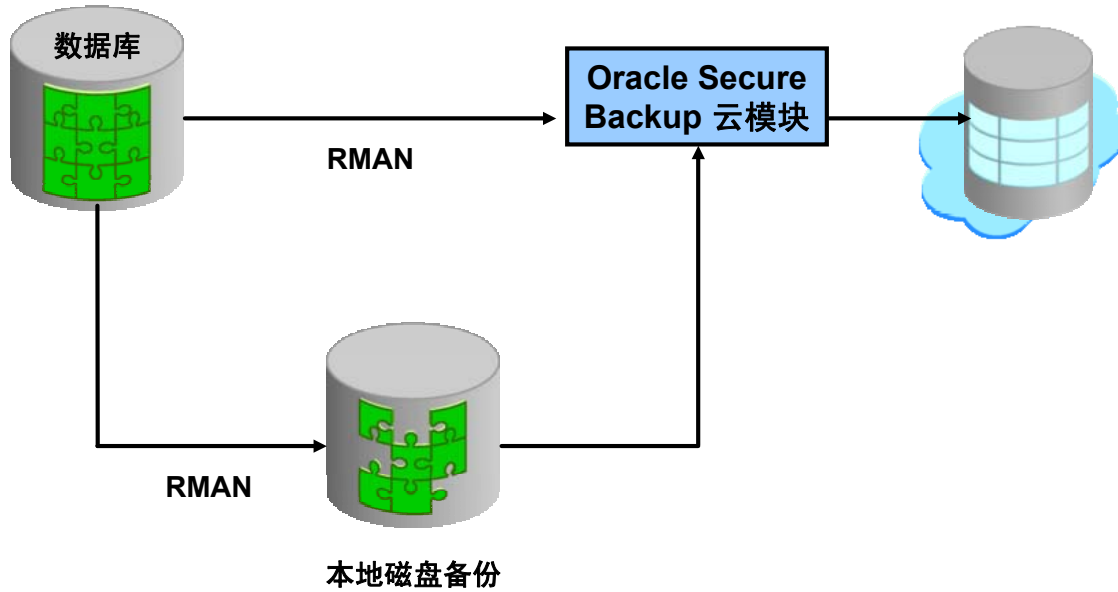
## 课程目标

学完本课后，您应能描述如何使用 **Oracle Secure Backup** 云模块

ORACLE®

版权所有 © 2010, Oracle。保留所有权利。

## 在云中备份 Oracle 数据库



ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 在云中备份 Oracle 数据库

使用 Oracle Secure Backup 云模块，可以创建在云中存储的异地备份。

Oracle Secure Backup 云模块使用 Oracle Recovery Manager (RMAN) SBT 接口实施。

可以使用 Oracle Secure Backup 云模块将备份以流的方式直接存储在云中。

也可以用该模块将本地磁盘备份发送至存储云，将其作为异地存储。对于本地数据库，Oracle 建议首先进行基于本地磁盘的备份，而将云作为可按需访问的异地备份位置。

## Oracle Secure Backup 云模块简介

- Oracle Secure Backup 云模块可用于备份所有支持的 Oracle DB 版本。



- 可以用现有界面创建备份。

ORACLE Enterprise Manager 11g  
Database Control

```
% rman @bu_command_file.txt
```

- 包括特殊用途许可：
  - 要使用 RMAN 备份加密，需要有 Oracle 高级安全选件
  - 要使用快速备份压缩，需要有 Oracle 高级压缩选件

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### Oracle Secure Backup 云模块简介

可以使用 Oracle Secure Backup 云模块备份所有支持的 Oracle DB 版本。

使用 Oracle Secure Backup 云模块时，可以继续使用 Enterprise Manager 和脚本创建备份。

Oracle Secure Backup 云模块包括特殊用途许可：

- 要使用 RMAN 备份加密，需要有 Oracle 高级安全选件
- 要使用快速备份压缩功能，需要有 Oracle 高级压缩选件

注：在 Oracle Database 9i 发行版 2 中，未提供加密和压缩功能。



## 备份至云的优点

与基于磁带的传统异地备份相比，备份至云具有以下优点：

- **持续的可访问性：**
  - 与本地磁盘备份类似，存储在云中的备份始终可以访问
- **更高的可靠性：**
  - 存储云基于磁盘，本质上比磁带更可靠
  - 云供应商通常会维护多个冗余数据副本，以实现可用性和可伸缩性
- **节约成本，降低或消除了以下各项：**
  - 前期资本支出
  - 异地存储成本

