

## Oracle Database 11g: 数据库 管理 — 课堂练习 II

学生指南第 2 册

D50079CN20

版本 2.0

2011 年 3 月

D71768

**ORACLE®**

## 作者

Maria Billings

## 技术撰稿人和审稿人

Christian Bauwens

Yanti Chang

Timothy Chien

Joe Fong

Andy Fortunak

Gerlinde Frenzen

Mark Fuller

Peter Fusek

Joel Goodman

Vimala Jacob

Dominique Jeunot

Pete Jones

Fukue Kawabe

Donna Keesling

Sean Kim

Achiel Langers

Gwen Lazenby

Jerry Lee

Deidre Matishak

Bill Millar

Lakshmi Naraparreddi

Ira Singer

Ranbir Singh

James Spiller

Matt Taylor

Branislav Valny

Jean-Francois Verrier

## 编辑

Nita Pavitrnan

Raj Kumar

## 制图员

Satish Bettogowda

## 出版商

Jayanthi Keshavamurthy

版权所有 © 2011, 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以供在 Oracle 培训课程中单独使用。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。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，请下列通知中的规定适用：

### 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/或其子公司的注册商标。其它名称可能是其各自拥有者的商标。

## 目录

### I 简介

课程目标 I-2

建议的日程表 I-3

Oracle Database 11g: “g” 代表 Grid (网格) I-4

用于单实例的 Oracle Grid Infrastructure I-6

课程示例: HR 示例方案 I-8

### 1 Oracle DB 的核心概念和工具

课程目标 1-2

列出 Oracle DB Server 的核心组件的名称 1-3

Oracle DB Server 体系结构概览 1-4

实例数据库配置 1-6

列出 Oracle DB 的内存结构的名称 1-7

Oracle DB 内存结构 1-8

进程体系结构 1-10

进程结构 1-11

添加进程名称 1-13

进程启动顺序 1-14

数据库存储体系结构 1-15

逻辑和物理数据库结构 1-17

自动存储管理 1-19

ASM 存储组件 1-20

ASM 实例 1-21

DBA 配置工具 1-23

管理框架和相关 DBA 工具 1-24

使用 Oracle Restart 简化数据库管理 1-25

测验 1-27

小结 1-28

### 2 进行配置以实现可恢复性

课程目标 2-2

备份和恢复功能的用途 2-3

典型的备份和恢复任务	2-4
Oracle 备份和恢复解决方案	2-5
Oracle 备份解决方案	2-6
术语回顾	2-7
您已了解的内容：Oracle 建议的备份	2-9
使用 Recovery Manager	2-10
RMAN 命令的类型	2-11
作业命令：示例	2-12
配置数据库进行备份和恢复操作	2-13
ARCHIVELOG 模式	2-14
配置 ARCHIVELOG 模式	2-15
配置归档日志目标位置	2-16
确保归档日志成功	2-18
指定保留策略	2-20
恢复窗口保留策略：示例	2-22
使用快速恢复区	2-23
定义快速恢复区	2-25
快速恢复区空间管理	2-26
快速恢复区空间使用情况	2-28
自动为您完成的任务	2-30
监视 FRA	2-31
使用快速恢复区的优点	2-32
测验	2-33
小结	2-35
练习 2 概览：进行配置以实现可恢复性	2-36

### **3 使用 RMAN 恢复目录**

课程目标	3-2
RMAN 资料档案库数据存储：比较选项	3-3
在恢复目录中存储信息	3-4
使用恢复目录的原因	3-5
创建恢复目录：三个步骤	3-6
配置恢复目录数据库	3-7
创建恢复目录所有者	3-8
创建恢复目录	3-9
管理恢复目录中的目标数据库记录	3-10

在恢复目录中注册数据库	3-11
使用 Oracle Enterprise Manager 注册数据库	3-12
从恢复目录中注销目标数据库	3-13
将其它备份文件列入目录	3-14
恢复目录重新同步：概念	3-16
手动重新同步恢复目录	3-17
使用 RMAN 存储脚本	3-18
执行 RMAN 存储脚本	3-19
维护 RMAN 存储脚本	3-20
备份恢复目录	3-21
重新创建无法恢复的恢复目录	3-22
导出和导入恢复目录	3-23
升级和删除恢复目录	3-24
IMPORT CATALOG 命令	3-25
创建和使用虚拟专用目录	3-27
使用 RMAN 虚拟专用目录	3-28
恢复目录概要	3-30
测验	3-32
小结	3-34
练习 3 概览：使用 RMAN 恢复目录	3-35
<b>4 配置备份设置</b>	
课程目标	4-2
配置 RMAN 的永久性设置	4-3
查看永久性设置	4-4
控制文件自动备份	4-5
管理永久性设置	4-7
使用介质管理器	4-8
指定备份目标	4-10
配置和分配通道	4-11
创建双向备份集	4-12
使用 CONFIGURE BACKUP COPIES 创建双向备份集	4-13
备份优化	4-14
通过压缩未使用的块节省备份空间	4-16
压缩备份	4-17
使用 RMAN 备份压缩	4-18

加密备份 4-19  
 测验 4-20  
 小结 4-22  
 练习 4 概览：配置备份规范 4-23

## 5 使用 RMAN 创建备份

课程目标 5-2  
 创建备份集 5-3  
 创建映像副本 5-4  
 创建整体数据库备份 5-6  
 RMAN 备份类型 5-8  
 快速增量备份 5-10  
 启用快速增量备份 5-11  
 监视块更改跟踪 5-12  
 执行代理副本 5-13  
 使用 BACKUP COPIES 创建双向备份集 5-14  
 创建备份集的备份 5-15  
 备份只读表空间 5-16  
 为超大型文件配置备份和还原 5-17  
 创建 RMAN 多部分备份 5-18  
 归档备份：概念 5-19  
 使用 EM 创建归档备份 5-21  
 使用 RMAN 创建归档备份 5-22  
 管理归档数据库备份 5-23  
 备份恢复文件 5-24  
 管理备份：报告 5-25  
 管理备份：动态性能视图 5-27  
 使用 Oracle Enterprise Manager 查看备份报告 5-28  
 管理备份：交叉检查和删除 5-29  
 测验 5-30  
 小结 5-32  
 练习 5 概览：创建备份 5-33

## 6 还原和恢复任务

课程目标 6-2  
 还原和恢复 6-3

文件丢失的原因	6-4
关键性与非关键性	6-5
自动恢复临时文件	6-6
日志组状态：综述	6-7
在丢失了重做日志组后进行恢复	6-8
清除日志文件	6-9
丢失了索引表空间后进行恢复	6-10
重新创建索引	6-11
面向数据库管理员的验证方法	6-13
重新创建口令验证文件	6-14
比较完全恢复和不完全恢复	6-16
完全恢复过程	6-17
时间点恢复过程	6-18
恢复只读表空间	6-20
恢复 NOLOGGING 数据库对象	6-21
在丢失了所有控制文件副本后进行恢复：概览	6-22
将控制文件恢复到默认位置	6-23
测验	6-24
小结	6-26

## 7 使用 RMAN 执行恢复

课程目标	7-2
使用 RMAN RESTORE 和 RECOVER 命令	7-3
执行完全恢复：在 ARCHIVELOG 模式下丢失了非关键数据文件	7-5
执行完全恢复：在 ARCHIVELOG 模式下丢失了系统关键数据文件	7-6
恢复映像副本	7-7
恢复映像副本：示例	7-8
执行到映像副本的快速切换	7-9
使用 SET NEWNAME 切换文件	7-10
SET NEWNAME 的替代变量	7-11
在 NOARCHIVELOG 模式下执行数据库还原和恢复	7-12
使用还原点	7-13
执行时间点恢复	7-14
使用备份控制文件执行恢复	7-16
恢复丢失的服务器参数文件	7-17
从控制文件自动备份还原服务器参数文件	7-18

从自动备份还原控制文件	7-19
使用增量备份恢复处于 NOARCHIVELOG 模式的数据库	7-21
在新主机上还原和恢复数据库	7-22
准备将数据库还原到新主机	7-23
将数据库还原到新主机	7-24
执行灾难恢复	7-28
测验	7-30
小结	7-32
练习 7 概览：使用 RMAN 执行恢复	7-33

## 8 监视和优化 RMAN

课程目标	8-2
并行备份集	8-3
监视 RMAN 会话	8-5
监视 RMAN 作业进度	8-7
解释 RMAN 消息输出	8-9
使用 DEBUG 选项	8-10
解释 RMAN 错误堆栈	8-11
优化 RMAN	8-12
RMAN 多路复用	8-13
分配磁盘缓冲区：示例	8-14
分配磁带缓冲区	8-15
比较同步 I/O 和异步 I/O	8-16
监视 RMAN 作业的性能	8-18
异步 I/O 瓶颈	8-19
同步 I/O 瓶颈	8-20
通道优化	8-21
优化 BACKUP 命令	8-23
优化 RMAN 备份性能	8-24
设置 LARGE_POOL_SIZE	8-25
优化 RMAN 磁带流性能瓶颈问题	8-26
测验	8-28
小结	8-30
练习 8 概览：监视和优化 RMAN	8-31



## 9 诊断数据库

- 课程目标 9-2
- 数据恢复指导 9-3
- 数据故障 9-6
- 数据故障：示例 9-7
- 数据恢复指导 RMAN 命令行界面 9-8
- 列出数据故障 9-9
- 修复建议 9-11
- 执行修复 9-12
- 分类（和关闭）故障 9-13
- 数据恢复指导视图 9-14
- 最佳做法：主动检查 9-15
- 什么是块损坏 9-16
- 块损坏故障现象：ORA-01578 9-17
- 如何处理损坏 9-18
- 设置检测损坏的参数 9-19
- 块介质恢复 9-21
- 块介质恢复的先决条件 9-22
- RECOVER...BLOCK 命令 9-23
- 自动诊断工作流 9-24
- 自动诊断资料档案库 9-25
- ADR 命令行工具 (ADRCI) 9-26
- V\$DIAG\_INFO 视图 9-27
- 诊断跟踪的位置 9-28
- 健康装款监视器：概览 9-29
- 手动运行健康检查：PL/SQL 示例 9-30
- 使用 ADRCI 实用程序查看 HM 报表 9-31
- 测验 9-32
- 小结 9-36
- 练习 9 概览：诊断数据库 9-37

## 10 使用闪回技术 I

- 课程目标 10-2
- 闪回技术 10-3
- 事务处理和还原 10-4
- 保证还原保留时间 10-5

准备数据库以进行闪回	10-6
使用闪回技术查询数据	10-8
闪回查询	10-9
闪回查询：示例	10-10
闪回版本查询	10-11
闪回版本查询：注意事项	10-12
测验	10-13
闪回表概览	10-15
闪回表	10-16
对表启用行移动	10-17
执行闪回表	10-18
闪回表：注意事项	10-20
测验	10-21
闪回事务处理查询	10-22
使用 Oracle Enterprise Manager 执行闪回事务处理查询	10-23
闪回事务处理查询：注意事项	10-24
闪回事务处理	10-25
先决条件	10-26
闪回事务处理	10-27
可能的工作流	10-28
闪回事务处理向导	10-29
选择其它回退选项	10-30
最后步骤（不使用 EM）	10-32
测验	10-33
小结	10-34
练习 10 概览：执行闪回事务处理回退	10-35

## 11 使用闪回技术 II

课程目标	11-2
Oracle Total Recall 概览	11-3
设置过程	11-4
Total Recall 的工作原理	11-5
Oracle Total Recall 场景	11-6
透明的方案演化	11-9
完整方案演化	11-10
限制	11-11

准则	11-12
查看闪回数据归档	11-13
测验	11-14
闪回删除和回收站	11-16
回收站	11-17
从回收站还原表	11-19
回收站：自动回收空间	11-20
回收站：手动回收空间	11-21
不使用回收站	11-22
查询回收站	11-23
测验	11-24
小结	11-25
练习 11 概览：使用闪回技术	11-26

## 12 执行闪回数据库

课程目标	12-2
闪回数据库	12-3
闪回数据库体系结构	12-4
配置闪回数据库	12-5
您需要执行的操作	12-6
闪回数据库：示例	12-7
闪回数据库注意事项	12-8
监视闪回数据库	12-9
使用 EM 监视闪回数据库	12-11
保证还原点	12-12
闪回数据库和保证还原点	12-13
测验	12-14
小结	12-16
练习 12 概览：使用闪回数据库	12-17

## 13 管理内存

课程目标	13-2
内存管理：概览	13-3
复查 Oracle DB 内存结构	13-4
缓冲区高速缓存	13-6
使用多个缓冲池	13-8

共享池	13-10
大型池	13-11
Java 池和流池	13-12
重做日志缓冲区	13-13
自动内存管理：概览	13-14
Oracle DB 内存参数	13-15
监视自动内存管理	13-16
有效使用内存：准则	13-18
库高速缓存的内存优化准则	13-20
自动共享内存管理：概览	13-22
ASMM 的工作原理	13-23
启用自动共享内存管理功能	13-24
禁用 ASMM	13-26
程序全局区 (PGA)	13-27
使用 V\$PARAMETER 视图	13-29
测验	13-30
小结	13-31
练习 13 概览：使用 AMM 纠正内存分配问题	13-32

## 14 管理数据库性能

课程目标	14-2
优化活动	14-3
性能规划	14-4
实例优化	14-6
性能优化方法	14-7
性能监视	14-8
性能优化数据	14-9
优化统计信息收集	14-10
统计信息首选项：概览	14-12
使用统计信息首选项	14-13
使用 Oracle Enterprise Manager 设置全局首选项	14-14
Oracle 等待事件	14-15
实例统计信息	14-16
监视会话性能	14-18
显示与会话有关的统计信息	14-19
显示与服务有关的统计信息	14-20

故障排除和优化视图	14-21
字典视图	14-22
自动工作量资料档案库	14-24
使用自动工作量资料档案库视图	14-26
Real Application Testing 概览：数据库重放	14-27
整体情况	14-28
测验	14-29
小结	14-30
练习 14 概览：监视实例性能	14-31

## 15 通过 SQL 优化管理性能

课程目标	15-2
SQL 优化	15-3
SQL 指导	15-4
自动 SQL 优化结果	15-5
实施自动优化建议	15-6
SQL 优化指导：概览	15-7
使用 SQL 优化指导	15-8
SQL 优化指导选项	15-9
SQL 优化指导建议	15-10
使用 SQL 优化指导：示例	15-11
重复的 SQL	15-13
SQL 访问指导：概览	15-14
典型的 SQL 访问指导会话	15-15
工作量来源	15-16
建议选项	15-17
复查建议	15-20
SQL 性能分析程序：概览	15-21
SQL 性能分析程序：使用情形	15-22
使用 SQL 性能分析程序	15-24
测验	15-25
小结	15-29
练习 15 概览：通过 SQL 优化管理性能	15-30

## 16 管理资源

课程目标 16-2

数据库资源管理器：概览 16-3

数据库资源管理器：概念 16-4

使用资源管理器的原因 16-5

默认维护资源管理器计划 16-7

示例：DEFAULT\_PLAN 16-8

潜在工作流 16-9

指定资源计划指令 16-11

资源计划的资源分配方法 16-12

EMPHASIS 和 RATIO 的比较 16-13

活动会话池机制 16-15

设置活动会话池 16-16

指定阈值 16-18

设置空闲超时 16-19

限制数据库级别的 CPU 占用率 16-20

限制服务器级别的 CPU 占用率：实例限制 16-22

实例限制示例 16-23

监视实例限制 16-24

资源使用者组映射 16-25

激活资源计划 16-27

数据库资源管理器信息 16-28

监视资源管理器 16-29

测验 16-32

小结 16-33

练习 16 概览：使用资源管理器 16-34

## 17 使用调度程序自动执行任务

课程目标 17-2

简化管理任务 17-3

核心组件 17-4

基本工作流 17-5

测验 17-7

持久轻量作业 17-8

使用基于时间的或基于事件的调度 17-9

创建基于时间的作业 17-10

创建基于事件的调度	17-12
使用 Oracle Enterprise Manager 创建基于事件的调度	17-13
创建基于事件的作业	17-14
基于事件的调度	17-15
创建复杂调度	17-17
测验	17-18
使用电子邮件通知	17-19
添加和删除电子邮件通知	17-20
创建作业链	17-22
链的示例	17-24
高级调度程序概念	17-25
作业类	17-26
窗口	17-28
区分窗口内作业的优先级	17-29
创建作业数组	17-30
测验	17-32
创建文件监视器和基于事件的作业	17-33
从远程系统启用文件到达事件	17-35
调度远程数据库作业	17-36
创建远程数据库作业	17-37
调度多个目标作业	17-38
查看调度程序元数据	17-39
测验	17-41
小结	17-42
练习 17 概览：使用调度程序自动执行任务	17-43

## 18 管理空间

课程目标	18-2
空间管理：概览	18-3
块空间管理	18-4
行链接和行迁移	18-5
测验	18-7
段内的空闲空间管理	18-8
段的类型	18-9
分配区	18-10
分配空间	18-11

创建没有段的表	18-12
控制延迟创建段	18-13
限制和例外	18-14
其它自动功能	18-15
测验	18-16
表压缩：概览	18-17
适用于直接路径插入操作的压缩	18-18
适用于 DML 操作的 OLTP 压缩	18-20
指定表压缩	18-21
使用压缩指导	18-22
使用 DBMS_COMPRESSION 程序包	18-23
压缩表数据	18-24
预先表空间监视	18-25
阈值和解决空间问题	18-26
监视表空间的空间使用量	18-27
收缩段	18-28
收缩操作的结果	18-29
回收 ASSM 段内的空间	18-30
段指导：概览	18-31
段指导	18-32
实施建议	18-33
自动段指导	18-34
使用 EM 手动收缩段	18-35
使用 SQL 收缩段	18-36
管理可恢复的空间分配	18-37
使用可恢复的空间分配	18-38
恢复挂起的语句	18-40
哪些操作是可恢复的	18-42
测验	18-43
小结	18-44
练习 18 概览：管理存储	18-45

## 19 管理数据库的空间

课程目标	19-2
数据库存储	19-3
支持 4 KB 扇区磁盘	19-4



使用 4 KB 扇区磁盘	19-5
指定磁盘扇区大小	19-6
测验	19-7
移动表空间	19-10
概念：最低兼容级别	19-11
最低兼容级别	19-12
可移动表空间过程	19-13
确定平台的 Endian 格式	19-14
使用 RMAN 的 CONVERT 命令	19-16
使用 Oracle Enterprise Manager 实施可移动表空间	19-17
移动数据库	19-20
数据库移动过程：源系统转换	19-21
数据库移动过程：目标系统转换	19-22
数据库移动：注意事项	19-23
测验	19-24
小结	19-25
练习 19 概览：管理数据库的空间	19-26

## 20 复制数据库

课程目标	20-2
使用副本数据库	20-3
选择数据库复制方法	20-4
复制活动数据库	20-5
通过目标连接复制数据库	20-6
在具有恢复目录但没有目标连接的情况下复制数据库	20-7
在没有恢复目录或目标连接的情况下复制数据库	20-8
创建基于备份的副本数据库	20-9
创建辅助实例的初始化参数文件	20-10
为目标指定新名称	20-11
使用 SET NEWNAME 子句	20-12
SET NEWNAME 的替代变量	20-13
指定用于文件命名的参数	20-14
在 NOMOUNT 模式下启动实例	20-16
确保备份和归档重做日志文件可用	20-17
分配辅助通道	20-18
了解 RMAN 复制操作	20-19

- 指定 DUPLICATE 命令的选项 20-21
- 使用其它 DUPLICATE 命令选项 20-22
- 使用 EM 克隆数据库 20-23
- 测验 20-24
- 小结 20-25
- 练习 20 概览：复制数据库 20-26

## 附录 A：练习与解答

### 附录 B：执行表空间时间点恢复

- 课程目标 B-2
- 表空间时间点恢复 (TSPITR)：概念 B-3
- 表空间时间点恢复 (TSPITR)：术语 B-4
- 表空间时间点恢复：体系结构 B-5
- 何时使用 TSPITR B-7
- 为 TSPITR 做准备 B-8
- 确定正确的目标时间 B-9
- 确定恢复集的表空间 B-10
- 确定跨越恢复集边界的关系 B-11
- 确定将丢失的对象 B-12
- 执行基本 RMAN TSPITR B-13
- 执行完全自动执行的 TSPITR B-14
- 使用映像副本提高 TSPITR 性能 B-15
- 使用 Oracle Enterprise Manager 执行 TSPITR B-16
- RMAN TSPITR 处理 B-17
- 使用 RMAN 管理的辅助实例执行 RMAN TSPITR B-19
- 使用您自己的辅助实例执行 RMAN TSPITR B-20
- RMAN TSPITR 故障排除 B-21
- 小结 B-22

### 附录 C：执行用户管理的备份和恢复

- 课程目标 C-2
- 备份和恢复的使用类型 C-3
- 执行用户管理的数据库备份 C-4
- 需要备份模式的场合 C-5
- 确定要手动备份的文件 C-6

手动备份 NOARCHIVELOG 数据库 C-7  
 手动备份 ARCHIVELOG 数据库 C-8  
 备份控制文件 C-9  
 执行用户管理的数据库完全恢复：概览 C-10  
 对关闭的数据库执行完全恢复：概览 C-11  
 确定与恢复相关的文件 C-12  
 还原与恢复相关的文件 C-13  
 应用重做数据 C-15  
 对打开的数据库执行完全恢复 C-16  
 执行用户管理的不完全恢复：概览 C-17  
 选择不完全恢复方法 C-18  
 执行用户管理的不完全恢复 C-19  
 执行用户管理的不完全恢复：步骤 C-21  
 用户管理的基于时间的恢复：示例 C-22  
 用户管理的基于取消的恢复：示例 C-24  
 小结 C-26

## 附录 D：管理 ASM 实例

课程目标 D-2  
 ASM 对于管理员的好处 D-3  
 ASM 实例 D-4  
 ASM 组件：ASM 实例 — 主进程 D-6  
 ASM 实例初始化参数 D-7  
 数据库实例与 ASM 之间的交互 D-9  
 ASM 实例：动态性能视图 D-10  
 ASM 系统权限 D-11  
 使用 Oracle Enterprise Manager 管理 ASM 用户 D-12  
 使用 SQL\*Plus 启动和停止 ASM 实例 D-13  
 使用 `srvctl` 启动和停止 ASM 实例 D-15  
 使用 `asmcmd` 启动和停止 ASM 实例 D-16  
 磁盘组概览 D-17  
 ASM 磁盘 D-18  
 分配单元 D-19  
 ASM 文件 D-20  
 区映射 D-21  
 条带化粒度 D-22

细粒度条带化	D-23
ASM 故障组	D-25
条带化和镜像示例	D-26
故障示例	D-27
管理磁盘组	D-28
使用 SQL*Plus 创建和删除磁盘组	D-29
将磁盘添加至磁盘组	D-30
其它 ALTER 命令	D-31
使用 Oracle Enterprise Manager 的 ASM 管理	D-32
ASM 磁盘组兼容性	D-33
ASM 磁盘组属性	D-35
使用 Oracle Enterprise Manager 编辑磁盘组属性	D-36
检索 ASM 元数据	D-37
ASM 快速镜像再同步概览	D-38
小结	D-39

# 11

## 使用闪回技术 II

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

华周 (286991486@qq.com) has a non-transferable license to use this Student Guide.

## 课程目标

学完本课后，应能完成以下工作：

- 描述和使用 Oracle Total Recall
  - 创建和启用闪回数据归档 (FDA)
  - 管理 FDA
  - 查看元数据
- 描述和使用闪回回收站
  - 从回收站还原删除的表
  - 管理回收站的空间使用量
  - 查询回收站

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

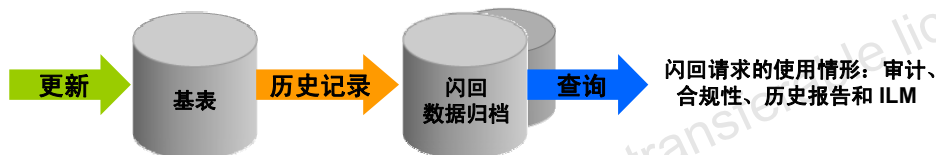
## Oracle Total Recall 概览

自动跟踪数据库历史更改：

- 在您指定的保留期内在表级别启用。
- 所有后续更改都是透明存储的并能够防止篡改。
- 早于保留期的记录会被自动删除。
- 使用闪回技术检索历史记录。

```
SELECT ... AS OF TIMESTAMP...
```

```
SELECT ... VERSIONS BETWEEN TIMESTAMP and TIMESTAMP...
```



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### Oracle Total Recall 概览

Oracle Database 11g 中的 Oracle Total Recall（也称为闪回数据归档）选项提供了一种用于跟踪生产数据库更改的机制，这种机制是一种安全、高效、易于使用 and 应用程序透明的机制。

使用 Oracle Total Recall 技术，您可以自动跟踪和存储启用了闪回数据归档的表中的数据。这可确保闪回查询能够获得对数据库对象版本的 SQL 级别访问权限，而不会出现快照太旧的错误。

利用闪回数据归档功能，可以在“跟踪的”表的整个生命期内跟踪并存储对该表的所有事务处理更改。不再需要将此智能功能内置到应用程序中。可将 Oracle Total Recall 用于合规性报告、审计报表、数据分析和决策支持系统。闪回数据归档后台进程随数据库一起启动。

使用情形示例：

- 审计支持：查找自去年以来重复的保险索赔。
- 合规性支持：监视平静期的股票交易情况。
- 信息生命周期管理 (ILM)：确保对病历的永久访问权。
- 保留策略执行：自动清除五年以前的记录。
- 历史报告：检索客户的信用及付款历史记录。
- 错误恢复：还原被错误地删除或更新的记录。

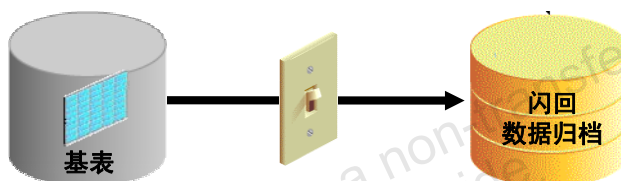
## 设置过程

1. 创建用于保存 FDA 的新表空间。
2. 使用 FLASHBACK ARCHIVE ADMINISTER 系统权限：创建闪回数据归档，将它分配到表空间，并指定其保留期。

```
CREATE FLASHBACK ARCHIVE fda1
TABLESPACE fda_tbs1 QUOTA 10M RETENTION 1 YEAR;
```

3. 使用 FLASHBACK ARCHIVE 对象权限：变更基表以启用归档并将其分配到一个闪回归档。

```
ALTER TABLE HR.EMPLOYEES FLASHBACK ARCHIVE fda1;
```



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

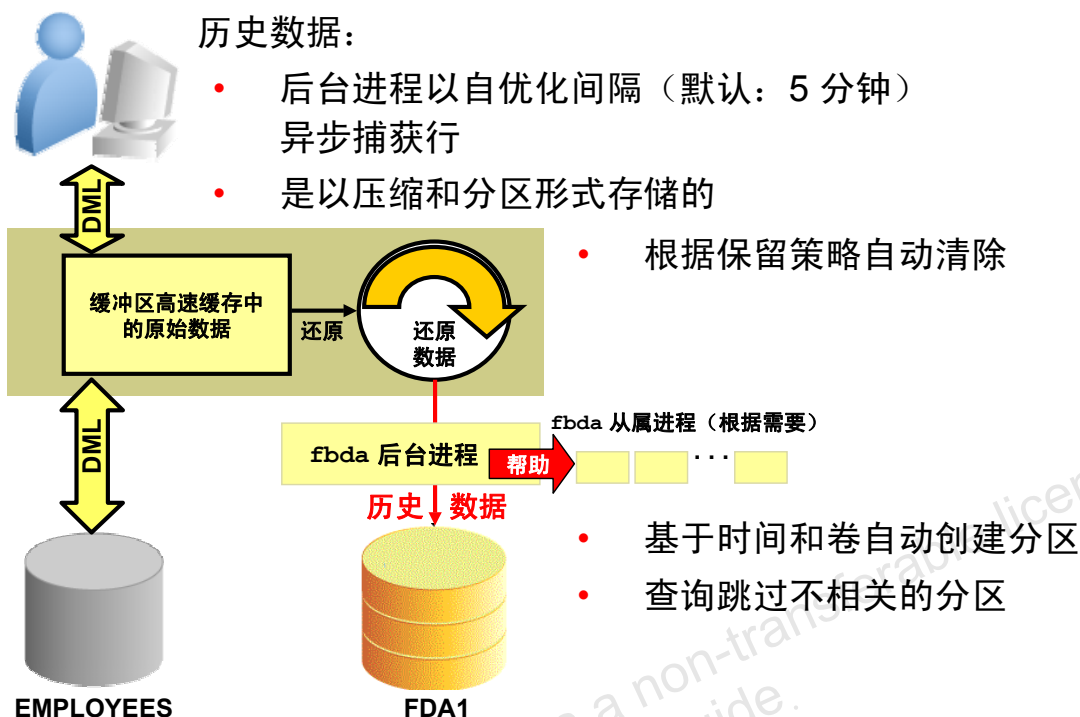
### 闪回数据归档的过程

一个闪回数据归档包含一个或多个表空间。可以拥有多个闪回数据归档，它们都配置有保留时间。您应根据保留时间的要求创建不同的闪回数据归档 — 例如，为必须保留两年的所有记录创建一个闪回数据归档，为必须保留五年的所有记录创建另一个闪回数据归档。数据库将在保留期到期后的第一天自动清除所有历史记录信息。

1. 为闪回数据归档创建表空间。空间大小取决于基表及预期的 DML 和 DDL 活动。
2. 创建一个闪回数据归档并设置保留时间。归档在闪回数据归档中的数据的保存时间等于配置的保留时间。此任务需要 FLASHBACK ARCHIVE ADMINISTER 系统权限。如果需要不同的保留期，必须创建不同的归档。
3. 为（整个）表启用闪回归档（然后再禁用此归档）。此任务需要 FLASHBACK ARCHIVE 对象权限。即使为某个表启用了闪回归档，也仍然不允许对该表执行某些 DDL 语句。默认情况下，闪回归档对任何表都处于关闭状态。



## Total Recall 的工作原理



版权所有 © 2011, Oracle. 保留所有权利。

### Total Recall 的工作原理

由 fbda 后台进程以自优化间隔从还原（及缓冲区高速缓存）中捕获历史数据。默认间隔为 5 分钟。不论更新的列有多少，都会存储所更新的整个基表行。

- 使用 OLTP 表压缩（而不是 Hybrid Columnar 压缩）对历史数据进行压缩。  
注：如果基表使用 Hybrid Columnar 压缩，则表无法启用闪回数据归档。
- 每个闪回归档分区至少为 1 天和 1 MB 数据，基于 ENDSCN 进行分区。对归档进行闪回查询可避免查询不相关的分区。
- fbda 进程最多可调用 10 个闪回归档从属进程。
- 如果闪回归档进程和从属进程太忙，则归档可能会按顺序执行，这对用户的响应时间有很大影响。

## Oracle Total Recall 场景

使用闪回数据归档访问历史数据：

```
-- create the Flashback Data Archive
CREATE FLASHBACK ARCHIVE DEFAULT fla1
    TABLESPACE tbs1 QUOTA 10G RETENTION 5 YEAR;
```

①

```
-- Specify the default Flashback Data Archive
ALTER FLASHBACK ARCHIVE fla1 SET DEFAULT;
```

②

```
-- Enable Flashback Data Archive
ALTER TABLE inventory FLASHBACK ARCHIVE;
ALTER TABLE stock_data FLASHBACK ARCHIVE;
```

③

```
SELECT product_number, product_name, count FROM inventory AS
    OF TIMESTAMP TO_TIMESTAMP ('2007-01-01 00:00:00', 'YYYY-MM-
    DD HH24:MI:SS');
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### Oracle Total Recall 场景

可以使用 CREATE FLASHBACK ARCHIVE 语句创建闪回数据归档。

- 可以为系统指定默认的闪回数据归档（可选）。
- 需要提供闪回数据归档的名称。
- 需要提供闪回数据归档的第一个表空间的名称。
- 可以确定闪回数据归档在表空间中允许使用的最大空间量。默认值为“Unlimited（无限制）”。如果第一个表空间中的空间限额不是无限制的，则必须指定该值，否则会引发 ORA-55621。
- 需要提供保留时间（即需要保证表的闪回数据归档数据存储多少天）。

在幻灯片所示的第一个示例中，创建了一个名为 fla1 的默认闪回数据归档，该归档最多可以使用 tbs1 表空间的 10 GB 空间，其数据的保留期为五年。在第二个示例中，指定了默认闪回数据归档。默认情况下，系统没有任何闪回数据归档。可以使用以下两种方式之一设置它：

- 在 ALTER FLASHBACK ARCHIVE 语句的 SET DEFAULT 子句中指定某个现有闪回数据归档的名称。
- 在创建闪回数据归档时，在 CREATE FLASHBACK ARCHIVE 语句中包括 DEFAULT。

### Oracle Total Recall 场景（续）

在第三个示例中，启用了闪回数据归档。如果自动还原管理处于禁用状态，则在尝试修改表时会出现 ORA-55614。

## Oracle Total Recall 场景

可以选择增加空间：

```
ALTER FLASHBACK ARCHIVE fla1
ADD TABLESPACE tbs3 QUOTA 5G;
```

④

可以选择更改保留时间：

```
ALTER FLASHBACK ARCHIVE fla1 MODIFY RETENTION 2 YEAR;
```

⑤

可以选择清除数据：

```
ALTER FLASHBACK ARCHIVE fla1 PURGE BEFORE TIMESTAMP
(SYS_TIMESTAMP - INTERVAL '1' day);
```

⑥

可以选择删除闪回数据归档：

```
DROP FLASHBACK ARCHIVE fla1;
```

⑦

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### Oracle Total Recall 场景（续）

要对表启用闪回归档，请在 CREATE TABLE 或 ALTER TABLE 语句中包括 FLASHBACK ARCHIVE 子句。在 FLASHBACK ARCHIVE 子句中，可指定用于存储表的历史数据的闪回数据归档。默认值为系统的默认闪回数据归档。要对表禁用闪回归档，请在 ALTER TABLE 语句中指定 NO FLASHBACK ARCHIVE。

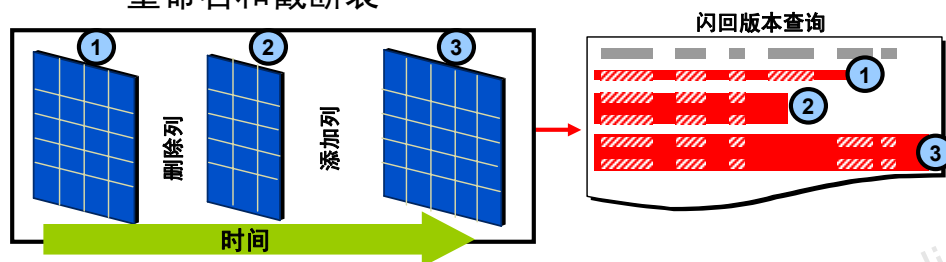
上一张幻灯片显示的最后一个语句显示了如何检索 2007 年年初的所有物料库存。继续使用上一示例：

- 示例 4 将 tbs3 表空间的最多 5 GB 空间添加至 fla1 闪回数据归档。
- 示例 5 将闪回数据归档 fla1 的保留时间更改为两年。
- 示例 6 从闪回数据归档 fla1 中清除了一天前的所有历史数据。通常会在保留时间到期后的第一天执行自动清除。也可以覆盖此设置进行临时清除。
- 示例 7 删除闪回数据归档 fla1 和历史数据，但不删除其表空间。使用 ALTER FLASHBACK ARCHIVE 命令可以：
  - 更改闪回数据归档的保留时间
  - 清除部分或全部数据
  - 增加、修改和删除表空间

注：删除闪回数据归档的全部表空间会引发错误。

## 透明的方案演化

- DDL 支持：
  - 添加、删除、重命名和修改列
  - 删除和截断分区
  - 重命名和截断表



- 闪回查询可以跨 DDL 更改运行。
- 不能自动支持所有其它 DDL（请见下一个幻灯片）。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 透明的方案演化

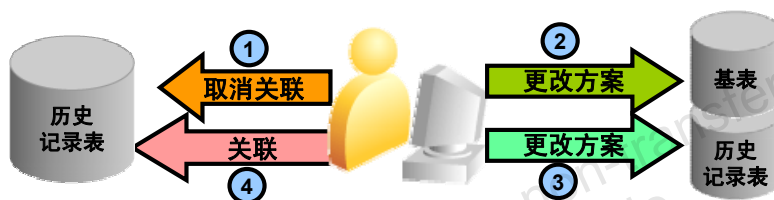
闪回数据归档可以使用最常用的 DDL 命令。如果方案以幻灯片中列出的任一方式进行了演化，Total Recall 会自动跟踪更改。闪回查询会随相应方案返回对应的一行或多行（如图中所示）。

## 完整方案演化

取消关联或关联 DBMS\_FLASHBACK\_ARCHIVE 程序包中的过程：

- 对指定的表禁用 Total Recall 并允许更复杂的 DDL（升级、拆分表，等等）。
- 在关联期间执行方案完整性检查。（基表和历史记录表必须为相同的方案。）

**注：**应谨慎使用此项功能，请注意此功能会导致归档不再保持不变，因为在取消关联期间历史记录可能会更改。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 完整方案演化

可以通过 DBMS\_FLASHBACK\_ARCHIVE 程序包执行不被自动支持的所有 DDL 更改。可以使用 DISASSOCIATE\_FBA 和 REASSOCIATE\_FBA 过程从闪回数据归档取消关联和重新关联给定的表。

**注：**应谨慎使用此项功能，请注意此功能会导致归档不再保持不变，因为在取消关联期间历史记录可能会更改。取消关联发生时，系统目录会有记录。

幻灯片中的图表显示了下列工作流：

- 如果有 FLASHBACK\_ARCHIVE\_ADMINISTER 权限，您可以从基表中取消关联归档。
- 为基表进行必要的更改。
- 为相应的归档进行必要的更改。
- 然后使用相同方案将表与归档关联。Total Recall 验证关联方案是否相同。

## 限制

- 如果使用 Hybrid Columnar 压缩，则对基表不能启用 Total Recall。
- 如果使用取消关联，则不能保证历史记录的永久性（但您总可以通过正确的权限清除之前的历史记录。）
- 历史记录表没有可移植性。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 限制

针对启用了闪回数据归档的表使用 DDL 语句时，某些 DDL 语句会导致 ORA-55610 错误。例如：

- ALTER TABLE 语句，包含 UPGRADE TABLE 子句并且使用或不使用 INCLUDING DATA 子句
- ALTER TABLE 语句，用于移动或交换分区或子分区操作
- DROP TABLE 语句

## 准则

- 使用 SCN 进行精确查询。  
**或者**
- 为方便可使用闪回技术。
- 闪回使用当前系统设置。
- 在查询过去的数据库之前，使用 COMMIT 或 ROLLBACK 操作确保数据库的一致性。
- 不能从动态性能 (V\$) 视图检索过去的数据库。它们包含当前数据库。
- 但是，可以在静态数据字典视图（如 \*\_TABLES）中执行对过去数据库的查询。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 准则

- 为了方便起见，可以对您编写的 SQL 代码使用闪回查询、闪回版本查询或闪回事务处理查询。
- 请记住，所有闪回处理过程都使用当前会话设置（如国家/地区语言和字符集），而不是在所查询的时间有效的设置。
- 要查询某个精确时间的过去数据库，可使用 SCN。如果使用时间戳，查询的实际时间可能会比指定的时间提前 3 秒钟。Oracle DB 在内部使用 SCN，并以 3 秒钟的粒度级别将它们映射到时间戳。
- 要获得稍后与闪回功能一起使用的 SCN，可以使用 DBMS\_FLASHBACK.GET\_SYSTEM\_CHANGE\_NUMBER 函数。
- 要计算或检索一个过去时间供查询使用，可以使用一个函数返回值作为时间戳或 SCN 参数。例如，将 SYSTIMESTAMP 函数的值加上或减去一个 INTERVAL 值。
- 要确保数据库的一致性，请始终在查询过去数据库之前执行 COMMIT 或 ROLLBACK 操作。
- 不能从动态性能 (V\$) 视图检索过去的数据库。对此类视图的查询总是返回当前数据库。但是，可以在静态数据字典视图（如 \*\_TABLES）中执行对过去数据库的查询。



## 查看闪回数据归档

查看结果：

视图名称 (DBA/USER)	说明
*_FLASHBACK_ARCHIVE	显示有关闪回数据归档的信息
*_FLASHBACK_ARCHIVE_TS	显示闪回数据归档的表空间
*_FLASHBACK_ARCHIVE_TABLES	显示有关启用了闪回归档的表的信息

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 查看闪回数据归档

可使用动态数据字典视图查看跟踪的表和闪回数据归档元数据。要访问 USER\_FLASHBACK\_\* 视图，您必须拥有表的所有权。要检查 DBA\_FLASHBACK\_\* 视图，您需要 SYSDBA 权限。

有关详细资料，请参阅《Advanced Application Developer's Guide》和《PL/SQL Packages and Types Reference》。

## 测验

不能删除但可以截断 Oracle Total Recall 跟踪的表。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 测验

选择有关 Oracle Total Recall 全面回忆的所有正确叙述：

1. 默认情况下启用 Oracle Total Recall。
2. 利用闪回数据归档功能，可以在“跟踪的”表的整个生命期内跟踪并存储对该表的所有事务处理更改。
3. 删除启用了闪回数据归档的表中的列会导致错误。
4. 闪回处理总是使用查询时有效的设置。
5. 闪回使用当前会话设置，例如国家/地区语言和字符集。

ORACLE

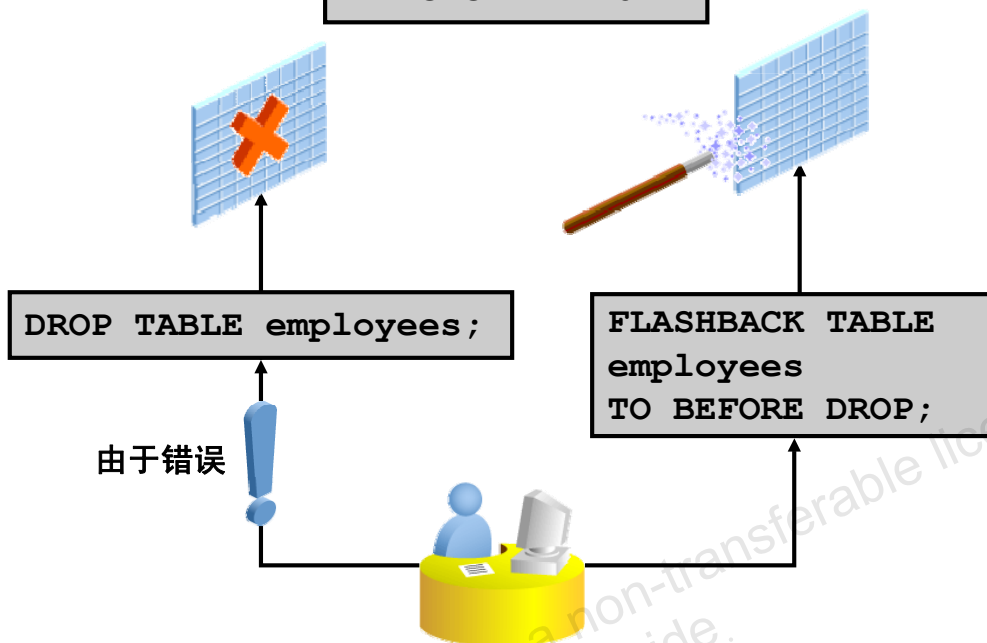
版权所有 © 2011, Oracle。保留所有权利。

答案：2、5

# 闪回删除和回收站

Total Recall  
> 闪回删除

RECYCLEBIN=ON



ORACLE

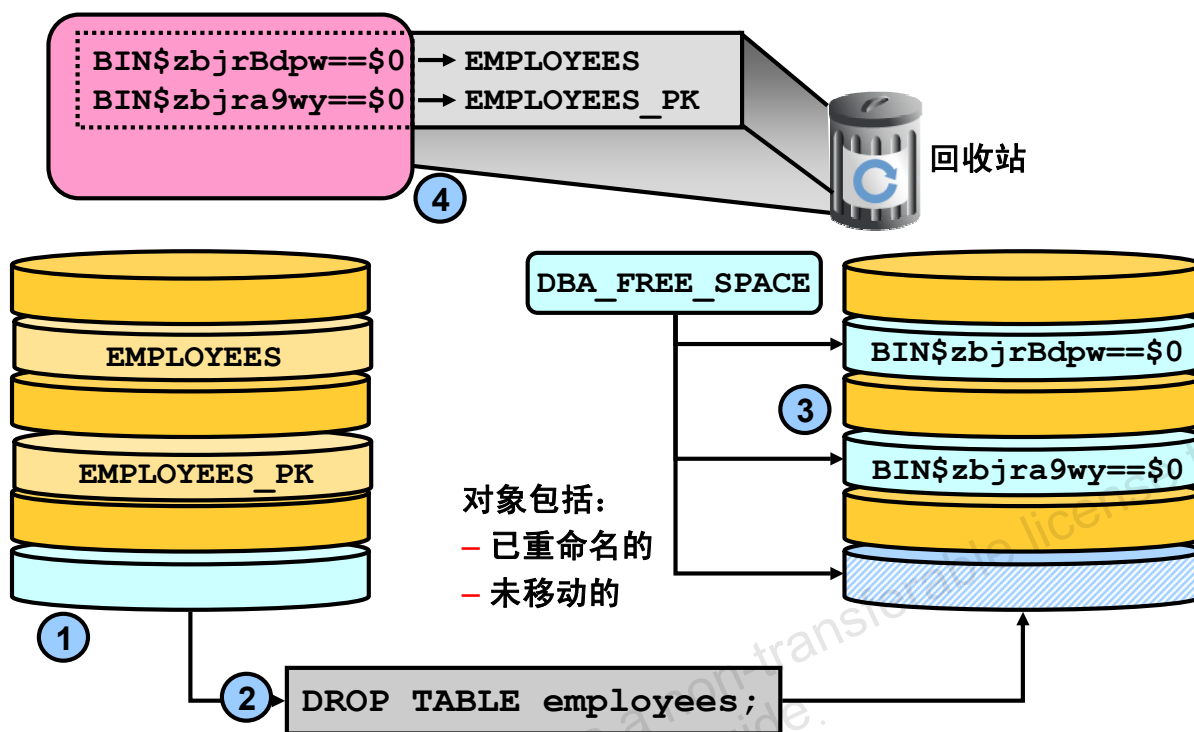
版权所有 © 2011, Oracle. 保留所有权利。

## 闪回删除和回收站

使用 `FLASHBACK TABLE` 命令，您可以在无需使用时间点恢复的情况下，还原 `DROP TABLE` 语句的结果。

**注：**初始化参数 `RECYCLEBIN` 用于控制闪回删除功能是打开 (ON) 还是关闭 (OFF)。如果将该参数设置为 OFF，则删除的表不会进入回收站。如果将该参数设置为 ON，则删除的表将进入回收站，并且可以进行恢复。默认情况下，`RECYCLEBIN` 设置为 ON。

## 回收站



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 回收站

如果不启用回收站，则删除表时，与该表及其从属对象关联的空间会立即变为可回收（也就是说，该空间可用于其它对象）。

如果启用了回收站，则删除表时，与该表及其从属对象关联的空间不会立即变为可回收，即使该空间确实显示在 `DBA_FREE_SPACE` 中。相反，会在回收站中引用删除的对象，这些对象仍属于其各自的所有者。在空间不紧张时，绝不会把回收站对象使用的空间自动回收。这样，您就能在尽可能长的期限内恢复回收站对象。

将删除的表“移动”到回收站时，将使用系统生成的名称对该表及其关联对象和约束条件进行重命名。重命名惯例如下：`BIN$unique_id$version`。

其中，`unique_id` 是该对象的全局唯一标识符，包含 26 个字符，用于在所有数据库之间唯一地标识回收站名称，而 `version` 是数据库分配的版本号。

## 回收站（续）

回收站本身是一个数据字典表，用于维护已删除对象的原始名称与各自系统生成名称之间的关系。可使用 `DBA_RECYCLEBIN` 视图查询回收站。上一张幻灯片中的图表说明了这种行为：

1. 您已经在表空间中创建了一个名为 `EMPLOYEES` 的表。
2. 删除 `EMPLOYEES` 表。
3. 此时，`EMPLOYEES` 占用的区被视为空闲空间。
4. `EMPLOYEES` 被重命名且新名称将记录到回收站中。

## 从回收站还原表

- 还原删除的表和从属对象。
- 如果多个回收站条目具有相同原始名称，则：
  - 使用系统生成的唯一名称来还原特定版本
  - 使用原始名称时，还原的表遵循后进先出 (LIFO) 的规则
- 重命名原始名称（如果当前正在使用该名称）。

```
FLASHBACK TABLE <table_name> TO BEFORE DROP  
[RENAME TO <new_name>];
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 从回收站还原表