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 云备份优于基于磁带的异地备份的方面

与基于磁带的传统异地备份相比，使用 OSB 云模块备份至云具有以下优点：

- **持续的可访问性：**存储在云中的备份始终可访问，与本地磁盘备份相似。您无需求助于他人来还原数据，在执行还原之前也无需运送或加载磁带。您可以使用标准工具启动还原操作，就像异地备份存储在本机一样。这有助于加快文件还原速度并可减少停机时间。
- **更高的可靠性：**存储云是基于磁盘的，本质上比磁带更可靠。此外，云供应商通常会维护多个冗余数据副本，以实现可用性和可伸缩性。
- **节约成本：**使用云存储可降低或消除前期资本支出以及异地存储成本。

## 什么是 Amazon Simple Storage Service (S3)

- **Amazon Simple Storage Service (S3)** 用于访问高度可伸缩的、可靠、快速、价格较低的 **Amazon** 数据存储基础结构，**Amazon** 使用该结构运行自己的全球 **Web** 站点网络。
- 使用 **Amazon S3** 可存储和检索来自 **Web** 上任意位置的任意数量的数据。
- **Amazon S3** 旨在最大化规模效益，并使订户从中受益。

**ORACLE**

版权所有 © 2010, Oracle。保留所有权利。

### 什么是 Amazon Simple Storage Service (S3)

自 2006 年以来，Amazon Web Services (AWS) 以云的形式为各种规模的公司提供基础结构性的 Web 服务平台。

Amazon Simple Storage Service (S3) 提供了一个高度可伸缩的、可靠、价格较低的数据存储基础结构，可用来存储来自 Web 上任意位置的任意数量的数据。

有关详细信息，请参阅 Amazon Web Services 网站：<http://aws.amazon.com>。

## 安装 Oracle Secure Backup 云模块

1. 在 Amazon Web Services 网站上注册使用 Amazon S3 Web 服务: <http://aws.amazon.com>
2. 注册 Oracle.com 或 Oracle 技术网 (OTN) 帐户: <http://otn.oracle.com>
3. 从 OTN 下载 Oracle Secure Backup 云模块安装程序: <http://www.oracle.com/technology/software/tech/cloud/index.html>
4. 运行该安装程序, 安装 Oracle Secure Backup 云模块并配置云备份

ORACLE

版权所有 © 2010, Oracle。保留所有权利。

### 使用 Oracle Secure Backup 云模块

要使用 Oracle Secure Backup 云模块, 请执行以下步骤:

1. 注册使用 Amazon S3 网络服务。您将获得 “Access Key ID (访问密钥 ID)” 和 “Secret Access Key (保密访问密钥)”。
2. 您需要有 Oracle.com 或 Oracle 技术网 (OTN) 帐户, 才能安装 Oracle Secure Backup 云模块。如果没有这些帐户, 可以在 OTN 网站上注册。
3. 从 OTN 上下载 Oracle Secure Backup 云模块安装程序 (osbws\_install.jar)。如果要备份的数据库在 Amazon Elastic Compute Cloud (EC2) 上, 您可以在 /home/oracle/scripts/osbws 目录中找到 osbws\_install.jar 文件。
4. 调用安装程序安装 Oracle Secure Backup 云模块并配置云备份。

有关 Oracle Secure Backup 云模块安装和配置的详细信息, 请参阅 “Oracle Database 11g R2: Availability - RMAN eStudy”。

## 小结

在本课中，您应该已经学会如何使用 **Oracle Secure Backup** 云模块。

ORACLE

版权所有 © 2010, Oracle。保留所有权利。