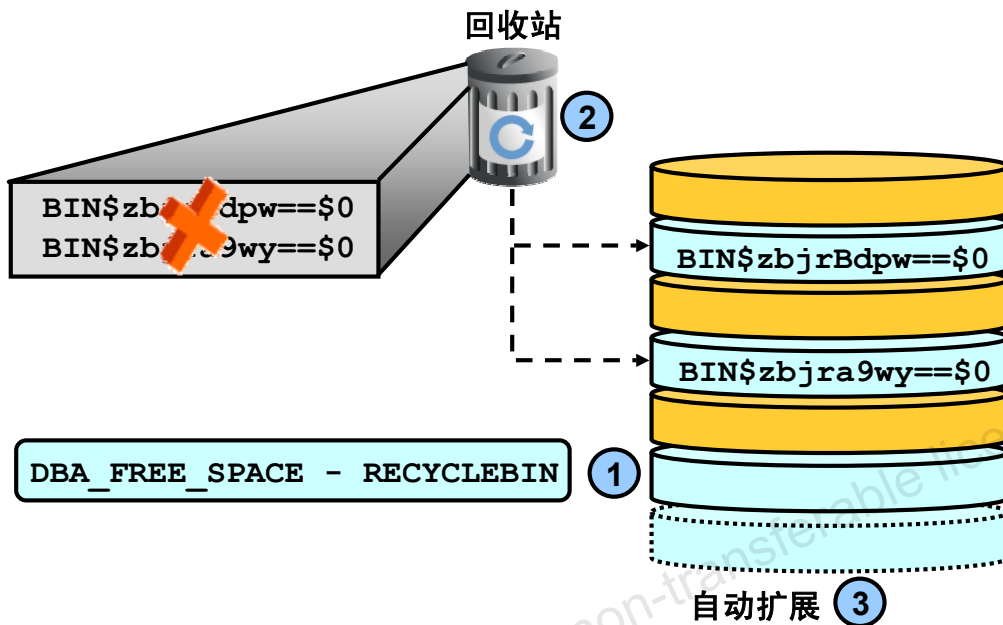
使用 `FLASHBACK TABLE ... TO BEFORE DROP` 命令从回收站恢复表及其所有可能的从属对象。可以指定表的原始名称或删除对象时分配给对象的系统生成名称。

如果指定原始名称，且回收站包含多个具有该名称的对象，则最先恢复最晚移动到回收站的对象（LIFO：后进先出）。如果需要检索表的较早版本，则可以指定要检索的表的系统生成名称，或发出其它 `FLASHBACK TABLE ... TO BEFORE DROP` 语句，直到检索到所需的表。

如果删除原始表后又在同一方案中创建了同名的新表，则会返回错误，除非还指定了 `RENAME TO` 子句。

**注：**闪回删除的表时，恢复的索引、触发器和约束条件将保留各自的回收站名称。因此，建议在闪回删除的表前查询回收站和 `DBA_CONSTRAINTS`。使用这种方法，可以将恢复的索引、触发器和约束条件重命名为更实用的名称。

## 回收站：自动回收空间



版权所有 © 2011, Oracle。保留所有权利。

### 回收站：自动回收空间

只要回收站对象使用的空间没有被回收，就可以使用闪回删除功能恢复这些对象。下面是回收站对象的回收策略：

- 显式发出 PURGE 命令时执行手动清除
- 空间不足时执行自动清除：对象在回收站中时，DBA\_FREE\_SPACE 也会报告其对应空间，因为这些空间是可以自动回收的。然后按以下顺序使用特定表空间中的空闲空间：
  1. 与回收站对象不对应的空闲空间。
  2. 与回收站对象对应的空闲空间。在这种情况下，将使用先进先出 (FIFO) 算法自动将回收站对象从回收站中清除。
  3. 自动分配的空闲空间（如果表空间是可以自动扩展的）假设在 TBS1 表空间内创建一个新表。如果向此表空间分配了与回收站对象不对应的空闲空间，则首先使用此空闲空间。如果该空间不够，则使用与 TBS1 内驻留的回收站对象对应的空闲空间。如果使用了某些回收站对象的空闲空间，则会从回收站中自动清除这些对象。此时，将无法再使用闪回删除功能恢复这些对象。如果仍未满足空间要求，则作为最后的方法，会在可能的情况下扩展 TBS1 表空间。

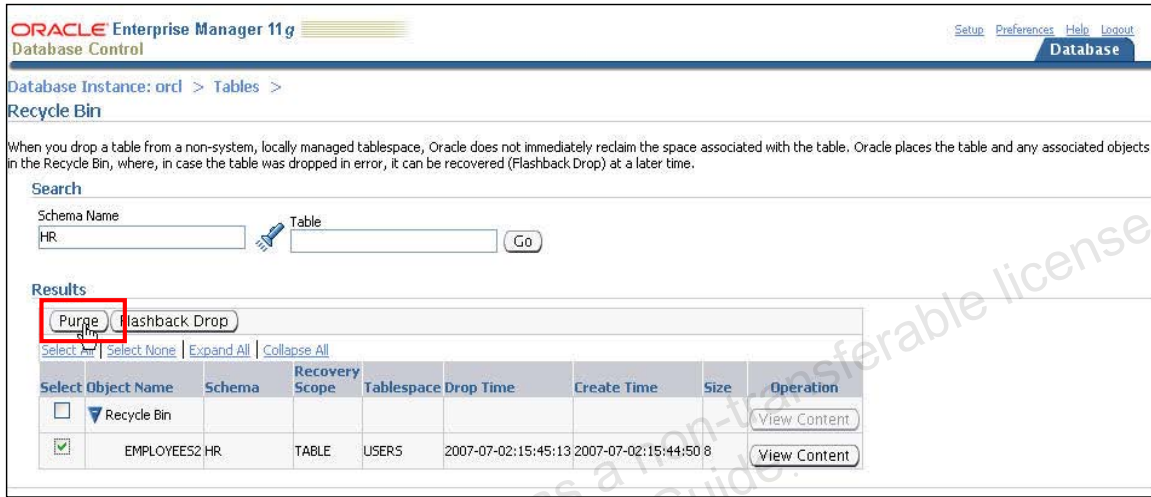


## 回收站：手动回收空间

```
PURGE {TABLE <table_name>|INDEX <index_name>}
```

```
PURGE TABLESPACE <ts_name> [USER <user_name>]
```

```
PURGE [USER_|DBA_]RECYCLEBIN
```



版权所有 © 2011, Oracle。保留所有权利。

### 回收站：手动回收空间

使用 PURGE 命令可从回收站中永久地删除对象。从回收站中清除某个对象时，会从数据库中永久地删除该对象及其从属对象。因此，将无法再使用闪回删除功能恢复从回收站中清除的对象。下面是可能使用的一些 PURGE 命令：

- PURGE TABLE 用于清除指定表。
- PURGE INDEX 用于清除指定索引。
- PURGE TABLESPACE 用于清除驻留在指定表空间中的所有对象。此外，也可能清除从属的、驻留在其它表空间中的对象。
- PURGE RECYCLEBIN 用于清除属于当前用户的所有对象。RECYCLEBIN 和 USER\_RECYCLEBIN 同义。
- PURGE DBA\_RECYCLEBIN 用于清除所有对象。要发出此命令，必须具有足够的系统权限或 SYSDBA 系统权限。

还可以使用 Oracle Enterprise Manager 从回收站中清除表。在“Schema folder（方案文件夹）”选项卡中，单击“Tables（表）”，选择删除的对象所在的方案，然后单击“Recycle Bin（回收站）”按钮。从结果列表中选择表并单击“Purge（清除）”按钮。

**注：**对于 PURGE TABLE 和 PURGE INDEX 命令，如果指定原始名称且回收站包含多个具有该名称的对象，则首先清除在回收站中的时间最长的对象 (FIFO)。

## 不使用回收站

```
DROP TABLE <table_name> [PURGE] ;
```

```
DROP TABLESPACE <ts_name>  
[INCLUDING CONTENTS] ;
```

```
DROP USER <user_name> [CASCADE] ;
```

### 回收站的安全注意事项：

```
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;
```


**ORACLE**

版权所有 © 2011, Oracle. 保留所有权利。

### 不使用回收站

使用 DROP TABLE PURGE 命令可从数据库中永久地删除表及其从属对象。使用此命令时，对应的对象不会移到回收站中。此命令提供的功能与以前版本中 DROP TABLE 命令的功能相同。

发出 DROP TABLESPACE ... INCLUDING CONTENTS 命令后，表空间中的对象不会移到回收站中。而且，回收站中属于该表空间的对象也会被清除。发出不带 INCLUDING CONTENTS 子句的以上命令时，表空间必须为空，之后才能成功执行命令。但回收站中可以有属于该表空间的对象。这种情况下，会清除这些对象。

发出 DROP USER ... CASCADE 命令后，将从数据库中永久地删除该用户及其拥有的所有对象。回收站中属于已删除用户的所有对象都将被清除。

为增加安全性，您可以决定禁用回收站。以 SYSDBA 身份连接，您可以：

- 使用以下命令查看回收站状态：  
SHOW PARAMETER RECYCLEBIN
- 使用以下命令禁用回收站：  
ALTER SYSTEM SET RECYCLEBIN=OFF SCOPE=SPFILE;

发出此命令后，您需要重新启动数据库。

## 查询回收站

```
SELECT owner, original_name, object_name,
       type, ts_name, droptime, related, space
FROM dba_recyclebin
WHERE can_undrop = 'YES';
```

```
SQL> SELECT original_name, object_name, ts_name, droptime
FROM user_recyclebin WHERE can_undrop = 'YES';
```

ORIGINAL_NAME	OBJECT_NAME	TS_NAME	DROPTIME
EMPLOYEES2	BIN\$NE4Rk64w...gbpQ==\$0	USERS	2007-07-02:15:45:13

```
SQL> SHOW RECYCLEBIN
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 查询回收站

通过查询 `user_recyclebin` 或 `RECYCLEBIN` 可查看所有已删除的对象。为便于使用，它具有同义词 `RECYCLEBIN`。

`dba_recyclebin` 视图可显示所有用户删除的且仍在回收站中的所有对象。

还可以使用 `SQL*Plus SHOW RECYCLEBIN` 命令。此命令只显示可“取消删除”的那些对象。

以下示例显示了如何从回收站中提取重要信息：

- `original_name` 是删除对象前的对象名称。
- `object_name` 是删除对象后系统生成的对象名称。
- `type` 是对象的类型。
- `ts_name` 是对象所属表空间的名称。
- `droptime` 是删除对象的日期。
- `related` 是删除对象的对象标识符。
- `space` 是对象当前使用的块数。

还可以使用 Database Control 查看回收站的内容。

注：有关 `DBA_RECYCLEBIN` 视图的详细信息，请参阅《Oracle Database Reference》指南。

## 测验

闪回删除的表时，恢复的索引、触发器和约束条件将保留各自的回收站名称。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 小结

在本课中，您应该已经学会：

- 设置和使用 Total Recall
- 从回收站还原删除的表
- 查询回收站

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 11 概览：使用闪回技术

本练习包含以下主题：

- 使用 Total Recall
- 回收站活动（可选）

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

# 12

## 执行闪回数据库

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

华周 (286991486@qq.com) has a non-transferable license to use this Student Guide.

## 课程目标

学完本课后，应能完成以下工作：

- 配置闪回数据库
- 执行闪回数据库操作
- 监视闪回数据库

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。



## 闪回数据库

闪回数据库操作：

- 作用类似于数据库的倒带按钮
- 可以在用户造成了逻辑数据损坏的情况下使用



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

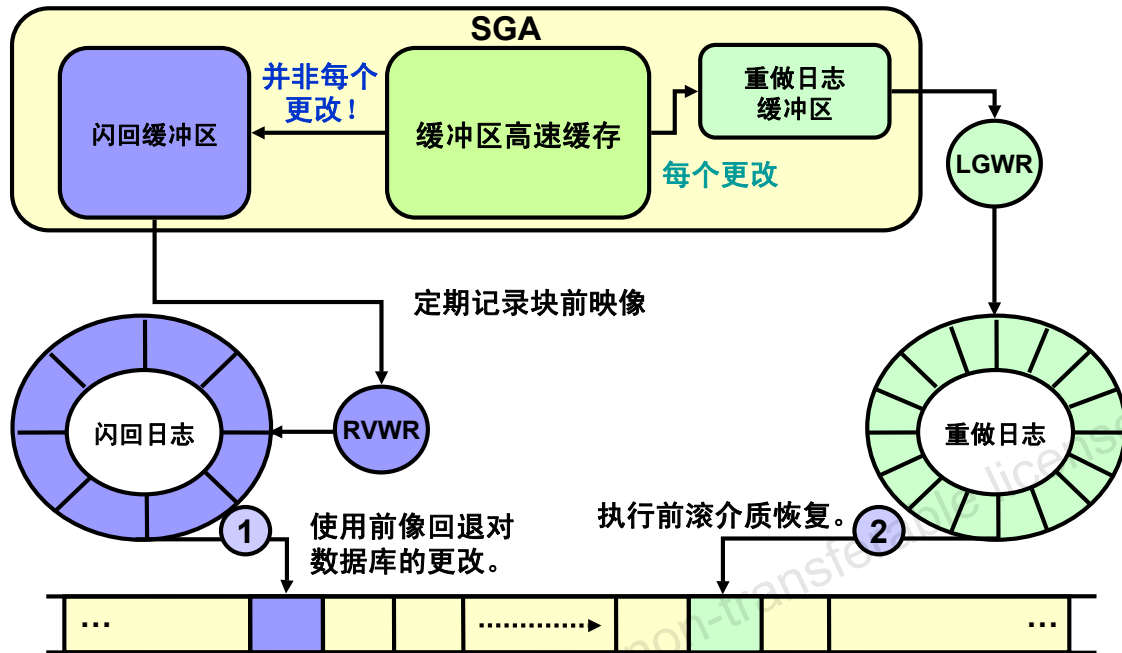
### 闪回数据库

使用闪回数据库，通过还原自先前某个时间点以来发生的所有更改，可快速将数据库恢复到那个时间的状态。因为不需要还原备份，所以此操作速度很快。可以使用此功能还原导致逻辑数据损坏的更改。

使用闪回数据库时，Oracle DB 可使用过去的块映像回退对数据库的更改。在正常数据库操作期间，Oracle DB 会不定期地将这些块映像记录在闪回日志中。闪回日志将按顺序写入并且不进行归档。Oracle DB 会自动在快速恢复区中创建、删除闪回日志和调整它的大小。您仅需出于监视性能目的而关注闪回日志，并确定为快速恢复区分配了多少磁盘空间以存放闪回日志。

使用闪回数据库倒回数据库所需的时间与需回退到多久以前以及目标时间之后发生的数据库活动量成比例。还原和恢复整个数据库所需的时间会长得多。闪回日志中的前像仅用于将数据库还原至过去的某一时间点，而前滚恢复则用于将数据库恢复到与过去某个时间一致的状态。Oracle DB 可将数据文件恢复至以前的时间点，但辅助文件除外，如初始化参数文件。闪回数据库还可用来辅助 Data Guard 和恢复指导，以及用来同步已复制的数据库。

## 闪回数据库体系结构



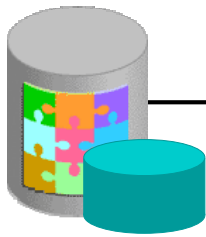
版权所有 © 2011, Oracle. 保留所有权利。

### 闪回数据库体系结构

启用闪回数据库功能时，会启动 RVWR（闪回写进程）后台进程。此后台进程按顺序将闪回数据库数据从闪回缓冲区写入闪回数据库日志，这些日志会被循环使用。随后，当发出 FLASHBACK DATABASE 命令时，系统使用闪回日志来还原到块的前像，然后使用重做数据前滚到所需的闪回时间。

启用闪回数据库的开销取决于数据库的读/写混合工作量。因为查询不需要记录任何闪回数据，所以工作量的写操作量越大，启用闪回数据库的开销就越高。

## 配置闪回数据库



1. 配置 FRA。



2. 设置保留目标。



3. 启用闪回数据库。

```
SQL> SHUTDOWN IMMEDIATE
SQL> STARTUP MOUNT
SQL> ALTER DATABASE ARCHIVELOG;
SQL> ALTER SYSTEM SET
      2 DB_FLASHBACK_RETENTION_TARGET=2880 SCOPE=BOTH;
SQL> ALTER DATABASE FLASHBACK ON;
SQL> ALTER DATABASE OPEN;
```

如果您的数据库处于  
ARCHIVELOG 模式，则  
无需重新启动该数据库。

使用打开的数据库。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 配置闪回数据库

可按以下步骤配置闪回数据库：

1. 配置快速恢复区。
2. 使用 DB\_FLASHBACK\_RETENTION\_TARGET 初始化参数设置保留目标。可指定一个上限（以分钟为单位），指示数据库能够闪回到多长时间以前。本示例使用了 2,880 分钟，相当于两天。此参数只是一个目标，并不提供任何保证。闪回时间间隔取决于快速恢复区中保存的闪回数据量。
3. 使用以下命令启用闪回数据库：

```
ALTER DATABASE FLASHBACK ON;
```

必须先针对归档配置数据库，之后才能发出命令来启用闪回数据库。

可以使用以下查询来确定是否已启用闪回数据库：

```
SELECT flashback_on FROM v$database;
```

可使用 ALTER DATABASE FLASHBACK OFF 命令来禁用闪回数据库。这样，会自动删除所有现有的闪回数据库日志。

**注：**只有在独占模式下装载数据库后才能启用闪回数据库，在打开状态下则不可以。

## 您需要执行的操作

### 配置 workflow:

1. 确保数据库处于 ARCHIVELOG 模式。
2. 启用闪回日志记录并指定快速恢复区。

### Flash Recovery

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location

Flash Recovery Area Size  GB

Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (GB)	2.07
Reclaimable Flash Recovery Area (GB)	1.38
Free Flash Recovery Area (GB)	6.55

☒ Enable Flashback Database\*

Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time  Hours

Current size of the flashback logs(GB) n/a

Lowest SCN in the flashback data n/a

Flashback Time n/a

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 您需要执行的操作

登录到 Oracle Enterprise Manager (EM)。在“Availability（可用性）”页上的“Backup/ Recovery（备份/恢复）”区域中，选择“Recovery Settings（恢复设置）”。确保数据库处于 ARCHIVELOG 模式。如果数据库未处于此模式，请选中“ARCHIVELOG Mode（ARCHIVELOG 模式）”，然后单击“Continue（继续）”。需要关闭并重新启动实例，更改才能生效。

配置了快速恢复区和归档后，USE\_DB\_RECOVERY\_FILE\_DEST 被配置为归档日志目标位置 10。选择“Enable Flashback Logging（启用闪回日志记录）”可启用闪回日志记录。还可以设置闪回保留时间，查看与闪回数据库窗口有关的重要信息。

复查快速恢复区的位置。快速恢复区是 Oracle DB 中所有与恢复相关的文件和活动的统一存储位置。在发生介质故障后完全恢复数据库所需的所有文件都包含在快速恢复区中。可在快速恢复区中创建的与恢复相关的文件包括：归档重做日志文件、控制文件、Recovery Manager (RMAN) 创建的备份、闪回日志和更改跟踪文件。通过分配存储位置并将与恢复相关的文件集中到特定区域内，Oracle DB Server 减轻了数据库管理员的负担，使他们不必管理由这些组件创建的磁盘文件。快速恢复区的默认位置为 \$ORACLE\_BASE/flash\_recovery\_area。如果希望其位于其它位置，可立即进行更改。向下滚动到“Recovery Settings（恢复设置）”页的底部，然后单击“Apply（应用）”。

## 闪回数据库：示例

- 闪回：以独占模式装载数据库

```

RMAN> FLASHBACK DATABASE TO TIME =
2> "TO_DATE('2009-05-27 16:00:00',
3> 'YYYY-MM-DD HH24:MI:SS')";

RMAN> FLASHBACK DATABASE TO SCN=23565;
RMAN> FLASHBACK DATABASE
2> TO SEQUENCE=223 THREAD=1;
  
```

通过 V\$SESSION\_LONGOPS  
视图监视闪回数据库的进度。

```

SQL> FLASHBACK DATABASE
2 TO TIMESTAMP(SYSDATE-1/24);
SQL> FLASHBACK DATABASE TO SCN 53943;
SQL> FLASHBACK DATABASE TO RESTORE POINT b4_load;
  
```

- 复查更改：以只读方式打开数据库
- 完成：使用 RESETLOGS 以读/写方式打开数据库

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 闪回数据库：示例

可以使用 RMAN FLASHBACK DATABASE 命令执行闪回数据库操作。可以使用 SEQUENCE 和 THREAD 指定作为下限的重做日志序列号和线程。RMAN 只选择可用于闪回到指定序列号（但不包括指定序列号）的文件。

也可以使用 SQL FLASHBACK DATABASE 命令将数据库返回到过去的某个时间或 SCN。如果使用 TO SCN 子句，则必须提供一个数字。如果指定 TO TIMESTAMP，则必须提供一个时间戳值。还可以指定还原点名称。

您可以通过 V\$SESSION\_LONGOPS 视图监视闪回数据库进度。

**注：**必须以独占模式装载数据库，之后才能发出 FLASHBACK DATABASE 命令；要复查更改，可以按只读方式打开数据库。完成后，必须用 RESETLOGS 选项以读/写方式打开数据库。

## 闪回数据库注意事项

- 闪回数据库操作完成后，打开数据库：
  - 在只读模式下验证使用的目标时间或 SCN 正确无误。
  - 使用 RESETLOGS 操作以允许 DML。
- “闪回”与“恢复”是对立的。
- 以下情况下不能使用闪回数据库操作：
  - 控制文件已还原或已重新创建。
  - 表空间已删除。
  - 数据文件的大小已被减小。
- 使用 TO BEFORE RESETLOGS 子句闪回至上一 RESETLOGS 操作前的那一刻。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 闪回数据库注意事项

在不能使用闪回数据库功能的情况下，应使用不完全恢复操作将数据库恢复至特定时间。闪回数据库操作完成后，可在只读模式下打开数据库，验证是否使用了正确的目标时间或 SCN。如果没有，可以再次闪回数据库，或者通过执行恢复操作来前滚数据库。所以，要还原闪回数据库操作，应前滚恢复数据库。

不能使用闪回数据库操作来恢复在闪回操作期间删除的数据文件。删除的数据文件会添加到控制文件中且标记为脱机，但是不会闪回。闪回数据库操作无法将数据文件闪回到其创建之后、调整大小之前的某个时间。如果某个文件在数据库要闪回到的期间调整了大小，则应在闪回数据库操作开始之前使该文件脱机。这适用于收缩的文件，而不是扩展的文件。可以将闪回数据库用于已经配置为自动扩展的数据文件。通过在 FLASHBACK DATABASE 命令中提供 TO BEFORE RESETLOGS 子句，可恰好闪回至上一 RESETLOGS 操作前的那一刻。

**注：**闪回保留目标不能完全保证闪回可用。如果必需的文件需要在快速恢复区中占用空间，可能会自动删除闪回日志。



## 监视闪回数据库

要针对保留目标监视容量：

- 查看快速恢复区的磁盘限额：

```
SQL> SELECT estimated_flashback_size,
2         flashback_size
3 FROM    V$FLASHBACK_DATABASE_LOG;
```

- 确定当前闪回窗口：

```
SQL> SELECT oldest_flashback_scn,
2         oldest_flashback_time
3 FROM    V$FLASHBACK_DATABASE_LOG;
```

- 监视闪回数据库日志中的日志记录：

```
SQL> SELECT *
2 FROM    V$FLASHBACK_DATABASE_STAT;
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 监视闪回数据库

监视快速恢复区的空间使用量是非常重要的，因为这样做可知道如何才能满足保留目标。使用 V\$FLASHBACK\_DATABASE\_LOG 视图可监视闪回数据库的保留目标：

- ESTIMATED\_FLASHBACK\_SIZE 使用先前记录的闪回数据，可估计快速恢复区中闪回日志需要多少磁盘空间才能满足当前闪回保留目标。估计的依据是自实例启动以来的工作量，或者是等于闪回保留目标的最近时间间隔，取较小的那个值。
- FLASHBACK\_SIZE 给出了闪回数据的当前字节大小（以字节为单位）。
- OLDEST\_FLASHBACK\_SCN 和 OLDEST\_FLASHBACK\_TIME 显示了数据库可闪回到的近似最低 SCN 和时间。V\$DATABASE 中的 CURRENT\_SCN 给出了当前数据库 SCN。

使用 V\$FLASHBACK\_DATABASE\_STAT 视图可监视闪回数据库日志中记录闪回数据的开销。此视图包含 24 小时的信息，每一行代表一小时的时间间隔。使用此视图可确定闪回数据的生成速率变化。

```
SQL> SELECT begin_time, end_time, flashback_data, db_data,
2         redo_data, estimated_flashback_size AS EST_FB_SZE
3 FROM    V$FLASHBACK_DATABASE_STAT;
```

## 监视闪回数据库（续）

BEGIN_TIME	END_TIME	FLASHBACK_DATA	DB_DATA	REDO_DATA	EST_FB_SIZE
12-FEB-09	12-FEB-09	16384	0	24576	0
12-FEB-09	12-FEB-09	6594560	7471104	1533440	815923200
12-FEB-09	12-FEB-09	17235968	12361728	5150920	839467008
12-FEB-09	12-FEB-09	311648256	37249024	10272768	855195648

基于这些信息，可能需要调整保留时间或快速恢复区大小。

FLASHBACK\_DATA 和 REDO\_DATA 分别代表时间间隔期间写入的闪回数据和重做数据的字节数，而且，DB\_DATA 提供了读写的数据块的字节数。此视图还包含一定时间间隔内预计需要的闪回空间。

查询 V\$RECOVERY\_FILE\_DEST 可查看与快速恢复区相关的信息。列说明如下：

- **NAME:** 快速恢复区的名称，指示位置字符串
- **SPACE\_LIMIT:** 在 DB\_RECOVERY\_FILE\_DEST\_SIZE 参数中指定的磁盘限制
- **SPACE\_USED:** 快速恢复区文件使用的空间（字节）
- **SPACE\_RECLAIMABLE:** 通过删除过时文件、冗余文件和其它空间管理算法确定的低优先级文件，可回收的空间量
- **NUMBER\_OF\_FILES:** 文件数量

```
SQL> SELECT name, space_limit AS quota,
2         space_used AS used,
3         space_reclaimable AS reclaimable,
4         number_of_files AS files
5 FROM v$recovery_file_dest;
```

NAME	QUOTA	USED	RECLAIMABLE	FILES
/u01/flash_recovery_area	5368707120	2507809104	203386880	226



## 使用 EM 监视闪回数据库

**Flash Recovery**

This database is using a flash recovery area. The chart shows space used by each file type that is not reclaimable by Oracle. Performing backups to tertiary storage is one way to make space reclaimable. Usable Flash Recovery Area includes free and reclaimable space.

Flash Recovery Area Location:

Flash Recovery Area Size:  GB

Flash Recovery Area Size must be set when the location is set.

Non-reclaimable Flash Recovery Area (MB): 294

Reclaimable Flash Recovery Area (B): 0

Free Flash Recovery Area (GB): 5.71

☐ Enable Flashback Database\*

Flashback database can be used for fast database point-in-time recovery, as it returns the database to a prior point-in-time without restoring files. Flashback is the preferred point-in-time recovery method in the recovery wizard when appropriate. The flash recovery area must be set to enable flashback database.

Flashback Retention Time:  Hours

Current size of the flashback logs(GB): n/a

Lowest SCN in the flashback data: n/a

Flashback Time: n/a

☐ Apply initialization parameter changes to SPFILE only. If not checked, parameter changes will be made to both the SPFILE and the running instance.

\* Changes to this setting or parameter require a database restart.

**Flash Recovery Area Usage**

File Type	Size (GB)	Percentage
Online Log	0.15	2.5%
Backup Piece	0.08	1.3%
Archived Redo Log	0.05	0.8%
Control File	0.01	0.2%
Image Copy	0	0%
Flashback Log	0	0%
Usable	5.71	95.2%

ORACLE

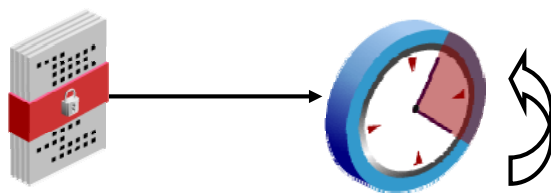
版权所有 © 2011, Oracle。保留所有权利。

## 使用 EM 监视闪回数据库

可在“Recovery Settings（恢复设置）”页中看到上面各页中提到的大多数闪回数据库统计信息。这些度量包括当前所有闪回日志使用的空间、最低 SCN 和闪回数据中最低 SCN 的时间。

## 保证还原点

使用保证还原点可确保在任何时间通过执行 FLASHBACK DATABASE 命令都可闪回到该 SCN。



```
SQL> CREATE RESTORE POINT before_upgrade  
2  GUARANTEE FLASHBACK DATABASE;
```

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 保证还原点

如普通还原点一样，保证还原点在恢复操作中可用作 SCN 的别名。主要区别是保证还原点绝不会因过期而从控制文件中删除，必须显式删除它。但是，保证还原点还提供与使用闪回数据库功能相关的特定功能。

在特定 SCN 创建保证还原点后，会通过强制执行闪回数据库操作将数据库恢复到该 SCN 所对应的状态，即使未对数据库启用闪回日志记录也是如此。如果启用了闪回日志记录，则创建保证还原点会强迫保留所需的闪回日志，使闪回数据库操作可恢复到最早的保证还原点创建后的任何时间点。

保证还原点可用于将整个数据库还原到几天或几周前的已知正常状态，只要快速恢复区中有足够的磁盘空间来存储所需的日志。与普通还原点相同，保证还原点可用于指定 RECOVER DATABASE 操作的时间点。

**注：**适用于闪回数据库的限制也适用于保证还原点。例如，收缩数据文件或删除表空间会导致受影响的数据文件无法闪回至保证还原点。

## 闪回数据库和保证还原点

要使用保证还原点，数据库必须满足以下先决条件：

- COMPATIBLE 初始化参数必须设置为 10.2 或更大值。
- 数据库必须在 ARCHIVELOG 模式下运行。
- FLASHBACK DATABASE 需要使用大约从还原点时间开始的归档重做日志。
- 必须配置快速恢复区。



版权所有 © 2011, Oracle。保留所有权利。

### 闪回数据库和保证还原点

要支持使用保证还原点，数据库必须满足以下先决条件：

- COMPATIBLE 初始化参数必须设置为 10.2 或更大值。
- 数据库必须在 ARCHIVELOG 模式下运行。
- 要将数据库倒回至保证还原点，FLASHBACK DATABASE 命令需要使用大约从还原点时间开始的归档重做日志。
- 必须配置快速恢复区。保证还原点使用类似于闪回日志记录的机制。与闪回日志记录一样，Oracle DB 必须将所需日志存储在快速恢复区中。
- 如果未启用闪回数据库，则创建第一个保证还原点时（或者如果已删除了先前创建的所有保证还原点），数据库必须处于装载状态而不是打开状态。

闪回数据库和保证还原点的日志记录需要捕获应用更改之前数据文件块的映像。

FLASHBACK DATABASE 命令可使用这些映像将数据文件恢复至以前的状态。普通闪回日志记录与保证还原点日志记录的主要区别在于何时记录块，以及是否在快速恢复区空间不足时删除日志。这些区别会影响日志的空间使用量和数据库性能。

如果启用闪回数据库并定义了一个或多个保证还原点，则数据库执行普通闪回日志记录。在这种情况下，恢复区会保留闪回至当前时间与目前最早定义的保证还原点之间的任意时间所需的闪回日志。如果保证还原点需要闪回日志，则不会因空间不足而删除闪回日志。

## 测验

当您想要执行下列操作时，可使用闪回数据库：

1. 修复逻辑数据损坏
2. 恢复已删除的表空间
3. 数据文件的大小已被减小之前恢复到某一时间点
4. 重新创建控制文件之前恢复到某一时间点

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 测验

归档闪回日志，以允许您倒回至 FRA 无法满足的时间点。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2

## 小结

在本课中，您应该已经学会：

- 配置闪回数据库
- 执行闪回数据库操作
- 监视闪回数据库

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 12 概览：使用闪回数据库

本练习包含以下主题：

- 通过执行闪回数据库功能撤消不需要的事务处理
- 监视闪回数据库的保留情况
- 确定闪回日志的大小

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

华周 (286991486@qq.com) has a non-transferable license to use  
this Student Guide.



# 13

## 管理内存

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 描述 SGA 中的内存组件
- 实施自动内存管理
- 手动配置 SGA 参数
- 配置自动 PGA 内存管理

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 内存管理：概览

DBA 必须将内存管理视为其工作中至关重要的部分，因为：

- 可用内存空间量有限
- 为某些类型的功能分配更多内存可提高整体性能
- 自动优化的内存分配通常是正确的配置，但特定环境甚至短期情况下可能需要特别注意

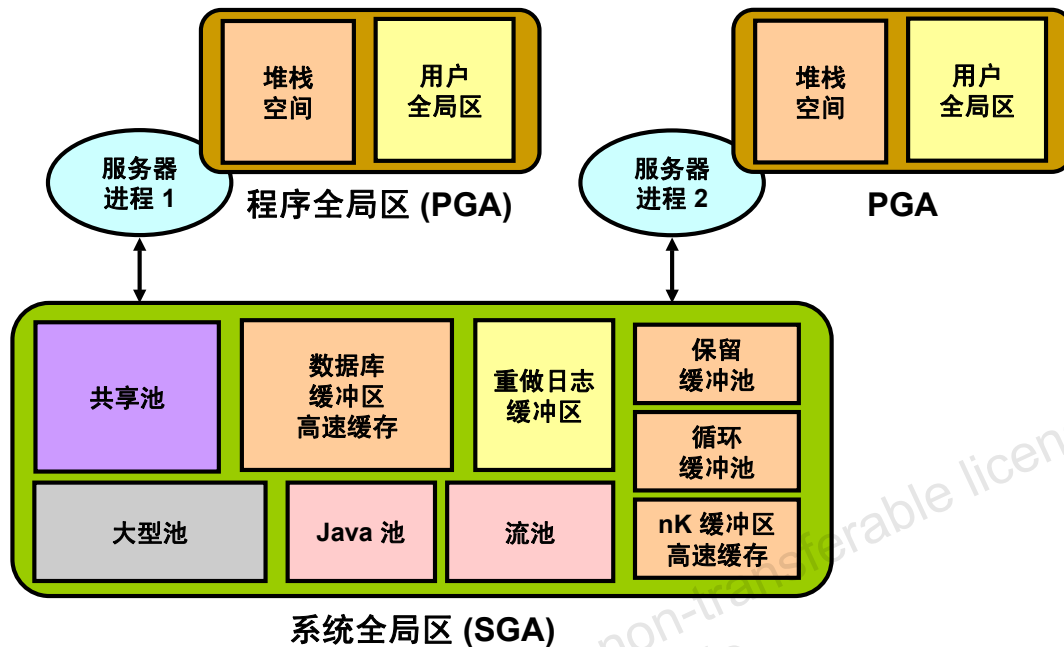
ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 内存管理：概览

由于数据库服务器上的可用内存量有限，因此，对于 Oracle DB 实例，必须注意内存的分配情况。如果将过多的内存分配给没有此需求的特定区域使用，则很可能导致其它功能区没有足够的内存，无法以最优方式工作。采用自动确定和维护内存分配的功能，极大地简化了该项任务。但是，要实现系统内存最佳利用，即使是自动优化的内存也需要进行监控，有时可能还需要某种程度的手动配置。

## 复查 Oracle DB 内存结构



版权所有 © 2011, Oracle. 保留所有权利。

### 复查 Oracle DB 内存结构

Oracle DB 创建并使用内存结构来满足多种用途。例如，内存可以存储正在运行的程序代码、用户间共享的数据以及每个已连接的用户的数据区域。

一个实例有两个关联的基本内存结构：

- **系统全局区 (SGA)：** 一组共享的内存结构（称为 SGA 组件），其中包含一个 Oracle DB 实例的数据和控制信息。SGA 在所有服务器和后台进程之间共享。SGA 中存储的数据的示例包括高速缓存的数据块和共享 SQL 区域。
- **程序全局区 (PGA)：** 包含服务器进程或后台进程的数据及控制信息的内存区域。PGA 是 Oracle DB 在服务器进程或后台进程启动时创建的非共享内存。服务器进程对 PGA 的访问是独占的。每个服务器进程和后台进程都具有自己的 PGA。

## 复查 Oracle DB 内存结构（续）

SGA 是包含实例的数据和控制信息的内存区。SGA 包含以下数据结构：

- **共享池：**用于缓存可在用户间共享的各种构造
- **数据库缓冲区高速缓存：**用于缓存从数据库中检索到的数据块
- **保留缓冲池：**一种经过优化的专用数据库缓冲区高速缓存，用于长时间在内存中保留数据块
- **回收缓冲池：**一种经过优化的专用数据库缓冲区高速缓存，用于从内存中快速回收或删除数据块
- **nK 缓冲区高速缓存：**几种专用数据库缓冲区高速缓存之一，用于存放与默认数据库块大小不同的块大小
- **重做日志缓冲区：**用来缓存用于恢复实例的重做信息，直到可以将其写入磁盘中存储的物理重做日志文件
- **大型池：**可选区域，用于为某些大型进程（如 Oracle 备份和恢复操作）和 I/O 服务器进程分配较大的内存空间
- **Java 池：**用于存储 Java 虚拟机 (JVM) 中特定会话的所有 Java 代码和数据
- **流池：**Oracle Streams 使用它来存储捕获和应用所需的信息

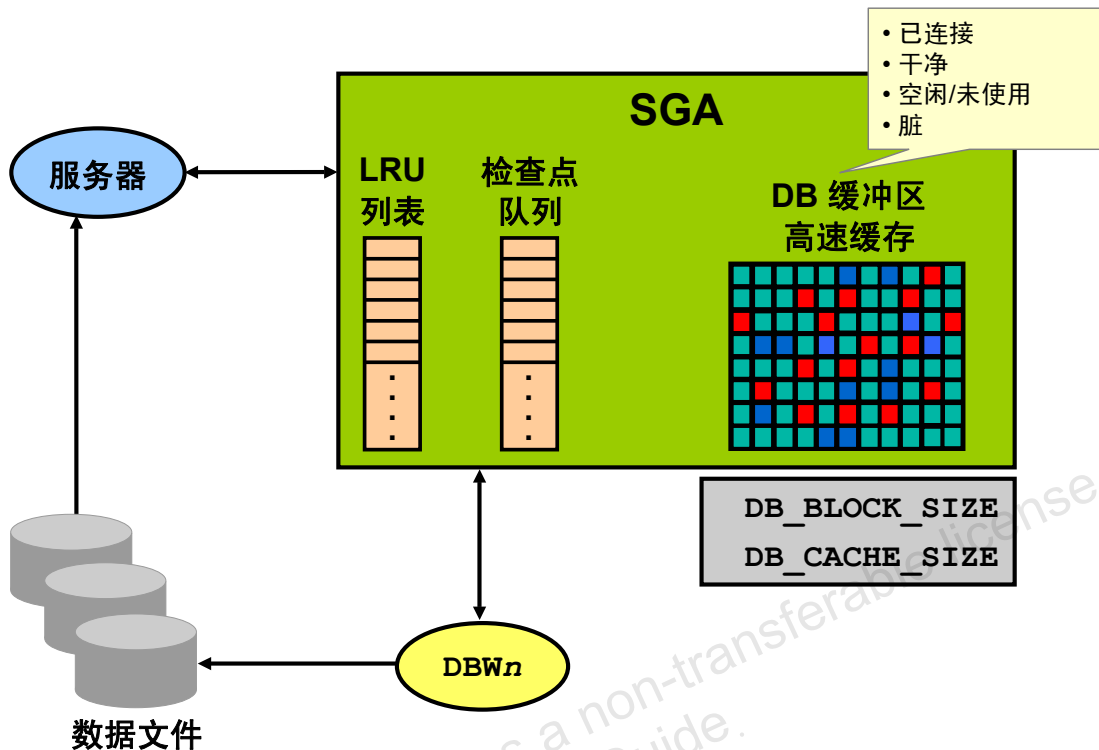
使用 Oracle Enterprise Manager 或 SQL\*Plus 启动实例时，会显示为 SGA 分配的内存量。

程序全局区 (PGA) 是一个内存区，其中包含每个服务器进程的数据及控制信息。Oracle Server 进程为客户机请求提供服务。每个服务器进程都有在服务器进程启动时创建的自己专用的 PGA。PGA 只能由相应的服务器进程访问，并且只有代表该服务器进程的 Oracle 代码可对其进行读取和写入。PGA 分为两个主要区域：堆栈空间和用户全局区 (UGA)。

使用动态 SGA 基础结构，可以在不关闭实例的情况下更改数据库缓冲区高速缓存、共享池、大型池、Java 池和流池的大小。

Oracle DB 使用初始化参数来创建和管理内存结构。管理内存的最简单方法是允许数据库自动管理和优化内存。在大多数平台上，要实现此目的，您只需设置目标内存大小初始化参数 (MEMORY\_TARGET) 和最大内存大小初始化参数 (MEMORY\_MAX\_TARGET)。

## 缓冲区高速缓存



版权所有 © 2011, Oracle。保留所有权利。

## 缓冲区高速缓存

通过指定 **DB\_CACHE\_SIZE** 参数的值，可以配置缓冲区高速缓存。缓冲区高速缓存可存放数据文件中块大小为 **DB\_BLOCK\_SIZE** 的数据块的副本。缓冲区高速缓存是 **SGA** 的一部分，因此所有用户都可以共享这些块。服务器进程将数据文件中的数据读入缓冲区高速缓存。为了提高性能，服务器进程有时在一个读操作中会读取多个块。然后由 **DBWn** 进程将数据从缓冲区高速缓存写入数据文件。为提高性能，**DBWn** 在一个写操作中会写入多个块。在任何给定时间，缓冲区高速缓存都可能会存放一个数据库块的多个副本。虽然该块只存在一个当前副本，但为了满足查询需要，服务器进程可能需要根据过去的映像信息构造读一致性副本。这称为读一致性 (CR) 块。

最近最少使用 (LRU) 列表可反映缓冲区的使用情况。缓冲区将依据其被引用时间的远近和引用频率进行排序。因此，最经常使用且最常用的缓冲区将列在最近最常使用一端。传入的块先被复制到最近最少使用一端的缓冲区中，然后该缓冲区将被指定到列表中央，作为起点。从这个起点开始，缓冲区根据使用情况在列表中上下移动。

## 缓冲区高速缓存（续）

缓冲区高速缓存中的缓冲区可以处于以下四种状态之一：

- **已连接：**当前正将该块读入高速缓存或正在写入该块。其它会话正等待访问该块。
- **干净的：**该缓冲区目前未连接，如果其当前内容（数据块）将不再被引用，则可以立即执行过期处理。这些内容与磁盘保持同步，或者缓冲区包含块的读一致性快照。
- **空闲/未使用：**缓冲区因实例刚启动而处于空白状态。此状态与“干净”状态非常相似，不同之处在于缓冲区未曾使用过。
- **脏：**缓冲区不再处于连接状态，但内容（数据块）已更改，因此必须先通过 DBWn 将内容刷新到磁盘，然后才能执行过期处理。

服务器进程使用缓冲区高速缓存中的缓冲区；而 DBWn 进程通过将更改的缓冲区写回数据文件，使高速缓存中的缓冲区变为可用状态。检查点队列中列出将要写出到磁盘的缓冲区。

Oracle DB 支持同一数据库中有多种块大小。标准块大小用于 SYSTEM 表空间。标准块大小可以通过设置初始化参数 DB\_BLOCK\_SIZE 来指定。其有效值介于 2 KB 到 32 KB 之间，默认值为 8 KB。非标准块大小的缓冲区的高速缓存大小通过以下参数指定：

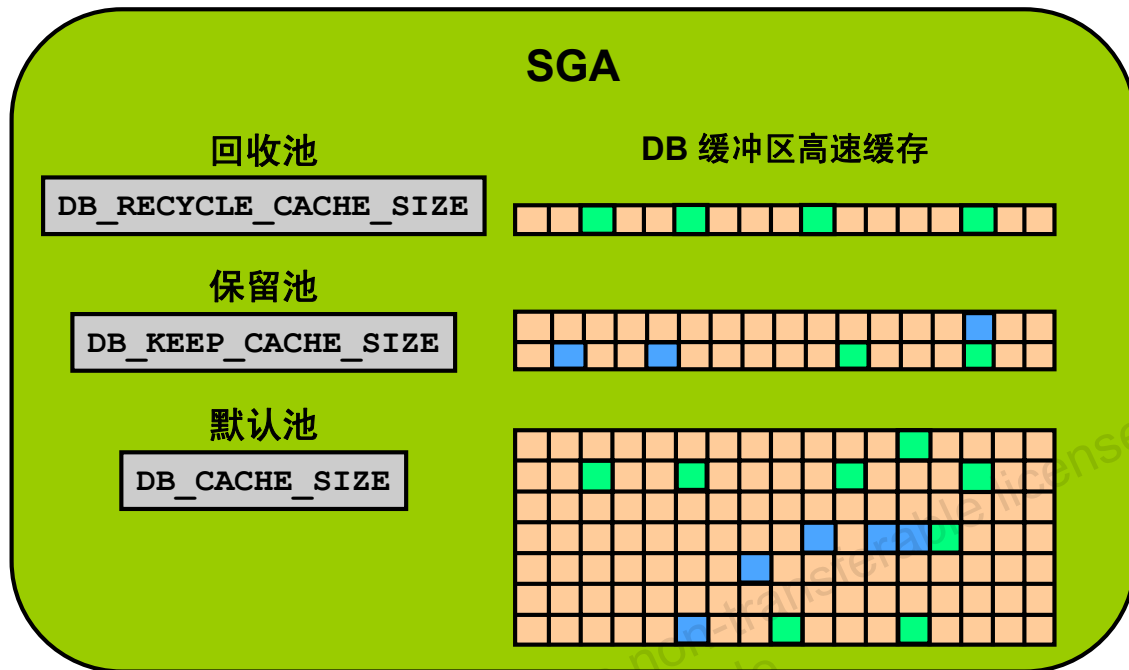
- DB\_2K\_CACHE\_SIZE
- DB\_4K\_CACHE\_SIZE
- DB\_8K\_CACHE\_SIZE
- DB\_16K\_CACHE\_SIZE
- DB\_32K\_CACHE\_SIZE

DB\_nK\_CACHE\_SIZE 参数不能用于调整标准块大小的高速缓存的大小。如果 DB\_BLOCK\_SIZE 的值为 nK，则设置 DB\_nK\_CACHE\_SIZE 是非法的。标准块大小的高速缓存的大小始终由 DB\_CACHE\_SIZE 的值确定。

由于每个缓冲区高速缓存的大小都有限制，因此，通常并非磁盘上的所有数据都能放在高速缓存中。当高速缓存写满时，后续高速缓存未命中会导致 Oracle DB 将高速缓存中已有的灰数据写入磁盘，以便为新数据腾出空间。（如果缓冲区中没有灰数据，则不需要写入磁盘即可将新块读入该缓冲区。）以后若对已写入磁盘的任何数据进行访问，则会导致再次出现高速缓存未命中现象。

数据请求导致高速缓存命中的几率会受到高速缓存大小的影响。高速缓存越大，包含所请求数据的几率也就越大。因此，增加高速缓存大小会提高引起高速缓存命中的数据请求的百分比。

## 使用多个缓冲池



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用多个缓冲池

数据库管理员 (DBA) 可以创建多个缓冲池来提高数据库缓冲区高速缓存的性能。您可以根据对象的访问情况将其分配给某个缓冲池。缓冲池有三种：

- **保留：**此池用于保留内存中可能要重用的对象。将这些对象保留在内存中可减少 I/O 操作。通过使池的大小大于分配给该池的各个段的总大小，可以将缓冲区保留在此池中。这意味着缓冲区不必执行过期处理。保留池可通过指定 `DB_KEEP_CACHE_SIZE` 参数的值来配置。
- **回收：**此池用于内存中重用几率很小的块。回收池的大小要小于分配给该池的各个段的总大小。这意味着读入该池的块经常需要在缓冲区内执行过期处理。回收池可通过指定 `DB_RECYCLE_CACHE_SIZE` 参数的值来配置。
- **默认：**此池始终存在。它相当于没有保留池和回收池的实例的缓冲区高速缓存，可通过 `DB_CACHE_SIZE` 参数进行配置。

**注：**保留池或回收池中的内存不是默认缓冲池的子集。



## 使用多个缓冲池

```
CREATE INDEX cust_idx ...  
  STORAGE (BUFFER_POOL KEEP);  
  
ALTER TABLE oe.customers  
  STORAGE (BUFFER_POOL RECYCLE);  
  
ALTER INDEX oe.cust_lname_idx  
  STORAGE (BUFFER_POOL KEEP);
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用多个缓冲池（续）

`BUFFER_POOL` 子句用于定义对象的默认缓冲池。它是 `STORAGE` 子句的一部分，对 `CREATE` 和 `ALTER` 表、集群和索引语句有效。未明确设置缓冲池的对象中的块将进入默认缓冲池。

语法为：`BUFFER_POOL [KEEP | RECYCLE | DEFAULT]`。

使用 `ALTER` 语句更改对象的默认缓冲池时，已缓存的块会一直保留在其当前缓冲区中，直到正常缓冲区管理活动将它们清除为止。从磁盘读取的块将被放置在为该段新指定的缓冲池中。

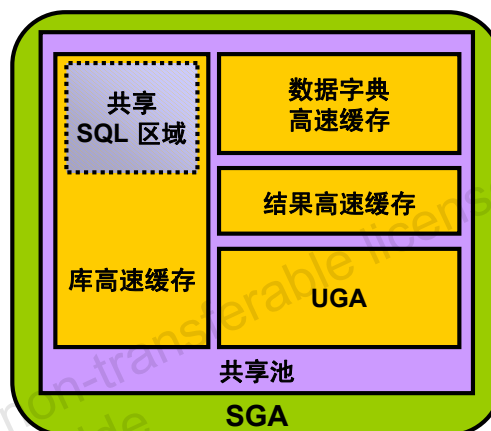
由于多个缓冲池被分配给某一个段，所以有多个段的对象可以将块放置在多个缓冲池中。例如，按索引组织的表在索引段和溢出段上可以有多个不同的池。

## 共享池

内容:

- 库高速缓存: 命令文本、已进行语法分析的代码和执行计划
- 数据字典高速缓存: 数据字典表中各表、列和权限的定义
- 结果高速缓存: SQL 查询和 PL/SQL 函数的结果
- 用户全局区 (UGA): Oracle 共享服务器的会话信息

SHARED\_POOL\_SIZE



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 共享池

可以使用 SHARED\_POOL\_SIZE 初始化参数指定共享池的大小。共享池是用于存储多个会话共享的信息的内存区。它包含不同类型的数据，如幻灯片中的图所示。

**库高速缓存:** 库高速缓存包含共享 SQL 区和 PL/SQL 区 — 经过完全语法分析或编译的 PL/SQL 块和 SQL 语句的表示法。PL/SQL 块包括:

- 过程和函数
- 程序包
- 触发器
- 匿名 PL/SQL 块

**数据字典高速缓存:** 数据字典高速缓存将字典对象的定义存放在内存中。

**结果高速缓存:** 结果高速缓存包括 SQL 查询结果高速缓存和 PL/SQL 函数结果高速缓存。此高速缓存用于存储 SQL 查询或 PL/SQL 函数的结果，以加快它们将来的执行速度。

**用户全局区:** UGA 包含 Oracle 共享服务器的会话信息。使用共享服务器会话时，如果尚未配置大型池，则 UGA 位于共享池中。

## 大型池

- 为以下对象提供大型内存分配：
  - 共享服务器和 Oracle XA 接口的会话内存
  - I/O 服务器进程
  - Oracle DB 备份和还原操作
  - 并行查询操作
  - 高级排队内存表存储
- 减少潜在的共享池碎片
- 是由 AMM 和 ASMM 管理的
- 大小是由 `LARGE_POOL_SIZE` 参数指定的

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 大型池

数据库管理员可以配置一个称为大型池的可选内存区，以便为以下对象提供大型内存分配：

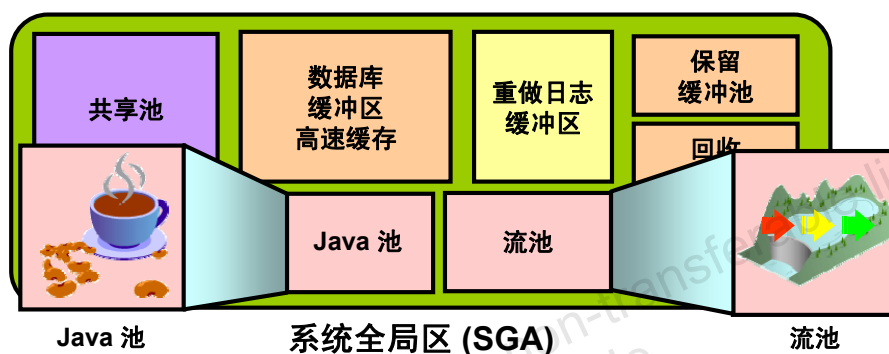
- 共享服务器和 Oracle XA 接口（在事务处理与多个数据库交互时使用）的会话内存
- I/O 服务器进程
- Recovery Manager (RMAN) I/O 从属进程的缓冲区
- 在语句并行执行中使用的消息缓冲区
- 高级排队内存表存储

通过为幻灯片中所列项目分配会话内存，减少了共享池中由于频繁分配和取消分配大对象而产生的碎片。将大对象从共享池中分离出来，可增加共享池内存的使用效率，这意味着，它可以将更多内存用于处理新的请求，以及在需要时用于保留现有数据。

大型池可以由 AMM 和 ASMM 自动管理。还可以用 `LARGE_POOL_SIZE` 参数设置其大小。

## Java 池和流池

- Java 池内存用于将 JVM 中特定于会话的所有 Java 代码和数据存储在服务器内存中。
- Oracle Streams 专门使用流池内存来：
  - 存储已缓冲的队列消息
  - 为 Oracle Streams 进程提供内存



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### Java 池和流池

Java 池内存用于将 JVM 中特定于会话的所有 Java 代码和数据存储在服务器内存中。Java 池内存的使用方式有多种，具体取决于 Oracle DB 运行的模式。

Java 池指导统计信息提供有关用于 Java 的库高速缓存内存的信息，并且预测如何更改 Java 池的大小就可以影响分析速率。将 `statistics_level` 设置为 TYPICAL 或更高级别时，则在内部打开 Java 池指导。关闭指导时，将重置这些统计信息。

流池专门由 Oracle Streams 使用。流池存储已缓冲的队列消息，并且为 Oracle Streams 捕获进程和应用进程提供内存。

除非专门配置流池，否则其大小以零开始。使用 Oracle Streams 时，池大小会根据需要动态地增长。

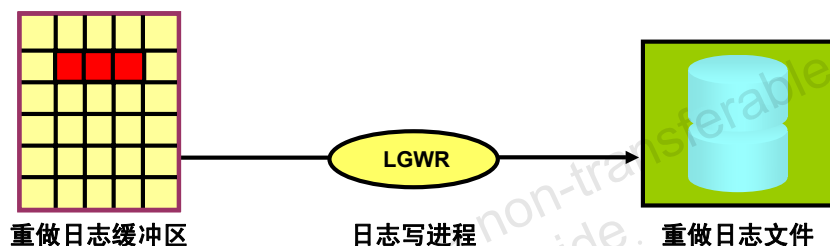
**注：**本课程对 Java 程序设计和 Oracle Streams 不作详细讨论。

## 重做日志缓冲区

- SGA 中的循环缓冲区
- 存放对数据库所做更改的信息
- 包含重做条目，重做条目中具有重做由诸如 DML 和 DDL 操作所做更改的信息

日志写进程 (LGWR) 传送的内容：

- 用户进程提交事务处理时
- 重做日志缓冲区满三分之一时
- DBWn 进程将修改的缓冲区写入磁盘之前



版权所有 © 2011, Oracle. 保留所有权利。

### 重做日志缓冲区

Oracle Server 进程将重做条目从用户的内存空间复制到每个 DML 或 DDL 语句的重做日志缓冲区。重做条目包含重建或重做 DML 和 DDL 操作对数据库的更改所必需的信息。它们用于数据库恢复，需要占用缓冲区中的连续空间。

重做日志缓冲区是一个回收缓冲区；服务器进程可以用新条目覆盖重做日志缓冲区中已写入磁盘的条目。LGWR 进程的写速度通常都很快，足以确保缓冲区中始终有存储新条目的空间。LGWR 进程将重做日志缓冲区写入磁盘上的活动联机重做日志文件（或活动组成员）中。LGWR 进程将 LGWR 上次写入磁盘以来进入缓冲区的所有重做条目复制到磁盘。

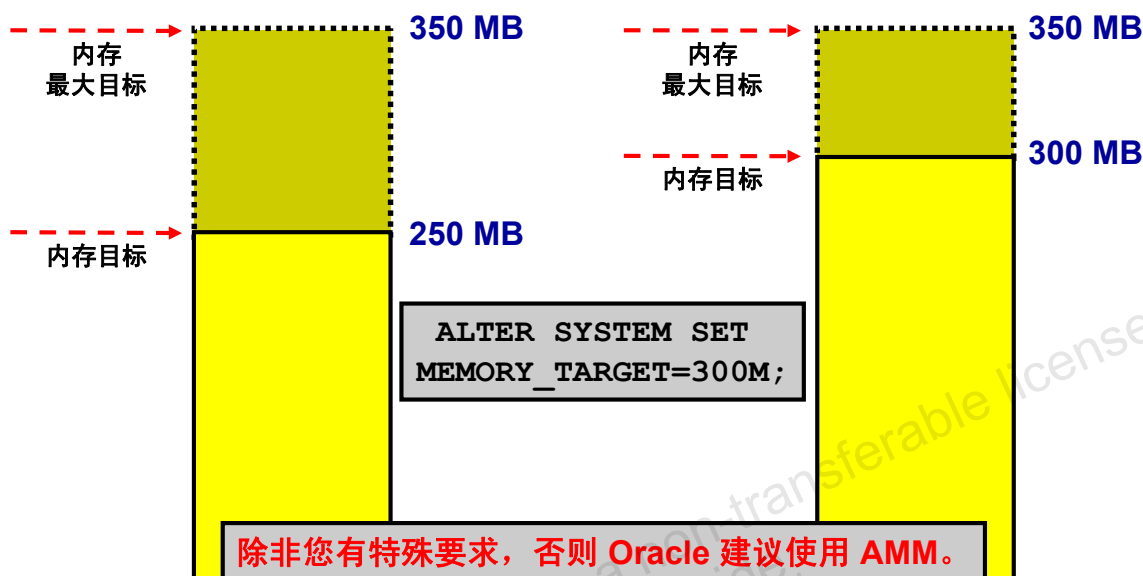
#### 什么导致 LGWR 执行写操作？

在以下情况下，LGWR 会从重做日志缓冲区中写出重做数据：

- 用户进程提交事务处理时
- 每隔三秒钟，或者重做日志缓冲区满三分之一时
- DBWn 进程将修改的缓冲区写入磁盘时（如果相应的重做日志数据尚未写入磁盘）

## 自动内存管理：概览

通过自动内存管理，数据库可以根据工作量自动调整 SGA 和 PGA 的大小。



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 自动内存管理：概览

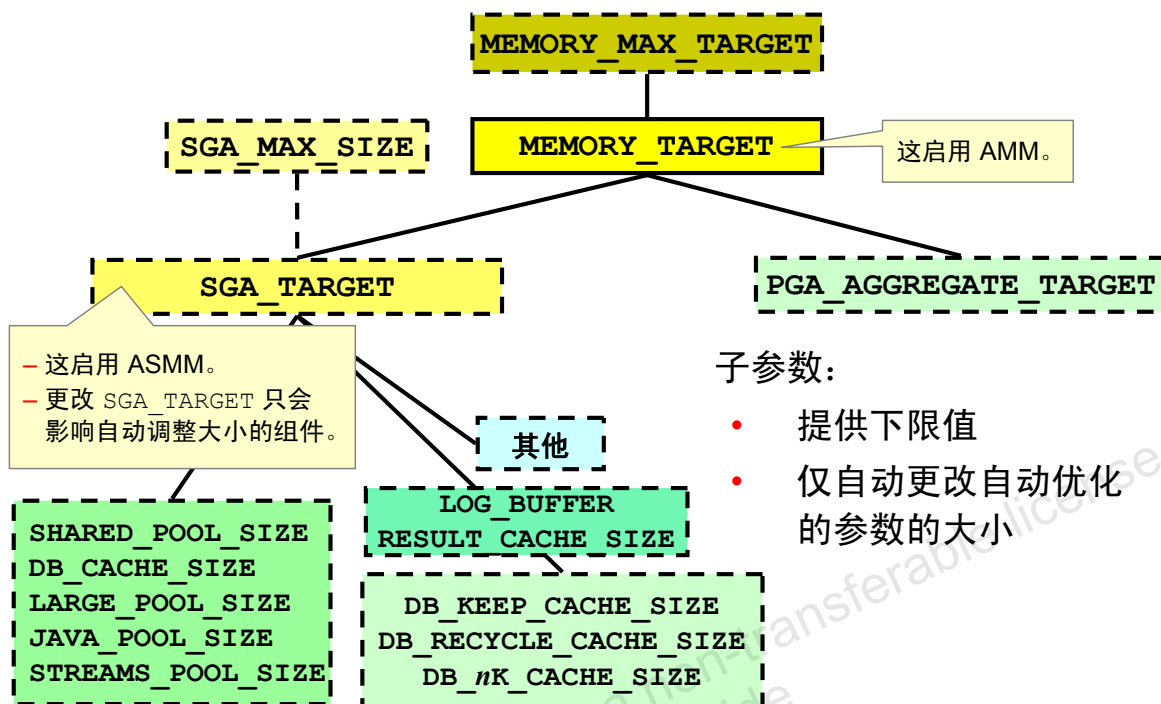
通过自动内存管理 (AMM)，Oracle DB 可以自动管理 SGA 内存以及实例 PGA 内存的大小。为此，在大多数平台上，只需要设置一个目标内存大小初始化参数 (MEMORY\_TARGET) 和一个最大内存大小初始化参数 (MEMORY\_MAX\_TARGET)，数据库就会根据处理需求在 SGA 与实例 PGA 之间动态交换内存。通过导航到“Server > Memory Advisors (服务器 > 内存指导)”（在“Database Configuration (数据库配置)”区域中），并单击“Enable (启用)”按钮，可启用 Oracle Enterprise Manager 中的 AMM。

通过这种内存管理方法，数据库还可以动态调整单个 SGA 组件的大小以及单个 PGA 的大小。

因为目标内存初始化参数是动态的，因此可以随时更改目标内存大小而不必重新启动数据库。最大内存大小相当于一个上限，以防您无意中将目标内存大小设置得太高。因为某些 SGA 组件的大小不容易收缩，或者其大小必须不低于某个下限值，所以数据库还要防止您将目标内存大小设置得太低。

这种间接的内存转移依赖于操作系统 (OS) 的共享内存释放机制。将内存释放给 OS 后，其它组件可以通过向 OS 请求内存来分配内存。目前，Linux、Solaris、HPUX、AIX 和 Windows 平台上已实施了自动内存管理。

## Oracle DB 内存参数



版权所有 © 2011, Oracle. 保留所有权利。

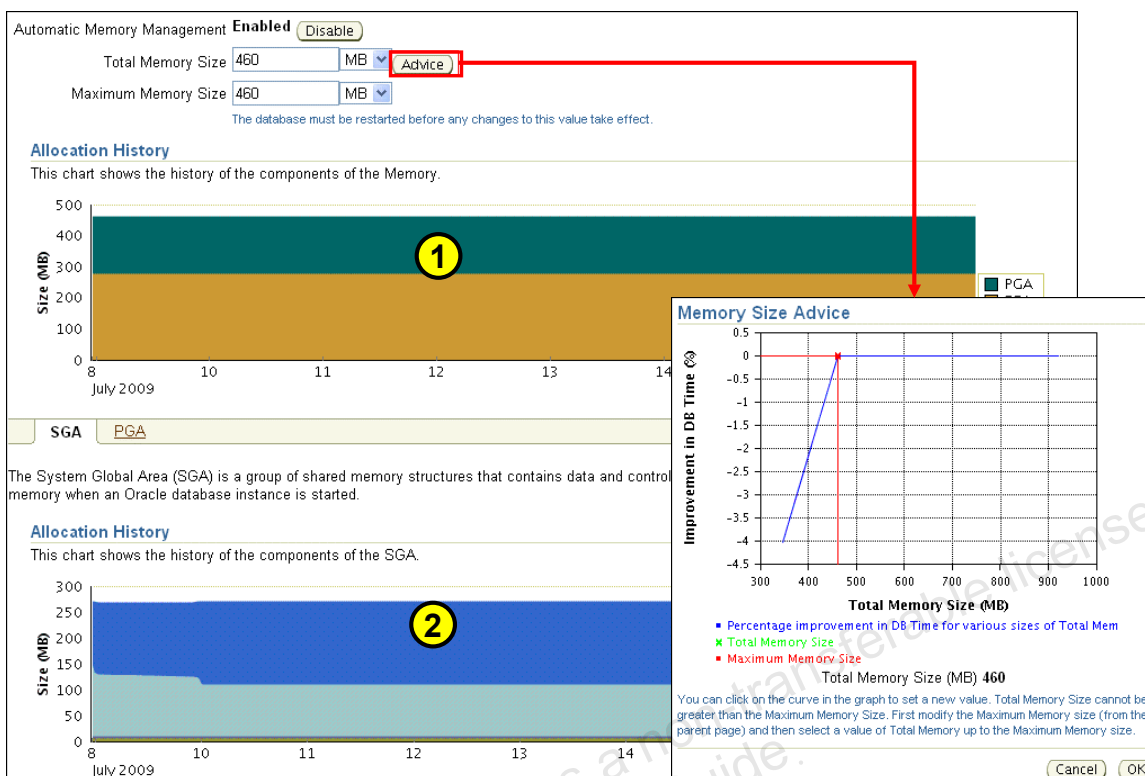
### Oracle DB 内存大小设置参数

幻灯片中的图展示了内存初始化参数的层次结构。虽然仅需要设置 MEMORY\_TARGET 来触发自动内存管理，但仍可以为各种高速缓存设置下限值。因此，如果子参数是用户设置的，则这些参数值将是 Oracle DB Server 自动优化该组件时的下限值。

- 如果将 SGA\_TARGET 和 PGA\_AGGREGATE\_TARGET 设置为非零值，则可将其分别视为 SGA 和 PGA 大小的下限值。MEMORY\_TARGET 可以采用从 SGA\_TARGET + PGA\_AGGREGATE\_TARGET 到 MEMORY\_MAX\_SIZE 的值。
- 如果设置了 SGA\_TARGET，则数据库将仅自动优化 SGA 的子组件的大小。PGA 的自动优化与是否显式设置 PGA 无关。但是，不会自动优化整个 SGA (SGA\_TARGET) 和 PGA (PGA\_AGGREGATE\_TARGET)，即不自动增长或收缩。



## 监视自动内存管理



版权所有 © 2011, Oracle。保留所有权利。

### 监视自动内存管理

在 EM 主页（“Related Links（相关链接）”部分）中，导航到“Advisor Central > Memory Advisors（指导中心 > 内存指导）”。此时，幻灯片中将显示“Memory Advisors（内存指导）”页。

启用了自动内存管理后，可以在“Memory Advisors（内存指导）”页的“Allocation History（分配历史记录）”部分看到以图形方式显示的内存大小组件历史记录。第一个矩形图的上部为可优化的那部分 PGA，其下部是所有 SGA。第二个矩形图的上部为共享池大小，其下部对应于缓冲区高速缓存大小。

在此页上，还可以通过单击“Advice（建议）”按钮访问内存目标指导。此指导将提供各种内存总大小可能实现的 DB 时间改善。

注：您也可以使用 V\$MEMORY\_TARGET\_ADVISOR 视图查看内存目标指导。



## 监视自动内存管理

如果要通过命令行监视自动内存管理做出的决定：

- V\$MEMORY\_DYNAMIC\_COMPONENTS 包含所有内存组件的当前状态
- V\$MEMORY\_RESIZE\_OPS 包含最近完成的 800 个内存大小调整请求的循环历史记录缓冲区
- V\$MEMORY\_TARGET\_ADVICE 提供针对 MEMORY\_TARGET 初始化参数的优化建议

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 监视自动内存管理（续）

动态性能视图 V\$MEMORY\_DYNAMIC\_COMPONENTS 显示所有动态优化的内存组件的当前大小，其中包括 SGA 和实例 PGA 的总大小。V\$MEMORY\_TARGET\_ADVICE 视图提供针对 MEMORY\_TARGET 初始化参数的优化建议。

查看 V\$MEMORY\_TARGET\_ADVICE 视图时，MEMORY\_SIZE\_FACTOR 为 1 的行显示当前的内存大小（由 MEMORY\_TARGET 初始化参数设置）以及完成当前工作量所需的 DB 时间量。在之前以及之后的行中，将显示使用一组替代 MEMORY\_TARGET 大小的结果。如果将 MEMORY\_TARGET 参数更改为替代大小中的任何一个，那么数据库将显示大小因子（当前大小的乘数）以及完成当前工作量的估计 DB 时间。请注意，如果内存总大小小于当前 MEMORY\_TARGET 大小，则估计的 DB 时间会增加。

## 有效使用内存：准则

- 使 SGA 适合物理内存。
- 优化以实现高缓冲区高速缓存命中率，但要注意以下几点：
  - 即使有效且必需的全表扫描也会降低命中率。
  - 可能存在因不必要地重复读取同一块而出现命中率虚升的情况。
- 使用内存指导。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 有效使用内存：准则

如果可能，最好使 SGA 适合物理内存，以便提供最快的访问速度。即使操作系统可能提供额外的虚拟内存，该内存也经常会因其性质而换出到磁盘。在某些平台上，可以使用 LOCK\_SGA 初始化参数将 SGA 锁定到物理内存中。此参数不能与 AMM 或 ASMM 一起使用。

执行 SQL 语句时，将请求数据块进行读、写或读写操作。这被认为是一个逻辑 I/O。请求某个块时，会先查看它是否已在内存中。如果不在内存中，则从磁盘中读取块，这称为物理 I/O。在内存中发现块的次数与逻辑 I/O 的总次数之比，称为缓冲区高速缓存命中率。通常，命中率越高越好，因为这意味着在内存中找到了更多的块，而不必进行磁盘 I/O 操作。

缓冲区高速缓存命中率高于 99% 的情况不罕见，但这并不总表示系统优化好了。如果某个查询的执行频率超过了必要的次数，而且它不断地反复请求相同的块，则命中率就会提高。如果这是个低效或不必要的查询，则会虚升命中率。这是因为它首先不应以这种方式执行或不应如此频繁地执行。

**有效使用内存：准则（续）**

此外，考虑到大型全表扫描（完全读取整个表）会降低此命中率，因为可能要从磁盘中读取整个表；此类扫描可能不会利用某些已在缓冲区高速缓存中的块。所以，如果应用程序中有一些必要的大型全表扫描，则即使是优化良好的数据库，其数据库缓冲区高速缓存命中率也可能始终很低。

使用 Oracle Enterprise Manager 内存指导。这些指导可以帮助您基于特定数据库中的活动调整 SGA 的大小。

## 库高速缓存的内存优化准则

- 为开发人员制定格式使用约定，以便 SQL 语句符合高速缓存的要求。
- 使用绑定变量。
- 消除不必要的重复 SQL。
- 考虑使用 CURSOR\_SHARING。
- 尽可能使用 PL/SQL。
- 缓存序列号。
- 连接库高速缓存中的对象。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 库高速缓存的内存优化准则

库高速缓存是共享池的一部分，它是 Oracle DB 用于存储所有 SQL、Java 代码、PL/SQL 过程和程序包以及控制结构（如锁定和库高速缓存句柄）的地方。这些代码进入此中央位置的目的是为了能够让所有用户共享。共享的好处在于，所有用户都能利用 SQL 代为执行的工作。因此，对于每条语句，不论它执行多少次，也不管有多少用户执行它，对语句进行语法分析和确定数据访问路径（也称为“解释计划”）之类的任务都只执行一次。如果库高速缓存过小，则没有空间容纳所有要执行的语句，因此对于某些语句，也就无法利用此工作共享的优势。如果库高速缓存过大，又会给系统带来管理其内容的负担。

因为库高速缓存中最终可能会填满一些看似不同、实际却是同一语句副本的语句。导致此问题的一个常见原因是每条语句的格式稍有不同。如果字符串不完全比较就没有匹配项。另一个原因是使用了文字而不是绑定变量。当两条语句之间的唯一差别是文字值时，如果用绑定变量替换这些文字，则大多数情况下，这些语句的每次执行和整个系统都将受益。

可以设置 CURSOR\_SHARING 初始化参数，指示系统在语句的其它部分都匹配时自动用绑定变量替换文字。通常，在适当时使用绑定变量更正应用程序之前，您应将此设置作为临时手段。与所有这些准则一样，使用此变量也可能有其它负面影响，这一点您应该调查清楚。

**库高速缓存的内存优化准则（续）**

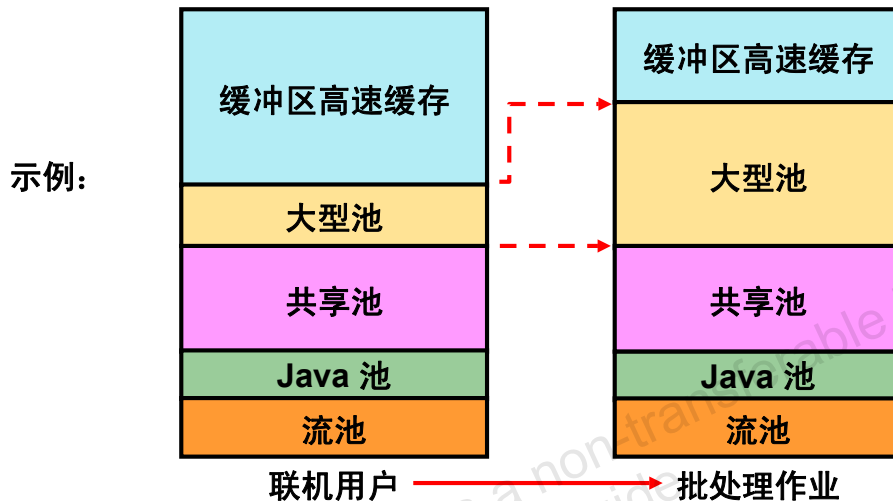
避免从应用程序中的几个不同位置发出同一 SQL 语句，而是使用 PL/SQL 将一条或多条语句放置在一个存储过程中。以后只需调用该存储过程。这样可保证该 SQL 语句是共享的，因为它只位于一个地方。同样，该 SQL 语句也已经过语法分析并有了解释计划，因为它位于一个已编译的存储过程中。

序列号可以缓存。因此，如果某些序列的活动频繁，请为其设置一个合适的高速缓存大小，然后利用它。

可以使用 DBMS\_SHARED\_POOL 程序包来连接库高速缓存中的对象。这样可减少重新加载和重新编译对象的几率。有关如何使用该程序包的详细信息，请参阅《PL/SQL Packages and Types Reference》一文。

## 自动共享内存管理：概览

- 自动根据工作量变化调整
- 最大程度地提高内存利用率
- 有助于消除内存不足的错误



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 自动共享内存管理：概览

如果 AMM 不能正常工作，因为需要一个固定的 PGA，请考虑使用可简化 SGA 内存管理的自动共享内存管理 (ASMM)。可以使用 `SGA_TARGET` 初始化参数指定实例的可用 SGA 内存总量，然后，Oracle DB 会自动在各个 SGA 组件间分配该内存，从而确保内存的高效利用。

例如，对于白天运行大型联机事务处理 (OLTP) 作业（要求大型缓冲区高速缓存）和夜晚运行并行批处理作业（要求大型池的内存空间比较大）的系统，就必须同时配置缓冲区高速缓存和大型池，以便适应峰值需求。

有了 ASMM 功能，当 OLTP 作业运行时，缓冲区高速缓存会获取大部分内存来保证良好的 I/O 性能。以后启动数据分析和报告批处理作业时，内存又会自动迁移到大型池，供并行查询操作使用，而不会产生内存溢出错误。

如果使用的是服务器参数文件 (SPFILE)，则 Oracle DB 会在每次关闭实例时记住自动优化的组件的大小。因此，系统不需要在每次启动实例时都重新了解工作量的特性。它可以利用从以前的实例中获取的信息，从上次关闭时中断的位置开始继续评估工作量。

## ASMM 的工作原理

- ASMM 以 MMON 在后台捕获的工作量信息为基础。
- MMON 使用内存指导。
- 将内存移到 MMAN 最迫切需要的地方。
- 如果使用 SPFILE（推荐）：
  - 在关闭时保存组件大小
  - 保存的值用于引导程序组件大小
  - 无需再确定最佳值

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### ASMM 的工作原理

自动共享内存管理功能使用 SGA 内存中介，此内存中介由可管理性监视器 (MMON) 和内存管理器 (MMAN) 这两个后台进程实施。统计信息和内存指导数据由 MMON 定期在内存中捕获。MMAN 根据 MMON 决策来协调内存组件的大小。SGA 内存中介会不断跟踪组件的大小和待处理的大小调整操作。

SGA 内存中介会观察系统和工作量，以便确定理想的内存分配方案。SGA 内存中介每隔几分钟就执行一次这种检查，使内存始终用在需要的地方。如果没有自动共享内存管理功能，必须分别预计各组件在峰值时的内存需求，然后对其内存大小进行调整。

在工作量信息基础上，自动共享内存管理功能会：

- 定期在后台捕获统计信息
- 使用内存指导
- 进行假设分析，确定最佳内存分配方案
- 将内存移到最迫切需要的地方
- 如果使用了 SPFILE，则在关闭时保存组件大小（这些大小可以在最后一次关闭前重新起用）

## 启用自动共享内存管理功能

要从手动共享内存管理模式下启用 ASMM，请执行以下操作：

1. 获取 **SGA\_TARGET** 的值：

```
SELECT ((SELECT SUM(value) FROM V$SGA) - (SELECT CURRENT_SIZE
FROM V$SGA_DYNAMIC_FREE_MEMORY)) "SGA_TARGET" FROM DUAL;
```

2. 使用该值设置 **SGA\_TARGET**。
3. 将自动设置大小的 SGA 组件的值设置为 0。

要从自动内存管理模式切换到 ASMM，请执行以下操作：

1. 将初始化参数 **MEMORY\_TARGET** 设置为 0。
2. 将自动设置大小的 SGA 组件的值设置为 0。

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 启用自动共享内存管理功能

启用 ASMM 的具体过程取决于是从手动共享内存管理模式更改为 ASMM，还是从自动内存管理模式更改为 ASMM。要从手动共享内存管理模式更改为 ASMM，请执行以下操作：

1. 运行以下查询获取 **SGA\_TARGET** 的值：

```
SELECT ((SELECT SUM(value) FROM V$SGA) - (SELECT CURRENT_SIZE FROM
V$SGA_DYNAMIC_FREE_MEMORY)) "SGA_TARGET" FROM DUAL;
```

2. 设置 **SGA\_TARGET** 的值：

```
ALTER SYSTEM SET SGA_TARGET=value [SCOPE={SPFILE|MEMORY|BOTH}]
```

其中 *value* 是在步骤 1 中计算的值，或者是介于所有 SGA 组件的总大小与 **SGA\_MAX\_SIZE** 之间的某个值。

3. 将自动设置大小的 SGA 组件的值设置为 0。为此，需要编辑文本初始化参数文件，或者发出 ALTER SYSTEM 语句。如果需要，重新启动实例。



## 启用自动共享内存管理功能（续）

要从自动内存管理模式更改为 ASMM，请执行以下操作：

1. 将初始化参数 MEMORY\_TARGET 设置为 0。

```
ALTER SYSTEM SET MEMORY_TARGET = 0;
```

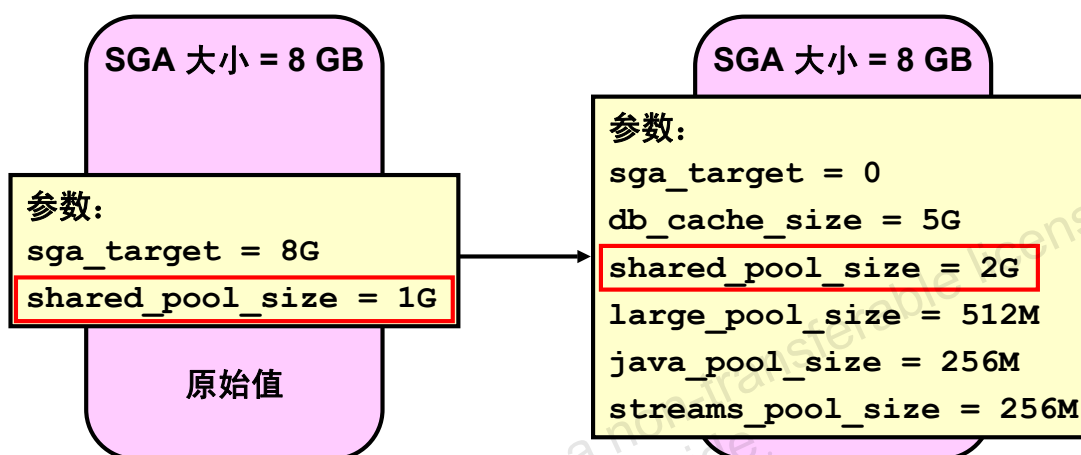
数据库将根据当前的 SGA 内存分配设置 SGA\_TARGET。

2. 将自动设置大小的 SGA 组件的值设置为 0。完成后，重新启动实例。

**注：**本课程稍后将讨论自动内存管理。

## 禁用 ASMM

- 将 `SGA_TARGET` 设置为 0 可禁用自动优化功能。
- 自动优化的参数设置为其当前大小。
- SGA 大小总体上不受影响。



ORACLE

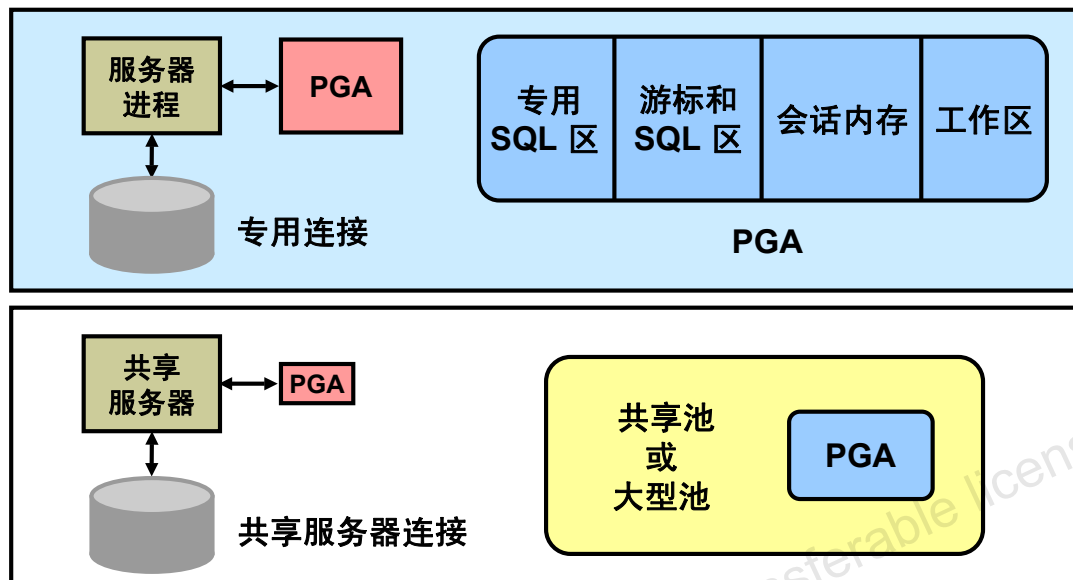
版权所有 © 2011, Oracle. 保留所有权利。

### 禁用 ASMM

通过将 `SGA_TARGET` 设置为 0，可以动态选择禁用自动共享内存管理。在此情况下，所有自动优化的参数的值都将设置为其对应组件的当前大小；即使用户早先为自动优化的参数指定了其它非零值，也是如此。

在幻灯片上的示例中，`SGA_TARGET` 的值为 8 GB，`SHARED_POOL_SIZE` 的值为 1 GB。如果系统将共享池组件的大小内部调整为 2 GB，则将 `SGA_TARGET` 设置为 0 会导致 `SHARED_POOL_SIZE` 被设置为 2 GB，从而覆盖用户定义的原有值。

## 程序全局区 (PGA)



默认情况下，启用自动 PGA 内存管理。

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 程序全局区 (PGA)

程序全局区 (PGA) 是包含某服务器进程的数据及控制信息的内存区。这是 Oracle Server 在服务器进程启动时创建的非共享内存，只有该服务器进程才能访问。由关联到某个 Oracle 实例的所有服务器进程分配的 PGA 总内存，也称为该实例分配的聚集 PGA 内存。使用共享服务器时，部分 PGA 可位于 SGA 中。

PGA 内存通常包含以下各项：

#### 专用 SQL 区

专用 SQL 区包含绑定信息和运行时内存结构等数据。这些信息是每个会话的 SQL 语句调用所特有的；在其它方面，绑定变量有不同的值，游标的状态也不同。发出 SQL 语句的每个会话都有一个专用 SQL 区。提交同一 SQL 语句的每个用户也都有其自己的专用 SQL 区，该专用 SQL 区使用一个共享 SQL 区。这样，许多专用 SQL 区可与同一个共享 SQL 区关联。专用 SQL 区的位置取决于为会话建立的连接类型。如果会话是通过专用服务器连接的，则专用 SQL 区位于该服务器进程的 PGA 中。不过，如果会话是通过共享服务器连接的，则部分专用 SQL 区将保留在 SGA 中。

## 程序全局区 (PGA) (续)

### 游标和 SQL 区

Oracle Pro\*C 程序或 Oracle 调用接口 (OCI) 程序的应用程序开发人员可以显式打开特定专用 SQL 区的游标或句柄，并在该程序的整个执行过程中将它们用作命名资源。数据库为某些 SQL 语句隐式发出的递归游标也使用共享 SQL 区。

### 工作区

对于复杂查询（例如，决策支持查询），会将大部分 PGA 供内存密集型运算符分配的工作区专用，例如：

- 基于排序的运算符（如 ORDER BY、GROUP BY 和 ROLLUP）和窗口函数
- 散列联接
- 位图合并
- 位图创建
- 批量加载操作使用的写缓冲区

排序运算符使用工作区（排序区），对一组行执行内存中排序。与此类似，散列联接运算符使用工作区（散列区），根据其左侧输入内容生成散列表。

工作区的大小是可以控制和优化的。通常，较大的工作区可以显著改进特定运算符的性能，不过代价是消耗较多的内存。

### 会话内存

会话内存是用于存放会话的变量（登录信息）以及与会话相关的其它信息的内存。对于共享服务器，会话内存是共享的，而不是专用的。

### 自动 PGA 内存管理

默认情况下，Oracle DB 会自动对供实例 PGA 专用的内存总量进行全局管理。可以通过设置初始化参数 PGA\_AGGREGATE\_TARGET 来控制此内存量。随后，Oracle DB 会努力确保分配给所有数据库服务器进程和后台进程的 PGA 内存总量始终不超过此目标值。

## 使用 V\$PARAMETER 视图

```
SGA_TARGET = 8G
```

```
DB_CACHE_SIZE = 0  
JAVA_POOL_SIZE = 0  
LARGE_POOL_SIZE = 0  
SHARED_POOL_SIZE = 0  
STREAMS_POOL_SIZE = 0
```

```
SELECT name, value, isdefault  
FROM v$parameter  
WHERE name LIKE '%size';
```

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 使用 V\$PARAMETER 视图

如果为 SGA\_TARGET 指定非零值，且未指定自动优化的 SGA 参数的值，则 V\$PARAMETER 视图中自动优化的 SGA 参数的值为 0，ISDEFAULT 列的值为 TRUE。

如果已为任何自动优化的 SGA 参数指定了值，则查询 V\$PARAMETER 时显示的值是您为该参数指定的值。

## 测验

为了获得最佳性能，应该通过设置 MEMORY\_TARGET 和 SGA\_TARGET 参数来启用自动内存管理 (AMM) 和自动共享内存管理 (ASMM)。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2

## 小结

在本课中，您应该已经学会：

- 描述 SGA 中的内存组件
- 实施自动内存管理
- 手动配置 SGA 参数
- 使用自动 PGA 内存管理

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 13 概览：使用 AMM 纠正内存分配问题

本练习包含以下主题：

- 诊断内存分配问题
- 启用和实施自动内存管理

ORACLE

版权所有 © 2011, Oracle。保留所有权利。



# 14

## 管理数据库性能

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

华周 (286991486@qq.com) has a non-transferable license to use this Student Guide.

## 课程目标

学完本课后，应能完成以下工作：

- 监视会话和服务的性能
- 描述数据库重放的优点

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 优化活动

性能管理中的三项活动为：

- 性能规划
- 实例优化
- SQL 优化



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 优化活动

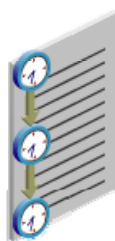
优化活动包括三个方面，即性能规划、实例优化和 SQL 优化。

- 性能规划是建立环境的过程，包括硬件、软件、操作系统、网络基础结构，等等。
- 实例优化是对 Oracle DB 参数和操作系统 (OS) 参数进行实际调整，从而使 Oracle DB 获得更好的性能。
- SQL 优化涉及的工作包括使应用程序提交有效的 SQL 语句。SQL 优化可针对整个应用程序执行，也可针对单条语句执行。在应用程序级别，需要确保应用程序的各个部分都能够利用彼此的工作，并且不会产生不必要的资源竞争。在本课中，您将学习用于优化特定 SQL 语句的常见操作。

**注：**有关性能优化的详细信息，请参阅《Oracle Database Performance Tuning Guide》。

## 性能规划

- 投资选项
- 系统体系结构
- 可伸缩性
- 应用程序设计原理
- 工作量测试、建模和实施
- 部署新的应用程序



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 性能规划

性能规划包括许多方面。规划时必须权衡性能（速度）、开销和可靠性。必须考虑在系统体系结构方面的投资：满足需求所需的硬件和软件基础结构。当然，这需要通过分析来确定适合于给定环境、应用程序和性能需求的值。例如，硬盘驱动器和控制器的数量会影响数据的访问速度。

应用程序的伸缩能力也是非常重要的。这意味着能够处理越来越多的用户、客户机、会话或事务处理，而不会对系统整体性能产生大的影响。限制可伸缩性的最明显因素是用户间的串行化操作。如果所有的用户同时都经过同一路径，则在添加更多用户时，毫无疑问地会对性能造成负面影响。这是因为会有越来越多的用户排队等待通过该路径。编写的 SQL 语句质量差也会影响可伸缩性。这会使许多用户都在等待低效 SQL 的完成；每个用户都在与其他用户竞争大量资源，但实际上他们并不需要这些资源。

应用程序的设计原理会对性能产生重大影响。设计的简洁性、视图和索引的使用以及数据建模都是非常重要的。

任何应用程序都必须在典型的生产工作量下进行测试。这需要对数据库大小和工作量进行估计，并生成测试数据和系统负载。

**性能规划（续）**

部署新的应用程序（或应用程序的新版本）时必须考虑到性能。有时，需要在推广新应用程序的过程中制定设计决策使其与旧系统保持兼容。新的数据库应该根据生产环境，为其承载的应用程序专门进行配置。

一项困难但必需完成的任务是在更改基础结构时测试现有的应用程序。例如，将数据库升级到更高版本，或者更换操作系统或服务器硬件。以新的配置将应用程序部署到生产中以前，您需要了解这样做的影响。一般几乎都需要对应用程序进行额外的优化。您需要确定关键的功能会正常运行，不会发生错误。

## 实例优化

- 有定义合理的目标。
- 将内存分配至数据库结构。
- 考虑数据库中各部分的 I/O 要求。
- 优化操作系统以获得最佳数据库性能。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 实例优化

在开始任何优化活动时，都必须有具体的目标。例如像“每分钟处理 500 个销售事务”这样的目标要比“使处理速度尽可能地快，然后就可以知道要达到什么程度”这类目标更容易实现。

必须为应用程序分配合适的 Oracle DB 内存，以获得最佳性能。可以使用的内存是有限的。为 Oracle DB 的某些部分分配的内存太少会导致后台操作的效率很低。如果不进行分析，您甚至可能都意识不到这个问题。

磁盘 I/O 通常会成为数据库的瓶颈，因此，任何数据库实施的开始阶段都需要特别注意。操作系统的配置也会影响 Oracle DB 的性能。有关详细信息，请参阅相应平台的《Oracle Database Installation Guide》。

## 性能优化方法

### 优化步骤：

- 自上而下进行优化。优化：
  1. 设计
  2. 应用程序代码
  3. 实例
- 对可以带来最大潜在好处的方面进行优化。确定和优化：
  - 使用资源最多的 SQL
  - 时间最长的等待
  - 最长的服务时间
- 达到目标时停止优化。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 性能优化方法

Oracle 根据多年的经验开发了一种优化方法。基本步骤如下：

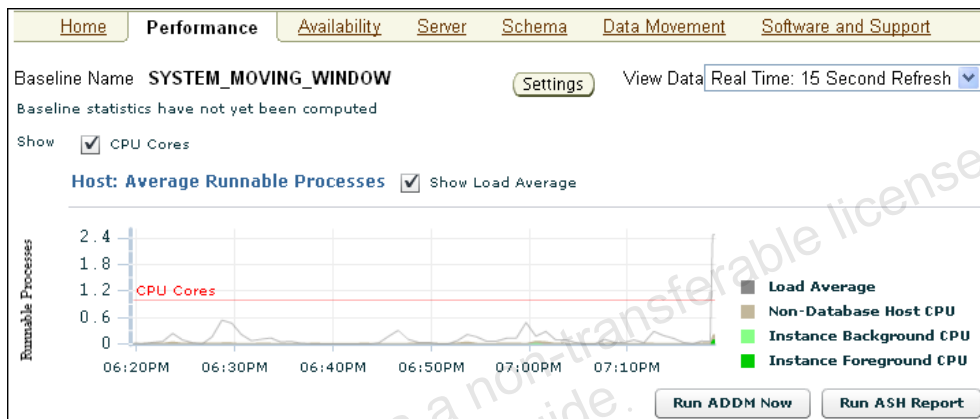
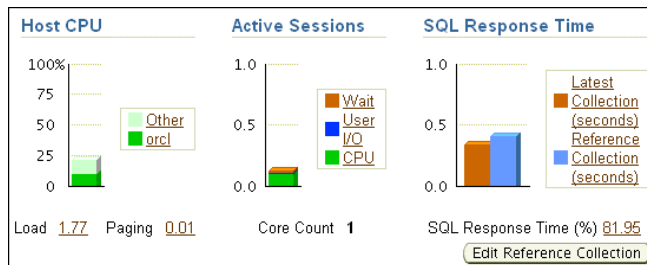
1. 在优化实例之前检查 OS 统计信息以及一般的计算机运行状况，以确保问题出现在数据库中。使用 Oracle Enterprise Manager 数据库主页。
2. 自上而下进行优化。从优化设计开始，然后优化应用程序，再优化实例。例如，在优化磁盘上的表空间布局之前，请尽量避免造成 I/O 争用的全表扫描。此活动通常需要访问应用程序代码。
3. 对可以带来最大潜在好处的方面进行优化。本课程中介绍的优化方法非常简单。确定最大的瓶颈并对其进行优化。重复此步骤。各种优化工具都有一些方法来确定占用时间最多的 SQL 语句、资源争用或服务。Oracle DB 提供时间模型和度量来自动确定瓶颈。Oracle Database 11g 中提供的指导严格地使用此方法。
4. 达到目标时停止优化。此步骤隐含的前提是您已设置了优化目标。

这是优化数据库实例的常用方法，可能需要进行多个来回。

## 性能监视

使用 Oracle Enterprise Manager 可以：

- 提供性能概览
- 以图形方式显示度量和详细资料



版权所有 © 2011, Oracle. 保留所有权利。

### 性能监视

只有知道性能已发生变化，才能响应性能的变化。Oracle Database 11g 提供了多种方式来监视数据库实例的当前性能。通过 Oracle Enterprise Manager (EM) 的数据库主页可以快速检查实例和服务器的健康状况，并查看用图形显示的 CPU 使用率、活动会话以及 SQL 响应时间。该主页还能显示已触发的所有预警。

EM 中的“Performance（性能）”选项卡通过多个图形从多个角度显示性能度量。您可以从 CPU、平均活动会话数、吞吐量、I/O 以及其它方面了解性能。在“Performance（性能）”页中，可以使用链接访问详细信息：包括会话和各条 SQL 语句。

EM 中显示的信息基于数据库中存在的性能视图。可以使用 SQL\*Plus 直接访问这些视图。有时，您可能需要访问这些视图以获取有关原始统计信息的一些详细资料。



## 性能优化数据

收集的数据类型：

- 累计统计信息：
  - 包含时间信息的等待事件
  - 时间模型
- 度量：统计率
- 抽样统计信息：活动会话历史记录
  - 按会话显示统计信息
  - 按 SQL 显示统计信息
  - 按服务显示统计信息
  - 其它维



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 性能优化数据

Oracle DB Server 软件可以获取与自身运行有关的信息。收集的数据主要有三种：累计统计信息、度量和抽样统计信息。

累计统计信息是数据库服务器中发生的各种事件的计数信息和计时信息。有些信息非常重要，如缓冲区忙等待。有些信息对优化几乎没有影响，例如索引块拆分。对于优化而言，最重要的事件通常是显示的累计时间值最大的事件。Oracle Database 11g 中的统计信息通过使用时间模型进行关联。时间模型统计信息基于数据库时间的百分比，为进行比较提供了一个共同的基础。

度量是每单位的统计计数。单位可能是时间（如秒）、事务处理或会话。度量为主动监视性能提供了一个基础。可以为导致生成预警的度量设置阈值。例如，可以将阈值设置为每毫秒的读取次数超过以前记录的峰值时，或归档日志区已占用 95% 时。

如果将 STATISTICS\_LEVEL 设置为 TYPICAL 或 ALL，则将自动收集抽样统计信息。利用抽样统计信息可以及时进行回顾。可以在不同的维查看过去收集的会话和系统统计信息，即使事先未考虑过指定对这些维收集数据。

## 优化统计信息收集

- SQL 性能优化：取决于能够收集到准确的统计信息
- 优化程序统计信息：
  - 对象统计信息
  - 操作系统统计信息
- 收集统计信息的方式：
  - 自动：自动维护任务
  - 手动：DBMS\_STATS 程序包
  - 通过设置数据库初始化参数进行收集
  - 通过从另一个数据库导入统计信息进行收集

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 优化统计信息收集

优化程序统计信息是有关数据库对象的特定详细资料的数据集合。查询优化程序必须使用这些统计信息来为每个 SQL 语句选择最佳的执行计划。会定期收集这些统计信息，在两次收集间隔之内数据不会发生变化。

建议您允许 Oracle DB 自动收集统计信息，以此来收集优化程序统计信息。自动维护任务是在创建数据库时自动创建的，并由调度程序进行管理。默认情况下，它会收集数据库中优化程序统计信息缺失或已过时的所有对象的统计信息。可以通过“Automatic Maintenance Tasks（自动维护任务）”页更改默认配置。

系统统计信息为查询优化程序描述系统的硬件特征，例如 I/O 及 CPU 性能和利用率。选择执行计划时，优化程序会估计每个查询所需的 I/O 和 CPU 资源。通过系统统计信息，查询优化程序可以更加准确地估计 I/O 和 CPU 开销，从而选择更好的执行计划。系统统计信息是使用 DBMS\_STATS.GATHER\_SYSTEM\_STATS 过程收集的。收集系统统计信息时，Oracle DB 对指定时段内的系统活动进行分析。系统统计信息不是自动收集的。Oracle Corporation 建议您使用 DBMS\_STATS 程序包来收集系统统计信息。

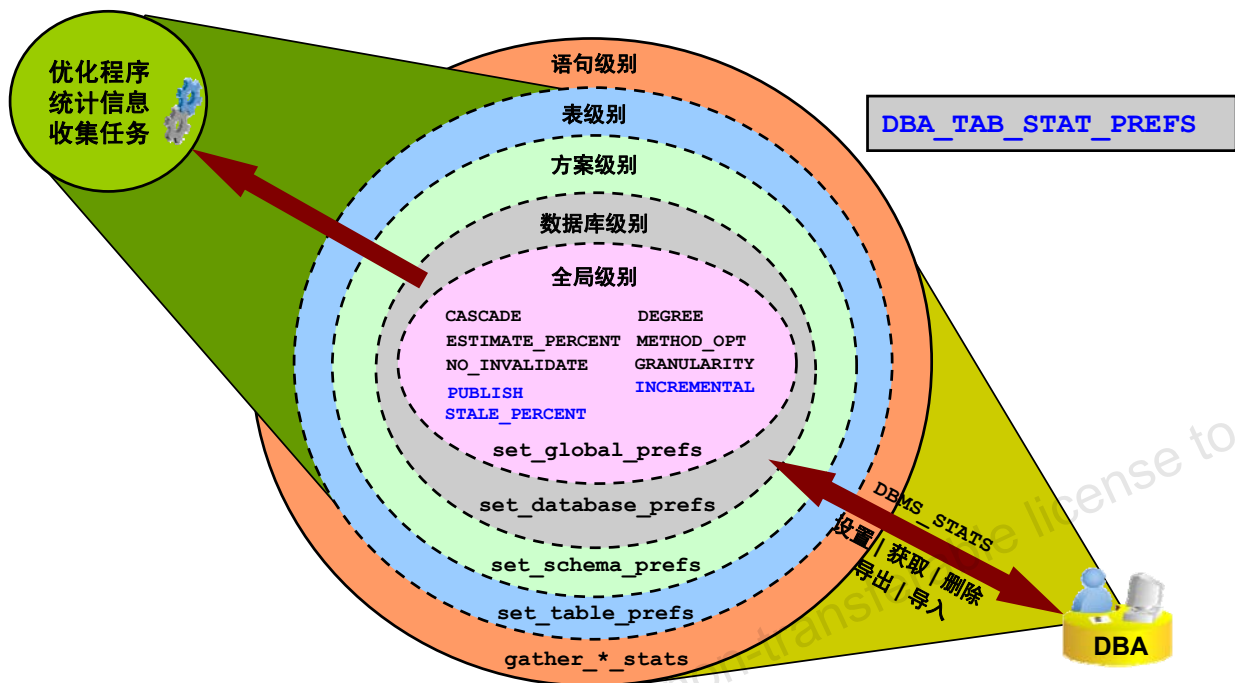
## 优化程序统计信息收集（续）

如果选择不使用统计信息自动收集功能，则必须手动收集所有方案中的统计信息，包括系统方案。如果数据库中的数据定期发生更改，则您也需要定期收集统计信息，以确保统计信息能够准确地反映数据库对象的特征。要手动收集统计信息，请使用 DBMS\_STATS 程序包。此 PL/SQL 程序包还可用于修改、查看、导出、导入和删除统计信息。

还可以通过数据库初始化参数来管理优化程序和系统统计信息收集。例如：

- OPTIMIZER\_DYNAMIC\_SAMPLING 参数控制优化程序执行的动态采样的级别。当表和相关索引的统计信息不可用或因为过于陈旧而不可信时，可以使用动态采样来进行估计。当收集到的统计信息无法使用或可能会导致重大估计错误时，动态采样也可以估计单表的谓词选择性。
- STATISTICS\_LEVEL 参数控制数据库中所有主要统计信息的收集或指导，并设置数据库的统计信息收集级别。此参数的值包括：BASIC、TYPICAL 和 ALL。通过查询 V\$STATISTICS\_LEVEL 视图可以确定受 STATISTICS\_LEVEL 参数影响的参数。  
**注：**将 STATISTICS\_LEVEL 设置为 BASIC 会禁用许多自动功能，因此建议不要使用该设置。

## 统计信息首选项：概览



版权所有 © 2011, Oracle. 保留所有权利。

### 统计信息首选项：概览

自动统计信息收集功能是在 Oracle Database 10g 发行版 1 中引入的，用于减轻维护优化程序统计信息的工作。但是，在有些情况下，您必须禁用该功能，并运行您自己的脚本。其中的一个原因是缺少对象级别的控制。只要发现一小部分对象的默认收集统计信息选项的效果不佳，就必须锁定统计信息，并使用您自己的选项单独对其进行分析。例如，对于其中数据常常发生更改的列，自动尝试确定足够样本大小的功能 (ESTIMATE\_PERCENT=AUTO\_SAMPLE\_SIZE) 的效果就不太好。解决此问题的唯一方法就是用自己的脚本手动指定样本大小。

**注：**可以使用 DBA\_TAB\_STAT\_PREFS 视图了解所有相关表的全部有效统计信息首选项设置。

## 使用统计信息首选项

- PUBLISH: 用于确定是将统计信息发布到字典还是先将其存储在临时等待区中
- STALE\_PERCENT: 用于确定赖以判断对象是否具有过时统计信息的阈值级别。该值是上次收集统计信息以来修改过的行数的百分比
- INCREMENTAL: 以增量方式收集分区表的全局统计信息

```
exec dbms_stats.set_table_prefs('SH','SALES','STALE_PERCENT','13');
```

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 使用统计信息首选项

Oracle Database 11g 中的统计信息首选项功能提高了灵活性, 即使有些对象需要不同于数据库默认设置的设置, 也可依赖自动统计信息收集功能来维护优化程序统计信息。

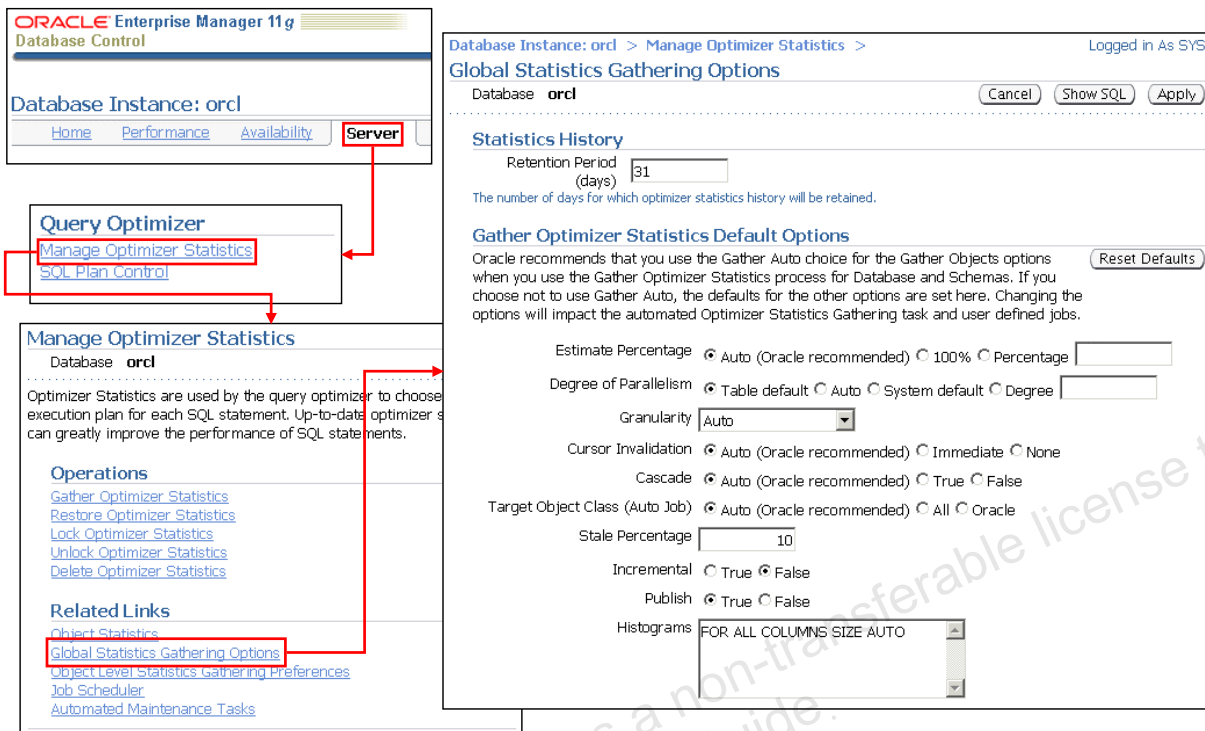
通过此功能, 您可以在对象级别或方案级别, 将覆盖 GATHER\_\*\_STATS 过程的默认行为的统计信息收集选项与自动优化程序统计信息收集任务关联起来。可以使用 DBMS\_STATS 程序包管理收集统计信息选项。

您可以在表级、方案级、数据库级和全局级设置、获取、删除、导出和导入这些首选项。全局首选项用于没有首选项的表, 而数据库首选项则用于设置针对所有表的首选项。

下列选项是 Oracle Database 11g 发行版 1 中的新增选项:

- PUBLISH 用于确定是将统计信息发布到字典还是先将其存储在临时等待区中。
- STALE\_PERCENT 用于确定判断对象是否具有过时统计信息所依据的阈值级别。该值是上次收集统计信息以来修改过的行数的百分比。示例仅将 SH.SALES 的默认值 10% 更改为 13%。
- INCREMENTAL, 以增量方式收集分区表的全局统计信息。

## 使用 Oracle Enterprise Manager 设置全局首选项



版权所有 © 2011, Oracle. 保留所有权利。

### 使用 Oracle Enterprise Manager 设置全局首选项

可以使用 Oracle Enterprise Manager 来控制全局首选项设置。可以在“Manage Optimizer Statistics（管理优化程序统计信息）”页上执行此操作；此页的访问方法如下：在“Database（数据库）”主页上单击“Server（服务器）”选项卡，然后单击“Manage Optimizer Statistics（管理优化程序统计信息）”链接，再单击“Global Statistics Gathering Options（全局统计信息收集选项）”链接。

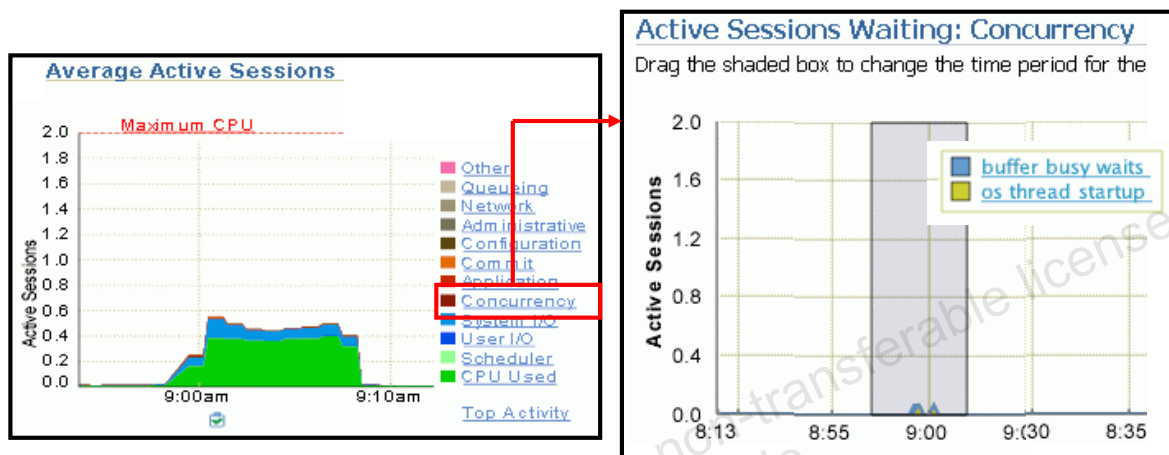
在“Global Statistics Gathering Options（全局统计信息收集选项）”页上，在“Gather Optimizer Statistics Default Options（收集优化程序统计信息默认选项）”部分更改全局首选项。完成后，单击“Apply（应用）”按钮。

**注：**要在对象级别或方案级别更改统计信息收集选项，请在“Manage Optimizer Statistics（管理优化程序统计信息）”页上单击“Object Level Statistics Gathering Preferences（对象级别统计信息收集首选项）”链接。



## Oracle 等待事件

- 等待事件集合提供了由于各种原因而不得不等待或必须等待的会话或进程的有关信息。
- V\$EVENT\_NAME 视图中列出了这些事件。



版权所有 © 2011, Oracle. 保留所有权利。

### Oracle 等待事件

等待事件是按服务器进程或线程递增的统计信息，指示必须等待事件完成之后才能继续处理。等待事件数据显示了可能会影响性能的问题的各种症状，如门锁争用、缓冲区争用和 I/O 争用。记住，这些只是问题的症状，而不是实际的原因。

等待事件按类别进行分组。等待事件的类别包括：管理、应用程序、簇、提交、并发、配置、空闲、网络、其它、调度程序、系统 I/O 和用户 I/O。

Oracle DB 中有 800 多种等待事件，包括“free buffer wait (空闲缓冲区等待)”、“latch free (门锁释放)”、“buffer busy waits (缓冲区忙等待)”、“db file sequential read (数据库文件顺序读取)”和“db file scattered read (数据库文件分散读取)”。

使用 EM，可以通过打开“Performance (性能)”页并查看“Average Active Sessions (平均活动会话数)”图（如幻灯片中所示）来查看等待事件。通过单击特定等待事件类别的链接，可以使用“Top Activity (顶级活动)”界面追溯至特定的等待事件。在本例中，有一个很小的“buffer busy waits (缓冲区忙等待)”集。

有关最常见 Oracle 事件的列表，请参阅《Oracle Database Reference 11g》一文。

## 实例统计信息

### V\$SYSSTAT

- statistic#
- name
- class
- value
- stat\_id

### V\$SYSTEM\_WAIT\_CLASS

- wait\_class\_id
- wait\_class#
- wait\_class
- total\_waits
- time\_waited

### V\$SGASTAT

- pool
- name
- bytes

### V\$EVENT\_NAME

- event\_number
- event\_id
- name
- parameter1
- parameter2
- parameter3
- wait\_class

### V\$SYSTEM\_EVENT

- event
- total\_waits
- total\_timeouts
- time\_waited
- average\_wait
- time\_waited\_micro

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 实例统计信息

要有效地诊断性能问题，必须有统计信息可供使用。Oracle DB 实例为系统、会话和单条 SQL 语句在实例级别生成许多类型的累计统计信息。Oracle DB 还跟踪段和服务的累计统计信息。分析其中任一范围内的性能问题时，通常需要查看所关注时段内统计信息的变化（差值）。

**注：**实例统计信息是动态的，在每个实例启动时进行重置。可以在某个时间点捕获这些统计信息，并将其以快照形式存储在数据库中。

### 等待事件统计信息

所有可能存在的等待事件均在 V\$EVENT\_NAME 视图中编录。

所有会话的累计统计信息都存储在 V\$SYSTEM\_EVENT 中，该视图显示特定事件从实例启动以来的等待事件合计。

在排除故障时，需要了解进程是否曾等待某项资源。

### 系统范围的统计信息

所有系统范围的统计信息均在 V\$STATNAME 视图中编录，Oracle Database 11g 提供了 400 多种统计信息。



## 实例统计信息（续）

服务器在 V\$SYSSTAT 视图中显示所有计算的系统统计信息。可以通过查询此视图，找到自实例启动以来的累计合计。

### 示例

```
SQL> SELECT name, class, value FROM v$sysstat;
NAME                                CLASS      VALUE
-----
...
table scans (short tables)          64         135116
table scans (long tables)           64           250
table scans (rowid ranges)          64            0
table scans (cache partitions)      64            3
table scans (direct read)           64            0
table scan rows gotten              64       14789836
table scan blocks gotten            64       558542
...
```

系统范围的统计信息按优化主题和调试用途分类。这些类别包括一般实例活动、重做日志缓冲区活动、锁定、数据库缓冲区高速缓存活动等等。每条系统统计信息都可能属于多种类别，因此您无法对 V\$SYSSTATS.CLASS 和 V\$SYSTEM\_WAIT\_CLASS.WAIT\_CLASS# 进行简单的联接。

您还可以通过查询 V\$SYSTEM\_WAIT\_CLASS 来查看特定等待类别的所有等待事件，如本例所示（已应用了格式）：

```
SQL> SELECT * FROM V$SYSTEM_WAIT_CLASS
2  WHERE wait_class LIKE '%I/O%';
CLASS_ID  CLASS# WAIT_CLASS      TOTAL_WAITS TIME_WAITED
-----
1740759767 8 User I/O           1119152      39038
4108307767 9 System I/O         296959       27929
```

## SGA 全局统计信息

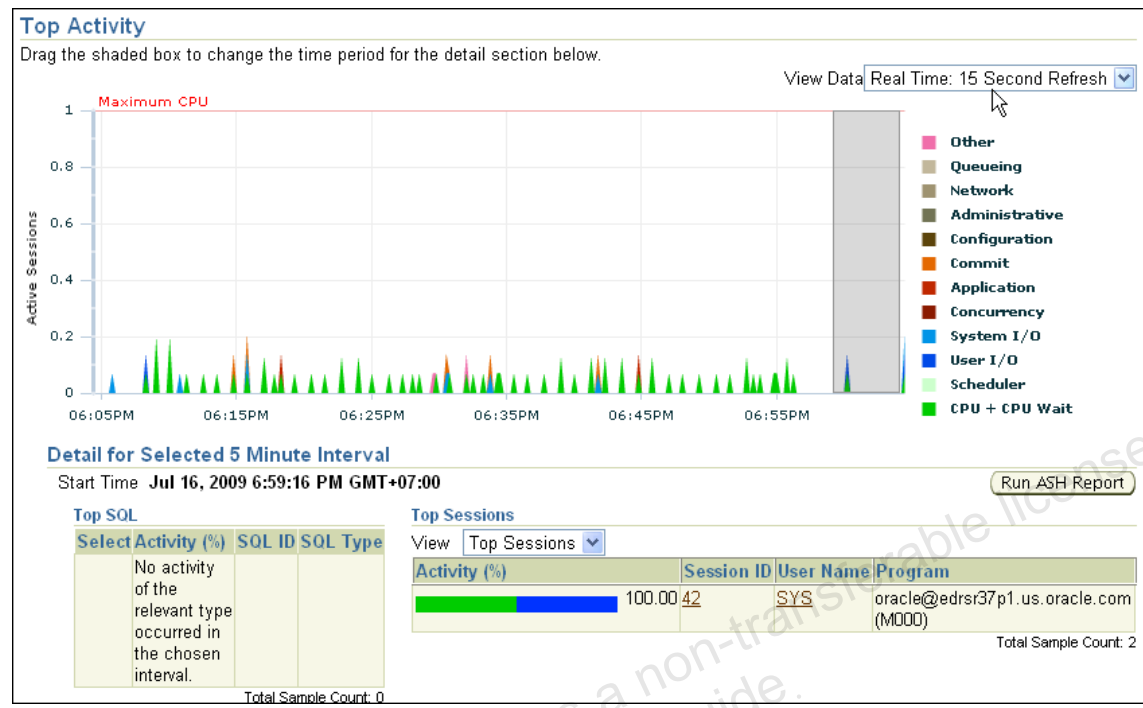
服务器在 V\$SGASTAT 视图中显示所有计算的内存统计信息。可以通过查询此视图，找到自实例启动以来关于 SGA 使用率的累计合计的详细资料，如下例所示：

```
SQL> SELECT * FROM v$sgastat;
POOL      NAME                                BYTES
-----
fixed_sga
buffer_cache
log_buffer
shared pool sessions
shared pool sql area
...
```

所示结果只是输出显示的一部分。

如将 STATISTICS\_LEVEL 参数设置为 BASIC，则 TIMED\_STATISTICS 参数的默认值为 FALSE。这样便不会收集等待事件的计时信息，并且将禁用数据库的许多性能监视功能。显式设置的 TIMED\_STATISTICS 值会覆盖从 STATISTICS\_LEVEL 派生得到的值。

# 监视会话性能

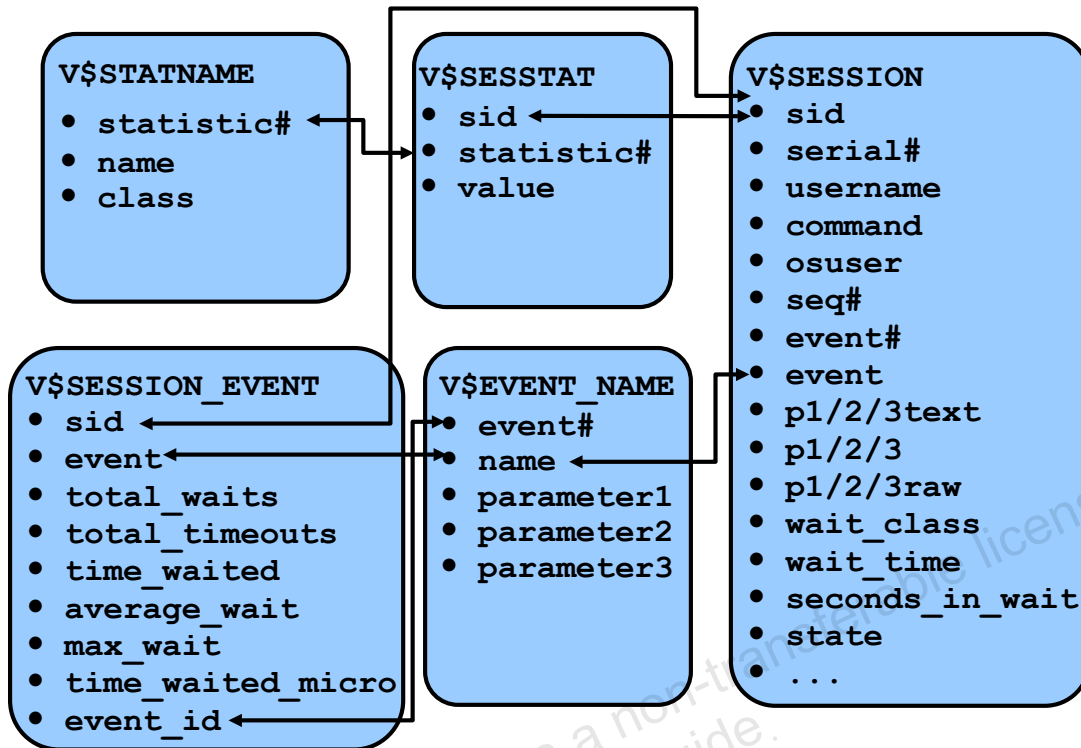


## 监视会话性能

Oracle Enterprise Manager 中有会话详细资料页面，因此您可以查看单个会话中出现的等待事件。在“Performance（性能）”选项卡页中，单击“Top Activity（顶级活动）”可查看所有会话的概要信息。在“Top Activity（顶级活动）”页的右下角列出了“Top Sessions（顶级会话）”。单击会话标识符可查看“Session Details（会话详细资料）”页。

“Top Activity（顶级活动）”页和“Session Details（会话详细资料）”页都基于数据库中的性能视图。

## 显示与会话有关的统计信息



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 显示与会话有关的统计信息

通过查询 **V\$SESSION** 可以查看每个已登录用户的当前会话信息。例如，可以使用 **V\$SESSION** 来确定会话是代表用户会话，还是由数据库服务器进程 (BACKGROUND) 创建。可以查询 **V\$SESSION** 或 **V\$SESSION\_WAIT** 来确定活动会话所等待的资源或事件。

可以在 **V\$SESSTAT** 中查看用户会话统计信息。**V\$SESSION\_EVENT** 视图列出了会话等待事件的信息。

实例统计信息的累计值通常通过动态性能视图（如 **V\$SESSTAT** 和 **V\$SYSSTAT**）提供。请注意，在关闭数据库实例时将重置动态视图中的累计值。

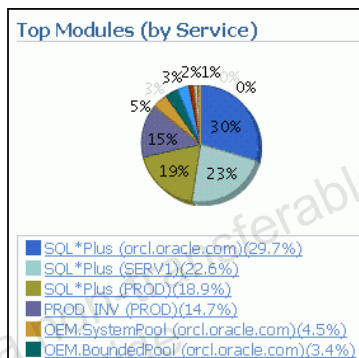
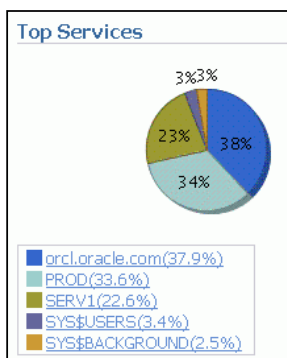
**V\$MYSTAT** 视图可显示当前会话的统计信息。

还可以通过查询 **V\$SESSMETRIC** 来显示所有活动会话的性能度量值。此视图列出了 CPU 利用率、物理读取数、硬分析数和逻辑读取比率等性能度量。

## 显示与服务有关的统计信息

对于  $n$  层环境，由于会话统计信息并非十分有用，因此可以在下列视图中查看服务级别的统计信息：

- **V\$SERVICE\_EVENT**：每个服务按事件统计的等待计数总计和等待时间总长
- **V\$SERVICE\_WAIT\_CLASS**：每个服务按等待类别统计的等待计数总计和等待时间总长



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 显示与服务有关的统计信息

在应用程序服务器共享数据库连接的  $n$  层环境中，查看会话可能无法获取分析性能所需的信息。将会话按服务名称分组可以更加准确地监视性能。以下这两个视图提供的信息与名称相似的对应会话视图所提供的信息相同，不同之处在于所提供的是服务层的信息而不是会话层的信息。

V\$SERVICE\_WAIT\_CLASS 显示每个服务的等待统计信息，这些信息按等待类别细分。

V\$SERVICE\_EVENT 显示的信息与 V\$SERVICE\_WAIT\_CLASS 的相同，只不过这些信息按照事件 ID 进一步细分。

Oracle Enterprise Manager 还按服务以及按模块和服务提供信息汇总。可以单击每个视图中的图例，查看每项服务的活动和统计信息。

可以使用 DBMS\_SERVICE 程序包在数据库中定义服务，并使用网络服务名将应用程序分配到某项服务。

## 故障排除和优化视图

### 实例/数据库

V\$DATABASE  
V\$INSTANCE  
V\$PARAMETER  
V\$SPPARAMETER  
V\$SYSTEM\_PARAMETER  
V\$PROCESS  
V\$BGPROCESS  
V\$PX\_PROCESS\_SYSSTAT  
V\$SYSTEM\_EVENT

### 磁盘

V\$DATAFILE  
V\$FILESTAT  
V\$LOG  
V\$LOG\_HISTORY  
V\$DBFILE  
V\$TEMPFILE  
V\$TEMPSEG\_USAGE  
V\$SEGMENT\_STATISTICS

### 内存

V\$BUFFER\_POOL\_STATISTICS  
V\$LIBRARYCACHE  
V\$SGAINFO  
V\$PGASTAT

### 争用

V\$LOCK  
V\$UNDOSTAT  
V\$WAITSTAT  
V\$LATCH

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 故障排除和优化视图

本幻灯片列出了一些视图，您可能需要访问这些视图来确定性能问题的原因或分析数据库的当前状态。其中的许多视图显示的数据与 Oracle Enterprise Manager 根据自动工作量资料档案库 (AWR) 生成报表时使用的数据相同。在有些情况下，需要使用原始数据来彻底诊断问题。

有关这些视图的完整说明，请参阅《Oracle Database Reference》手册。

## 字典视图

- 以下字典视图和特殊视图显示了使用 DBMS\_STATS 程序包后生成的对象统计信息：
  - DBA\_TABLES、DBA\_TAB\_COLUMNS
  - DBA\_CLUSTERS
  - DBA\_INDEXES
  - DBA\_TAB\_HISTOGRAMS
- 这些统计信息是静态的，直到再次执行 DBMS\_STATS 中的相应过程。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 字典视图

在需要详细查看特定数据库对象的优化程序统计信息时，请使用 DBMS\_STATS 程序包，该程序包将收集统计信息并填充一些 DBA\_XXX 视图中的列。

DBMS\_STATS 将填充视图中有关下列内容的列：

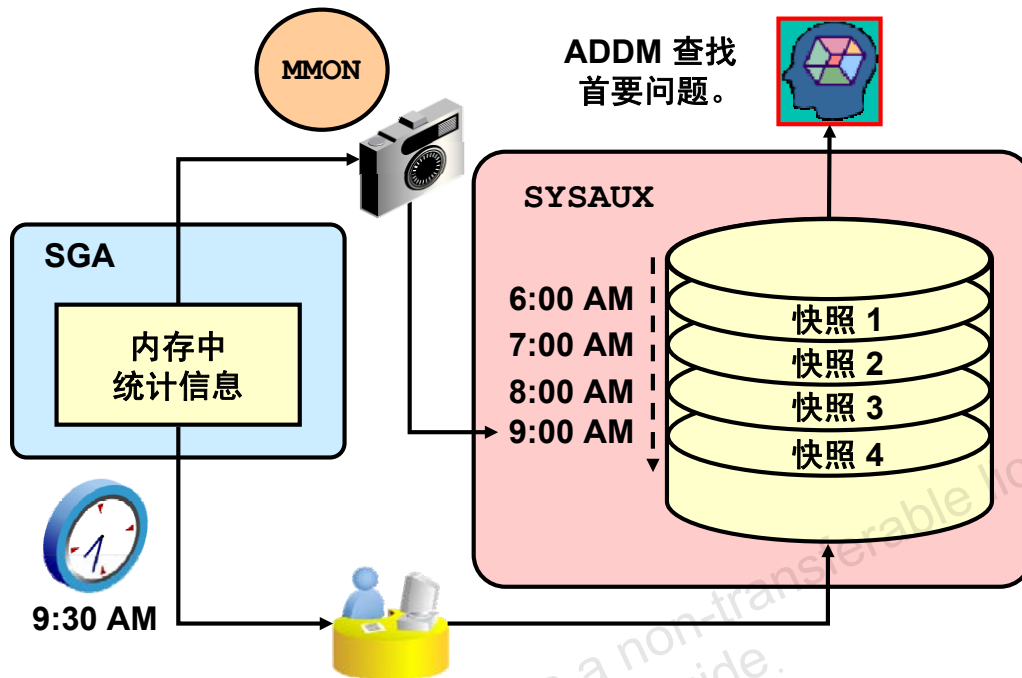
- 区和块中的表数据存储：
  - DBA\_TABLES
  - DBA\_TAB\_COLUMNS
- 区和块中的簇数据存储：
  - DBA\_CLUSTERS
- 区和块中的索引数据存储以及索引有效性：
  - DBA\_INDEXES
- 非索引列和索引列的数据分布：
  - DBA\_TAB\_HISTOGRAMS

有关使用 DBMS\_STATS 程序包的详细信息，请参阅《Oracle Database Performance Tuning Guide》。

### 字典视图（续）

执行 `ANALYZE INDEX ... VALIDATE STRUCTURE` 命令可填充包含索引统计信息的 `INDEX_STATS` 视图和 `INDEX_HISTOGRAM` 视图。

## 自动工作量资料档案库



版权所有 © 2011, Oracle. 保留所有权利。

### 自动工作量资料档案库

自动工作量资料档案库 (AWR) 是 SYS 拥有的永久性系统性能统计信息的集合。AWR 驻留在 SYS\_AUX 表空间中。

快照是在特定时间捕获并存储在 AWR 中的一组性能统计信息。每个快照都用一个快照序列号 (snap\_id) 进行标识，该序列号在 AWR 中是唯一的。默认情况下，快照每 60 分钟生成一次。通过更改快照 INTERVAL 参数可以调整此频率。由于数据库指导依赖于这些快照，因此应注意，调整间隔设置可能会影响诊断精确度。例如，如果将 INTERVAL 设置为 4 小时，则可能会遗漏在设置为 60 分钟间隔时会发觉的临时事件。

可以使用 DBMS\_WORKLOAD\_REPOSITORY.MODIFY\_SNAPSHOT\_SETTINGS 存储过程或 Database Control 来更改控制快照收集的设置。在 Database Control 中，单击“Server（服务器）”选项卡页的“Statistics Management（统计信息管理）”区域中的“Automatic Workload Repository（自动工作量资料档案库）”。然后，单击“Edit（编辑）”进行更改。与 Database Control 相比，存储过程在定义 INTERVAL 值方面更为灵活。

使用 Database Control 或 DBMS\_WORKLOAD\_REPOSITORY.CREATE\_SNAPSHOT 存储过程可以手动获取快照。支持手动获取快照的同时，也支持系统生成的自动快照。如果希望在与自动计划不一致的两个特定时间点捕获系统行为，则需要使用手动快照。



### 自动工作量资料档案库（续）

Statspack 是一种捆绑的实用程序，提供 AWR 的一部分收集和报表功能。但是，尚不支持将 Statspack 数据迁移到工作量资料档案库中。同样，工作量资料档案库也不能兼容 Statspack 方案。不能通过 Oracle Enterprise Manager 访问 Statspack；Statspack 需要设置，没有自动保留设置和自动清除功能。Statspack 实用程序不提供用于设置、自动快照收集和报表的脚本。可标记 Statspack 快照将其保留，也可将其纳入 Statspack 基线，还可用提供的脚本将其清除。

Statspack 的说明资料在 `$ORACLE_HOME/rdbms/admin/spdoc.txt` 文件中。

## 使用自动工作量资料档案库视图

- DBA\_HIST\_DB\_CACHE\_ADVICE
- DBA\_HIST\_DISPATCHER
- DBA\_HIST\_DYN\_REMASTER\_STATS
- DBA\_HIST\_IOSTAT\_DETAIL
- DBA\_HIST\_SHARED\_SERVER\_SUMMARY

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用新增自动工作量资料档案库视图

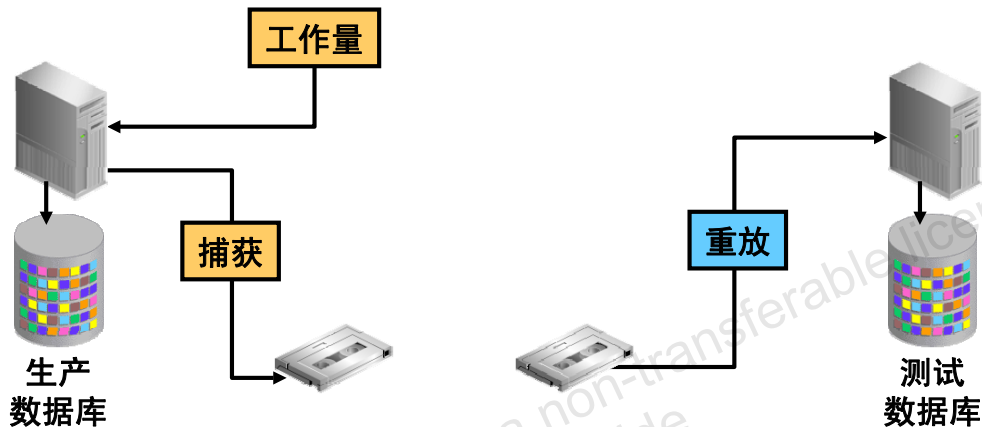
Oracle Database 11g R2 中提供了以下 AWR 统计信息视图：

- DBA\_HIST\_DB\_CACHE\_ADVICE：显示与每行对应的高速缓存大小的物理读取数的历史预测
- DBA\_HIST\_DISPATCHER：显示在快照时每个分派程序进程的历史记录信息
- DBA\_HIST\_DYN\_REMASTER\_STATS：显示有关动态重新配备进程的统计信息
- DBA\_HIST\_IOSTAT\_DETAIL：显示文件类型和函数汇总的历史 I/O 统计信息
- DBA\_HIST\_SHARED\_SERVER\_SUMMARY：显示共享服务器的历史记录信息，如共享服务器活动、公用队列和分派程序队列

## Real Application Testing 概览：数据库重放

数据库重放：

- 捕获生产工作量
- 使用实际工作量进行测试
- 每次测试中都对相同数据重放相同的 SQL



版权所有 © 2011, Oracle. 保留所有权利。

### Real Application Testing 概览：数据库重放

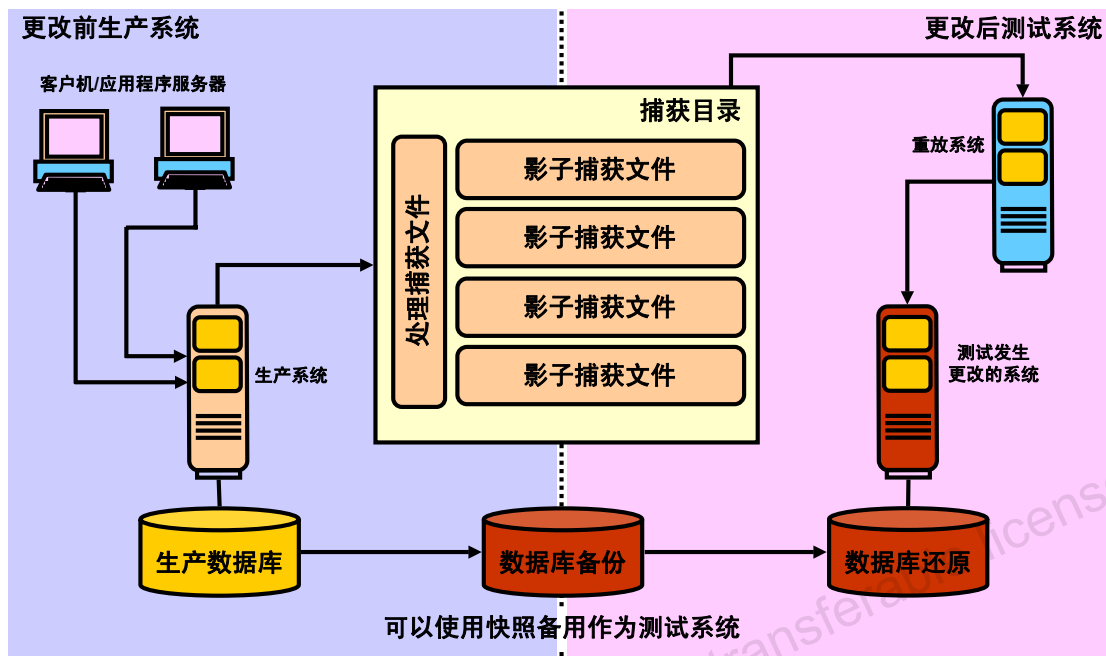
系统将发生更改。必须升级硬件、操作系统或数据库版本；或者必须同时升级这些项目。必须向用户保证应用程序在升级后仍然正常运行，并且通常要保证性能至少与升级前一样。构建一个测试环境，并使用工作量测试应用程序。反复进行更改和测试，直到确信应用程序在新环境中能正常运行。

上述整个过程依赖于测试工作量完整反映了实际生产工作量。如果测试工作量缺少生产工作量中的一条语句，那么就可能导致生产系统升级失败。

数据库重放可捕获实际生产工作量，从而使您可以在测试环境中重放该工作量。可以对重放进行修改，使其适用于更高的吞吐量、异步应用程序以及加快或放慢重放速度。数据库重放也可用于反复运行一组完全相同的语句，以达到微调的目的。可以更改环境并观察产生的影响。此功能还可用于捕获和重放测试案例以解决错误。

使用实际生产工作量进行测试，可使您对测试过程的完整性和准确性充满信心。

## 整体情况



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 整体情况

使用 Oracle Database 11g 管理系统更改的一个显著优点是：增加了企业对成功执行更改的信心。记录和重放功能使数据库服务器升级过程中的升级变得更加容易。数据库重放的一种有用的应用是测试新服务器配置的性能。假定您使用的是单实例数据库，并且希望移到 Oracle Real Application Clusters (RAC) 设置。您可以记录所关注的某个期间的工作量，然后设置一个 RAC 测试系统进行重放。在重放过程中，您可以通过比较记录系统的性能来监视新配置的性能改进。

数据库重放可用于调试。可以记录和重放模拟某个环境的会话，提高 Bug 的重现能力。可管理性功能测试是另一个优点。自我管理和自愈系统需要自动实施此项建议（“自动计算模型”）。使用多次重放迭代可以测试和微调控制策略，提高这些策略的有效性和稳定性。

要了解有关该技术的更多内容，请参加“Oracle Database 11g：性能”课程。

有关详细信息，另请参阅《Oracle Database Performance Tuning Guide》和《Oracle Database PL/SQL Packages and Types Reference》。

## 测验

选择关于统计信息收集的正确陈述：

1. 可使用 DBMS\_STATS 程序包手动收集统计信息。
2. 可通过启用自动维护任务自动收集统计信息。
3. 可从另一个数据库导入统计信息。
4. 可通过设置数据库初始化参数收集统计信息。
5. 可通过手动更新数据字典收集统计信息。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1、2、3、4

## 小结

在本课中，您应该已经学会：

- 监视会话和服务的性能
- 描述数据库重放的优点

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 14 概览：监视实例性能

本练习包含以下主题：

- 监视顶级服务和会话

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

Unauthorized reproduction or distribution prohibited. Copyright© 2014, Oracle and/or its affiliates.

华周 (286991486@qq.com) has a non-transferable license to use  
this Student Guide.



# 15

## 通过 SQL 优化管理性能

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 将 SQL 优化指导用于：
  - 确定使用资源最多的 SQL 语句
  - 优化使用资源最多的 SQL 语句
- 使用 SQL 访问指导优化工作量

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

# SQL 优化

## SQL 优化进程

- 确定没有很好地优化的 SQL 语句。
- 优化各条语句。
- 优化整个应用程序。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## SQL 优化

一般情况下，效果最明显的优化工作是 SQL 优化。没有很好地优化的 SQL 会不必要地使用过多资源。这种低效率会降低可伸缩性、使用更多的 OS 和数据库资源并增加响应时间。要对没有很好地优化的 SQL 语句进行优化，必须先确定这些语句，然后再进行优化。可以分别对各条 SQL 语句进行优化，但优化一条语句的解决方法经常会影响多条其它语句的性能。

顾名思义，使用资源最多的 SQL 语句是需要优化的语句。这些语句的用时最长，使用的 CPU 资源最多，或者物理或逻辑读取次数最多。

优化各条语句的方法是：检查优化程序统计信息，检查最高效的访问路径的解释计划，测试替代 SQL 构造，以及测试可能的新索引、实体化视图和分区。

使用优化后的 SQL 语句测试整个应用程序的性能。整体性能是否有所提高？

方法很可靠，但有些拖沓。优化单条语句并不难。测试各条语句优化对应用程序的整体影响可能非常困难。

在 Oracle Database 11g 中，可以使用一组 SQL 指导来确定和优化语句（单独优化或集体优化）。

## SQL 指导

**Advisor Central**

Advisors Checkers View Data Real Time: 15 Second Refresh

**Advisors**

[ADDM](#)
[Automatic Undo Management](#)
[Data Recovery Advisor](#)  
[Memory Advisors](#)
[MTTR Advisor](#)
[Segment Advisor](#)  
[SQL Advisors](#)
[SQL Performance Analyzer](#)
[Streams Performance Advisor](#)

**Advisor Tasks** [Change Default Parameters](#)

**Search**  
Select an advisory type and optionally enter a task name to filter the data that is displayed in your results set.

Advisory Type: **SQL Performance Analyzer** Task Name:  Advisor Runs: Last 31 Days Status: All   
 All Types  
 ADDM  
 Compression Advisor  
 Segment Advisor  
 SQL Access Advisor  
**SQL Performance Analyzer**  
 SQL Repair Advisor  
 SQL Tuning Advisor  
 Undo Advisor

Re-schedule  Previous 1-25 of 55 Next 25

Description	User	Status	Start Time	Duration (seconds)	Expires (days)
1161_1_49	ADDM auto run:	SYS COMPLETED	Aug 25, 2009 2:00:49 AM	1	3

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### SQL 指导

Oracle Database 11g 提供了一组 SQL 指导：SQL 访问指导、SQL 优化指导、SQL 性能分析器和 SQL 修复指导。AWR 可确定最近的高负荷 SQL 语句并记录有关的统计信息。

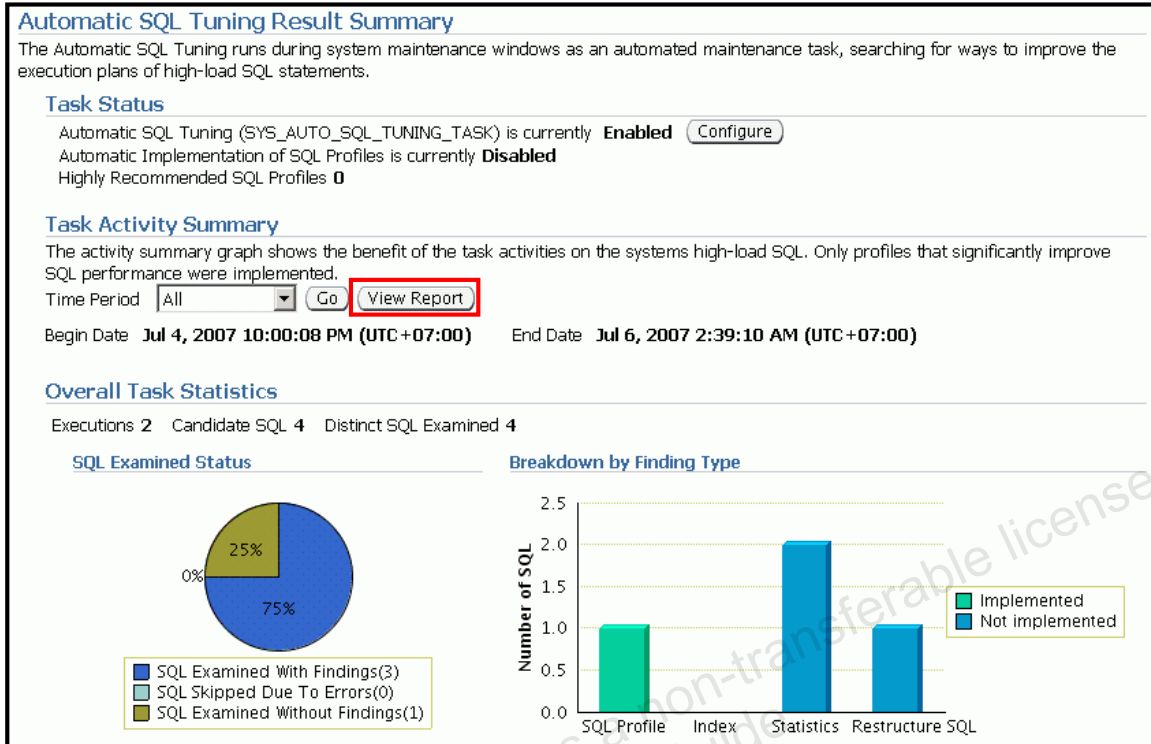
SQL 优化指导可分析一个或多个 SQL 语句（一次分析一个）。它会检查统计信息、SQL 概要文件、索引、实体化视图以及调整后的 SQL。SQL 优化指导可随时通过手动方式运行；但主要是在每个维护窗口期间针对最近的高负荷 SQL 语句运行。单击“Automatic SQL Tuning Results（自动 SQL 优化结果）”可查看和实施建议。可以对此自动作业进行配置，自动为高负荷语句实施建议的 SQL 概要文件。

SQL 访问指导会考察应用到一组 SQL 语句的更改，确定性能是否有净提高。这组语句可能是一组假想的 SQL 语句，以往的语句或手动创建的语句。

SQL 性能分析器可用于预测和防止任何影响 SQL 执行计划结构的数据库环境更改所带来的潜在性能问题。

SQL 修复指导是在某个 SQL 语句因严重错误而失败时从支持工作台运行的。严重错误还会产生意外事件。修复指导会尝试查找并推荐一个 SQL 补丁程序。如果未找到任何补丁程序，您可以接着在支持工作台中打包意外事件，并将意外事件程序包作为服务请求 (SR) 提交给 Oracle 技术支持部门。

## 自动 SQL 优化结果



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 自动 SQL 优化结果

默认情况下，自动 SQL 优化任务在每天晚上运行。自动 SQL 优化结果链接会显示结果概要页。如果单击“View Report（查看报表）”，则可查看接受检查的每条 SQL 语句。

单击“Configure（配置）”按钮会显示一个页面；在此页面中，可以更改自动优化任务的默认设置，并启用 SQL 概要文件自动实施功能。

## 实施自动优化建议

Automatic SQL Tuning Result Details

Begin Date Jul 4, 2007 10:00:08 PM (UTC+07:00)

End Date Jul 6, 2007 2:39:10 AM (UTC+07:00)

Recommendations

Only profiles that significantly improve SQL performance were implemented.

[View Recommendations](#)

[Implement All](#)

Select	SQL Text	Parsing Schema	SQL ID	Statistics	SQL Profile	Index	Restructure SQL	Miscellaneous	Error	Date
<input checked="" type="radio"/>	SELECT NULL AS table_cat, t.owner...	SYSMAN	<a href="#">0prhvnya3f97z</a>	✓	(82.9%) ✓					7/5/07

Recommendations for SQL ID:0prhvnya3f97z							
Only one recommendation should be implemented.							
<b>SQL Text</b>							
<a href="#">SELECT NULL AS table_cat, t.owner AS table_schem, t.table_name AS table_name, t.column_name AS column_name, DECODE (t.data_type, 'CHAR', 1, 'VARCHAR2', 12, 'NUMBER', 3, ...</a>							
<b>Select Recommendation</b>							
<a href="#">Original Explain Plan (Annotated)</a>							
<a href="#">Implement</a>							
Select	Type	Findings	Recommendations	Rationale	New Benefit Explain (%)	Explain Plan	Compare Explain Plans
<input checked="" type="radio"/>	SQL Profile	A potentially better execution plan was found for this statement.	Consider accepting the recommended SQL profile.		82.87	<a href="#">Explain</a>	<a href="#">Compare</a>

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 实施自动优化建议

如果单击“Automatic Tuning Results Summary（自动优化结果概要）”页上的“View Report（查看报表）”按钮，将显示“Automatic SQL Tuning Result Details（自动 SQL 优化结果详细资料）”页。您可以实施所有建议，或细化以查看或实施单个建议。在“Recommendations（建议）”页上，可以单击右侧的眼镜图标查看实施 SQL 概要文件会对解释计划产生的影响。

## SQL 优化指导：概览



SQL 优化指导

### 全面 SQL 优化

检测过时或缺少的统计信息

优化 SQL 计划  
(SQL 概要文件)

添加缺少的索引

调整 SQL 结构

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### SQL 优化指导：概览

SQL 优化指导是优化过程中使用的主要驱动程序。它可以执行多种类型的分析：

- **统计信息分析：**检查每个查询对象是否缺少统计信息或统计信息是否过时，然后提出建议以收集相关的统计信息。
- **SQL 概要分析：**优化程序会验证它自身的估计值并收集辅助信息以消除估计错误。它使用辅助信息构建 SQL 概要文件，并提出创建 SQL 概要文件的建议。创建 SQL 概要文件后，查询优化程序便可以通过此文件生成合理优化的计划。
- **访问路径分析：**考察新索引是否会显著改善对查询中的每个表的访问性能。如果合适，则会建议创建此类对象。
- **SQL 结构分析：**确定使用了错误计划的 SQL 语句，并提出对这些语句进行结构调整的相关建议。建议的更改可能涉及语法方面的更改，也可能涉及语义方面的更改。

SQL 优化指导会分别考察指导任务中包括的每个 SQL 语句。创建一个新索引可能会对查询有帮助，但也可能增加 DML 的响应时间。因此，应使用 SQL 访问指导检查建议的索引或其它对象对工作量（一组 SQL 语句）的影响，以确定性能是否有净提高。

## 使用 SQL 优化指导

- 使用 SQL 优化指导可分析 SQL 语句，并获得性能建议。
- SQL 优化指导分析的来源：
  - 顶级活动：分析当前处于活动状态的顶级 SQL 语句
  - SQL 优化集：分析用户提供的一组 SQL 语句
  - 以往的 SQL (AWR)：分析 AWR 快照收集的 SQL 语句中的语句

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 SQL 优化指导

SQL 优化指导会在每晚以自动 SQL 优化任务的形式自动运行。有时，可能需要对某个 SQL 语句立即进行优化操作。您可以随时使用 SQL 优化指导分析 SQL 语句，并获得性能建议。通常，运行此指导类似于使用 ADDM 查找性能问题的操作。

此外，可以运行 SQL 优化指导来分析哪些 SQL 语句占用的 CPU 时间、I/O 和内存最多。

即使在一项任务中提交多条语句进行分析，对每条语句的分析仍是分别进行的。要获得考虑了一组 SQL 语句的整体性能优化建议，请使用 SQL 访问指导。



## SQL 优化指导选项

**Schedule SQL Tuning Advisor** [Cancel] [Submit]

Specify the following parameters to schedule a job to run the SQL Tuning Advisor.

\* Name:

Description:

\* SQL Tuning Set:

SQL Tuning Set Description:

SQL Statements Counts:

**Scope**

Total Time Limit:  (minutes)

Scope of Analysis: ☐ Limited ☒ Comprehensive

**Overview**

The SQL Tuning Advisor analyzes individual SQL statements, and suggests indexes, SQL profiles, restructured SQL, and statistics that improve the performance of the SQL statements.

The SQL Tuning Advisor operates on a collection of SQL. You can choose a SQL Tuning Set to run the advisor. If you do not have a SQL Tuning Set with the desired SQL for running the advisor, you can create a new one.

You can click on one of the following sources, which will lead you to a data source where you can tune SQL statements using the SQL Tuning Advisor.

[Top Activity](#) [Historical SQL \(AWR\)](#) [SQL Tuning Sets](#)

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### SQL 优化指导选项

在“Schedule SQL Tuning Advisor（调度 SQL 优化指导）”页上，可以选择要包括的 SQL 语句，以及更改优化任务的自动默认设置。您可以设置 SQL 语句的来源；如果您有 ADVISOR 系统权限，可以提交任务。随后，Oracle Enterprise Manager 会为 SQL 优化指导创建一项优化任务。

通过 SQL 语句选项，可以从以下来源中选择一条或多条 SQL 语句：最近的顶级活动、AWR 中存储的以往的 SQL 语句或者已创建的 SQL 优化集。

选择优化任务的适当范围很重要。如果选择“Limited（有限制）”选项，SQL 优化指导会根据统计信息检查结果、访问路径分析结果和 SQL 结构分析结果来生成建议。“Limited（有限制）”选项不会产生 SQL 概要文件建议。如果选择“Comprehensive（综合）”选项，SQL 优化指导不仅会生成“Limited（有限制）”选项生成的所有建议，在 SQL 概要分析模式下还会调用优化程序来构建 SQL 概要文件。使用“Comprehensive（综合）”选项时，您还可以指定优化任务的时间限制，该时间限制的默认值是 30 分钟。选择“Run SQL Tuning Advisor（运行 SQL 优化指导）”之后，请使用“SQL Tuning Options（SQL 优化选项）”页配置优化任务。

# SQL 优化指导建议

SQL Tuning Results:SQL\_TUNING\_1183667475959

Page Refreshed Jul 6, 2007 3:31:31 AM GMT+07:00

Refresh

Status COMPLETED

Tuning Set Owner SYS

Started Jul 6, 2007 3:31:23 AM

Tuning Set Name TOP\_SQL\_1183667475554

Completed Jul 6, 2007 3:31:28 AM

Time Limit (seconds) 1800

Running Time (seconds) 5

Recommendations

View

Select	SQL Text	Parsing Schema	SQL ID	Statistics	SQL Profile	Index	Restructure SQL	Miscellaneous	Error
<input checked="" type="radio"/>	DECLARE table_nonexistent EXCEPTION; PRAGMA EXCEPTION_INIT (table_nonexistent, -942); BEGIN	HR	<a href="#">6t4uxuxdaxpff</a>					✓	
<input type="radio"/>	delete from sh.sales_copy	SYSTEM	<a href="#">0gqwcxx1quwux</a>	✓					
<input type="radio"/>	insert into sh.sales_copy select * from sh.sales	SYSTEM	<a href="#">axn4pkyyvt51a</a>	✓					

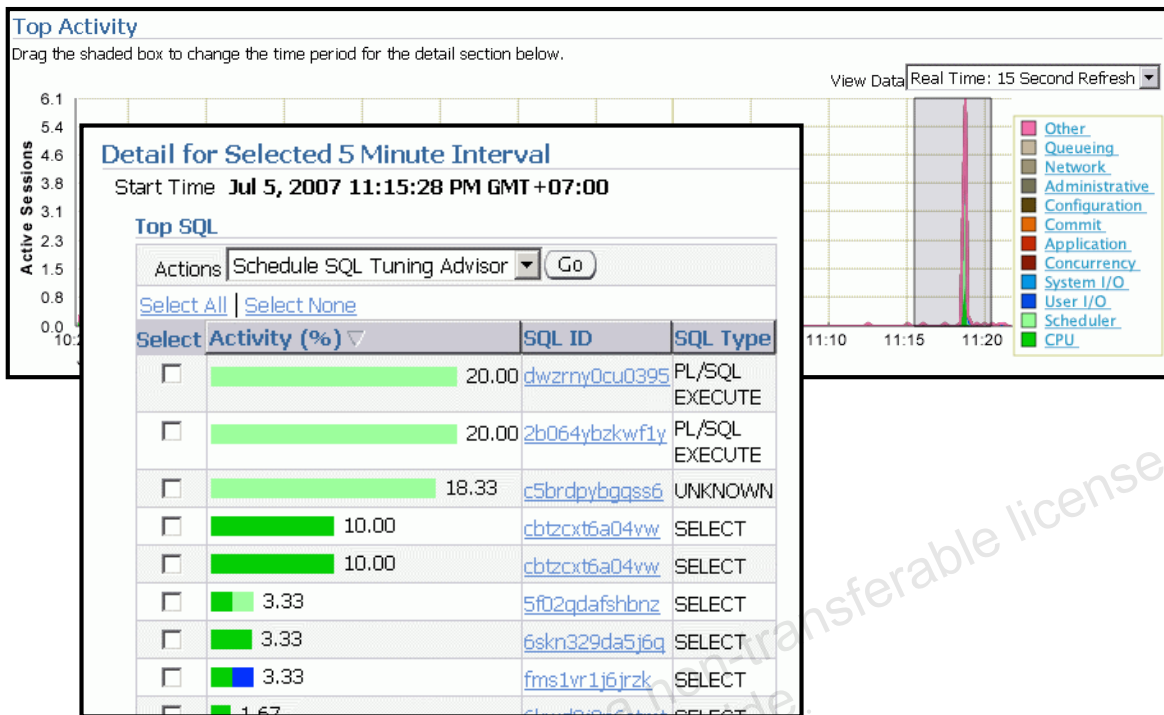
ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## SQL 优化指导建议

任务的 SQL 优化结果会在任务完成后即刻显示，也可以在以后通过“Advisor Central（指导中心）”页访问这些优化结果。此时还会显示建议的概要。您可以复查和实施单个建议。选择语句并单击“View（查看）”。

## 使用 SQL 优化指导：示例



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 使用 SQL 优化指导：示例

可以通过执行以下步骤来调用 SQL 优化指导：

1. 在“Database（数据库）”主页中，单击“Related Links（相关链接）”区域中的“Advisor Central（指导中心）”。
2. 单击“SQL Advisors（SQL 指导）”。此时将出现“SQL Tuning Advisor Links（SQL 优化指导链接）”页。

可以对以下任一来源运行该指导：

- **Active SQL（活动 SQL）**：分析当前活动的顶级 SQL 语句
- **SQL Tuning Sets（SQL 优化集）**：分析用户提供的一组 SQL 语句
- **Historical SQL (AWR)（以往的 SQL (AWR)）**：分析 AWR 快照捕获的 SQL 语句

3. 选择“Active SQL（活动 SQL）”。选择要分析的五分钟间隔，方法是将灰色框拖到目标时间段上。请选择要在选定期间内分析的一条或多条语句。
4. 单击“Run SQL Tuning Advisor（运行 SQL 优化指导）”。此时会出现“SQL Tuning Options（SQL 优化选项）”页，显示此时间间隔内的 SQL 语句。提供任务的名称和描述，选择“Comprehensive（综合）”作为范围，选择“Immediately（立即）”作为启动时间。单击“OK（确定）”。

### 使用 SQL 优化指导：示例（续）

5. 重新导航到 “Advisor Central（指导中心）” 页。指导任务的状态列于 “Results（结果）” 区域中此标题的下面。请一直等待，直至任务状态变为已完成。通过单击浏览器中的 “Refresh（刷新）” 检查状态。选择任务并单击 “View Result（查看结果）”。此时会显示 “SQL Tuning Result（SQL 优化结果）” 页。
6. 选择 SQL 语句并单击 “View Recommendations（查看建议）”。

# 重复的 SQL

Duplicate SQL

Page Refreshed Jul 6, 2007 4:49:59 AM GMT +07:00 Refresh

Applications can cause database to consume excessive CPU by parsing SQL statements that are similar and that can share the same SQL text. Such applications can also cause slow performance by creating contention in the database for Library Cache or Shared Pool.

CPU Consumption Since Instance Started

CPU Used as percentage of Total CPU (%) 2.19  
CPU Used for Parsing as percentage of CPU Used(%) 17.20

Duplicate SQL Statements

This report identifies similar SQL statements that could be shared by a single SQL statement if the database application used bind variables to replace literals and SQL coding conventions to remove differences based only on character case or whitespace. You can re-write the SQL statements to gain the efficiency of using a single, shared statement.  
Note: Only the first 2000 SQL statements that are executed only once are examined. The actual number of SQL statements that are duplicates can be more than 2000.

Expand All Collapse All

Duplicates	Plan Hash Value	SQL Text
▼ Duplicates		
▼ 5	1445457117	select * from hr.employees where employee_id = 148
	1445457117	select * from hr.employees where employee_id = 148
	1445457117	select * from hr.employees where employee_id = 145
	1445457117	select * from hr.employees where employee_id = 133
	1445457117	select * from hr.employees where employee_id = 100
	1445457117	select * from hr.employees where employee_id = 132

绑定变量  
候选项



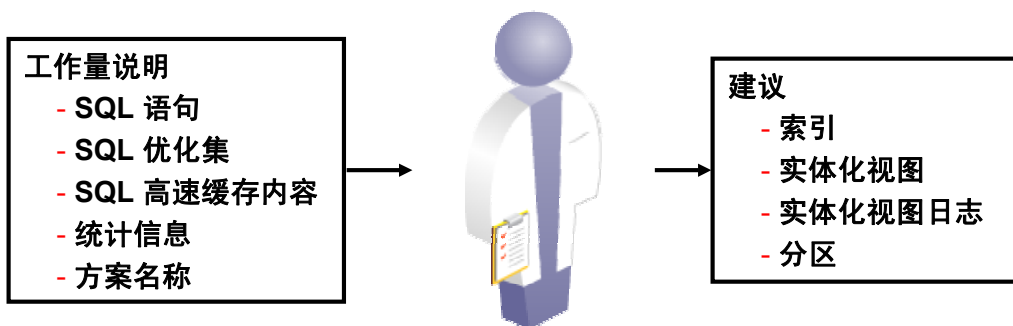
版权所有 © 2011, Oracle。保留所有权利。

## 重复的 SQL

重复的 SQL 语句是指只是所用的文字值或格式有所不同的语句。每个不同的语句在库高速缓存中都会有一个单独的游标。如果将文字替换为绑定变量，并统一格式，则重复的语句可以使用相同的游标。

通过在“Additional Monitoring Links（其他监视链接）”区域中单击“Performance（性能）”选项卡页上的“Duplicate SQL（重复的 SQL）”，可以确定重复的 SQL 语句。已确定为重复的 SQL（不考虑格式或文字差异）会列在一起。这将有助于确定应用程序中的哪个 SQL 可以合并，从而降低对库高速缓存的需求，提高语句的执行速度。

## SQL 访问指导：概览



### SQL 访问指导

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### SQL 访问指导：概览

SQL 访问指导能够为给定工作量建议一组适当的实体化视图、实体化视图日志、分区和索引。了解和使用这些结构在优化 SQL 时至关重要，因为这些结构在数据检索时能够极大地改善性能。

SQL 访问指导建议使用位图索引、基于函数的索引和 B 树索引。位图索引可减少许多类型的即席查询的响应时间，并且相对于其它索引技术而言降低了存储要求。B 树索引在数据仓库中最为常用，用于为唯一关键字或接近唯一的关键字编制索引。

SQL 访问指导的另一个组件还提供了优化实体化视图的建议，使这些视图可以快速刷新并利用常规查询重写。

**注：**有关实体化视图和查询重写的详细信息，请参阅《Oracle Database Performance Tuning Guide》。

## 典型的 SQL 访问指导会话

ORACLE Enterprise Manager 11g Database Control

Advisor Central > SQL Access Advisor: Initial Options

Select a set of initial options.

☐ Verify use of access structures (indexes, materialized views, partitioning, etc) only  
☒ Recommend new access structures  
☐ Inherit Options from a previously saved Task or Template

**Overview**  
 The SQL Access Advisor evaluates SQL statements in a workload Source, and can suggest indexes, partitioning, materialized views and materialized view logs that will improve performance of the workload as a whole.

**TIP** You are selecting the starting point for the wizard. All options can be changed from within the wizard.

Cancel Continue

版权所有 © 2011, Oracle. 保留所有权利。

### 典型的 SQL 访问指导会话

启动 SQL 访问指导会话时，可以选择“Use Default Options（使用默认选项）”，以一组建议的预定义指导选项开始。此外，还可以选择“Inherit Options from a Task or Template（从任务或模板继承选项）”来启动任务，并使该任务继承一组由模板或任务定义的选项值。这些模板包括多个针对通用环境、OLTP 和数据仓库数据库设计的通用模板。您可以保存先前任务中的定制模板并在需要时重新使用。

单击“Continue（继续）”启动 SQL 访问指导向导。

注：您可以从 Database Control 的“Advisor Central（指导中心）”页访问 SQL 访问指导。



## 工作量来源

**SQL Access Advisor: Workload Source**

Database **orcl** Cancel Step 1 of 4 **Next**

Logged In As **SYS**

---

Select the source of the workload that you want to use for the analysis. The best workload is one that fully represents all the SQL statements that access the underlying tables.

☒ **Current and Recent SQL Activity**  
SQL will be selected from the cache.

☐ **Use an existing SQL Tuning Set**  
SQL Tuning Set

☐ **Create a Hypothetical Workload from the Following Schemas and Tables**  
The advisor can create a hypothetical workload if the tables contain dimension or primary/foreign key constraints.  
Schemas and Tables  Add

Comma-separated list

☒ **TIP** Enter a schema name to specify all the tables belonging to that schema.

[Filter Options](#)

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 工作量来源

使用 SQL 访问指导向导的“Workload Source（工作量源）”页提供已定义的工作量，以便访问指导可以生成建议。支持的工作量源包括：

- **Current and Recent SQL Activity（当前和最近的 SQL 活动）**：使用高速缓存中的当前 SQL 作为工作量
- **Use an existing SQL Tuning Set（使用现有的 SQL 优化集）**：可以将先前创建的 SQL 优化集指定为工作量源
- **Create a Hypothetical Workload from the Following Schemas and Tables（从下列方案和表创建假想工作量）**：提供方案以便访问指导可以搜索维表并生成工作量

通过应用“Filter Options（过滤器选项）”部分中的可用过滤器，可以进一步缩小工作量的范围。使用这些选项，可以缩小出现在工作量中的 SQL 语句的范围。指导将过滤器应用于工作量后，可以对重点部分进行优化。可用的过滤器选项包括：

- 消耗资源最多的 SQL 语句
- 用户、模块标识符或操作
- 表



## 建议选项

**SQL Access Advisor: Recommendation Options**

Database **orcl** Cancel Back Step 2 of 4 Next

Logged In As **SYS**

---

**Access Structures to Recommend**

☒ Indexes

☒ Materialized Views

☐ Partitioning

**Scope**

The advisor can run in one of two modes, Limited or Comprehensive. Limited Mode is meant to return quickly after processing the statements with the highest cost, potentially ignoring statements with a cost below a certain threshold. Comprehensive Mode will perform an exhaustive analysis.

☐ Limited  
Analysis will focus on highest cost statements

☒ Comprehensive  
Analysis will be exhaustive

[▶ Advanced Options](#)

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 建议选项

使用“Recommendations Options（建议选项）”页可以选择是否限制指导基于单个访问方法提出建议。从“Access Structures to Recommend（要推荐的访问结构）”部分选择“Indexes（索引）”、“Materialized Views（实体化视图）”、“Partitioning（分区）”或者上述各项的任意组合。可以选择“Evaluation Only（仅评估）”，从而只评估现有的访问结构。在此模式下，指导不生成新的建议，而是就现有结构的使用提出意见。此项在跟踪当前索引、实体化视图的有效性和 MV 日志使用率随时间变化方面非常有用。

您可以使用“Advisor Mode（指导模式）”部分，以两种模式之一运行指导。这些模式会影响建议的质量和所需的时间。在“Comprehensive（综合）”模式中，指导将搜索候选的大型池，以便得到最高质量的建议。在“Limited（有限制）”模式中，指导执行速度很快，但是会限制候选建议。

## 建议选项

**Advanced Options**

**Workload Categorization**

Workload Volatility

☐ Consider only queries  
Choose this option if this is a Data Warehouse workload

Workload Scope

☐ Recommend dropping unused access structures  
Select when workload represents all access structure use cases

**Space Restrictions**

Do you want to limit additional space used by recommended indexes and materialized views?

☒ No, show me all recommendations (unlimited space)

☐ Yes, limit additional space to  MB

Set to zero or negative to recommend dropping existing access structures to make room for better ones.

**Tuning Prioritization**

Prioritize tuning of SQL statements by

SQL statements will be analyzed in descending order of the value of the prioritized statistic.

☒ Consider access structures creation costs recommendations  
If checked, the SQL Access Advisor will weigh the cost of creation of access structures against the frequency and potential improvement of SQL statement execution time. If unchecked, the cost of creation will be ignored. Check this box if you do not want specific recommendations generated for statements that are not executed frequently.

**Default Storage Locations**

By default, indexes will be placed in the schema and tablespace of the table they reference, materialized views will be placed in the schema and tablespace of the first table referenced in the query, and materialized view logs will be placed in the default tablespace of the schema of the table they reference. These fields allow you to change these default locations.

Index Tablespace	<input type="text"/>	
Index Schema	<input type="text"/>	
Materialized View Tablespace	<input type="text"/>	
Materialized View Schema	<input type="text"/>	
Materialized View Log Tablespace	<input type="text"/>	
Partitioning Tablespaces	<input type="text"/>	

Comma-separated list

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 建议选项（续）

您可以选择“Advanced Options（高级选项）”来显示或隐藏选项，这些选项可用于设置空间限制、优化选项和默认存储位置。使用“Workload Categorization（工作量类别）”部分可以设置“Workload Volatility（工作量不稳定性）”和“Workload Scope（工作量范围）”选项。生成建议时，可以选择关注只读操作，或者考察被引用对象的不稳定性。您还可以选择“Partial Workload（部分工作量）”，其中不包括删除未使用的访问结构的建议；或者选择“Complete Workload（全部工作量）”，其中包括删除未使用的访问结构的建议。

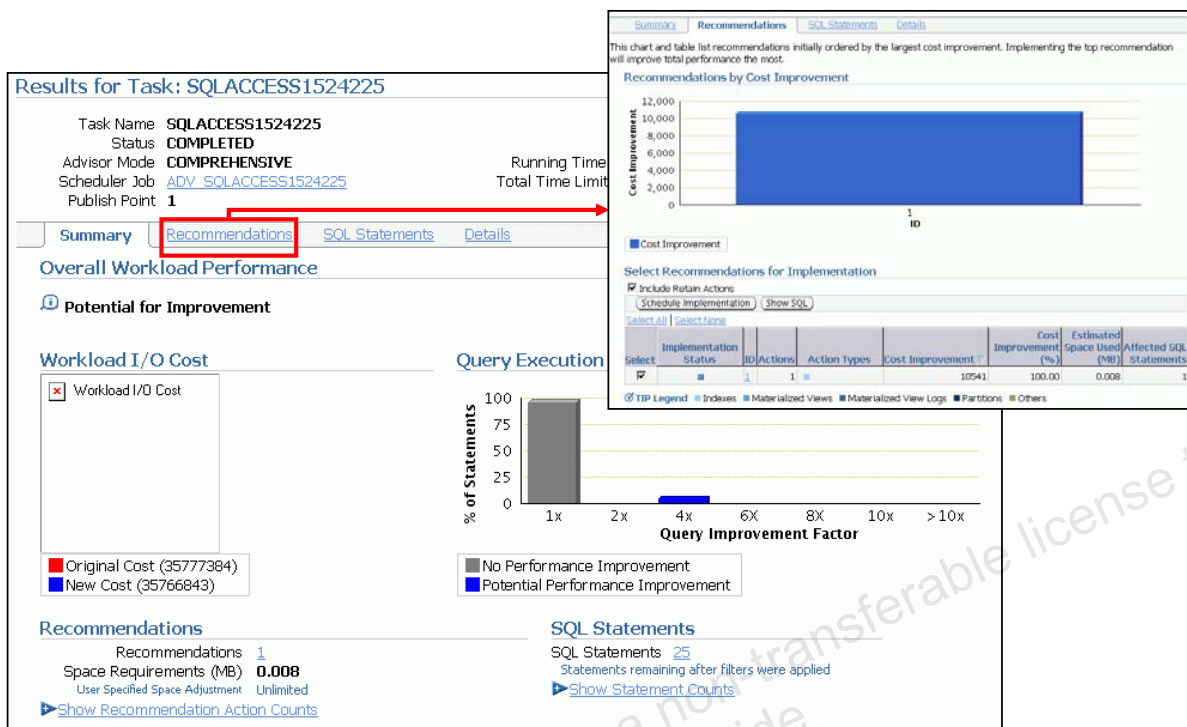
使用“Space Restrictions（空间限制）”部分可指定硬性空间限制，强制指导仅使用不超过指定限制的总空间要求生成建议。

使用“Tuning Options（优化选项）”部分可指定相关选项对指导生成的建议进行定制。使用“Prioritize Tuning of SQL Statements by（确定优化 SQL 语句优先级的依据）”下拉列表，可以按“Optimizer Cost（优化程序开销）”、“Buffer Gets（缓冲区获取数）”、“CPU Time（CPU 时间）”、“Disk Reads（磁盘读取数）”、“Elapsed Time（用时）”和“Execution Count（执行计数）”划分优先级。

**建议选项（续）**

使用“Default Storage Locations（默认存储位置）”部分可以覆盖为方案和表空间位置定义的默认值。默认情况下，索引放在所引用表的方案和表空间中。实体化视图放在相应用户的方案和表空间中，该用户执行的某一个查询为实体化视图建议提供了信息。定义了这些参数以后，可以调度和复查优化任务。

## 复查建议



版权所有 © 2011, Oracle。保留所有权利。

### 复查建议

使用“**Advisor Central**（指导中心）”页可以列出所有已完成的 SQL 访问指导任务。选择要查看建议的任务，然后单击“**View Result**（查看结果）”按钮。使用“**Results for Task**（任务结果）”的“**Summary**（概要）”页，可以访问指导查找结果的概览。该页中给出了图表和统计信息，提供了建议可能对整体工作量性能和查询执行时间产生的改进。使用该页可以显示语句计数和建议操作计数。

要查看指导任务结果的其它方面，可单击该页上其它三个选项卡之一：“Recommendations（建议）”、“SQL Statements（SQL 语句）”或“Details（详细资料）”。

“Recommendations（建议）”页中提供了一个图表和一个表，其中按建议对整个工作量总开销的改进百分比显示顶级建议。顶级建议对总体性能的改进幅度最大。

单击“**Show SQL（显示 SQL）**”按钮，可以查看为所选建议生成的 SQL 脚本。在表中单击相应的建议标识符，可以查看实施建议所需执行的操作列表。在“**Actions（操作）**”页上，可以查看实施操作需要执行的所有相应 SQL 语句。对于不希望实施的建议，请使用对应的复选框处于未选中状态。然后，单击“**Schedule Implementation（安排实施）**”按钮来实施保留的操作。此步骤以调度程序作业的形式执行。

## SQL 性能分析程序：概览

- 目标用户：DBA、QA、应用程序开发人员
- 帮助预测系统更改对 SQL 工作量响应时间的影响
- 构建不同版本的 SQL 工作量性能数据（即 SQL 执行计划和执行统计信息）
- 以串行方式执行 SQL（不考虑并发）
- 分析性能差异
- 针对单个 SQL 提供细粒度性能分析
- 与 SQL 优化指导相集成以优化回归

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### SQL 性能分析程序：概览

Oracle Database 11g 包括 SQL 性能分析程序，该工具可以准确地评估更改对构成工作量的 SQL 语句的影响。SQL 性能分析程序有助于预测潜在的更改对 SQL 查询工作量的性能影响。此功能可为 DBA 提供有关 SQL 语句性能的详细信息，如执行前后的统计信息，以及性能有所提高或降低的语句。这样您便可以（例如）在测试环境中进行更改，以确定数据库的升级是否会改进工作量性能。

## SQL 性能分析程序：使用情形

SQL 性能分析程序在以下使用场合中很有用：

- 数据库升级
- 实施优化建议
- 方案更改
- 收集统计信息
- 更改数据库参数
- 更改操作系统和硬件

可通过 Oracle Enterprise Manager 和 DBMS\_SQLPA 程序包访问



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### SQL 性能分析程序：使用情形

SQL 性能分析程序可用于预测和防止任何影响 SQL 执行计划结构的数据库环境更改所带来的潜在性能问题。这些更改可包括（但不限于）以下任何更改：

- 数据库升级
- 实施优化建议
- 方案更改
- 收集统计信息
- 更改数据库参数
- 更改操作系统和硬件

您可以使用 SQL 性能分析程序分析最复杂的环境更改所导致的 SQL 性能变化。随着应用程序在开发生命周期中的推进，数据库应用程序开发人员可以测试对方案、数据库对象和重写应用程序的更改，以减轻任何潜在的性能影响。

SQL 性能分析程序还可以比较 SQL 性能统计信息。

您可以通过 Oracle Enterprise Manager 或使用 DBMS\_SQLPA 程序包访问 SQL 性能分析程序。

### SQL 性能分析程序：使用场合（续）

有关详细信息，请参阅《Oracle Database 11g：性能优化》和《Oracle Database 11g：面向管理员的新增功能》课程。有关 DBMS\_SQLPA 程序包的详细资料，请参阅《Oracle Database PL/SQL Packages and Types Reference Guide》。

## 使用 SQL 性能分析程序

1. 捕获生产环境的 SQL 工作量。
2. 将 SQL 工作量传输至测试系统。
3. 构建“更改前”性能数据。
4. 进行更改。
5. 构建“更改后”性能数据。
6. 比较步骤 3 和步骤 5 的结果。
7. 优化回归的 SQL。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 SQL 性能分析程序

1. **收集 SQL:** 在此阶段中，将收集代表生产系统中的 SQL 工作量的 SQL 语句集。
2. **传输:** 必须将得到的工作量结果传输到测试系统。将 STS 从生产系统中导出，然后再导入测试系统。
3. **计算“之前版本”性能:** 在进行任何更改之前，先执行 SQL 语句，收集评估未来更改对工作量性能可能产生的影响所需的基线信息。
4. **进行更改:** 获得了之前版本数据后，即可实施计划的更改，然后开始查看对性能的影响。
5. **计算“之后版本”性能:** 在数据库环境中进行了更改之后，执行此步骤。SQL 工作量中的每个语句都在虚拟执行模式下运行（仅收集统计信息），收集与步骤 3 所捕获的相同的信息。
6. **比较和分析 SQL 性能:** 在获得了两个版本的 SQL 工作量性能数据后，可以比较之后与之前版本的数据，进行性能分析。
7. **优化回归的 SQL:** 在此阶段中，已经准确地确认了哪些 SQL 语句在进行数据库更改时可能导致性能问题。您可以使用任何一种数据库工具来优化系统。在实施了任何优化操作后，应重复该过程来创建新的之后版本数据，然后分析性能差异以确保新的性能是可接受的。



## 测验

即使在启用自动维护任务的情况下，仍要始终单独启动 SQL 优化指导。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2

## 测验

您可以收到对 AWR 快照收集的以往 SQL 语句的性能建议。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 测验

SQL 访问指导能够为给定工作量建议一组适当的实体化视图、实体化视图日志、分区和索引。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 测验

SQL 性能分析程序可为您提供有关 SQL 语句性能的详细信息，例如，执行前后的统计信息，以及性能有所提高或降低的语句。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 小结

在本课中，您应该已经学会：

- 将 SQL 优化指导用于：
  - 确定使用资源最多的 SQL 语句
  - 优化使用资源最多的 SQL 语句
- 使用 SQL 访问指导优化工作量

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 15 概览：通过 SQL 优化管理性能

本练习包含以下主题：

- 使用 SQL 优化指导

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

# 16

## 管理资源

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成下列工作：

- 配置数据库资源管理器
- 访问和创建资源计划
- 创建使用者组
- 指定用于向使用者组分配资源的指令
- 将使用者组映射到计划
- 激活资源计划
- 监视资源管理器

ORACLE®

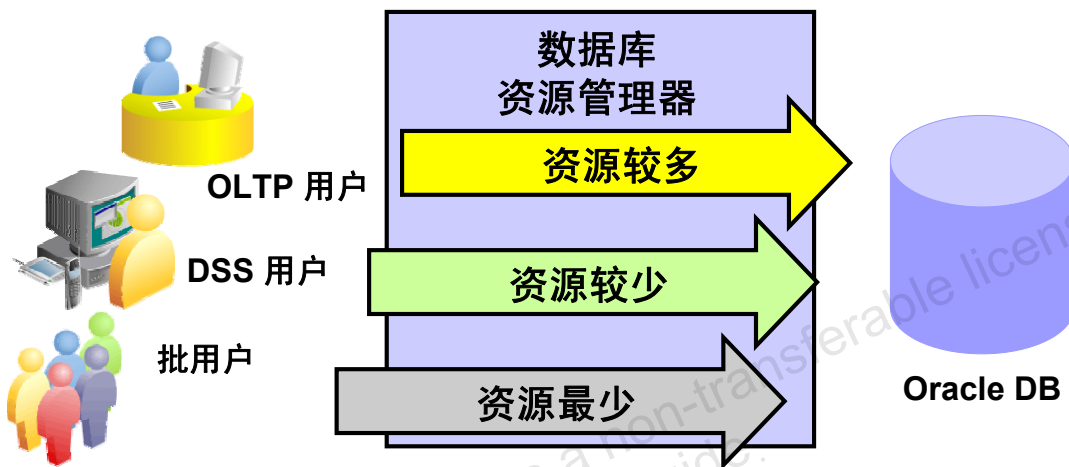
版权所有 © 2011, Oracle。保留所有权利。



## 数据库资源管理器：概览

资源管理器可用于：

- 管理混合工作量
- 控制系统性能



版权所有 © 2011, Oracle. 保留所有权利。

### 数据库资源管理器：概览

与通常只通过操作系统管理资源相比，使用数据库资源管理器（也称为资源管理器）可以对计算机资源的分配进行更多的控制。如果由操作系统来决定资源的管理，则可能会导致如下问题：

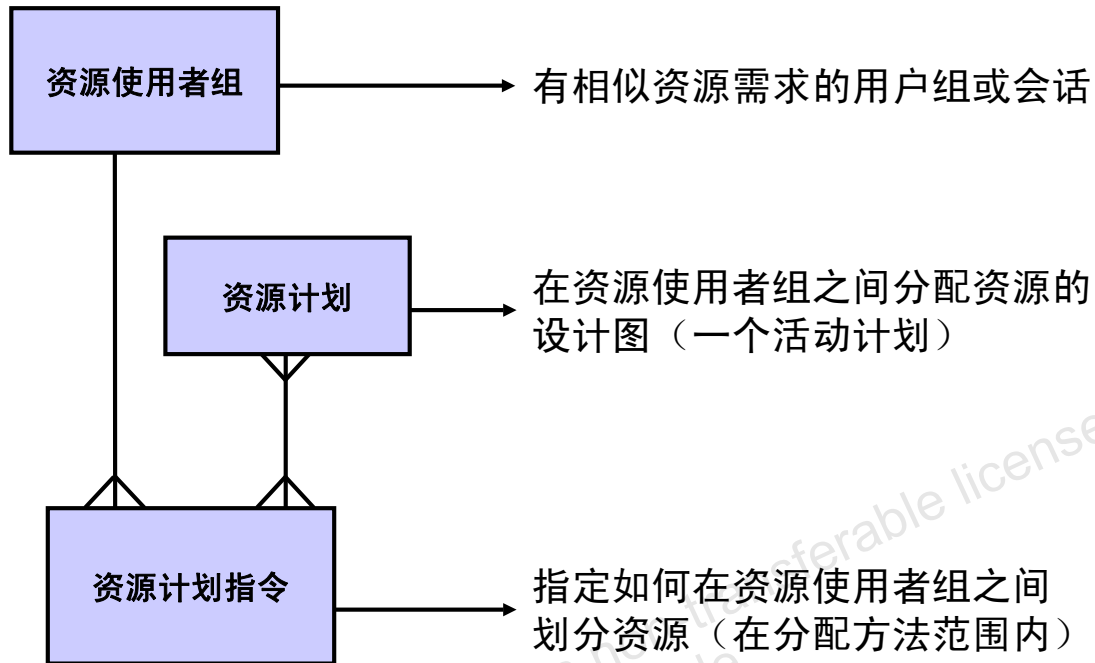
- 如果服务器进程非常多，则 Oracle DB Server 进程会频繁进行操作系统上下文切换，从而导致过多的开销
- 保持锁定的数据库服务器进程会暂停
- 资源在所有 Oracle DB 进程之间的分布不均衡，并且不能划分任务的优先级
- 不能管理特定于数据库的资源，例如并行执行服务器和会话

数据库资源管理器通过控制数据库内部的执行调度来控制资源在各个会话之间的分布。通过控制所要运行的会话以及会话运行的时间长度，数据库资源管理器可以确保资源分布与计划指令相匹配，因此也符合业务目标。利用数据库资源管理器，无论系统的负载和用户的数量如何，都可以保证用户组的处理资源达到最低限度。

DBMS\_RESOURCE\_MANAGER\_PRIVS 程序包包含授予和撤销

ADMINISTER\_RESOURCE\_MANAGER 系统权限的过程，该权限是调用资源管理器的先决条件。

## 数据库资源管理器：概念



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 数据库资源管理器：概念

使用数据库资源管理器来管理系统需要用到资源计划、资源使用者组和资源计划指令。

“资源使用者组”定义了具有相似的系统或数据库资源使用需求的一组用户或会话。

“资源计划”指定如何在各个资源使用者组之间分配资源。使用数据库资源管理器还可以在计划中创建计划，这称为“子计划”。

“资源计划指令”指定如何在使用者组或子计划之间共享某个特定资源。通过计划指令，可以将资源使用者组及子计划与特定的资源计划关联起来。

“资源分配方法”确定分配任何特定资源时使用的策略。资源分配方法由资源计划和资源使用者组使用。

## 使用资源管理器的原因

- 可以管理数据库和操作系统资源，例如：
  - CPU 使用率
  - 并行度
  - 活动会话数
  - 生成还原
  - 操作执行时间
  - 空闲时间
  - 数据库合并
  - 服务器合并
- 还可以指定标准，如果满足该标准，会话将自动切换到另一个使用者组。

通过以下渠道访问：

- EM

**Resource Manager**  
[Getting Started](#)  
[Consumer Groups](#)  
[Consumer Group Mappings](#)  
[Plans](#)  
[Settings](#)  
[Statistics](#)

- DBMS\_RESOURCE\_MANAGER 程序包

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用资源管理器的原因

数据库资源管理器提供了多种分配资源的方式：

- **CPU 方法：**可以指定在使用者组和子计划之间如何分配 CPU 资源。
- **并行度限制：**可以控制使用者组中任何操作的最大并行度。
- **具有队列的活动会话池：**可以限制使用者组或子计划的并发活动会话数。如果某个组的会话数超过了允许的最大值，则新的会话将放在队列中，等待某个活动会话完成。您还可以指定时间限制，超过此限制后会话将退出并返回错误。
- **还原池：**可以控制使用者组或子计划能够生成的还原操作的总数。当还原空间的总数超过 UNDO\_POOL 指定的数量时，在同一组中其它会话释放还原空间或者增大使用者组的还原池之前，不允许执行任何新的 INSERT、UPDATE 或 DELETE 命令。如果使用者组在执行 DML 语句时超过了限额，则中止操作并返回错误。此时仍可查询，即便使用者组已经超出其还原阈值。
- **执行时间限制：**可以指定操作所允许的最大执行时间。Oracle DB 使用基于成本的优化程序统计信息来估计操作所需的时间。如果耗时超过了所允许的最大时间 (MAX\_EST\_EXEC\_TIME)，则操作返回错误并且不会启动。如果资源使用者组有多个指定了 MAX\_EST\_EXEC\_TIME 的计划指令，则资源管理器将选择所有传入值中限制性最强的那个值。

## 使用资源管理器的原因（续）

- **空闲时间限制：**可以指定会话的空闲时间，超过该时间后将终止会话 (MAX\_IDLE\_TIME)。您可以进一步限制资源管理器，使其只终止阻止其它会话的会话 (MAX\_IDLE\_TIME\_BLOCKER)。
- **使用者组切换：**初始使用者组是在会话刚登录时所属的组。顶层调用被定义为将整个 PL/SQL 块视为一个调用，或类似地，将客户机单独发出的 SQL 语句视为单独调用。在中间层服务器实施会话共享的三层应用模型中，此功能的优势是最显而易见的。在这种情况下，中间层在为某个最终用户执行一个调用后，可使用相同的会话为另一个最终用户执行调用。因此，工作的分界线实际上为调用，并且上一个最终用户的操作不会影响下一个最终用户。可以创建一个计划指令，使资源管理器在顶层调用结束时将用户自动切换回初始使用者组。

**注：**不能在同一指令中同时指定 SWITCH\_TIME\_IN\_CALL 和 SWITCH\_TIME 这两个参数。SWITCH\_TIME 参数主要用于客户机/服务器应用程序，而 SWITCH\_TIME\_IN\_CALL 参数则用于三层应用程序。

- **数据库合并：**使用资源管理器，可以优化资源在并发数据库会话之间的分配。数据库合并要求应用程序之间彼此孤立。当一个应用程序的工作量增加时，这种增加不应影响其它应用程序。此外，每个应用程序的性能还应保持一致。数据库合并的有效候选应用程序是自动维护任务，因为目前这些应用程序可以 100% 利用服务器的 CPU 资源。
- **服务器合并：**因为许多测试、开发和小型生产数据库无法充分利用它们所在的服务器，所以服务器合并提供了一种可能的替代方法。使用服务器合并，可以通过在服务器上运行多个数据库实例来更充分地利用资源。在运行多个数据库实例的多 CPU 服务器上管理 CPU 分配的方法叫做“实例限制”。因为实例限制容易配置，并且不需要授权或安装任何新软件，因此它是其它服务器合并工具（如虚拟化和 O/S 工作量管理器）的出色替代方案。

您可以使用 Oracle Enterprise Manager 的图形化界面或 DBMS\_RESOURCE\_MANAGER 程序包的命令行来访问资源计划。

## 默认维护资源管理器计划

```
SQL> show parameter resource_manager_plan
```

NAME	TYPE	VALUE
resource_manager_plan	string	SCHEDULER[0x2843]: <b>DEFAULT_MAINTENANCE_PLAN</b>

Group/Subplan	Level 1	Level 2
ORA\$AUTOTASK_SUB_PLAN	0	25
ORA\$DIAGNOSTICS	0	5
OTHER_GROUPS	0	70
SYS_GROUP	100	0

Group/Subplan	Level 1	Level 2	Level 3
ORA\$AUTOTASK_HIGH_SUB_PLAN	0	100	0
ORA\$AUTOTASK_MEDIUM_GROUP	0	0	100
ORA\$AUTOTASK_URGENT_GROUP	100	0	0

Group/Subplan	Percentage
ORA\$AUTOTASK_HEALTH_GROUP	25
ORA\$AUTOTASK_SPACE_GROUP	25
ORA\$AUTOTASK_SQL_GROUP	25
ORA\$AUTOTASK_STATS_GROUP	25

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 默认维护资源管理器计划

自动维护任务依赖于在维护窗口期间启用的资源管理器。维护窗口打开时，会自动设置 DEFAULT\_MAINTENANCE\_PLAN 资源管理器计划，以控制自动维护任务使用的 CPU 数量。为了能够给维护窗口中的各项可能任务指定不同的优先级，为 DEFAULT\_MAINTENANCE\_PLAN 分配了多个使用者组。组和计划之间的层次结构如幻灯片所示。

对于高优先级任务，请参阅其组分配：

- “优化程序统计信息收集”任务位于 ORA\$AUTOTASK\_STATS\_GROUP 使用者组中。
- “段指导”自动任务位于 ORA\$AUTOTASK\_SPACE\_GROUP 使用者组中。
- “自动 SQL 优化”任务位于 ORA\$AUTOTASK\_SQL\_GROUP 使用者组中。

**注：**如有必要，可以更改分配给 ORA\$AUTOTASK\_HIGH\_SUB\_PLAN 中的各个自动维护任务使用者组的 CPU 资源百分比。

## 示例：DEFAULT\_PLAN

资源使用者组	分配方法		
	MGMT_P1	MGMT_P2	MGMT_P3
SYS_GROUP	100%	0%	0%
OTHER_GROUPS	0%	90%	0%
ORA\$AUTOTASK_SUB_PLAN	0%	5%	0%
ORA\$DIAGNOSTICS	0%	5%	0%

用于自动维护任务

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 示例：DEFAULT\_PLAN

DEFAULT\_PLAN 资源计划是所提供的默认计划之一。它包含的指令适用于下面提供的使用者组：

- **SYS\_GROUP**：SYS 和 SYSTEM 用户的初始使用者组。
- **OTHER\_GROUPS**：用于在活动资源计划之外的所有使用者组拥有的会话。任何活动计划中都必须有针对 OTHER\_GROUPS 的计划指令。
- **ORA\$AUTOTASK\_SUB\_PLAN**：此计划中优先级低于 SYS\_GROUP 和 OTHER\_GROUPS 的组。
- **ORA\$DIAGNOSTICS**：此计划中优先级与 ORA\$AUTOTASK\_SUB\_PLAN 相同的组。  
ORA\$ 组的优先级低可防止自动维护工作消耗过多的系统资源。

用户的初始使用者组是该用户创建的任何会话最初所属的使用者组。如果没有为用户设置初始使用者组，则 DEFAULT\_CONSUMER\_GROUP 将自动成为用户的初始使用者组。

可以使用，也可以不使用 DEFAULT\_PLAN 及关联的资源使用者组。它可以用作新资源计划的模板；可对其进行修改或删除。请根据实际环境使用。

## 潜在工作流

必需的资源管理器对象的工作流：

- 创建新的资源计划
- 创建使用者组
- 将用户分配至组
- 指定资源计划指令
- 激活资源计划

```
DBMS_RESOURCE_MANAGER.CREATE
_CONSUMER_GROUP (
  CONSUMER_GROUP => 'APPUSER',
  MGMT_MTH => 'ROUND-ROBIN',
  COMMENT => '' );
```

```
DBMS_RESOURCE_MANAGER_PRIVS.
GRANT_SWITCH_CONSUMER_GROUP
(
  grantee_name => 'PM',
  consumer_group => 'APPUSER',
  grant_option => FALSE );
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 潜在工作流

要创建新计划，必须配置多个资源管理器对象。

#### 创建新的资源计划

调度程序可以在调度程序窗口范围内自动更改资源管理器计划。如果不希望使用自动切换，则取消选中默认的“Automatic Plan Switching Enabled（已启用自动计划切换）”复选框。

#### 创建使用者组

使用 EM 中的“Resource Consumer Groups（资源使用者组）”页创建或编辑使用者组及说明，添加或删除其用户（成员）以及定义或编辑其数据库角色。

指定一种资源分配方法，以确定如何在使用者组中的会话之间分配 CPU 资源。“Round Robin（循环）”调度可确保所有会话均能合理执行。因此，默认分配方法是“Round Robin（循环）”。“Run to Completion（运行至完成）”分配方法指定优先处理活动时间最长的会话。使用 DBMS\_RESOURCE\_MANAGER.CREATE\_CONSUMER\_GROUP 过程的 MGMT\_MTH 选项可以实现等效的功能。

## 潜在工作流（续）

### 将用户分配至组

- 应将用户分配至资源使用者组。用户的默认使用者组是用户创建的任何会话最初所属的组。如果没有为用户设置使用者组，则用户的初始使用者组默认为 `DEFAULT_CONSUMER_GROUP`。必须先直接向用户（或者 `PUBLIC`）授予切换到某个使用者组的切换权限，然后才能使该使用者组成为用户的默认使用者组。切换权限不能从授予给用户角色继承。
- `DBMS_RESOURCE_MANAGER_PRIVS` 程序包中包含将资源使用者组分配至用户的过程。授予用户切换权限后，用户可以切换到其它使用者组。



# 指定资源计划指令

1

2

3

4

5

6

Ac

reate

Show SQL

Revert

Apply

General

Parallelism

Session Pool

Undo Pool

Threshold

Idle Time

A Resource Plan contains directives that specify how resources are allocated to Consumer Groups. For each Consumer Group, a directive specifies the amount of CPU resources are allocated. It also specifies limits, such as the maximum degree of parallelism, execution time, and amount of I/O, that each session in the Consumer Group can consume. You can enable a Resource Plan manually or automatically, using Scheduler Windows.

Plan **DEFAULT\_PLAN**

Description

☐ Activate this plan

☐ Automatic Plan Switching Enabled

**Resource Allocations**

Mode: ☐ Percentage ☒ Advanced

Group/Subplan	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8
APPUSER			60					
LOW_GROUP			40					
ORA\$AUTOTASK_SUB_PLAN		5						
ORA\$DIAGNOSTICS		5						
OTHER_GROUPS		90						
SYS_GROUP	100							

Modify

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

## 指定资源计划指令

如果不使用 Oracle Enterprise Manager 创建资源计划或资源使用者组，则必须先创建一个临时等待区。这是一个暂存区，可以在所做的更改生效之前存放并验证这些更改。

在 Oracle Enterprise Manager 中，有多个可用于指定计划指令的属性页：

1. 在“General（一般信息）”页上，将使用者组与计划关联，并通过 MGMT\_MTH 值指定每个使用者组或子计划分得的 CPU 资源。
2. 指定并行度限制来控制使用者组中任何操作的最大并行度。
3. 可以控制使用者组中允许的并发活动会话的最大数量。整个并行执行会话计为一个活动会话。
4. 可以控制使用者组能够生成的还原总数。
5. 您可以指定阈值，如执行时间限制（以秒为单位）、I/O 限制（以 MB 为单位）和 I/O 请求限制（以请求数为单位）。
6. 可以指定会话处于空闲状态的时间量，超过后将终止会话。可以进一步限制此类终止，使其只应用于那些阻塞其它会话的会话。

**注：**对于本幻灯片中使用标签编号的指令，后面的幻灯片会进一步详细阐述。如果在后面的幻灯片中看到“指令标签 *n*”的字样，请参阅本幻灯片。

## 资源计划的资源分配方法

参数 (注释)	可能值
MGMT_MTH	EMPHASIS、RATIO
分配 CPU 使用率	
PARALLEL_DEGREE_LIMIT_MTH	PARALLEL_DEGREE_LIMIT_ABSOLUTE
限制任何操作的并行度	
ACTIVE_SESS_POOL_MTH	PARALLEL_DEGREE_LIMIT_ABSOLUTE
限制活动会话和排队的非活动会话的数量	
QUEUEING_MTH	FIFO_TIMEOUT
控制队列及非活动会话进入活动会话池的方式	

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 资源计划的资源分配方法

资源分配方法确定了资源管理器如何向资源使用者组或资源计划分配特定资源。创建资源计划时，需要为以下资源分配方法指定值。

通过 MGMT\_MTH 参数指定 CPU 分配的方式有两种：

- EMPHASIS 是单级计划的默认方法。它也用于多级计划，它以百分比形式指定 CPU 如何在使用者组之间分配。
- RATIO 适用于单级计划，它使用比率来指定如何分配 CPU。

PARALLEL\_DEGREE\_LIMIT\_MTH 限制任何操作的最大并行度。只能为资源使用者组，而不能为子计划指定此方法。PARALLEL\_DEGREE\_LIMIT\_ABSOLUTE 方法是唯一的可能值，该方法指定可以为一个操作分配的进程数量。如果有多个计划指令引用了相同的子计划或使用组，则所有可能值中的**最小值**将用作该子计划或使用组的并行度限制。

ACTIVE\_SESS\_POOL\_MTH 参数限制活动会话的数量。所有其它会话均为非活动的，在队列中等待激活。此参数的唯一值（即唯一可用的方法）是 PARALLEL\_DEGREE\_LIMIT\_ABSOLUTE，该值也是此参数的默认值。

QUEUEING\_MTH 控制队列中非活动会话的执行顺序。FIFO\_TIMEOUT 是默认且唯一可用的方法。

## EMPHASIS 和 RATIO 的比较

EMPHASIS	RATIO
该值指定某个使用者组可以使用的 CPU 资源百分比的最大值。	该值指定一个数字，此数字表示分配到使用者组的 CPU 资源比率。
最多可以进行 8 种不同级别的资源分配。	只能为一个级别指定值。
任何给定级别的百分比总和必须小于等于 100。	必须使用整数，但是对于值的总和没有限制。
默认值是 NULL。	默认值是 NULL。

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### EMPHASIS 和 RATIO 的比较

EMPHASIS 这一 CPU 分配方法确定在资源计划中对不同使用者组中的会话的重视程度。CPU 使用率的分配级别为从 1 到 8，级别 1 的优先级最高。百分比指定如何将 CPU 资源分配给每一级中的各个使用者组。

以下规则适用于 EMPHASIS 资源分配方法：

- CPU 资源在给定级别按指定的百分比分配。为资源使用者组指定的 CPU 百分比是该使用者组在给定级别可以使用的最大值。
- 给定级别上未使用的使用者资源可供下一级别的使用者组使用。例如，如果级别 1 的使用者组只使用了 60% 的可用资源，则其余的 40% 可供级别 2 的使用者组使用。
- 任何给定级别的百分比总和必须小于等于 100。
- 对于没有明确指定计划指令的所有级别，其所有子计划或使用者组的默认资源是 0%。
- EMPHASIS 资源分配方法避免了资源缺乏问题，该问题导致优先级较低的使用者没有运行的机会。

**EMPHASIS 和 RATIO 的比较（续）**

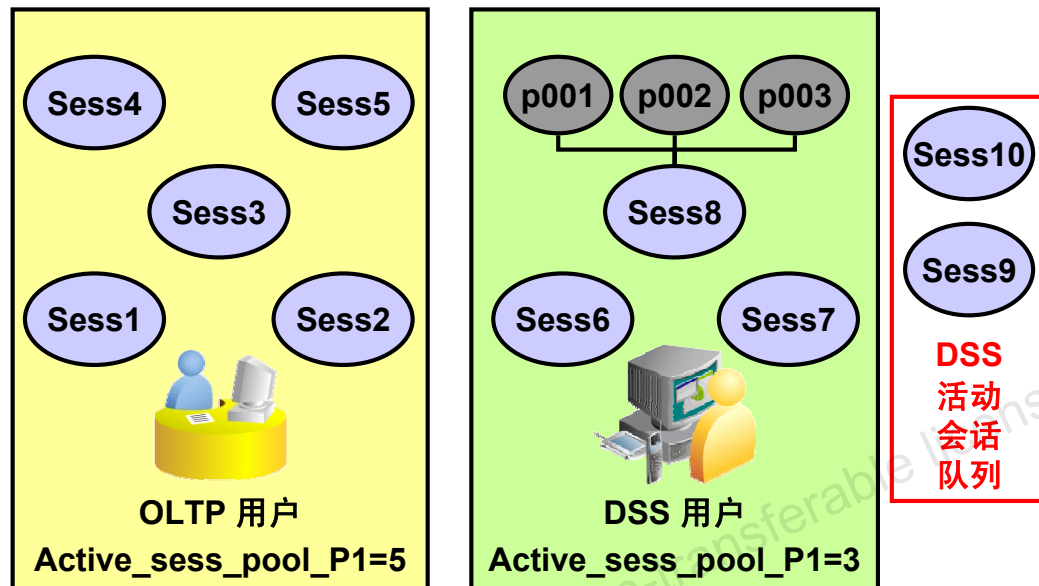
RATIO 策略是一种单级别的 CPU 分配方法。将指定要为用户组分配的 CPU 比率相对应的数字，而不是百分比。例如，假定有三个用户组 OLTP\_USERS、DSS\_USERS 和 BATCH\_USERS，您可以指定下列比率：

- OLTP\_USERS: 4
- DSS\_USERS: 3
- BATCH\_USERS: 2
- OTHER: 1

这就类似于让 OLTP 用户获得 40% 的资源、DSS 用户获得 30% 的资源、批用户获得 20% 的资源、所有其它用户组获得 10% 的可用资源。

如果 OTHER 或 DSS\_USERS 用户组中当前都没有用户在使用 CPU 资源，则 OLTP\_USERS 用户组将获得三分之二的可用资源，而 BATCH\_USERS 用户组将获得其余三分之一的可用资源。

## 活动会话池机制



版权所有 © 2011, Oracle. 保留所有权利。

### 活动会话池机制

使用活动会话池功能，您可以控制每个资源使用者组的最大并发活动会话数。使用此功能，由于资源的消耗与活动会话的数量成比例，所以 DBA 能间接控制任何资源使用者组使用的资源量。使用活动会话池有助于减少从系统中获取资源的服务器数量，因而可以避免由于试图同时运行过多作业而导致的低效的分页、交换和其它资源损耗（如内存）。

使用活动会话填充活动会话池后，资源管理器对尝试成为活动会话的所有后续会话进行排队，直到其它活动会话完成或成为不活动会话。活动会话是事务处理、查询或并行操作中当前涉及的会话。单独的并行从属进程不被视为会话；而将整个并行操作视为一个活动会话。

每个资源使用者组只有一个队列，排队方法是先进先出 (FIFO)，并带有超时。队列采用内存结构，不能直接查询。

## 设置活动会话池

**Edit Resource Plan: DEFAULT\_PLAN**

Actions:

[General](#) [Parallelism](#) **Session Pool** [Undo Pool](#) [Threshold](#) [Idle Time](#)

Specify a limit on the maximum number of concurrently active sessions for a consumer group. All other sessions will wait in an activation queue.

Group	Maximum Number of Active Sessions	Activation Queue Timeout (sec)
APPUSER	<b>50</b>	UNLIMITED
LOW_GROUP	UNLIMITED	UNLIMITED
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED
OTHER_GROUPS	UNLIMITED	UNLIMITED
SYS_GROUP	UNLIMITED	UNLIMITED



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 设置活动会话池

使用 Oracle Enterprise Manager 可以方便地配置资源计划的活动会话池设置。

例如，如果将 APPUSER 使用者组的活动会话的最大数量限制为 50：

```
BEGIN
  dbms_resource_manager.clear_pending_area();
  dbms_resource_manager.create_pending_area();
  dbms_resource_manager.update_plan_directive(
    plan => 'DEFAULT_PLAN',
    group_or_subplan => 'APPUSER',
    new_comment => '',
    new_active_sess_pool_p1 => 50,
    new_queueing_p1 => NULL,
    new_parallel_degree_limit_p1 => NULL,
    new_switch_group => '',
    new_switch_time => NULL,
    new_switch_estimate => false,
```

## 设置活动会话池（续）

```
new_max_est_exec_time => NULL,  
new_undo_pool => NULL,  
new_max_idle_time => NULL,  
new_max_idle_blocker_time => NULL,  
mgmt_p1 => NULL,  
mgmt_p2 => NULL,  
mgmt_p3 => 60,  
mgmt_p4 => NULL,  
mgmt_p5 => NULL,  
mgmt_p6 => NULL,  
mgmt_p7 => NULL,  
mgmt_p8 => NULL,  
switch_io_megabytes => NULL,  
switch_io_reqs => NULL,  
switch_for call);  
dbms_resource_manager.submit_pending_area();  
END;
```

## 指定阈值

指定执行时间限制：

- 预先估计操作执行时间（通过基于成本的优化程序统计信息），默认值为：UNLIMITED
- 在资源使用者组级别指定最大估计执行时间
- 如果估计时间超过 MAX\_EST\_EXEC\_TIME：  
(ORA-07455)，则不允许启动大工作量作业

指定其它阈值：

- 使用 SWITCH\_IO\_MEGABYTES 限制会话 I/O（以 MB 表示）
- 使用 SWITCH\_IO\_REQS 限制会话 I/O 请求

使用 SWITCH\_FOR\_CALL 返回原始使用者组（默认值：  
FALSE，使用者组未还原）

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 指定阈值

通过设置资源计划指令的 MAX\_EST\_EXEC\_TIME 参数，可以定义任何给定时间发生的任何操作的最大估计执行时间。

- 设置了此参数后，数据库资源管理器将估计特定作业消耗的时间，该时间通过基于成本的优化程序的统计信息计算得出。
- 如果有多个计划指令引用了某个资源使用者组，则该组可能会有多个 MAX\_EST\_EXEC\_TIME。数据库资源管理器将选择所有传入值中限制性最强的那个值。
- 如果操作的估计时间超过 MAX\_EST\_EXEC\_TIME，则不启动操作并发出 ORA-07455 错误。这样可以消除任何占用过多系统资源的异常大的作业。
- SWITCH\_IO\_MEGABYTES 指令指定在执行某项操作之前会话可发出的 I/O 的量（以 MB 为单位）。默认值为 NULL，表示无限制。
- SWITCH\_IO\_REQS 指令指定在执行某项操作之前会话可发出的 I/O 请求的数量。默认值为 NULL，表示无限制。
- SWITCH\_FOR\_CALL 指令指定如果是因为 SWITCH\_TIME、SWITCH\_IO\_MEGABYTES 或 SWITCH\_IO\_REQS 参数而执行某项操作，则在顶级调用结束时使用者组将还原至原始的使用者组。默认值是 FALSE，表示在顶级调用结束时不还原原始的使用者组。



## 设置空闲超时

**Edit Resource Plan: DEFAULT\_PLAN**

Actions:

[General](#) [Parallelism](#) [Session Pool](#) [Undo Pool](#) [Threshold](#) [Idle Time](#)

Specify the maximum time a session in the consumer group can be idle.

Group	Max Idle Time (sec)	Max Idle Time if Blocking Another Session (sec)
APPUSER	600	300
LOW_GROUP	UNLIMITED	UNLIMITED
ORA\$DIAGNOSTICS	UNLIMITED	UNLIMITED
OTHER_GROUP		
SYS_GROUP		

```

DBMS_RESOURCE_MANAGER.UPDATE_PLAN_DIRECTIVE
(PLAN => 'DAY_PLAN',
 GROUP_OR_SUBPLAN => 'APPUSER',
 COMMENT => 'Limit Idle Time Example',
 NEW_MAX_IDLE_TIME => 600,
 NEW_MAX_IDLE_BLOCKER_TIME => 300);
  
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 设置空闲超时

使用资源计划的“Idle Time（空闲时间）”选项卡可以设置资源计划的最大空闲超时。

“Max Idle Time (sec)（最长空闲时间(秒)）”和“Max Idle Time if Blocking Another Session (sec)（阻塞其他会话时的最大空闲时间(秒)）”分别等效于

DBMS\_RESOURCE\_MANAGER.UPDATE\_PLAN\_DIRECTIVE 过程中的 NEW\_MAX\_IDLE\_TIME 和 NEW\_MAX\_IDLE\_BLOCKER\_TIME 资源指令。这两项均以秒为单位。

NEW\_MAX\_IDLE\_TIME 指定会话既不处于执行状态也不处于等待 I/O 状态的时间。会话超过指定的时限后，PMON 进程将强制终止会话并清除其状态。除了限制会话的最大空闲时间外，还可以限制空闲会话阻塞其它会话的时间。通过 NEW\_MAX\_IDLE\_BLOCKER\_TIME 资源指令可以设置允许某个阻塞其它会话的会话处于空闲状态的秒数，以便强制实施此限制。您还可以指定 UNLIMITED 值，表示不设置最大时间。默认值为 NULL，表示无限制。这些设置提供了比概要文件更为精细的控制。在概要文件中，单一的值无法区分阻塞会话和非阻塞会话。

在幻灯片示例中，PMON 进程会终止空闲时间超过 600 秒的会话。PMON 进程还会终止空闲时间超过 300 秒并且阻塞其它会话的会话。PMON 每分钟检查一次这些限制，如果发现有会话超出了其中一个限制，便强制终止该会话并清除其所有资源。

## 限制数据库级别的 CPU 占用率

数据库合并要求：

- 应用程序彼此孤立
- 性能一致

CPU 指令可用于：

- 为每个应用程序指定一个最小的 CPU 分配
- 指定应该如何重新分配未使用的分配
- 指定 MAX\_UTILIZATION\_LIMIT 属性以便对 CPU 占用率实施绝对上限（这将覆盖计划内部的所有 CPU 重新分配）
- 良好的候选应用程序：自动维护任务

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 限制数据库级别的 CPU 占用率

对于并行数据库会话：数据库合并要求应用程序彼此孤立。当一个应用程序的工作量增加时，这种增加不应影响其它应用程序。此外，每个应用程序的性能还应保持一致。

#### 固定策略 CPU 资源管理

使用资源计划指令的 MAX\_UTILIZATION\_LIMIT 属性，可以对资源使用者组的 CPU 占用率设置绝对上限。该绝对限制会覆盖计划内部的所有 CPU 重新分配。

**注：**数据库合并的良好候选应用程序是自动维护任务，因为当前这些应用程序可占用 100% 的服务器 CPU 资源。您可以为每个自动任务使用者组设置一个最大限制。

## 限制数据库级别的 CPU 占用率

指定 CPU 占用率的最小和最大限制。

仅指定 CPU 占用率的最大限制。

DB 合并计划 #1		
	CPU 分配	最大占用率限制
App 1	50%	60%
App 2	20%	30%
App 3	20%	30%
App 4	10%	20%

DB 合并计划 #2		
	CPU 分配	最大占用率限制
App 1	null	50%
App 2	null	20%
App 3	null	20%
App 4	null	10%

```
EXEC DBMS_RESOURCE_MANAGER.CREATE_PLAN_DIRECTIVE( -
    plan                => 'db_consolidation_plan',
    group_or_subplan    => 'App_1',
    mgmt_pl             => 50,
    max_utilization_limit => 60);
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 限制数据库级别的 CPU 占用率（续）

MAX\_UTILIZATION\_LIMIT 指令可限制应用程序的 CPU 消耗。您可以设置最小和最大边界，如幻灯片中所示。

幻灯片中的 PL/SQL 示例在级别 1 上为 APP\_1 使用者组指定了一个 CPU 分配资源的最小值百分比 (50%)。该示例还指定了同一使用者组允许的最大绝对 CPU 占用率百分比 (60%)。示例使用了 DB\_CONSOLIDATION\_PLAN 计划。

可以为示例表中的每个使用者组执行类似命令。

注：在 Oracle Database 11gR2 之前的发行版中，隐式的最大占用率限制设置为 100%。

## 限制服务器级别的 CPU 占用率：实例限制

- 在运行多个数据库实例的多 CPU 服务器上管理 CPU 分配
- 启用实例限制功能：
  - 启用任意 CPU 资源计划。

```
alter system set resource_manager_plan = 'default_plan';
```

- 指定实例可随时使用的 CPU 的最大数目。

```
alter system set cpu_count=4;
```

两种方法：

- 超量配置：每个实例的 CPU 限制的总和超过实际的 CPU 数目。
- 分区：每个实例的 CPU 限制的总和等于实际的 CPU 数目。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 限制服务器级别的 CPU 占用率：实例限制

因为许多测试、开发和小型生产数据库无法充分利用它们所在的服务器，所以服务器合并提供了一种可能的替代方法。使用服务器合并，可以通过在服务器上运行多个数据库实例来更充分地利用资源。但是，这可能会导致 CPU 争用以及由于一个实例上的工作量激增而带来的负面影响。

实例限制是一种方法，该方法使用 CPU\_COUNT 初始化参数来限制实例可使用的 CPU 数目。此外，还可以利用资源管理器基于实例的资源计划为数据库会话分配 CPU。

通过启用以下内容，分两步配置实例限制：

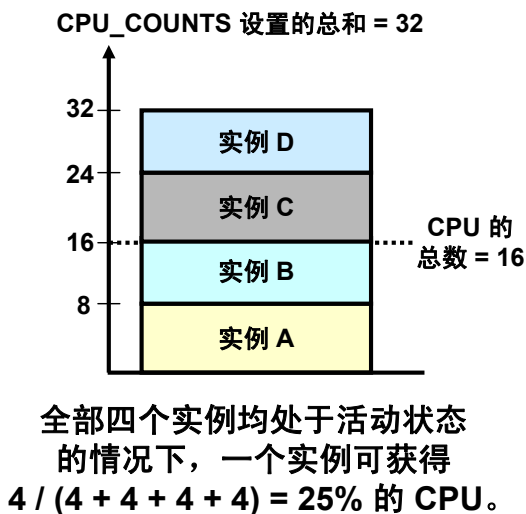
- 资源管理器，用于限制数据库实例消耗的 CPU 的量
- CPU\_COUNT 参数，指定数据库实例可在任意时间使用的 CPU 的最大量（限制），而不是实际的量

默认情况下，CPU 资源管理器假定数据库实例可使用服务器上的所有 CPU。要启用实例限制，任何具有 CPU 指令的资源计划皆可使用。

## 实例限制示例

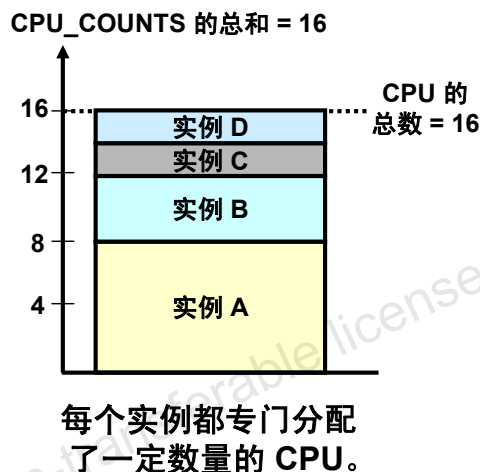
### 超量配置方法:

一个数据库实例  
仍可影响其它实例。



### 分区方法:

一个数据库实例  
无法影响其它实例。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 实例限制示例

**超量配置方法:** 此方法适用于非关键性数据库和低负载、非关键性的生产系统。虽然实例相互影响彼此的性能，但是在任意给定的时间，都会有一个或多个实例可能处于闲置状态或者负载较低。

虽然数据库实例可相互影响彼此的性能，但是实例限制能够限制这种影响，并有助于提供可预测的性能。在左边的示例中，全部四个实例均将 CPU\_COUNT 设置为 4，数据库实例在任意时间点可占用的 CPU 的最大百分比为数据库实例自己的限值除以所有活动数据库的限值总和。在本示例中，一个实例将能够消耗  $4 / (4 + 4 + 4 + 4) = 25\%$  的 CPU。如果只有两个实例处于活动状态，则一个实例将能够消耗  $4 / (4 + 4) = 50\%$  的 CPU。

**分区方法:** 此方法适用于关键生产系统。它能够防止实例互相影响，并且提供可预测的性能。

实例限制可对 CPU 资源进行分区，确保所有 CPU 限值的总和不超过 CPU 的总数。在右边的示例中，如果四个数据库实例共享一个具有 16 个 CPU 的服务器，则它们的限值可设置为 8、4、2 和 2。通过将 CPU 资源专用于某个数据库实例，分区提供两个优点：

- 一个数据库实例的 CPU 负载不会影响另一个数据库实例的 CPU 负载。
- 每个数据库实例的 CPU 资源是固定的，从而使性能的预测更加准确。

## 监视实例限制

查看 CPU\_COUNT 参数的值:

```
SELECT value FROM v$parameter WHERE name = 'cpu_count'
AND (isdefault = 'FALSE' OR ismodified != 'FALSE');
```

确认资源管理器的状态:

```
SELECT name FROM v$rsrc_plan
WHERE is_top_plan = 'TRUE' AND cpu_managed = 'ON';
```

管理限制:

```
SELECT begin_time, consumer_group_name,
       cpu_consumed_time, cpu_wait_time
FROM v$rsrcmgrpmetric_history
ORDER BY begin_time;
```

```
SELECT name, consumed_cpu_time, cpu_wait_time
FROM v$rsrc_consumer_group;
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 监视实例限制

- 如果未设置 CPU\_COUNT 参数, 第一个查询不会返回任何值。
- 如果幻灯片中的第二个查询未返回任何行, 表明资源管理器未管理 CPU。如果返回了行, 则表明存在活动的计划。

实例限制通过限制前台进程来限制前台进程的 CPU 消耗。当前台进程在 “resmgr:cpu quantum” 等待事件中等待时, 将限制该前台进程。

您可以通过两种方法监视限制的量:

- V\$RSRCMGRP\_METRIC\_HISTORY 视图显示过去一小时中每一分钟 CPU 的消耗量 (CPU\_CONSUMED\_TIME) 和限制量 (CPU\_WAIT\_TIME)。值以毫秒为单位进行显示。
- V\$RSRC\_CONSUMER\_GROUP 视图显示自 CPU 资源管理启用以来 CPU 的消耗量 (CPU\_CONSUMED\_TIME) 和限制量 (CPU\_WAIT\_TIME)。时间以毫秒为单位进行显示。

注: 有关案例研究, 请参阅名为《Database Instance Caging: A Simple Approach to Server Consolidation》的 Oracle 白皮书。



## 资源使用者组映射

**Consumer Group Mappings**

General **Priorities**

Create rules to enable the resource manager to automatically assign sessions to

View: All

Add Rule for Selected Type

Select	Priority	View	Value	Consumer
<input checked="" type="radio"/>	1	Service Module and Action	No Mappings Specified	
<input type="radio"/>	2	Service and Module	No Mappings Specified	
<input type="radio"/>	3	Module and Action	No Mappings Specified	
<input type="radio"/>	4	Module	No Mappings Specified	
<input type="radio"/>	5	Service	No Mappings Specified	
<input type="radio"/>	6	Oracle User	HR	APPUSER
			SCOTT	LOW_GRO
			SYS, SYSTEM	SYS_GRO
<input type="radio"/>	7	Client Program	No Mappings Specified	
<input type="radio"/>	8	Client OS User	ORACLE	SYS_GRO
<input type="radio"/>	9	Client Machine	No Mappings Specified	

**Attribute Mappings**

Service Module and Action  
Service and Module  
Module and Action  
Module  
Service  
Oracle User  
Client Program  
Client OS User  
Client Machine

版权所有 © 2011, Oracle. 保留所有权利。

### 资源使用者组映射

通过提供会话属性和使用者组之间的映射，可以将数据库资源管理器配置为自动将使用者组分配到会话。此外，还可以指定映射的优先级，从而指示在冲突时优先使用哪个映射。有两种类型的会话属性：登录属性和运行时属性。登录属性（幻灯片所示的“Attribute Mappings（属性映射）”列表中的最后五项）只在会话登录时有意义，此时数据库资源管理器将确定会话的初始使用者组。而会话登录后可以根据其运行时属性分配到其它使用者组。

在 Database Control 主页中，导航到“Server（服务器）”选项卡页，然后单击“Resource Manager（资源管理器）”部分中的“Resource Consumer Group Mappings（资源使用者组映射）”链接。对于每个属性，设置由标识会话的方法（如用户名）和使用者组所组成的映射。根据需要，添加或删除每个资源使用者组类别所对应的行，并在相应组中输入用于标识用户、客户机、模块或服务的文本。使用“Priorities（优先级）”选项卡，可以建立冲突的属性映射之间的优先级顺序。使用导航箭头（幻灯片中突出显示）可以按重要性从高到低的顺序设置优先级。列表顶部的映射优先级最高。

使用 EM Database Control，可以单击“Show SQL（显示 SQL）”按钮，方便地查看操作生成的 SQL。

## 资源使用者组映射（续）

以下示例为客户机 OS 用户授予了比客户机程序更高的优先级：

```
BEGIN
  dbms_resource_manager.clear_pending_area();
  dbms_resource_manager.create_pending_area();
  dbms_resource_manager.set_consumer_group_mapping(
    dbms_resource_manager.oracle_user,
    'SCOTT',
    'LOW_GROUP'
  );
  dbms_resource_manager.set_consumer_group_mapping_pri(
    EXPLICIT => 1,  SERVICE_MODULE_ACTION => 2,
    SERVICE_MODULE => 3,
    MODULE_NAME_ACTION => 4,
    MODULE_NAME => 5,
    SERVICE_NAME => 6,
    ORACLE_USER => 7,
    CLIENT_OS_USER => 8,
    CLIENT_PROGRAM => 9,
    CLIENT_MACHINE => 10
  );
  dbms_resource_manager.submit_pending_area();
END;
```



## 激活资源计划

<input type="button" value="Edit"/>	<input type="button" value="View"/>	<input type="button" value="Delete"/>	<input type="button" value="Actions"/>	<input type="button" value="Activate"/>	<input type="button" value="Go"/>
Select	Plan	Status	Description	Scheduler Windows	
<input checked="" type="radio"/>	<a href="#">DEFAULT MAINTENANCE PLAN</a>		Default plan for maintenance windows that prioritizes SYS_GROUP operations and allocates the remaining 5% to diagnostic operations and 25% to automated maintenance operations.	<a href="#">MONDAY WINDOW</a> <a href="#">TUESDAY WINDOW</a> <a href="#">WEDNESDAY WINDOW</a> <a href="#">THURSDAY WINDOW</a> <a href="#">FRIDAY WINDOW</a> <a href="#">SUNDAY WINDOW</a> <a href="#">SATURDAY WINDOW</a>	

EM > Server > Settings (EM > 服务器 > 设置) (在 “Resource Manager (资源管理器)” 部分中)

ORACLE Enterprise Manager 11g Database Control

Database Instance: orcl > Resource Manager Settings

Active Resource Plan **No Active Plan**

Available Resource Plans: MIXED\_WORKLOAD\_PLAN

Buttons: [Activate selected Resource Plan] [View selected Resource Plan]

Configure properties of Resource Management that apply to all Resource Plans.

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 激活资源计划

使用 Oracle Enterprise Manager 的 “Plans (计划)” 页可以管理资源计划。要激活计划，请选择所要激活的计划，在 “Actions (操作)” 下拉列表中选择 “Activate (激活)”，然后单击 “Go (执行)”。选定的计划将成为实例的当前最高计划。

#### 使用 RESOURCE\_MANAGER\_PLAN 初始化参数

实例的计划是使用 RESOURCE\_MANAGER\_PLAN 数据库初始化参数定义的。此参数指定用于此实例的最高计划。如果没有指定计划，则不为实例激活资源管理器。

使用 ALTER SYSTEM 语句可以激活、停用或更改当前最高计划。使用此命令更改资源计划时，更改将立即生效。

如果在参数文件中设置了参数，并且未在数据库中定义指定的计划，则不能使用该参数文件打开数据库。此时将返回以下错误：

```
ORA-07452: specified resource manager plan does not exist in
the data dictionary
```

如果遇到此错误，则必须先修改参数以显示正确值，然后才能重新启动实例。

## 数据库资源管理器信息

视图名称	信息
DBA_RSRC_PLANS	计划和状态
DBA_RSRC_PLAN_DIRECTIVES	计划指令
DBA_RSRC_CONSUMER_GROUPS	使用者组
DBA_RSRC_CONSUMER_GROUP_PRIVS	用户/角色
DBA_RSRC_GROUP_MAPPINGS	使用者组映射
DBA_RSRC_MAPPING_PRIORITY	映射优先级
DBA_USERS	列 initial_rsrc_consumer_group
DBA_RSRC_MANAGER_SYSTEM_PRIVS	用户/角色

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 数据库资源管理器信息

若干数据字典视图可用于检查实例中声明的资源计划、使用者组和计划指令。本节讨论了可以从这些视图中获取的一些有用信息。有关其中每个视图的内容的详细信息，请参阅《Oracle Database Reference》手册。

使用下列查询可获取有关数据库中定义的资源计划的信息：

```
SQL> SELECT plan, num_plan_directives, status, mandatory
2  FROM dba_rsrc_plans;
PLAN                                NUM_PLAN_DIRECTIVES STATUS      MAN
-----
DEFAULT_PLAN                        3 ACTIVE    NO
INTERNAL_QUIESCE                    2 ACTIVE    YES
INTERNAL_PLAN                       1 ACTIVE    YES
BUGDB_PLAN                          4 ACTIVE    NO
MAILDB_PLAN                         3 ACTIVE    NO
MYDB_PLAN                           3 ACTIVE    NO
```

ACTIVE 状态表示计划已经提交并且可供使用，而 PENDING 状态表示计划已经创建，但仍在暂挂区。

如果为 mandatory 列分配了 YES 值，则不能删除计划。

## 监视资源管理器

Queued Sessions						
Consumer Group	Configured Limits		Statistics			
	Active Session Limit	Maximum Time in Queue (sec)	Current Queued Sessions	Total Queued Sessions	Time in Queue (sec)	Queue Time Outs
<a href="#">APPUSER</a>	UNLIMITED	UNLIMITED	0	0	0	0
<a href="#">SYS_GROUP</a>	UNLIMITED	UNLIMITED	0	0	0	0
<a href="#">OTHER_GROUPS</a>	UNLIMITED	UNLIMITED	0	0	0	0
<a href="#">ORA\$DIAGNOSTICS</a>	UNLIMITED	UNLIMITED	0	0	0	0
<a href="#">LOW_GROUP</a>	UNLIMITED	UNLIMITED	0	0	0	0

Automatic Reprioritization						
Consumer Group	Configured Limits			Statistics		
	CPU Time Limit (ms)	I/O Requests	I/O (MB)	Switch into Group	Switch out of the Group	Active Sessions Killed
<a href="#">APPUSER</a>	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0
<a href="#">SYS_GROUP</a>	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0
<a href="#">OTHER_GROUPS</a>	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0
<a href="#">ORA\$DIAGNOSTICS</a>	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0
<a href="#">LOW_GROUP</a>	UNLIMITED	UNLIMITED	UNLIMITED	0	0	0

Idle				
Consumer Group	Idle Blocker Limit		Idle Limit	
	Maximum Idle Blocker Time (sec)	Blocker Sessions Killed	Maximum Idle Time (sec)	Idle Sessions Killed
<a href="#">APPUSER</a>	UNLIMITED	0	UNLIMITED	0
<a href="#">SYS_GROUP</a>	UNLIMITED	0	UNLIMITED	0
<a href="#">OTHER_GROUPS</a>	UNLIMITED	0	UNLIMITED	0
<a href="#">ORA\$DIAGNOSTICS</a>	UNLIMITED	0	UNLIMITED	0
<a href="#">LOW_GROUP</a>	UNLIMITED	0	UNLIMITED	0

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 监视资源管理器

可以在会话级别监视数据库资源管理器的运行。此功能已集成在自动数据库诊断监视程序 (ADDM) 中。

使用 EM Database Control 可以用不同的方法来管理和监视资源管理器。在“Server（服务器）”选项卡页上，单击“Resource Manager（资源管理器）”部分中的“Statistics（统计信息）”链接。

“Resource Monitors Statistics（资源监视器统计信息）”页显示一组描述活动资源计划当前状态的统计信息和图表。您可以查看当前活动计划的统计信息。

对于资源使用率，可以查看“消耗的 CPU”、“每秒 I/O 请求”和“每秒发出的 I/O MB”。另一图表显示“资源管理器导致等待”。还有“Queued Sessions（排队的会话）”、

“Automatic Reprioritization（自动重设优先级）”以及空闲时间的统计信息。

## 监视资源管理器

- V\$SESSION: 包含显示会话的当前组的 resource\_consumer\_group 列
- V\$RSRC\_PLAN: 显示活动资源计划的视图
- V\$RSRC\_CONSUMER\_GROUP: 包含所有活动组统计信息的视图

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 监视资源管理器（续）

#### CPU 使用率

可以提供有关 Oracle DB CPU 占用率信息的视图至少有以下三种：

- 如果您运行的是 Oracle DB 资源管理器，V\$RSRC\_CONSUMER\_GROUP 可以基于每个使用者组显示 CPU 占用率的统计信息。此视图显示与当前活动的资源使用者组相关的数据。
- V\$SYSSTAT 显示所有会话的 Oracle DB CPU 使用率。“CPU used by this session（此会话使用的 CPU）”统计信息显示所有会话使用的 CPU 总计。
- V\$SESSTAT 显示每个会话的 Oracle DB CPU 使用率。您可以使用此视图确定占用 CPU 最多的特定会话。

#### V\$RSRC\_CONSUMER\_GROUP 视图

下面简要描述了此视图中的部分列：

- **name:** 使用者组的名称。
- **active\_sessions:** 此使用者组中的当前活动会话数。
- **execution\_waiters:** 等待时间片断的活动会话数。

### 监视资源管理器（续）

- **requests:** 此使用者组中累计执行的请求数。
- **cpu\_wait\_time:** 会话等待 CPU 的累计时间。
- **consumed\_cpu\_time:** 所有会话累计消耗的 CPU 时间。

没有任何视图可以直接显示活动会话池队列，但是可以通过以下对象获取一些信息：

- **V\$SESSION:** current\_queue\_duration 列显示会话的排队时间，如果会话当前没有排队，则显示 0（零）。
- **V\$RSRC\_CONSUMER\_GROUP:** queue\_length 列显示每个使用者组中当前排队的会话数。

## 测验

选择关于资源管理器及其功能的正确说法：

1. 只能为执行时间设置阈值，不能为会话 I/O 设置阈值。
2. 可以限制数据库级别的 CPU 占用率，从而使应用程序彼此孤立。
3. 在运行多个数据库实例的多 CPU 服务器上，可以启用实例限制来限制每个服务器的 CPU 占用率。
4. 如果 SWITCH\_TIME、SWITCH\_IO\_MEGABYTES 或 SWITCH\_IO\_REQS 参数导致使用者组发生切换，则永远无法返回至原始使用者组。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、3

## 小结

在本课中，您应该已经学会如何执行以下任务：

- 配置数据库资源管理器
- 访问和创建资源计划
- 创建使用者组
- 指定用于向使用者组分配资源的指令
- 将使用者组映射到计划
- 激活资源计划
- 监视资源管理器

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 16 概览：使用资源管理器

本练习包含以下主题：

- 创建资源使用者组
- 为使用者组指定 CPU 资源分配指令
- 将用户与资源使用者组关联起来
- 激活资源计划
- 在 SQL\*Plus 中测试
- 停用资源计划

ORACLE

版权所有 © 2011, Oracle。保留所有权利。



# 17

## 使用调度程序自动执行任务

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 使用调度程序来简化管理任务
- 创建作业、程序和调度
- 监视作业执行
- 使用基于时间或基于事件的调度来执行调度程序作业
- 描述窗口、窗口组、作业类和使用用户组的用途
- 使用电子邮件通知
- 使用作业链来执行一系列相关任务
- 描述远程系统上的调度程序作业
- 使用高级调度程序概念来区分作业的优先级

ORACLE

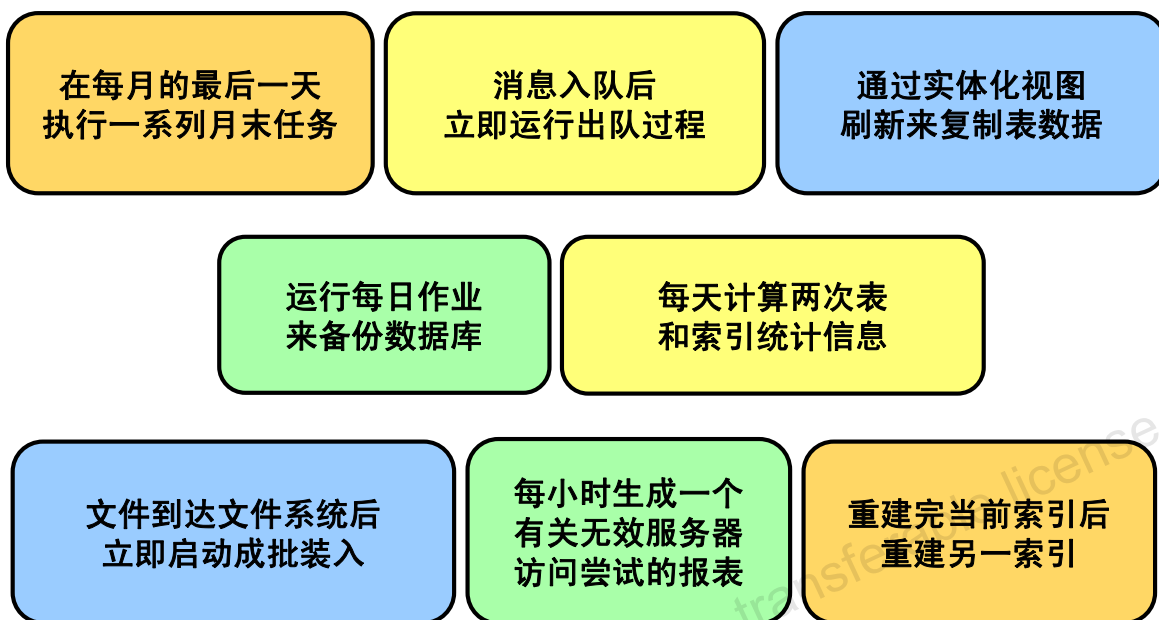
版权所有 © 2011, Oracle。保留所有权利。

### 课程目标

有关各个调度程序组件及其交互的信息，请参阅《Oracle 数据库管理员指南》。

有关 DBMS\_SCHEDULER 程序包的详细信息，请参阅《Oracle Database PL/SQL Packages and Types Reference》。

## 简化管理任务



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 简化管理任务

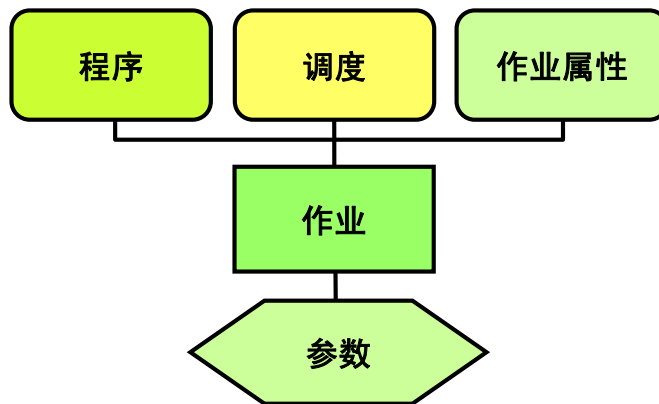
Oracle 环境中的许多任务都需要作业调度功能。例行的数据库维护和应用程序逻辑要求定期调度并运行作业。企业到企业 (B2B) 应用程序要求对其业务事件进行调度。DBA 需要在指定时间窗口中调度定期维护作业。

Oracle DB 通过数据库调度程序提供高级调度功能，此调度程序是 DBMS\_SCHEDULER 程序包中函数和过程的集合。可以在任何 SQL 环境中，或者通过 Oracle Enterprise Manager (EM) 来调用此调度程序。

使用此调度程序，数据库管理员和应用程序开发者可以控制在数据库环境中执行各种任务的时间和位置。这些任务可能既耗时又复杂；可以使用调度程序来管理和计划这些任务。

可以根据时间或者在发生指定事件时启动调度程序作业，而且调度程序还可以在作业状态更改时（例如，从 RUNNING 变为 COMPLETE）引发事件。您还可以使用为达到组合目标而链接在一起的一系列已命名的程序。

## 核心组件



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 核心组件和主要步骤

一个作业包含两个必需组件：需要执行的操作，操作的发生时间或调度。“操作”是由命令区域和作业属性中的 `job_type` 和 `job_action` 参数表示的。“时间”是在调度中表示的，调度可以基于时间或事件，或者从属于其它作业的结果。

调度程序使用以下基本组件：

- “作业”指定要执行的操作。它可以是 PL/SQL 过程、纯二进制可执行文件、Java 应用程序或 Shell 脚本。可以将程序（内容）和调度（时间）指定为作业定义的一部分，也可以改用现有的程序或调度。可以使用作业的参数来定制其运行时行为。
- “调度”指定作业的执行时间和次数。调度可以基于时间或事件。可以为作业定义调度，方法是使用一系列日期、一个事件，或两者相结合，以及表示重复间隔的附加说明。可以单独存储作业的调度，然后对多个作业使用同一个调度。
- “程序”是有关特定可执行文件、脚本或过程的元数据集合。自动作业将执行某个任务。使用程序，您无需修改作业本身即可修改作业任务或者“内容”。您可以定义程序的参数，使用户可以修改任务的运行时行为。

## 基本工作流

使用调度程序来简化管理任务：

1. 创建程序（启用或禁用）— 可选
  - 在多个作业中重用此操作
  - 在无需重新创建 PL/SQL 块的情况下更改作业的调度
2. 创建并使用调度
3. 创建并提交作业

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 基本工作流

您可以在 Oracle Enterprise Manager 的图形环境中执行所有步骤，或者通过命令行使用 DBMS\_SCHEDULER PL/SQL 程序包执行所有步骤。

#### 1. 创建程序

使用 CREATE\_PROGRAM 过程来创建程序。使用调度程序时，创建程序是一个可选部分。您还可以对操作进行编码，使其在 CREATE\_JOB 过程的匿名 PL/SQL 块中执行。通过单独创建程序，可以定义一次操作，然后在多个作业中重用此操作。使用这种方法，您无须重新创建 PL/SQL 块即可更改作业的调度。

默认情况下，程序是以禁用状态创建的（除非 enabled 参数设为 TRUE）。在将禁用的程序启用之前，作业无法执行此程序。您可以通过将 enabled 的值指定为 TRUE 来指定应以启用状态创建程序。

#### 2. 创建并使用调度

作业的调度可以是预定义的调度（用 CREATE\_SCHEDULE 过程创建的），也可以是在创建作业时定义的。

## 基本工作流（续）

### 创建并使用调度（续）

调度指定有关作业运行时间的属性，例如：

- 起始时间，定义作业从哪一时间开始执行；结束时间，指定作业在哪一时间之后失效且不再进行调度
- 指定作业重复间隔的表达式
- 通过组合现有调度创建的复杂调度
- 启动作业之前必须满足的条件或状态变化（称为事件）

通过使用调度（而不是在作业定义中指定作业的执行次数），您可以管理多个作业的预定执行，而无须更新多个作业定义。如果修改了某个调度，则使用该调度的每个作业都将自动使用新调度。

### 3. 创建并运行作业

作业是一个组合，其中包括调度、要执行的操作的说明以及作业需要的所有附加参数。可以为作业设置许多属性。属性将控制作业的执行方式。

## 测验

选择关于调度程序的所有正确语句：

1. 要使用调度程序，创建程序是一个必需步骤。
2. 作业操作在程序中（而不是直接在作业中）时，您无需重新创建 PL/SQL 块即可更改作业调度。
3. 要使用调度程序，创建作业是一个可选步骤。
4. 每个作业必须有一个调度。调度可以是预定义的，也可以在创建作业的过程中定义。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、4

## 持久轻量作业

持久轻量作业：

- 减少启动作业所需的开销和时间
- 作业元数据和运行时数据在磁盘上占用很小的空间
- 是使用作业模板（在命令行中）创建的

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
  job_name          => 'my_lightweight_job2',
  program_name      => 'MY_PROG',
  schedule_name     => 'MY_SCHED',
  job_style         => 'LIGHTWEIGHT');
END;
/
```

选择合适的作业类型：

- 使用常规作业可提供最大灵活性。
- 需要在很短的时间内创建大量作业时，请使用持久轻量作业。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 轻量作业

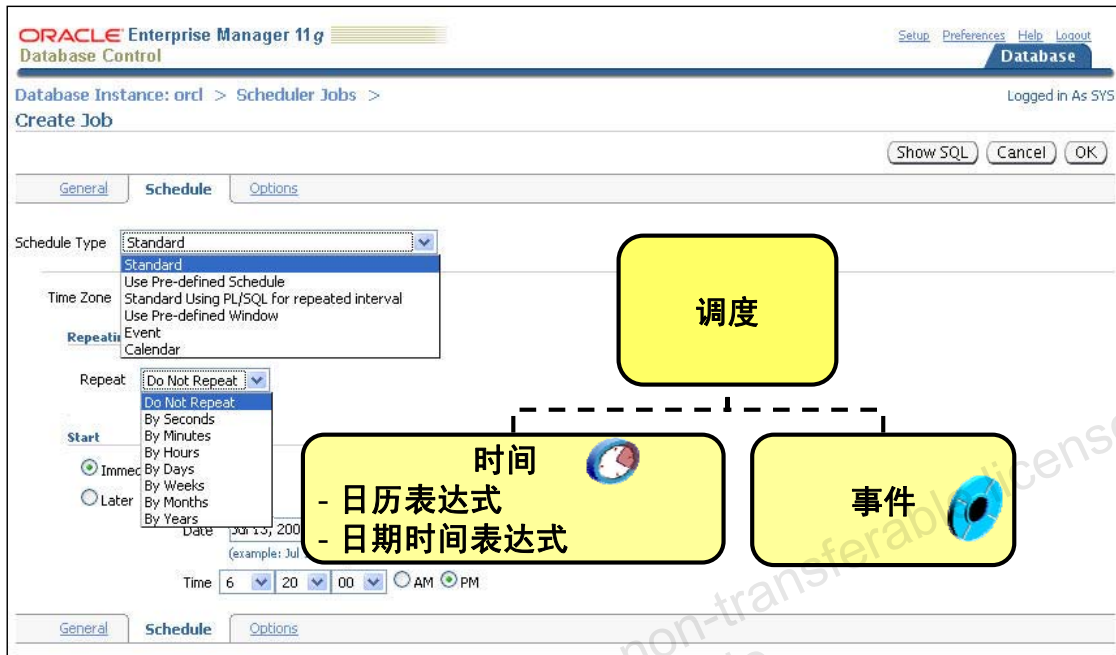
轻量作业：

- 适合需要在一秒内创建数百个作业的客户。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。此种类型作业需求的相关开销是很大的。在 Oracle DB 调度程序中，有一种“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。将为作业创建极少的元数据。这可以减少启动作业时所需的时间和创建的重做。
- 作业元数据和运行时数据在磁盘上占用很小的空间。在磁盘上占用的空间小还可实现在 RAC 环境中进行负载平衡。
- 始终是使用作业模板创建的。作业模板必须是一个存储过程或一个程序。存储过程可以保存作业所需的所有信息，包括权限。只可以指定少量作业属性：作业参数和调度。
- 必须在命令行中创建。JOB\_STYLE 参数在 EM 中不可用。

在示例中，MY\_PROG 是作业模板，调度是通过已命名的调度应用的。



## 使用基于时间的或基于事件的调度



版权所有 © 2011, Oracle. 保留所有权利。

### 使用基于时间的或基于事件的调度

要为作业指定基于时间的调度，可以指定日历表达式或日期时间表达式。使用日历表达式时，将使用作业的重复间隔和起始日期来计算作业的下一启动时间。使用日期时间表达式时，指定的表达式确定作业下次应运行的时间。如果没有指定重复间隔，作业将只在指定的起始日期运行一次。

如果作业使用基于事件的调度，作业将在事件发生时运行。在较高层次上，可以将事件视为状态的更改。布尔条件的状态从 FALSE 更改为 TRUE，或者从 TRUE 更改为 FALSE 时，将发生事件。


调度程序使用 Oracle Streams Advanced Queuing (AQ) 来引发和使用事件。

**注：**调度程序不保证作业恰好在计划的时间执行，因为可能会由于系统过载而造成资源不可用。

## 创建基于时间的作业

示例：创建一个作业，从今晚起在每晚 11:00 调用备份脚本。

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB (
    job_name=>'HR.DO_BACKUP',
    job_type => 'EXECUTABLE',
    job_action =>
      '/home/usr/dba/rman/nightly_incr.sh',
    start_date=> SYSDATE,
    repeat_interval=>'FREQ=DAILY;BYHOUR=23',
                      /* next night at 11:00 PM */
    comments => 'Nightly incremental backups');
END;
/
```



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建基于时间的作业

可以使用 DBMS\_SCHEDULER 程序包的 CREATE\_JOB 过程来创建作业。默认情况下将以禁用状态创建作业，仅当显式启用时，这些作业才生效并可以调度。所有作业名称都采用以下形式：[schema.]name。

应该使用 SYSTIMESTAMP 并指定时区，这样当时间因夏时制更改时，作业能够自动调整其执行时间。

默认情况下，将在当前方案中创建作业。可以通过指定方案的名称在另一个方案中创建作业，如幻灯片中的示例所示。作业所有者是在其方案中创建作业的用户，而作业创建者是创建作业的用户。作业将按作业所有者的权限来执行。作业运行时的国家语言支持 (NLS) 环境与创建作业时的环境相同。job\_type 参数指示作业将要执行的任务的类型。可能的值包括：

- **PLSQL\_BLOCK**：匿名 PL/SQL 块
- **STORED\_PROCEDURE**：命名的 PL/SQL、Java 或外部过程
- **EXECUTABLE**：可以从操作系统 (OS) 命令行执行的命令

## 创建基于时间的作业（续）

`job_action` 参数可以是要运行的过程的名称、脚本的名称或操作系统命令的名称，也可以是匿名的 PL/SQL 代码块，具体取决于 `job_type` 参数的值。

在幻灯片中的示例中，`job_type` 被指定为 `EXECUTABLE`，`job_action` 是所需的外部可执行文件加上任何命令行参数（可选）的操作系统相关完整路径。

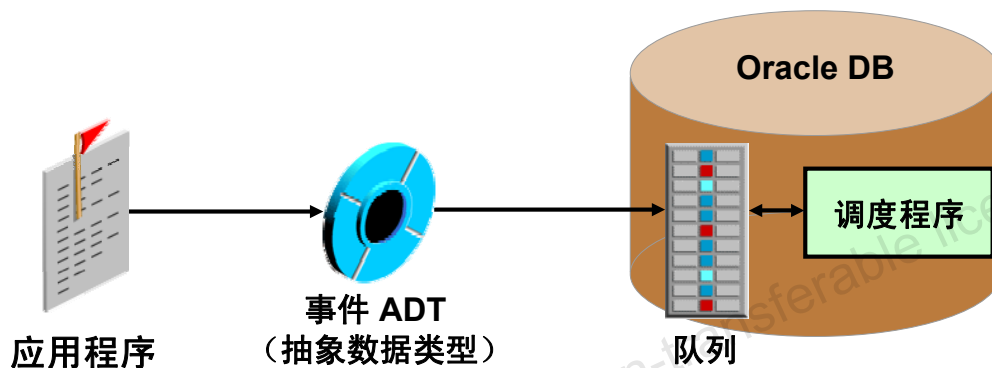
外部作业是指在数据库外部运行的作业。所有外部作业均作为低权限的来宾用户运行，这一点已在数据库管理员配置外部作业支持时确定。因为可执行文件作为低权限的来宾帐户运行，所以应确保其有权访问必要的文件和资源。大多数（但不是所有）平台都支持外部作业。对于不支持外部作业的平台，如果将作业或程序的属性创建或设置为 `EXECUTABLE` 类型，将返回错误。

有关使用调度程序配置外部程序运行环境的详细信息，请参阅 Oracle DB 平台特定的文档。

## 创建基于事件的调度

要创建基于事件的作业，必须设置：

- 队列说明（应用程序将消息入队以启动作业）
- 一个事件条件（与 Oracle Streams AQ 规则条件的语法相同），如果为 TRUE 则启动作业



版权所有 © 2011, Oracle。保留所有权利。

### 创建基于事件的调度

可以基于事件触发作业。应用程序可以通知调度程序启动作业，方法是将消息入队到 Oracle Streams 队列中。以这种方法启动的作业称为基于事件的作业。要创建基于事件的作业，必须用 CREATE\_JOB 过程设置以下两个附加属性：

- **queue\_spec:** 对队列的指定，包括应用程序将消息放入其中以引发作业启动事件的队列的名称；对于安全队列，则为 <queue\_name>、<agent\_name> 对。
- **event\_condition:** 基于消息属性的条件表达式，此表达式的求值结果必须为 TRUE，消息才能启动作业。只要消息有效负载是用户定义的对象类型，并且将表达式中的对象属性加了 tab.user\_data 前缀，就可以在表达式中包含用户数据属性。

可以将 queue\_spec 和 event\_condition 指定为内嵌作业属性，或者用这两个属性创建基于事件的调度，然后创建一个引用此调度的作业。

## 使用 Oracle Enterprise Manager 创建基于事件的调度

The screenshot shows the 'Schedule' configuration page in Oracle Enterprise Manager. It includes the following fields and options:

- Time Zone:** A text box containing 'America/New\_York' with a calendar icon to its right.
- Schedule Type:** A dropdown menu set to 'Event'.
- Event Parameters:**
  - Queue Name:** 'SYS.ALERT\_QUE' with a 'Change Queue' button.
  - Agent Name:** 'ADMIN\_AGNT1' with a calendar icon.
  - Condition:** 'tab.user\_data.event\_type = "DISK\_FULL" '.

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 Oracle Enterprise Manager 创建基于事件的调度

使用“Create Schedule（创建调度）”页，可以从标准的、基于时间的调度和基于事件的调度之中选择一种来创建。如果选择基于事件的调度，则界面将更改，然后您可以指定队列名、代理名和事件条件以及其它调度属性。

**注：**对于与 `event_condition` 匹配的事件的每个实例，调度程序都会运行基于事件的作业。但是，将忽略作业已经运行后发生的事件；事件将被使用，但不会触发作业再次运行。

#### 参考：

- 有关如何创建队列和将消息入队的信息，请参阅《Oracle Streams Advanced Queuing User's Guide and Reference》。
- 有关 Oracle Streams AQ 规则和事件条件的详细信息，请参阅《Oracle Database PL/SQL Packages and Types Reference 11g》手册中的 `DBMS_AQADM.ADD_SUBSCRIBER` 过程。

## 创建基于事件的作业

示例：创建一个作业，如果成批装入的数据文件在上午 9:00 之前到达文件系统，则运行此作业。

```
BEGIN
  DBMS_SCHEDULER.CREATE_JOB(
    job_name=>'ADMIN.PERFORM_DATA_LOAD',
    job_type => 'EXECUTABLE',
    job_action => '/loadaddir/start_my_load.sh',
    start_date => SYSTIMESTAMP,
    event_condition => 'tab.user_data.object owner =
event_condition => 'tab.user_data.object_owner =
  'HR' and tab.user_data.object_name = 'DATA.TXT'
  and tab.user_data.event_type = 'FILE_ARRIVAL'
  and tab.user_data.event_timestamp < 9 ',
    queue_spec => 'HR.LOAD_JOB_EVENT_Q');
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建基于事件的作业

要将事件信息指定为作业属性，请使用 CREATE\_JOB 的替代语法，其中包括 queue\_spec 和 event\_condition 属性。作业可以包含作为作业属性的内嵌事件信息，或可以通过指向事件调度来指定事件信息。幻灯片中显示的示例使用了内嵌的基于事件的调度。

幻灯片中的示例显示了一个作业，只要有文件在上午 9:00 前到达操作系统，就会启动此作业。假定消息有效负载是一个对象，该对象包括四个名称分别为 object\_owner、object\_name、event\_type 和 event\_timestamp 的属性。

本例使用的是用户定义的事件。因此，文件在到达了文件系统后，必须有一个程序或过程将包含正确信息的事件对象类型入队到指定的事件队列中，才能启动此作业。

HR.LOAD\_JOB\_EVENT\_Q 队列必须与用于将事件的发生通知调度程序的事件对象类型相同。即，HR.LOAD\_JOB\_EVENT\_Q 队列必须是类型化队列，其中的类型包括四个名称为 object\_owner、object\_name、event\_type 和 event\_timestamp 的属性。

有关如何创建队列和将消息入队的详细信息，请参阅《Oracle Streams Advanced Queuing User's Guide and Reference》文档。

## 基于事件的调度

事件类型：

- 用户或应用程序生成的事件
- 调度程序生成的事件

由调度程序作业引发的事件：

- |                 |                       |
|-----------------|-----------------------|
| • JOB_STARTED   | • JOB_SCH_LIM_REACHED |
| • JOB_SUCCEEDED | • JOB_DISABLED        |
| • JOB_FAILED    | • JOB_CHAIN_STALLED   |
| • JOB_BROKEN    | • JOB_ALL_EVENTS      |
| • JOB_COMPLETED | • JOB_RUN_COMPLETED   |
| • JOB_STOPPED   | • JOB_OVER_MAX_DUR    |

引发事件的示例：

```
DBMS_SCHEDULER.SET_ATTRIBUTE('hr.do_backup',
'raise_events', DBMS_SCHEDULER.JOB_FAILED);
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 基于事件的调度

您可以创建一个作业，但不向作业分配调度，而是通过直接引用事件来启动作业。有两种类型的事件：

- **用户或应用程序生成的事件：**应用程序可以引发由调度程序使用的事件。调度程序通过启动作业来回应事件。此类事件的示例：正在运行的作业完成；文件到达文件系统；数据库内的帐户被锁定；以及库存到达下限阈值。
- **调度程序生成的事件：**调度程序可以引发事件来指示在调度程序自身内发生的状态更改。例如，在作业启动时、作业完成时、作业超出其分配的运行时间时等条件下，调度程序均可以引发事件。事件的使用者是一个应用程序，该应用程序将执行某个操作以便对事件做出响应。

您可以对作业进行配置，使调度程序在作业的状态更改时引发事件。可以通过设置 `raise_events` 作业属性来完成此操作。默认情况下，在变更作业的 `raise_events` 属性之前，作业不会引发任何状态更改事件。要变更此属性，必须先使用 `CREATE_JOB` 过程创建作业，然后使用 `SET_ATTRIBUTE` 过程修改此属性的默认值。本例显示 `hr.do_backup` 作业已变更，使得该作业在失败时引发事件。

为作业启用作业状态更改事件后，调度程序将引发这些事件，方法是将消息入队到默认的事件队列 `SYS.SCHEDULER$_EVENT_QUEUE` 中。



**基于事件的调度（续）**

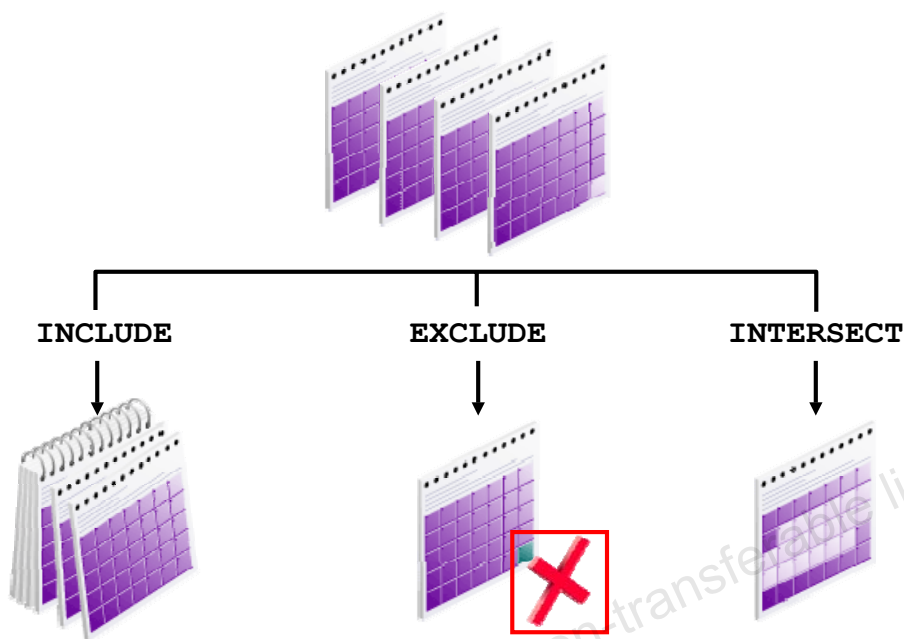
默认的调度程序事件队列是一个安全队列。可能需要配置队列，才能允许特定用户在此队列上执行操作，具体取决于应用程序。有关安全队列的信息，请参阅《Oracle Streams Concepts and Administration》文档。

默认的调度程序事件队列主要用于调度程序生成的事件。Oracle 建议您不要将此队列用于用户应用程序或用户定义的事件。

事件类型	说明
JOB_STARTED	作业已启动。
JOB_SUCCEEDED	作业已成功完成。
JOB_FAILED	作业因引发错误或异常终止而失败。
JOB_BROKEN	作业被禁用并更改为 BROKEN 状态，因为该作业超出了由 MAX_FAILURES 作业属性定义的失败次数。
JOB_COMPLETED	作业已完成，因为该作业达到了 MAX_RUNS 或 END_DATE 作业属性所设置的值。
JOB_STOPPED	作业由对 STOP_JOB 过程的调用停止。
JOB_SCH_LIM_REACHED	达到了作业调度的限制。未启动作业，因为启动作业过程中的延迟超出了 SCHEDULE_LIMIT 作业属性的值。
JOB_DISABLED	作业由调度程序禁用，或者由对 SET_ATTRIBUTE 过程的调用禁用。
JOB_CHAIN_STALLED	运行链的作业被置于 CHAIN_STALLED 状态。如果没有正在运行或计划运行的步骤，并且链的 EVALUATION_INTERVAL 设为 NULL，则正在运行的链将停止。该链将等待人工干预。
JOB_ALL_EVENTS	JOB_ALL_EVENTS 不是事件，而是一个常数，使您能够轻松启用所有事件。
JOB_OVER_MAX_DUR	作业的运行时间已达到了其设置所允许的最长时间。
JOB_RUN_COMPLETED	作业运行已完成。该作业可能已失败、已成功或者已停止。



## 创建复杂调度



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建复杂调度

调度是数据库中的对象。创建调度时，将自动保存这些调度。可以使用调度的组合来创建更复杂的调度。通过组合调度，可以向日历表达式中添加特定日期，或者从日历表达式中排除特定日期。

定义调度的重复间隔时可以使用下列选项：

- **INCLUDE:** 向日历表达式结果中添加日期列表
- **EXCLUDE:** 从日历表达式结果中删除日期列表
- **INTERSECT:** 只使用两个或多个调度共有的日期

创建要组合使用的调度时，可以对日期列表进行编码，方法是包含形式为 [YYYY]MMDD 的硬编码日期，或者包含用 CREATE\_SCHEDULE 过程创建的已命名调度。例如，可以对调度的重复间隔使用下列值来指定日期列表：

```
0115,0315,0325,0615,quarter_end_dates,1215
```

此字符串表示日期 1 月 15 日、3 月 15 日、3 月 25 日、6 月 15 日、12 月 15 日以及 QUARTER\_END\_DATES 调度所指定的日期列表。

如果未在调度中指定硬编码日期的可选年份，则包括每一年的这些日期。

## 测验

选择关于持久轻量作业的正确语句：

1. 持久轻量作业的作业元数据和运行时数据在磁盘上占用很小的空间。
2. 使用持久轻量作业可提供最大灵活性。
3. 持久轻量作业是使用作业模板创建的。
4. 可以在 Oracle Enterprise Manager 中或通过命令行创建持久轻量作业。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1、3

## 使用电子邮件通知

- 作业状态更改电子邮件通知
- 是由作业状态事件触发的
- 多个通知，多个收件人
- \*\_SCHEDULER\_NOTIFICATIONS 视图

使用调度程序电子邮件通知：

1. 指定您将用来发送电子邮件的 SMTP 服务器的地址：

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE
('email_server','host[:port]');
```

2. 还可以设置默认的发件人电子邮件地址：

```
DBMS_SCHEDULER.SET_SCHEDULER_ATTRIBUTE
('email_sender','valid email address');
```

3. 为指定作业添加电子邮件通知。 (续)

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用电子邮件通知

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，可以为不同的事件添加通知。可以将电子邮件通知发送给多个收件人。

要启用电子邮件通知功能，您必须：

1. 设置 email\_server 调度程序属性。
2. 此外，还可以使用 email\_sender 调度程序属性指定电子邮件通知的默认发件人电子邮件地址。
3. 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。

数据字典通过 \*\_SCHEDULER\_NOTIFICATIONS 视图支持电子邮件通知。

## 添加和删除电子邮件通知

```
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);
```

```
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 © 2011, Oracle. 保留所有权利。

### 添加和删除电子邮件通知

使用 DBMS\_SCHEDULER.ADD\_JOB\_EMAIL\_NOTIFICATION 过程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 中的规范匹配的事件将生成电子邮件。

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。调用该过程的用户必须是作业的所有者，必须具有 CREATE ANY JOB 系统权限，或已经被授予了对作业的 ALTER 权限。

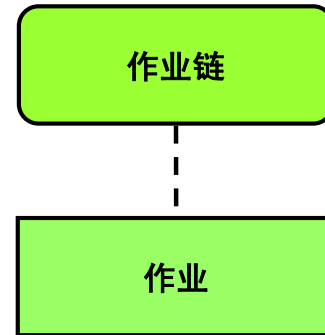
- 通知电子邮件的**主题**可以包含值将被替换的下列变量：%job\_owner%、%job\_name%、%event\_type%、%event\_timestamp%、%log\_id%、%error\_code%、%error\_message%、%run\_count%、%failure\_count%、%retry\_count%、%job\_subname%、%job\_class\_name%。
- 通知电子邮件的**正文**可以包含 SUBJECT 中有效的任何变量。
- 逗号分隔的**事件**列表不能为空。请参阅 JOBS 的 RAISE\_EVENTS 属性的事件列表以了解有效的事件。
- 如果 **filter\_condition** 为空（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。

### 添加和删除电子邮件通知（续）

使用 `DBMS_SCHEDULER.REMOVE_JOB_EMAIL_NOTIFICATION` 过程删除指定作业的一个或多个电子邮件通知。

## 创建作业链

1. 创建链对象。
2. 定义链步骤。
3. 定义链规则。
4. 启动链：
  - 启用链。
  - 创建指向链的作业。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建作业链

链是为达到组合目标而链接在一起的一系列已命名的程序。这被称为“依存调度”。链的示例如下所示：

运行程序 A 然后运行程序 B，如果程序 A 和程序 B 成功完成，则只运行程序 C，否则运行程序 D。

在相互依赖的程序构成的链中，每个位置都称为一个步骤。通常情况下，链的一系列初始步骤启动之后，后续步骤的执行依赖于一个或多个之前步骤的完成。要创建并使用链，请依次完成下列步骤。如果没有特别说明，则提到的所有过程都是 DBMS\_SCHEDULER 程序包的一部分。

1. 使用 CREATE\_CHAIN 过程**创建链**。可以选择用方案名称来限定链名称（如 `myschema.myname`）。
2. **定义**（一个或多个）**链步骤**。定义步骤时需要命名此步骤并指定步骤中发生的操作。每个步骤都可以指向下列项之一：
  - 程序
  - 另一个链（嵌套链）
  - 事件

## 创建作业链（续）

通过调用 `DEFINE_CHAIN_STEP` 过程来定义指向程序或嵌套链的步骤。

要定义等待事件发生的步骤，请使用 `DEFINE_CHAIN_EVENT_STEP` 过程。过程参数可以指向事件调度，也可以包括内嵌队列说明和事件条件。指向事件的步骤会一直等待，直到指定事件被引发。如果此事件发生，则步骤将成功完成。

3. 创建链对象后，请定义**链规则**。链规则定义各步骤运行的时间，并定义各步骤之间的依赖关系。每个规则都有一个“条件”和一个“操作”：

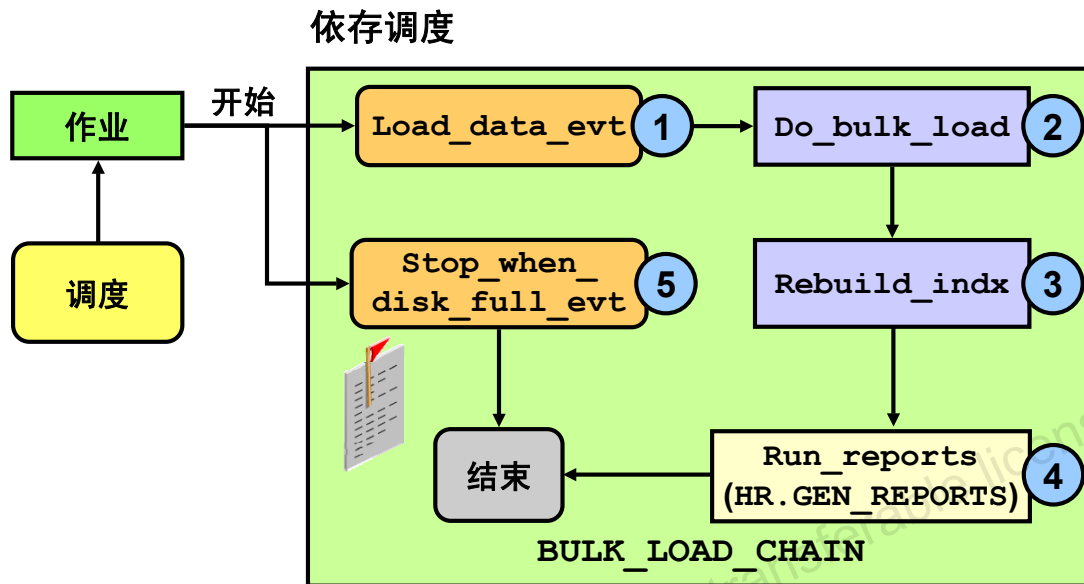
- 如果条件的求值结果为 `TRUE`，则执行操作。条件中可以包含在 `SQL WHERE` 子句中有效的任何语法。条件通常基于前面一个或多个步骤的结果。例如，您可能希望使某步骤在前两个步骤全部成功的情况下运行，而使另一个步骤在前两个步骤未全部成功的条件下运行。
- 操作指定在触发规则时要执行的内容。  
典型操作是运行指定的步骤。可能的操作包括启动或停止步骤。还可以选择结束作业链的执行、返回一个值或者返回步骤名和错误代码。

添加到链中的所有规则共同确定了链的整体行为。在作业启动时以及在每个步骤结束时，将对所有规则求值以确定接下来发生的操作。请使用 `DEFINE_CHAIN_RULE` 过程向链中添加规则。向链中添加每个规则时均需要调用一次此过程。

4. **启动链**包括两个操作：

- 用 `ENABLE` 过程启用链。（链总是以禁用状态创建，因此可以在任何作业执行该链之前向其中添加步骤和规则。）启用一个已启用的链不会返回错误。
- 要运行链，必须创建 '`CHAIN`' 类型的作业。作业操作必须引用链名称。可以为此作业使用基于事件的调度或基于时间的调度。

## 链的示例



ORACLE

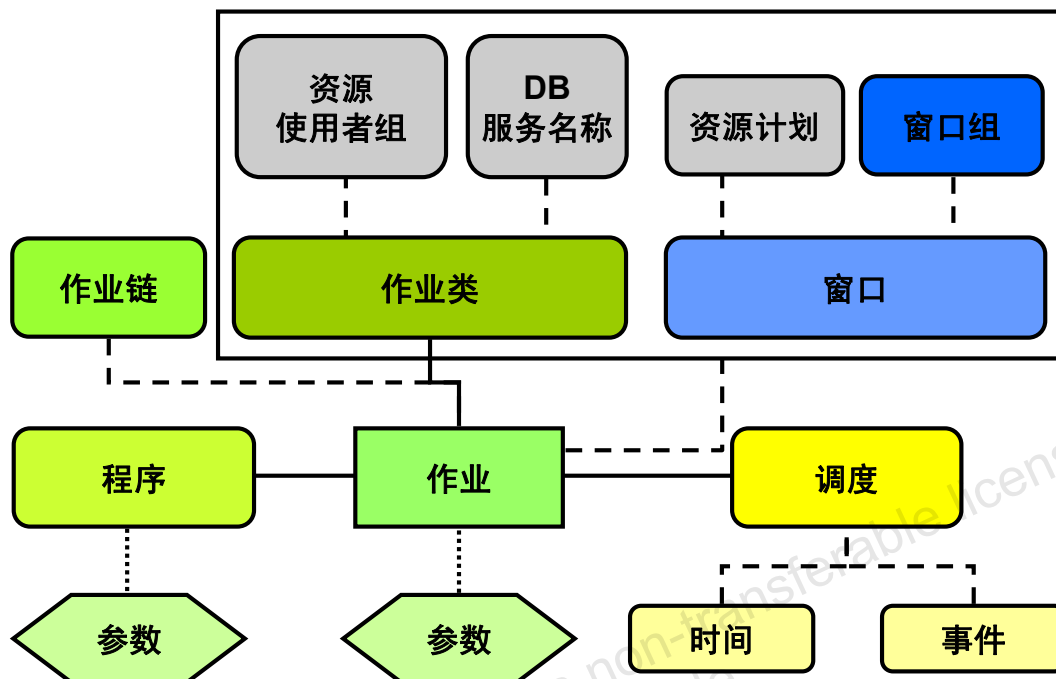
版权所有 © 2011, Oracle。保留所有权利。

### 链的示例

在此链示例中，请考虑在批量数据加载期间出现的所有任务和条件。首先，必须有要加载的数据。然后加载数据，同时观察文件系统以确保在加载期间不会发生空间不足的情况。在数据加载完成后，需要重建在更新后的表上定义的索引。然后针对新加载数据运行报表。该幻灯片显示了一个依存调度示例。



## 高级调度程序概念



版权所有 © 2011, Oracle。保留所有权利。

### 高级调度程序概念

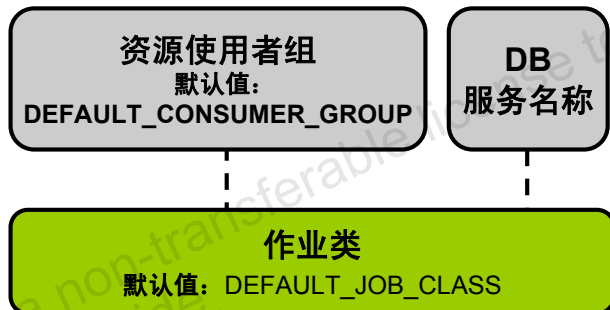
使用高级调度程序功能，可以对调度的各个方面施加更强的控制，例如作业窗口和区分作业优先级。下面汇总了这些高级功能，在后面的几张幻灯片中将对其进行详细讨论。

- “窗口”由定义好起始时间和结束时间的时间间隔表示，用于在不同时间激活不同的资源计划。这使您可以更改某一时段（如一天或销售年度内的某一时间段）的资源分配。
- “窗口组”表示一系列窗口，使用它可以更轻松地管理窗口。可以将窗口或窗口组用于作业的调度，以确保作业仅在窗口及其关联资源计划有效时运行。
- “作业类”定义了一类作业，这些作业具有共同的资源使用要求和其它特性。作业类将作业分组到更大的实体中。
- 与作业类关联的“资源使用者组”确定分配给作业类中作业的资源。
- 使用“资源计划”，用户可以在各资源使用者组中区分资源（特别是 CPU）的优先级。

注：灰色对象不是调度程序对象。

## 作业类

- 为其中的成员作业分配一组相同的属性值
- 是由 CREATE\_JOB\_CLASS 过程创建的
- 在作业类中指定作业（使用 SET\_ATTRIBUTE 过程）
- 属于 SYS 方案
- 为成员作业设置资源分配
- 将服务属性设置为所需的数据库服务名称
- 将作业分组以区分优先级



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 作业类

作业类是参与的成员作业的策略。每个作业类指定一组属性，例如日志记录级别。为作业类分配作业时，作业可以继承这些属性。例如，可以指定用相同的策略来清除所有工资单作业的日志项。

- 可以使用 CREATE\_JOB\_CLASS 过程来创建作业类。类始终属于 sys 方案。要创建类，必须具有 MANAGE\_SCHEDULER 权限。存在一个名为 DEFAULT\_JOB\_CLASS 的默认作业类，此作业类是随数据库一起创建的。
- 创建了作业类之后，可以在创建作业时或者在创建作业之后，将作业指定为此作业类的成员，方法是使用 DBMS\_SCHEDULER 程序包中的 SET\_ATTRIBUTE 过程。如果没有将作业与某个作业类相关联，则作业属于该默认作业类。
- 将作业类的服务属性设置为所需的数据库服务名称。这决定了 Real Application Clusters 环境中运行成员作业的实例和指定给成员作业的系统资源（后者可选）。

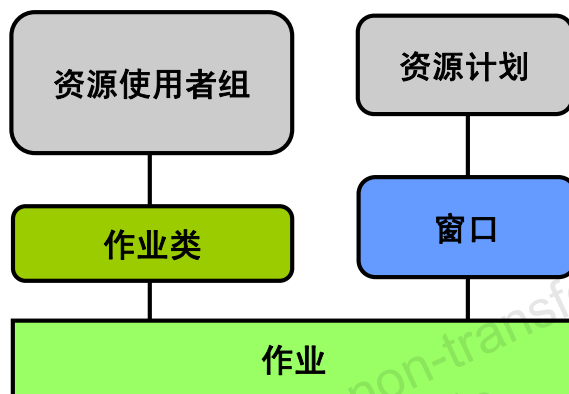
## 作业类（续）

- 为成员作业设置资源分配。作业类可以将数据库资源管理器与调度程序关联起来，因为每个作业类可以将一个资源使用者组指定为一个属性。这样成员作业就属于指定的使用者组，并根据当前资源计划中的设置分配资源。此外，还可以将 `resource_consumer_group` 属性保留为 NULL 并将作业类的服务属性设置为所需的数据库服务名称。然后，该服务可以映射到资源使用者组。如果同时设置了 `resource_consumer_group` 和服务属性并且指定的服务映射到资源使用者组，则优先使用 `resource_consumer_group` 属性中指定的资源使用者组。如果创建作业类时未指定资源使用者组，则作业类将映射到 `DEFAULT_CONSUMER_GROUP` 资源使用者组。启用了资源管理器后，对于默认作业类中的作业或者与默认资源使用者组关联的作业类中的作业，可能无法为其分配足够的资源以完成各自的任務。
- 为作业分组以区分优先级。可以为同一作业类中的单个作业分配 1 到 5 之间的优先级值，这样如果将该类中的两个作业安排在同一时间启动，优先级较高的作业优先。这样可以确保次要作业不会阻碍重要作业及时完成。如果为两个作业指定的优先级值相同，启动日期较早的作业优先。如果没有为作业指定优先级值，则其优先级默认为 3。

## 窗口

调度程序窗口：

- 可以在不同时间段启动作业或更改资源在作业间的分配
- 一次只有一个是活动的
- 是使用 CREATE\_WINDOW 过程创建的



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 窗口

作业的优先级可以随时间而变化。例如，在夜间可能要将大部分数据库资源分配给数据仓库加载作业，而在白天则将较大一部分资源分配给应用程序作业。要实现这一点，可以使用调度程序窗口来更改数据库资源计划。

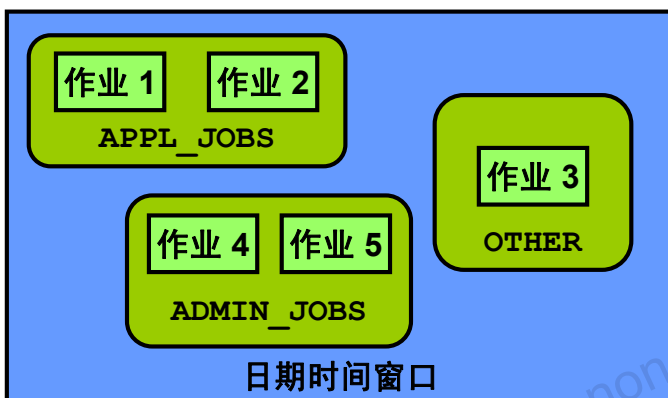
- 调度程序窗口可以在一天、一周等的不同时间段自动启动作业或更改资源在作业间的分配。窗口由定义好起始时间和结束时间的时间间隔表示，例如“从凌晨 12:00 到早上 6:00”。
- 在任意给定时间内，只能存在一个有效的窗口。
- 可以使用 CREATE\_WINDOW 过程创建窗口。

调度程序窗口使用作业类来控制资源分配。每个窗口指定了在该窗口打开（生效）时激活的资源计划，而每个作业类指定了资源使用者组或可以映射到使用者组的数据库服务。因此，在窗口中运行的作业所拥有的资源是根据其作业类的使用者组和窗口的资源计划分配的（如幻灯片中的图形所示）。

## 区分窗口内作业的优先级

区分作业优先级：

- 在类级别（通过资源计划）
- 在作业级别（使用作业优先级属性）
- 不能担保不同作业类中的作业的优先级



作业	优先级
作业 1	1
作业 2	2
作业 3	3
作业 4	5
作业 5	2

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 区分窗口内作业的优先级

在一个数据库中创建多个作业时，您需要通过某种方式来根据您的业务需求安排作业处理，并指定哪些作业具有最高优先级。对于特定窗口，可能有多种类别的作业在运行，每一类都具有各自的优先级。

可以在两个级别区分优先级：类级别和作业级别。

- 首先是在类级别区分优先级，使用的是资源计划。完全根据类资源分配情况来区分属于不同类的作业的优先级。
- 其次是在类内区分优先级，使用的是作业的作业优先级属性。

仅当属于同一个类的两个作业同时启动时，才会考虑优先级。优先级较高的作业先启动。不能确保不同作业类中的作业按其优先级运行。例如，APPL\_JOBS 作业类中的高优先级作业不一定在 ADMIN\_JOBS 作业类中的低优先级作业之前启动，即使它们共享同一个计划时也是如此。如果 APPL\_JOBS 作业类有较低级别的资源可用，则该类中的高优先级作业必须等待资源可用，即使在其它作业类中存在可用于低优先级作业的资源时也是如此。

## 创建作业数组

### 1. 声明 `sys.job` 和 `sys.job_array` 类型的变量:

```
DECLARE
  newjob sys.job;
  newjobarr sys.job_array;
```

### 2. 初始化作业数组:

```
BEGIN
  newjobarr := SYS.JOB_ARRAY();
```

### 3. 调整作业数组大小以存储所需的作业数量:

```
newjobarr.EXTEND(100);
```

(...续)

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 创建作业数组

使用作业数组是一种效率较高的创建作业集的方式。这也适用于轻量作业。在幻灯片的示例中，在一个作业数组中创建了 100 个作业说明，并且将这些说明提交到了单个事务处理的作业队列中。请注意，对于轻量作业，所需的信息量非常有限。在此示例中，`start_time` 参数默认为 NULL，因此作业被安排为立即启动。

1. 声明用于存储作业定义的变量和一个作业数组变量。
2. 使用 `SYS.JOB_ARRAY` 构造器初始化作业数组。这将为数组中的每个作业创建一个位置。
3. 将数组大小设置为预期的作业数。
4. 创建每个作业，并将其放入数组中。在本幻灯片示例中，唯一的差异是作业的名称。作业的 `start_time` 变量将被省略并默认为 NULL，表示将立即运行该作业。
5. 使用 `CREATE_JOBS` 过程将数组中的所有作业作为一个事务处理提交。

**注：**如果数组很小，则性能不会显著优于提交单个作业时的性能。

## 创建作业数组

### 4. 将作业放入作业数组:

```
FOR i IN 1..100 LOOP
    newjob := SYS.JOB(job_name => 'LWTJK' || to_char(i),
                     job_style => 'LIGHTWEIGHT',
                     job_template => 'MY_PROG',
                     enabled => TRUE );
    newjobarr(i) := newjob;
END LOOP;
```

### 5. 将作业数组作为一个事务处理提交:

```
DBMS_SCHEDULER.CREATE_JOBS(newjobarr,
                             'TRANSACTIONAL');
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建作业数组（续）

此示例的全部代码如下:

```
DECLARE
    newjob sys.job;
    newjobarr sys.job_array;
BEGIN
    -- Create an array of JOB object types
    newjobarr := sys.job_array();
    -- Allocate sufficient space in the array
    newjobarr.extend(100);
    -- Add definitions for jobs
    FOR i IN 1..100 LOOP
        -- Create a JOB object type
        newjob := sys.job(job_name => 'LWTJK' || to_char(i),
                         job_style => 'LIGHTWEIGHT',
                         job_template => 'PROG_1',
                         enabled => TRUE );
        -- Add job to the array
        newjobarr(i) := newjob;
    END LOOP;
    -- Call CREATE_JOBS to create jobs in one transaction
    DBMS_SCHEDULER.CREATE_JOBS(newjobarr, 'TRANSACTIONAL');
END;
```

## 测验

选择与高级调度程序概念和功能有关的所有正确语句：

1. 可以使用作业数组创建轻量作业。
2. 在类级别区分作业优先级（通过资源计划）和在作业级别区分作业优先级（使用作业优先级属性）是两种互斥的操作。
3. 调度程序窗口使用作业类来控制资源分配。
4. 作业链用于实施“依存调度”。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1、3、4



## 创建文件监视器和基于事件的作业

请执行以下任务：

1. 创建调度程序身份证明对象并授予 EXECUTE 权限。
2. 创建文件监视器并授予 EXECUTE 权限。
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。（可选）使作业能够针对文件到达事件的每个实例运行。
5. 启用文件监视器、程序和作业。

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 创建文件监视器和基于事件的作业

执行以下任务来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：

1. 创建调度程序身份证明对象（身份证明），用于通过主机操作系统验证对文件的访问权限，并将对身份证明的 EXECUTE 权限授予文件监视器将启动的基于事件的作业所属的方案。
2. 创建文件监视器并将对文件监视器的 EXECUTE 权限授予引用文件监视器的基于事件的作业所属的任何方案。
3. 创建其元数据参数引用事件消息的调度程序程序对象。
  - 使用 EVENT\_MESSAGE 属性定义元数据参数。
  - 创建具有一个类型为 SYS.SCHEDULER\_FILEWATCHER\_RESULT 的参数的存储过程，供该程序调用。存储过程必须具有一个 SYS.SCHEDULER\_FILEWATCHER\_RESULT 类型（事件消息的数据类型）的参数。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。

## 创建文件监视器和基于事件的作业（续）

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` 过程使作业能够针对文件到达事件的每个实例运行，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`。

```
BEGIN
  DBMS_SCHEDULER.SET_ATTRIBUTE('','PARALLEL_INSTANCES', TRUE);
END;
```

这样作业将作为轻量作业运行，因此可以快速启动作业的多个实例。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。

5. 启用文件监视器、程序和作业。

## 从远程系统启用文件到达事件

执行以下任务在远程系统上启用文件到达事件触发功能：

1. 设置数据库以运行远程外部作业。
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. 为其余的每个远程系统重复步骤 2。



版权所有 © 2011, Oracle。保留所有权利。

### 从远程系统启用文件到达事件

要接收来自远程系统的文件到达事件，您必须在系统上安装调度程序代理，且必须在数据库中注册代理。该远程系统不需要 Oracle DB 实例来生成文件到达事件。

有关详细信息，请参阅《Oracle Database Administrator's Guide 11g Release 2》。

## 调度远程数据库作业

- 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。
- DBMS\_SCHEDULER.CREATE\_DATABASE\_DESTINATION 和 DBMS\_SCHEDULER.CREATE\_CREDENTIAL 可以用于远程数据库作业。
- 作业类型为 PLSQL\_BLOCK 和 STORED\_PROCEDURE 的作业可以是 SET\_ATTRIBUTE 为 DESTINATION 和 CREDENTIAL 属性调用的目标。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 调度远程数据库作业

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。以下几页提供了详细信息。

## 创建远程数据库作业

执行以下任务以创建远程作业：

1. 设置远程作业的发起数据库。
2. 使用 `DBMS_SCHEDULER.CREATE_JOB` 创建作业。
3. 使用 `DBMS_SCHEDULER.CREATE_CREDENTIAL` 创建身份证明。
4. 使用 `DBMS_SCHEDULER.SET_ATTRIBUTE` 设置作业 `CREDENTIAL_NAME`。
5. 使用 `DBMS_SCHEDULER.SET_ATTRIBUTE` 设置作业的 `DESTINATION` 属性。
6. 使用 `DBMS_SCHEDULER.ENABLE` 启用作业。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建远程数据库作业

您可以执行幻灯片中列出的任务来创建远程数据库作业。

要设置远程作业的发起数据库，请执行以下步骤：

1. 验证是否已安装 XML DB。
2. 启用至数据库的 HTTP 连接。

```
BEGIN
  DBMS_XDB.SETHTTPPORT(port);
END;
```

3. 执行 `prvtrsch.plb` 脚本。
4. 设置调度程序代理的注册口令。

```
BEGIN
  DBMS_SCHEDULER.SET_AGENT_REGISTRATION_PASS('password');
END;
```

有关详细示例，请参阅《Oracle Database Administrator's Guide 11g Release 2》。

## 调度多个目标作业

- 该功能可用于指定要在其上执行作业的多个目标。
- 它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 © 2011, Oracle。保留所有权利。

### 调度多个目标作业

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作为单个实体进行修改
- 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。

## 查看调度程序元数据

主要调度程序管理视图，显示：

- \*\_SCHEDULER\_JOBS：所有作业，包括启用的和禁用的
- \*\_SCHEDULER\_SCHEDULES：所有调度
- \*\_SCHEDULER\_PROGRAMS：所有程序
- \*\_SCHEDULER\_RUNNING\_JOBS：活动的作业状态
- \*\_SCHEDULER\_JOB\_LOG：所有作业状态更改
- \*\_SCHEDULER\_JOB\_RUN\_DETAILS：所有已完成的作业运行

```
SELECT job_name, status, error#, run_duration
FROM USER_SCHEDULER_JOB_RUN_DETAILS;
```

JOB_NAME	STATUS	ERROR#	RUN_DURATION
GATHER_STATS_JOB	SUCCESS	0	+000 00:08:20
PART_EXCHANGE_JOB	FAILURE	6576	+000 00:00:00

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 查看调度程序元数据

作业表是一个包含所有作业的容器，每个数据库有一个作业表。作业表用于存储所有作业的信息，例如所有者名称和日志记录级别。可以在 \*\_SCHEDULER\_JOBS 视图中查看这些信息。

作业是数据库对象，因此可能会累积起来并占用过多的空间。为避免发生这种情况，默认情况下会在作业完成后自动删除作业对象。该行为由 auto\_drop 作业属性控制。

DBA 和授权用户可以通过多个视图查看与调度程序、作业、调度和窗口等有关的重要操作信息。这些视图包括：

- \*\_SCHEDULER\_PROGRAM\_ARGS：显示为所有程序定义的所有参数及默认值（如果存在）
- \*\_SCHEDULER\_JOBS：显示所有的作业，不管是已启用还是禁用的作业
- \*\_SCHEDULER\_JOB\_RUN\_DETAILS 显示所有已完成（失败或成功）的作业运行。每个作业实例都有对应的一行。每一行中都包含有关该实例的作业执行情况的信息。ERROR# 是遇到的第一个错误的编号
- \*\_SCHEDULER\_GLOBAL\_ATTRIBUTE：显示调度程序属性的当前值
- \*\_SCHEDULER\_JOB\_ARGS：显示所有作业的所有已设置的参数值
- \*\_SCHEDULER\_JOB\_CLASSES：显示所有作业类

## 调度程序数据字典视图（续）

对于作业链：

- \*\_SCHEDULER\_RUNNING\_CHAINS：显示所有活动链
- \*\_SCHEDULER\_CHAIN\_STEPS：显示所有链的所有步骤
- \*\_SCHEDULER\_CHAINS：显示所有链
- \*\_SCHEDULER\_CHAIN\_RULES：显示所有链的所有规则

对于其它高级对象窗口：

- \*\_SCHEDULER\_WINDOWS 显示所有窗口
- \*\_SCHEDULER\_WINDOW\_GROUPS 显示所有窗口组
- \*\_SCHEDULER\_WINDOWGROUP\_MEMBERS 显示所有窗口组的成员，一行对应一个组成员
- \*\_SCHEDULER\_JOB\_LOG 显示作业的所有状态更改
- \*\_SCHEDULER\_CREDENTIALS 显示数据库中的身份证明的列表，其中密码进行了模糊处理
- \*\_SCHEDULER\_JOB\_ROLES 按数据库角色显示所有作业

轻量作业与常规作业是通过相同的视图查看的：

- \*\_SCHEDULER\_JOBS：显示所有作业，包括 JOB\_STYLE='LIGHTWEIGHT' 的作业
- \*\_SCHEDULER\_JOB\_ARGS：还显示轻量作业的所有已设置的参数值
- 因为轻量作业不是数据库对象，所以无法通过 \*\_OBJECTS 视图查看轻量作业

从 Oracle Database 11gR2 开始：

- \*\_SCHEDULER\_NOTIFICATIONS 显示已经设置的电子邮件通知
- \*\_SCHEDULER\_FILE\_WATCHERS 显示文件监视器的配置信息

以下视图显示与多目标作业有关的信息：

- \*\_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且可用作远程外部作业目标的代理
- \*\_SCHEDULER\_DB\_DESTS：显示您可在其上调度远程数据库作业的所有数据库
- \*\_SCHEDULER\_GROUPS：显示您的方案中的组或数据库中的所有组
- \*\_SCHEDULER\_GROUP\_MEMBERS：显示您的方案中的组成员或数据库中的所有组成员
- \*\_SCHEDULER\_JOB\_DESTS：显示远程数据库中的作业的状态

**注：**在上面列出的视图中，视图名称开头的星号会被替换为 DBA、ALL 或 USER。



## 测验

选择关于 Oracle 调度程序的所有正确语句：

1. 创建远程数据库作业是一个手动任务，需要使用特定于操作系统的命令。
2. 调度程序身份证明是一个对象，用于通过主机操作系统验证对文件的访问权限。
3. 您可以指定应在其上执行作业的多个目标，并从创建作业的数据库上监视这些作业。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、3

## 小结

在本课中，您应该已经学会：

- 使用调度程序来简化管理任务
- 创建作业、程序和调度
- 监视作业执行
- 使用基于时间或基于事件的调度来执行调度程序作业
- 描述窗口、窗口组、作业类和使用组的目的
- 使用电子邮件通知
- 使用作业链来执行一系列相关任务
- 描述远程系统上的调度程序作业
- 使用高级调度程序概念来区分作业的优先级

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 17 概览：使用调度程序自动执行任务

本练习包含以下主题：

- 创建在数据库外运行程序的作业
- 创建程序和调度
- 创建使用程序和调度的作业
- 创建轻量作业
- 监视作业运行情况

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 附注

此练习同时使用了 Oracle Enterprise Manager Database Control 和 SQL\*Plus。

华周 (286991486@qq.com) has a non-transferable license to use  
this Student Guide.

# 18

## 管理空间

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 介绍 Oracle DB Server 如何自动管理空间
- 使用压缩节省空间
- 主动监视和管理表空间的空间使用量
- 介绍 Oracle DB 中的段的创建
- 控制延迟创建段
- 使用“段指导”
- 使用段收缩功能从表和索引中回收浪费的空间
- 管理可恢复的空间分配

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 空间管理：概览

空间由 Oracle DB Server 自动管理。Oracle DB Server 可以生成有关潜在问题的预警，并推荐可能的解决方案。这些功能包括：

- Oracle Managed Files (OMF)
- 用位图进行的空闲空间管理（“本地管理”）和数据文件自动扩展
- 主动的空间管理（默认阈值和服务器生成的预警）
- 空间回收（收缩段、联机重新定义表）
- 容量计划（增长报表）

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 空间管理：概览

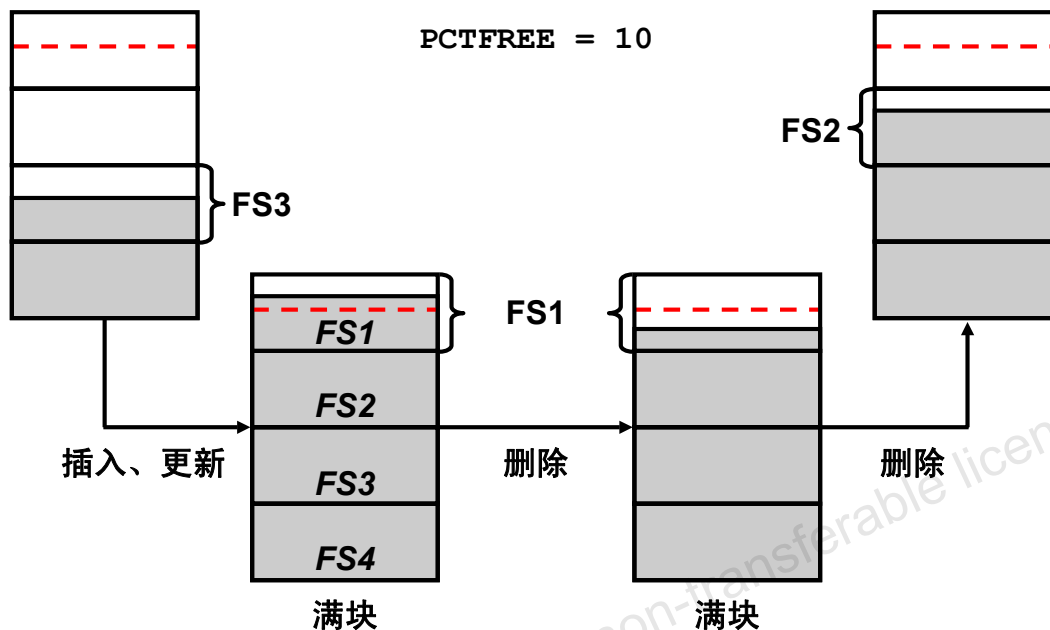
通过使用 Oracle Managed Files (OMF)，您可以根据数据库对象（而不是文件名）来指定操作。Oracle DB Server 可以用位图来管理表空间内的空闲空间。这称为“本地管理的”表空间。此外，位于本地管理的表空间中的段内的空闲空间可以使用位图进行管理。这称为“自动段空间管理”。位图化的实施可以避免很多与空间相关的表优化工作，同时能在高峰负载期间提供更高的性能。另外，Oracle DB Server 还可以自动扩展数据文件，因此文件可以根据其中的数据量自动增长。

创建数据库时，默认情况下将启用预先空间监视。（这不会对性能产生任何影响。）Oracle DB Server 在正常空间分配和取消分配操作期间监视空间使用情况，如果空闲空间的可用性低于预定义的阈值（可以覆盖此阈值），将发出预警。指导和向导可帮助您回收空间。

计划容量时，Oracle DB Server 根据表结构和行数进行空间估计，并根据自动工作量资料档案库 (AWR) 中存储的历史空间使用情况来提供增长趋势报表。

（本课和后面的课程将介绍这些主题。）

## 块空间管理



版权所有 © 2011, Oracle。保留所有权利。

### 块空间管理

空间管理包括块级别的空闲空间管理。使用“自动段空间管理”时，每个块分为四部分，分别称为 FS1（空闲空间介于 0 到 25% 之间）、FS2（25% 到 50% 空闲）、FS3（50% 到 75% 空闲）和 FS4（75% 到 100% 空闲）。

块的状态将根据其中空闲空间的级别自动更新。这样，您就可以根据所插入行的长度来判断特定块能否满足插入操作的要求。请注意，“满”状态表示不能再向块中插入。

在幻灯片示例中，左侧的块是一个 FS3 块，因为其空闲空间介于 50% 和 75% 之间。执行了一些插入和更新语句后，将达到 PCTFREE（虚线处），并且不能再向该块中插入新行。现在该块被视为“满”或 FS1 块。块的空闲空间级别一旦降至低于下一部分，就会考虑将该块再次用于插入。在上面这种情况中，空闲空间一旦超过 25%，状态即变为 FS2。

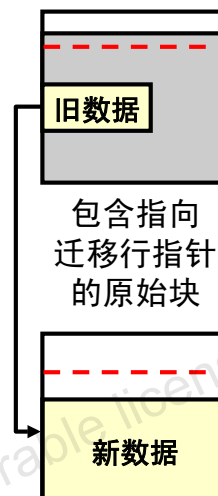
**注：**大对象 (LOB) 数据类型 (BLOB、CLOB、NCLOB 和 BFILE) 不使用 PCTFREE 存储参数。未压缩块和 OLTP 压缩块的默认 PCTFREE 值为 10；基本压缩块的默认 PCTFREE 值为 0。



## 行链接和行迁移

示例：

- **更新时：** 行的长度增加，超过了块中的可用空闲空间。
- 需要将数据存储在新块中。
- 将保留行的原始物理标识符 (ROWID)。
- Oracle DB Server 需要读取两个块来检索数据。
- “段指导” 查找包含迁移行的段。
- 可以对块中的碎片空闲空间进行自动合并。



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 行链接和行迁移

在以下两种情况下，表中某行的数据可能太大，无法包含在单个数据块中。第一种情况：在第一次插入行时，就发现行太大，无法包含在一个数据块中。在这种情况下，Oracle DB Server 将在为该段保留的一系列数据块（一个或多个）中存储该行的数据。较大的行通常要进行行链接，例如包含数据类型为 LONG 或 LONG RAW 的列的行。在这些情况下，行链接是无法避免的。

但在第二种情况中，最初可以包含在一个数据块中的行在更新后，整个行的长度增加了，而块的空闲空间已经完全填满。在这种情况下，Oracle DB Server 假定整个行可以包含在新数据块中，从而将整个行的数据迁移到新数据块中。数据库将保留迁移行的原始行片段，以便指向包含迁移行的新块。迁移行的 ROWID 不变。

当行链接或迁移后，与此行关联的输入/输出 (I/O) 性能会降低，因为 Oracle DB Server 必须扫描多个数据块来检索该行的信息。

“段指导” 查找包含由 UPDATE 操作产生的迁移行的段。

### 行链接和行迁移（续）

Oracle DB 在以下情况下会以自动透明方式合并数据块的空闲空间：

- INSERT 或 UPDATE 语句尝试使用包含足够空闲空间的块来容纳新的行片段
- 空闲空间已成为碎片，以致于行片段无法插入到块的相邻部分

合并之后，空闲空间量与操作前的空间量相同，但空间现在是连续的。

## 测验

当行链接或迁移后，与此行关联的 I/O 性能会降低，因为 Oracle DB Server 必须扫描多个数据块来检索该行的信息。

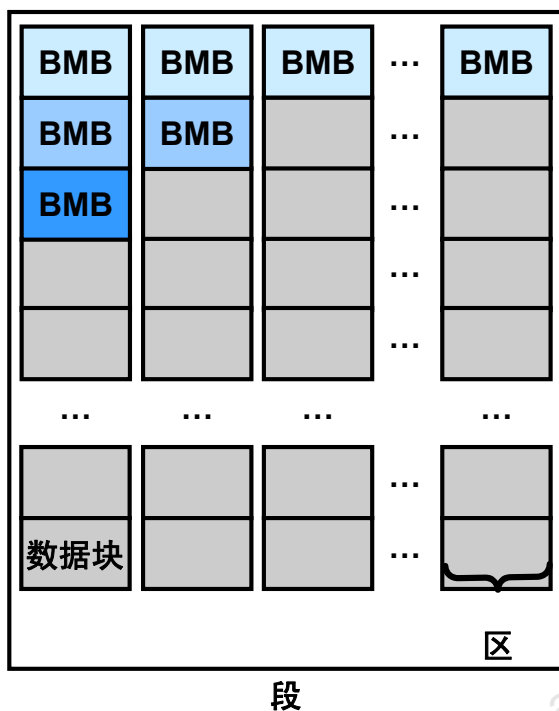
1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 段内的空闲空间管理



- 由段中的位图跟踪
- 好处:
- 空间使用更加灵活
  - 运行时调整
  - BMB 的多进程搜索

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 空闲空间管理

可以自动管理数据库段内的空闲空间。段内的空闲空间或已用空间由位图跟踪。要利用此功能，请在创建本地管理的表空间时，指定“自动段空间管理”。然后，指定的设置将应用于随后在此表空间中创建的所有段。

自动空间管理段具有一组位图块 (BMB)，用于描述该段中数据块的空间使用情况。BMB 以树形层次结构组织。层次的根层存储在段头部中，其中包含对所有中间层 BMB 的引用。此层次的各个叶节点代表属于该段的一组连续数据块的空间信息。此层次的最大层数为三。

使用自动空间管理的好处包括：

- 改善空间使用情况，尤其是各行大小差异很大的对象的空间使用情况
- 对并行访问发生的变化进行更好的运行时调整
- 在性能方面或空间使用方面改善多实例行为

因此，DBA 可以少做一些工作。

## 段的类型

段是为某个逻辑结构分配的一组区。段包括以下不同类型：

- 表和簇段
- 索引段
- 还原段
- 临时段

段由 Oracle DB Server 动态分配。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 段的类型

**表和簇段：**每个非聚簇表都有一个数据段。所有表数据都存储在表段的区中。对于分区表，每个分区都有一个数据段。每个集群也都有一个数据段。集群中每个表的数据都存储在集群的数据段中。

**索引段：**每个索引都有一个索引段，存储其所有数据。对于分区索引，每个分区都有一个索引段。

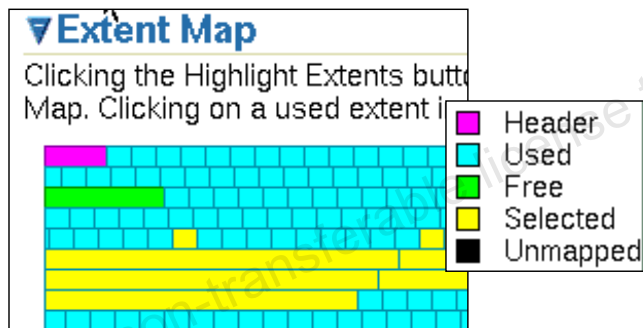
**还原段：**Oracle DB 会维护用于回退对数据库所做更改的信息。此信息包括事务处理操作的记录；这些操作统称为还原。还原信息存储在还原表空间的还原段中。

**临时段：**临时段是在 SQL 语句需要临时数据库区域来完成执行时由 Oracle DB Server 创建的。语句完成执行后，临时段中的区将返回到系统，以备将来使用。

如果段的现有区变满，Oracle DB Server 将动态分配空间。因为区是根据需要分配的，所以段中的区在磁盘上可能连续，也可能不连续。

## 分配区

- 搜索数据文件的位图，以便获取所需数目的相邻空闲块
- 可用以下存储子句调整区的大小：
  - UNIFORM
  - AUTOALLOCATE
- 查看区映射
- 获得取消分配建议



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 分配区

对于本地管理的表空间，Oracle DB Server 将查找要分配给新区的空闲空间，具体方法是：先在表空间中确定候选数据文件，然后搜索数据文件的位图，以便获取所需数目的相邻空闲块。如果该数据文件没有足够的相邻空闲空间，则 Oracle DB Server 将在另一数据文件中查找。

以下两个子句会影响区的大小调整：

- 使用 UNIFORM 子句，数据库可以按您指定的统一大小（或默认大小）为表空间中创建的所有对象创建所有区。
- 使用 AUTOALLOCATE 子句，数据库可以为表空间确定区大小调整策略。

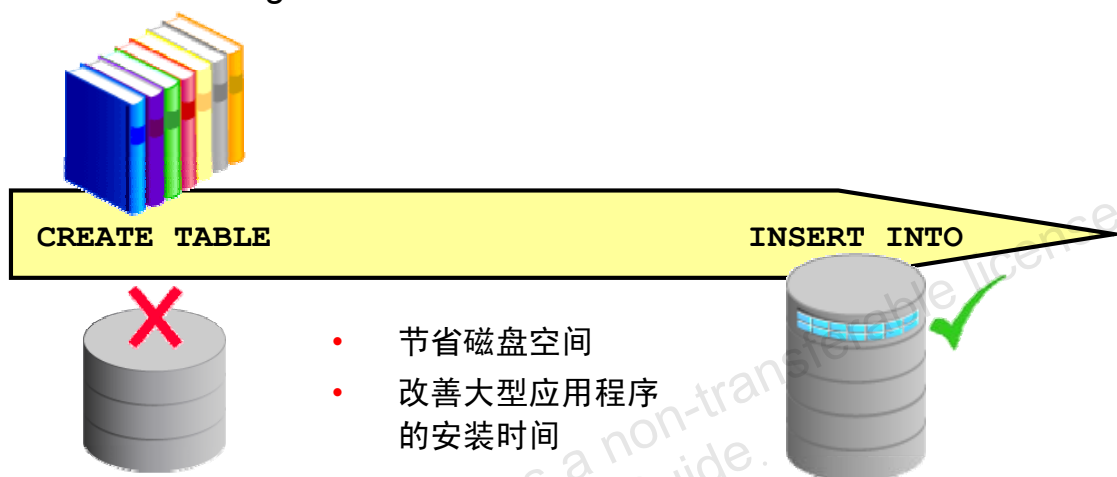
要在 Oracle Enterprise Manager 中查看区映射，请选择“Server > Tablespaces > View Tablespace > Show Tablespace Contents（服务器 > 表空间 > 查看表空间 > 显示表空间内容）”。

Oracle DB Server 提供了“Segment Advisor（段指导）”，使用该指导可以根据对象中空间碎片的级别来帮助确定对象是否有空间可回收。

## 分配空间

新的空间分配方法：

- DEFERRED\_SEGMENT\_CREATION = TRUE (默认值)
1. Table creation > Data dictionary operation (创建表 > 数据字典操作)
  2. DML > Segment creation (DML > 创建段)



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 分配空间

Oracle Database 11gR2 包含了一种新的空间分配方法。创建非分区堆表时，表段创建将延迟，直至插入第一行。该功能是默认启用的，即 DEFERRED\_SEGMENT\_CREATION 初始化参数设置为 TRUE。

此新增的空间分配方法的优点包括：

- 对于安装时就会创建成百或上千个表（其中大多数可能永远不会填入内容）的应用程序而言，可以节省大量的磁盘空间。
- 缩短了应用程序安装时间。

在向表中插入第一行后，系统将为基表、其 LOB 列及其索引创建段。在创建段期间，表上的游标将失效。这些操作会对性能产生较小的额外影响。

**注：**使用该新增的分配方法时，务必制定合适的容量计划以便有足够的磁盘空间用于处理填充表时的段创建操作，这一点非常重要。有关详细资料，请参阅《Oracle 数据库管理员指南》。

## 创建没有段的表

```
SQL> SHOW PARAMETERS deferred_segment_creation
NAME                                TYPE                                VALUE
-----
deferred_segment_creation           boolean                             TRUE

SQL> CREATE TABLE seg_test(c number, d varchar2(500));
Table created.
SQL> SELECT segment_name FROM user_segments;
no rows selected
```

插入行并创建段：

```
SQL> INSERT INTO seg_test VALUES(1, 'aaaaaaa');
1 row created.

SQL> SELECT segment_name FROM user_segments;
SEGMENT_NAME
-----
SEG_TEST
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建没有段的表

本幻灯片显示了如何检查 DEFERRED\_SEGMENT\_CREATION 参数。然后创建一个没有段的表，可通过查询 USER\_SEGMENTS 数据字典视图进行验证。插入一行后，再次查询该视图以查看现在是否存在段。

您还可以查询 USER\_TABLES、USER\_INDEXES 或 USER\_LOBS 视图的 SEGMENT\_CREATED 列。对于非分区表、索引和 LOB，如果创建了段，该列将显示 YES。数据字典中增加的另一个对象是 SYS.SEG\$ 表，该表用于存储在创建表或索引期间指定的存储参数。



## 控制延迟创建段

在以下项中使用 DEFERRED\_SEGMENT\_CREATION 参数：

- 初始化文件
- ALTER SESSION 命令
- ALTER SYSTEM 命令

使用 SEGMENT CREATION 子句：

- IMMEDIATE
- DEFERRED (Oracle Database 11gR2 中的默认值)

```
CREATE TABLE SEG_TAB3 (C1 number, C2 number)
    SEGMENT CREATION IMMEDIATE TABLESPACE SEG_TBS;
CREATE TABLE SEG_TAB4 (C1 number, C2 number)
    SEGMENT CREATION DEFERRED;
```

注：索引继承表的特征。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 控制延迟创建段

可通过以下两种方式控制段的创建：

- 将 DEFERRED\_SEGMENT\_CREATION 初始化参数设置为 TRUE 或 FALSE。可在初始化文件中设置该参数。还可以通过 ALTER SESSION 或 ALTER SYSTEM 命令进行控制。

示例：

```
ALTER SESSION SET DEFERRED_SEGMENT_CREATION = TRUE;
ALTER SYSTEM SET DEFERRED_SEGMENT_CREATION = FALSE;
```

- 使用 CREATE TABLE 命令的 SEGMENT CREATION 子句：
  - SEGMENT CREATION DEFERRED: 如果指定，则系统将延迟创建段直至表中插入第一行。这是 Oracle Database 11gR2 的默认行为。
  - SEGMENT CREATION IMMEDIATE: 如果指定，则在表创建期间对段进行实体化。这是早于 Oracle Database 11gR2 的 Oracle DB 中的默认行为。

该子句优先于 DEFERRED\_SEGMENT\_CREATION 参数。

可以使用 ALTER TABLE ... MOVE 命令为已创建的表强制创建段。

不过，对于从属对象（如索引），无法直接控制延迟创建段。它们继承其父对象（在本例中为表）的这一特征。

## 限制和例外

按需创建段：

- 仅适用于非分区表和索引
- 不适用于 IOT、聚簇表或其它特殊表
- 不适用于字典管理的表空间中的表

注：如果计划将没有段的表从本地管理的表空间迁移到字典管理的表空间，则必须将其删除并重新进行创建。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 限制和例外

- 在 Oracle Database 11gR2 中，延迟创建段被限制于非分区表和非分区索引。不支持 IOT 和其它特殊表。
- 对于聚簇表和在字典管理的表空间中创建的表，不支持按需创建段。如果尝试这样做，系统将创建段。
- 如果在本地管理的表空间中创建了一个采用延迟创建段的表，则该表没有段。如果稍后将该表空间迁移到字典管理的表空间，则任何创建段的尝试都将产生错误。此时，必须将该表删除并重新进行创建。
- 对于聚簇表、全局临时表、特定于会话的临时表、内部表、类型化表、AQ 表、SYS 拥有的表、外部表、位图联接索引以及域索引，不支持按需创建段。归 SYSTEM、PUBLIC、OUTLN 以及 XDB 所有的表也不在支持范围内。

## 其它自动功能

没有用户干预：

- 不能用的索引和索引分区没有段
- 创建没有段的索引：

```
CREATE INDEX test_i1 ON seg_test(c) UNUSABLE;
```

- 删除索引的任何已分配空间：

```
ALTER INDEX test_i UNUSABLE;
```

- 为索引创建段：

```
ALTER INDEX test_i REBUILD;
```

```
SELECT segment_name, partition_name,
       segment_type
FROM   user_segments
WHERE  segment_name like '%DEMO';
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 其它自动功能

Oracle Database 11g 发行版 2 中还实施了其它增强功能（与延迟创建段无关）以节省空间：创建的所有 UNUSABLE 索引和索引分区都没有段。该功能对您是完全透明的。该功能默认为启用，即 COMPATIBILITY 初始刷参数设置为 11.2.0.0。

**示例：**如果您有一个 DEMO 表，它有三个分区和一个本地索引，则在执行查询时会看到三个表段和三个索引段，如幻灯片中所示。

如果将一个表分区移至新的表空间，然后执行同样的查询，您将看到三个表段和两个索引段，这是因为不可用的一个已经自动删除。

## 测验

在下列关于 Oracle Database 11g 发行版 2 的说法中，  
哪一个是正确的？

1. 延迟创建段始终处于启用状态。您无法对其进行控制。
2. 可使用 CREATE TABLE 命令的 SEGMENT CREATION 子句控制延迟创建段。
3. 按需创建段适用于所有类型的表，包括 SYS 用户拥有的表。
4. 按需创建段适用于非分区表。



版权所有 © 2011, Oracle。保留所有权利。

答案：2、4

## 表压缩：概览

通过压缩所有数据减少存储成本：

- 基本压缩适用于直接路径插入操作：10x
- OLTP 压缩适用于所有 DML 操作：2–4x

```
{ COMPRESS [ BASIC | FOR { OLTP } ] | NOCOMPRESS }
```

压缩方法	压缩比率	CPU 开销	CREATE 和 ALTER TABLE 语法	典型应用
基本表压缩	高	最小	COMPRESS [BASIC]	DSS
OLTP 表压缩	高	最小	COMPRESS FOR OLTP	OLTP、DSS

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 表压缩：概览

Oracle DB 支持以下三种表压缩方法：

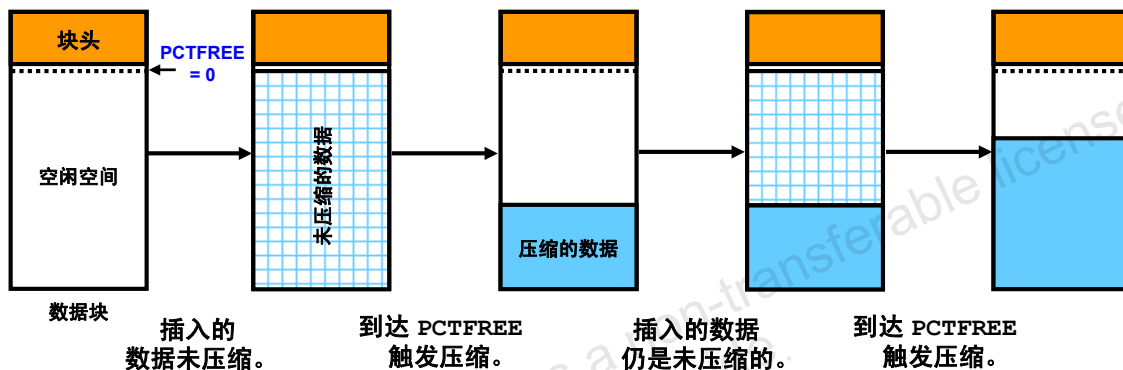
- 基本表压缩
- OLTP 表压缩
- Hybrid Columnar 压缩（通过 Exadata）

Oracle Corporation 建议压缩所有数据以减少存储成本。Oracle DB 可使用表压缩来消除数据块中的重复值。对于具有高度冗余数据的表，压缩可节省磁盘空间并减少数据库缓冲区高速缓存中的内存使用。表压缩对数据库应用程序是透明的。

- `table_compression` 子句仅对按堆组织的表有效。COMPRESS 关键字用于启用表压缩。NOCOMPRESS 关键字用于禁用表压缩。NOCOMPRESS 为默认值。
- 通过基本压缩，Oracle DB 在使用诸如直接加载或 CREATE TABLE AS SELECT 等操作执行批量加载时压缩数据。
- 通过 COMPRESS FOR OLTP，Oracle DB 在对表执行所有 DML 操作期间压缩数据。

## 适用于直接路径插入操作的压缩

- 是使用 `CREATE TABLE ... COMPRESS BASIC ...` 启用的
- 建议为批量加载数据仓库使用该压缩方法
- 替换已废弃的 `COMPRESS FOR DIRECT_LOAD OPERATIONS`
- 最大化块中的连续空闲空间



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 适用于直接路径插入操作的压缩

使用 `COMPRESS` 或 `COMPRESS BASIC` 可启用基本的表压缩。

- Oracle DB 在执行下列直接路径插入操作期间尝试压缩数据（如果这样做有利）：
  - 直接路径 `SQL*Loader`
  - `CREATE TABLE AS SELECT` 语句
  - 并行 `INSERT` 语句
  - 带有 `APPEND` 提示的 `INSERT` 语句
- 原始导入实用程序 (`imp`) 不支持直接路径 `INSERT`，因此无法以压缩的格式导入数据。
- 在早期版本中，这种类型的压缩称为 `DSS` 表压缩，是使用 `COMPRESS FOR DIRECT_LOAD OPERATIONS` 启用的。该语法已废弃。
- 压缩消除了块中因删除操作而形成的空隙，使块中连续可用空间达到最大化。

该幻灯片显示了压缩表中的数据块的演变过程，应按从左到右的顺序阅读。开始的时候，该数据块是空的，可以插入数据。开始在该块中插入数据时，数据以未压缩的格式存储（就像在未压缩的表中一样）。不过，只要该块达到其 `PCTFREE` 设置规定的填满标准，数据就会自动进行压缩，以减少其原来占据的空间。

**适用于直接路径插入操作的压缩（续）**

这样一来，可以在同一块中插入新的未压缩数据，直到再一次达到 PCTFREE 设置规定的填满标准。此时，会再一次触发压缩以减少块中的已使用空间量。

**注：**采用 COMPRESS 或 COMPRESS BASIC 的表使用 PCTFREE 值 0 以最大化压缩，除非显式设置 PCTFREE 子句的值。

采用 COMPRESS FOR OLTP 或 NOCOMPRESS 的表使用 PCTFREE 的默认值 10 以最大化压缩，同时仍允许以后对数据进行一些 DML 更改，除非显式覆盖该默认值。

## 适用于 DML 操作的 OLTP 压缩

- 是使用 `CREATE TABLE ... COMPRESS FOR OLTP ...;` 启用的
- 建议为活动的 OLTP 环境使用该压缩方法
- 替换已废弃的 `COMPRESS FOR ALL OPERATIONS`

	Y		Y		Y
G		Y		G	
	G		Y	Y	G

未压缩的块

G	Y				
	Y		Y		Y
G		Y		G	
	G		Y	Y	G

在块的开头包含  
符号表的 OLTP 压缩

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 适用于 DML 操作的 OLTP 压缩

使用 `COMPRESS FOR OLTP` 可启用 OLTP 表压缩。

- Oracle DB 在对表执行所有 DML 操作期间压缩数据。建议为活动的 OLTP 环境使用这种形式的压缩。
- 在早期版本中，OLTP 表压缩是使用 `COMPRESS FOR ALL OPERATIONS` 启用的。该语法已废弃。

使用 OLTP 压缩，一个数据块中行和列中的重复值只在块的开头存储一次（存储在符号表中）。重复值被替换为对符号表的简短引用（如幻灯片中所示）。因此，重新创建未压缩的数据所需的信息存储在块中。

为了说明 OLTP 压缩的原理，幻灯片中的图表显示了两个矩形。第一个灰色矩形包含四个标记为“G”的绿色小正方形和六个标记为“Y”的黄色正方形。它们表示未压缩的块。在第二个灰色矩形的开头，仅有一个标记为“G”的绿色正方形和一个标记为“Y”的黄色正方形，表示符号表。第二个灰色图表在与绿色和黄色正方形相同的位置还显示了 10 个白色的正方形。它们是白色的，因为它们现在仅是引用，重复值不占用空间。



## 指定表压缩

可以为以下项指定表压缩：

- 整个按堆组织的表
- 分区表（每个分区可具有一种不同类型或级别的压缩。）
- 嵌套表的存储

您无法：

- 为列数多于 255 的表指定基本压缩和 OLTP 压缩
- 如果表是针对直接加载而压缩的，则您无法删除列；但如果表使用 OLTP 压缩，则可删除

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 指定表压缩

可以为以下项指定表压缩：

- 整个按堆组织的表（在 `relational_table` 或 `object_table` 的 `physical_properties` 子句中）
- 分区表（每个分区可具有一种不同类型或级别的压缩。）
- 嵌套表的存储（在 `nested_table_col_properties` 子句中）

表压缩具有以下限制：

- 不支持对列数多于 255 的表使用 `COMPRESS FOR OLTP` 和 `COMPRESS BASIC`
- 无法从针对直接加载操作压缩的表中删除列，不过可以将这类列设置为未使用。  
`ALTER TABLE ... drop_column_clause` 的所有操作对使用 OLTP 压缩的表有效
- 使用 Hybrid Columnar 压缩的表不支持逻辑备用、Streams 和 LogMiner

## 使用压缩指导

### 压缩指导：

- 分析对象，估计不同压缩方法所节省的空间
- 帮助确定应用程序的正确压缩级别
- 推荐各种压缩策略
  - 为特定的数据集选取正确的压缩算法
  - 按特定列进行排序，以提高压缩比率
  - 提供各种压缩算法的折衷方案
- 适用于 OLTP 压缩（通过 EM）

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用压缩指导

压缩指导分析数据库对象，并确定每个压缩级别可实现的预期压缩比率。因此，它可帮助您确定应用程序的正确压缩级别。该指导推荐各种压缩策略。当从 EM 访问压缩指导时，它将确定 OLTP 压缩。

## 使用 DBMS\_COMPRESSION 程序包

确定最佳压缩比率：

```
BEGIN
DBMS_COMPRESSION.GET_COMPRESSION_RATIO ('USERS','SH','SALES',
  NULL,DBMS_COMPRESSION.COMP_FOR_OLTP, blkcnt_cmp, blkcnt_uncmp,
  rowcnt_cmp, rowcnt_uncmp, comptype);
DBMS_OUTPUT.PUT_LINE('Blk count compressed = ' || blkcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Blk count uncompressed = ' ||
blkcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Row count per block compressed = ' ||
rowcnt_cmp);
DBMS_OUTPUT.PUT_LINE('Row count per block uncompressed = ' ||
rowcnt_uncmp);
DBMS_OUTPUT.PUT_LINE('Compression type = ' || comptype);
DBMS_OUTPUT.PUT_LINE('Compression ratio =
'||blkcnt_uncmp/blkcnt_cmp||' to 1');
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 DBMS\_COMPRESSION 程序包

DBMS\_COMPRESSION 程序包提供的压缩指导帮助您为指定表确定可预期的压缩比率。该指导将分析数据库中的对象，确定可能达到的压缩比率，并建议最佳压缩级别。除了 DBMS\_COMPRESSION 程序包之外，还可以在现有指导框架内使用该压缩指导（通过 DBMS\_ADVISOR 程序包）。

为了确定压缩比率，DBMS\_COMPRESSION 程序包提供了以下子程序：

- GET\_COMPRESSION\_RATIO 过程，可针对未压缩表给出可行的压缩比率。
- GET\_COMPRESSION\_TYPE 过程，针对给定行返回压缩类型。

有关详细资料，请参阅《Oracle Database PL/SQL Packages and Types Reference》。

## 压缩表数据

压缩方法	压缩比率	CPU 开销	CREATE 和 ALTER TABLE 语法	典型应用
基本表压缩	高	最小	COMPRESS [BASIC]	DSS
OLTP 表压缩	高	最小	COMPRESS FOR OLTP	OLTP、 DSS

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 压缩表数据

Oracle DB 支持以下三种表压缩方法：

- 基本表压缩
- OLTP 表压缩
- Hybrid Columnar 压缩（通过 Exadata）

## 预先表空间监视

**Tablespaces**

Object Type: Tablespace

**Search**  
Enter an object name to filter the data that is displayed in your results set.  
Object Name:   
Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode: Single Go Create

Edit View Delete Actions Add Datafile Go

Select Name	Allocated Size(MB)	Space Used(MB)	Allocated Space Used(%)	Allocated Free Space(MB)	Status	Datafiles	Type	Extent Management	Segment Management
EXAMPLE	100.0	77.4		22.6	✓	1 PERMANENT	LOCAL	AUTO	

**Oracle Enterprise Manager 11g**

Database Control  
Database Instance: orcl > Tablespaces >  
Edit Tablespace: EXAMPLE  
Logged In As SYS

**General** **Storage** **Thresholds**

Available Space (MB) 32,767.90  
Space Used (MB) 77.36  
Space Used (%) 0.24  
Available Free Space (MB) 32,690.61

**Tablespace Full Metric Thresholds**  
Monitor the fullness of the tablespace using either of the metrics below.

**Space Used (%)**  
A warning or critical alert will be generated if the percentage of space used exceeds the corresponding threshold.  
☒ Use Database Default Thresholds Modify  
 Warning (%) 85  
 Critical (%) 97  
☐ Specify Thresholds  
 Warning (%) 85  
 Critical (%) 97  
☐ Disable Thresholds

**Free Space (MB)**  
A warning or critical alert will be generated if the remaining free space falls below the corresponding threshold. This metric is especially useful for large tablespaces.  
☒ Use Database Default Thresholds Modify  
 Warning (MB) Not Defined  
 Critical (MB) Not Defined  
☐ Specify Thresholds  
 Warning (MB)  
 Critical (MB)  
☐ Disable Thresholds

**Extent Allocation**  
Allocation Type Automatic

**Segment Space Management**  
Type Automatic

**Options**  
Compress ☐ Yes  
☒ No  
 Enable data segment compression to reduce disk usage. This option is especially useful in environments such as data warehouses, where the amount of insert and update operations is small.

**ORACLE**

版权所有 © 2011, Oracle. 保留所有权利。

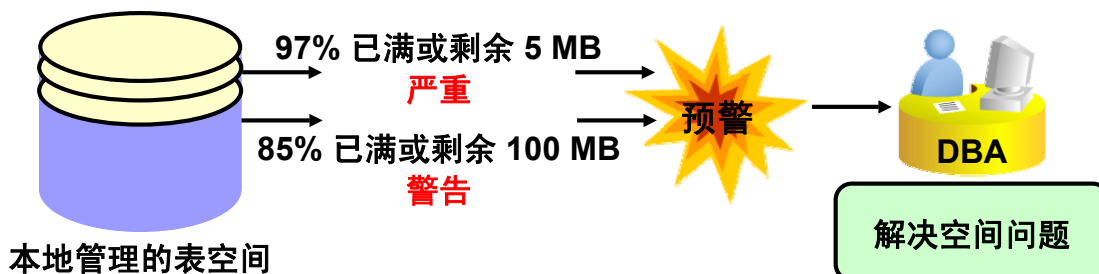
### 预先表空间监视

数据库以下列方式预先管理表空间的磁盘空间使用量：

- 表空间的可用磁盘空间变低，以及特定段空间不足时，系统会通过数据库预警通知您。然后您就可以为表空间提供更多磁盘空间，以避免空间不足的情况发生。
- 收集的信息被存储在自动工作量资料档案库 (AWR) 中，用于执行增长趋势分析以及计划数据库的容量。

要在 Oracle Enterprise Manager 中查看和修改表空间信息，请在“Database（数据库）”主页中选择“Server（服务器）”，然后选择“Tablespaces（表空间）”。选择需要的表空间，然后单击“Edit（编辑）”按钮。

## 阈值和解决空间问题



通过以下方法解决空间问题：

- 添加数据文件或调整数据文件大小
- 将 AUTOEXTEND 设置为 ON
- 收缩对象
- 减少 UNDO\_RETENTION
- 检查临时表空间中是否存在长时间运行的查询

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 阈值和解决空间问题

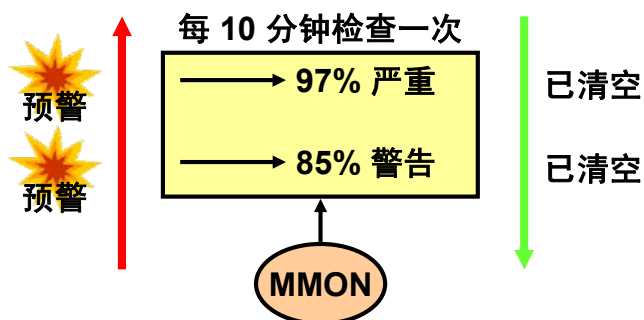
表空间阈值定义为“满”，或者定义为表空间中的可用空间。严重阈值和警告阈值是应用于表空间的两个阈值。DBMS\_SERVER\_ALERT 程序包中包含用于设置和获取阈值的过程。达到表空间限制大小时，将引发相应的预警。阈值以表空间大小的百分比，或剩余的空闲字节数表示。此值在内存中计算。可以为表空间同时定义百分比和字节形式的阈值。它们中的任何一个或两者都可以生成预警。

理想的警告阈值触发值设置会生成这样的预警：既能确保有足够时间来解决问题，使其不会变成严重状态，又不至于在空间尚未成为问题时干扰您。

预警指示 DBA 可以通过执行以下一项或多项操作来解决问题：

- 向表空间添加更多空间，方法是添加一个文件或调整现有文件的大小，或者使现有文件可自动扩展
- 在包含任何可自动扩展文件的磁盘上释放空间
- 收缩表空间中的稀疏对象

## 监视表空间的空间使用量



- 只读表空间和脱机表空间：不要设置预警。
- 临时表空间：阈值对应于会话当前使用的空间。
- 还原表空间：阈值对应于活动区和未到期区所使用的空间。
- 可自动扩展的文件：阈值基于最大文件大小。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

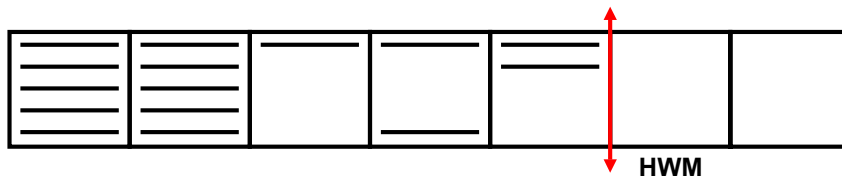
### 监视表空间的空间使用量

数据库在执行常规的空间管理活动时跟踪空间使用情况。MMON 进程每 10 分钟汇总一次此信息。达到或超过了表空间的阈值时，将触发预警。

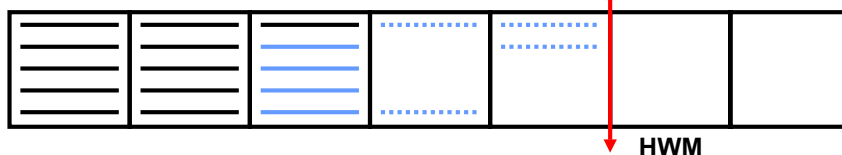
- 不应在处于只读模式的表空间或已脱机的表空间上标记预警，因为在这些表空间上无法进行太多操作。
- 在临时表空间中，阈值必须定义为对表空间中已用空间的限制。
- 对于还原的表空间，如果某个区中不包含活动或未到期的还原操作，则该区可以重用。计算阈值违规时，活动区和未到期区总体被视为已用空间。
- 对于包含可自动扩展文件的表空间，将根据您指定的最大文件大小或最大 OS 文件大小来计算阈值。



## 收缩段

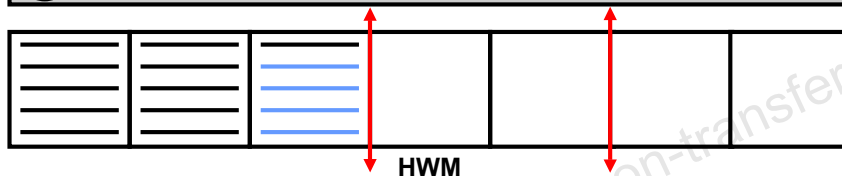


① ALTER TABLE employees SHRINK SPACE COMPACT;



可以在压缩期间进行 DML 操作和查询。

② ALTER TABLE employees SHRINK SPACE;



调整 HWM 时将阻塞 DML 操作。

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

## 收缩段

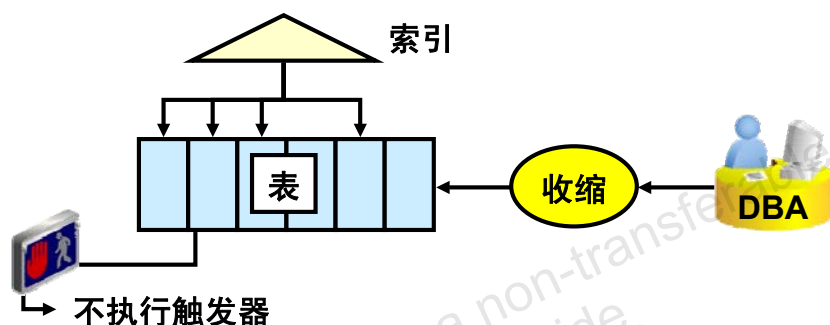
幻灯片中的图表描述了表收缩操作的两个阶段。第一个阶段执行压缩。在此阶段，行将尽可能移动到段的左侧部分。在内部，行将由数据包移动，以避免锁定问题。移动完行后，将启动收缩操作的第二阶段。在此阶段中，将调整高水位标记 (HWM)，并释放未使用的空间。

如果有长时间运行的查询，而且这些查询可能跨越收缩操作，并尝试从已回收的块中读取数据，则 COMPACT 子句会很有用。指定 SHRINK SPACE COMPACT 子句时，收缩操作的进度将保存在相应段的位图块中。这意味着下次在同一个段上执行收缩操作时，Oracle DB Server 可以记住已经执行过的操作。然后您就可以在非峰值时段重新发出 SHRINK SPACE 子句，无需使用 COMPACT 子句即可完成第二阶段。



## 收缩操作的结果

- 改善性能和空间使用情况
- 维护索引
- 不执行触发器
- 可以减少迁移行的数量
- 建议在 IOT 上重建二级索引



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 收缩操作的结果

收缩稀疏填充的段可以提高对该段执行扫描操作和 DML 操作的性能。这是因为在收缩段后，需要查看的块减少了。尤其体现在下列方面：

- 全表扫描（块变少、变密）
- 改善索引访问（由于树变得更紧凑，因此减少了 ROWID 范围扫描时的 I/O 次数）

另外，通过收缩稀疏填充的段，还可以提高数据库内空间使用效率，因为在对象需要空间时有更多空闲空间可以使用。

在段收缩操作期间将保证索引依赖关系。收缩了相应表后，索引处于可用状态。因此，不需要进一步维护。

实际收缩操作在内部作为 INSERT/DELETE 操作进行处理。但是，不执行任何 DML 触发器，因为数据本身并未发生变化。

段收缩操作的一个可能结果是迁移行的数量减少。但是，您不应总是依赖段收缩来减少迁移行数量。因为段收缩操作可能不会触及段中的所有块。因此，不能保证所有迁移行都得到处理。

**注：**建议执行收缩操作之后在索引表 (IOT) 上重建二级索引。

## 回收 ASSM 段内的空间

- 联机原地操作
- 只适用于位于 ASSM 表空间中的段
- 候选段类型包括：
  - 按堆组织的表和索引表
  - 索引
  - 分区和子分区
  - 实体化视图和实体化视图日志

ORACLE

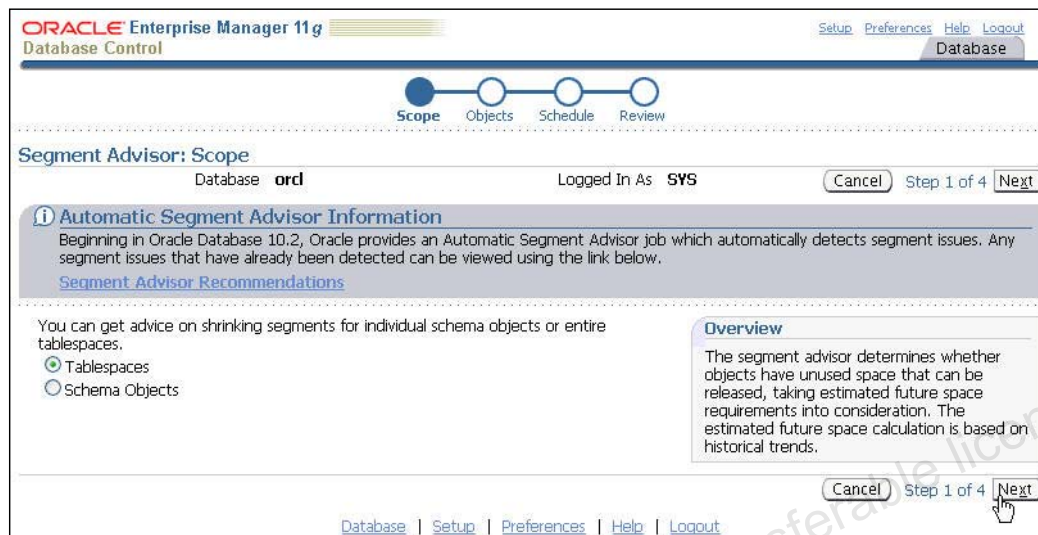
版权所有 © 2011, Oracle。保留所有权利。

### 使用 ASSM 回收空间

收缩操作是一个联机原地操作，因为不需要额外的数据库空间即可执行此操作。

- 无法在由空闲列表管理的段上执行收缩操作。可以收缩自动段空间管理的表空间中的段。但是，不能收缩 ASSM 表空间中存储的下列对象：
  - 集群中的表
  - 包含 LONG 列的表
  - 包含提交时实体化视图的表
  - 包含基于 ROWID 的实体化视图的表
  - IOT 映射表
  - 包含基于函数的索引的表
- 必须为按堆组织的段启用 ROW MOVEMENT。

## 段指导：概览



版权所有 © 2011, Oracle。保留所有权利。

### 段指导：概览

段指导可标识具有可以回收的空间的段。它通过检查自动工作量资料档案库 (AWR) 中的使用情况和增长统计数据以及对段中的数据进行采样来执行分析。可以将其配置为定期自动运行，也可以根据需求（手动）运行它。定期调度的段指导运行称为“自动段指导”。提供建议之后，您可以选择实施这些建议。可以在段级别或表空间级别调用收缩指导。

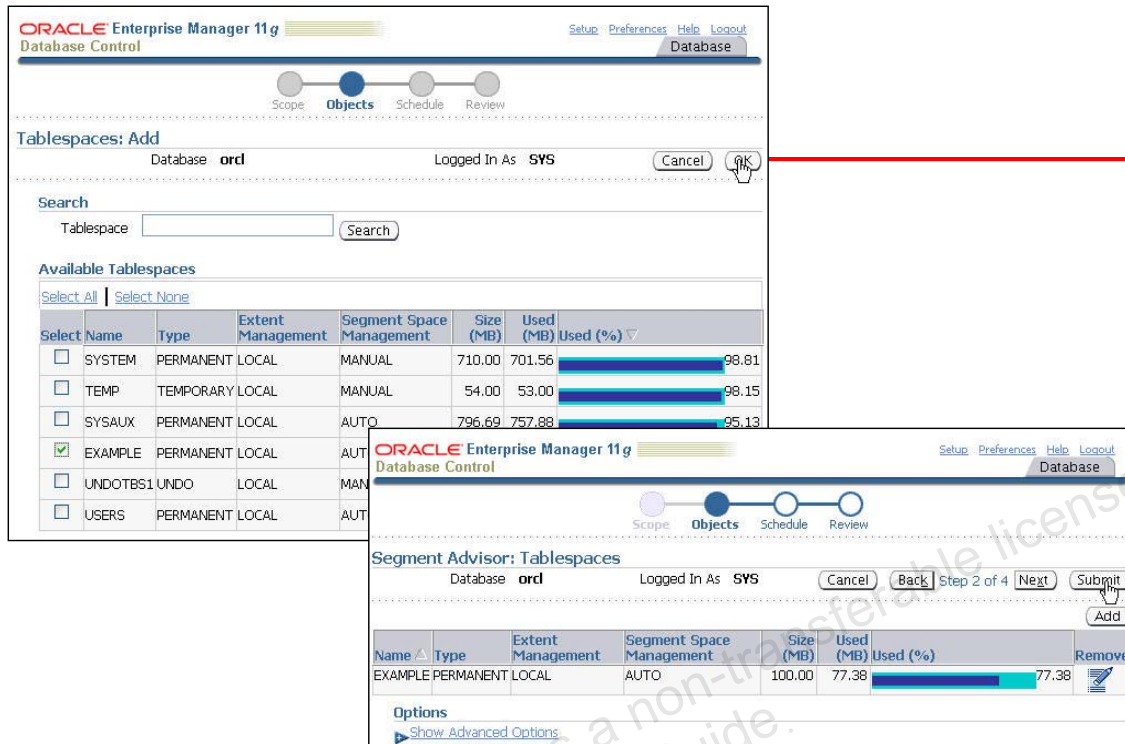
EM 数据库控制台是段指导的接口。可以从 EM 中的几个位置访问段指导：

- “Advisor Central（指导中心）”页
- “Tablespaces（表空间）”页
- 方案对象页

使用数据库控制台可以选择不同的输入信息，并可以调度一个调用段指导的作业来获得收缩建议。可以在表空间环境中或方案对象环境中调用段指导向导，没有相关环境也可以调用。

段指导可以根据采样分析、历史信息和未来增长趋势来提供建议。

## 段指导



版权所有 © 2011, Oracle. 保留所有权利。

### 段指导

在“Server（服务器）”页中的“Storage（存储）”部分，选择“Tablespaces（表空间）”。在“Tablespaces（表空间）”页上，选择要对其执行收缩分析的表空间，然后在“Actions（操作）”下拉列表中选择“Run Segment Advisor（运行段指导）”。单击“Go（执行）”可以打开“Segment Advisor（段指导）”初始页。必须从“comprehensive（综合）”和“limited（有限制）”分析模式中选择一种。在综合模式下，分析时间较长，因为指导会对段进行采样以确定正确的目标。

一直单击“Continue（继续）”，回答指导的各个问题。您在“Segment Advisor: Review（段指导: 复查）”页上结束操作，可以在该页上复查分析内容的详细资料。段指导分析作为调度作业运行，因此可以从“Advisor Central（指导中心）”页检查已调度的任务。完成后，可以检查指导的建议。

**注：**在段指导中，可以指定分析操作的持续时间。这样，可以限制指导用于生成建议的时间。一般说来，分析时段越长，得到的结果就越全面。结果存储在 AWR 中，可以稍后查看。使用“Number of days to retain（保留天数）”选项可以告知 Oracle DB Server 这些结果应在 AWR 中保留多少天后再清除。

# 实施建议

Advisor Central

Advisors

Checkers

Page Refreshed Jul 12, 2007 7:18:54 AM EDT Refresh

Advisors

ADDM

Memory Advisors

SQL Advisors

Automatic Undo Management

MTTR Advisor

SQL Performance Analyzer

Data Recovery Advisor

Segment Advisor

Advisor Tasks

Change Default Parameters

Search

Select an advisory type and optionally enter a task name to filter the data that is displayed in your results set.

Advisory Type

Task Name

Advisor Runs

Status

Segment Advisor

All

All

Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Results

View Result

Delete

Actions

Re-schedule

Go

Previous

1-25 of 45

Next 20

Select	Advisory Type	Name	Description	User	Status	Start Time	Duration (seconds)	Expires In (days)
<input checked="" type="checkbox"/>	Segment Advisor	SEGMENTADV_5412893	Get shrink advice based on object growth trend	SYS	COMPLETED	Jul 12, 2007 7:13:44 AM	134	30

Select

Schema

Segment

Recommendation

Reclaimable Space (MB)

(MB)

(MB)

Type

<input checked="" type="checkbox"/>	SYS	EMPLOYEES1	Shrink	alter table "SYS"."EMPLOYEES1"	shrink space
<input checked="" type="checkbox"/>	SYS	EMPLOYEES2	Shrink	alter table "SYS"."EMPLOYEES2"	shrink space
<input checked="" type="checkbox"/>	SYS	EMPLOYEES3	Shrink	alter table "SYS"."EMPLOYEES3"	shrink space

版权所有 © 2011, Oracle。保留所有权利。

## 实施建议

段指导完成其作业后，您可以查看建议的详细资料并直接实施建议。

注：在收缩按堆组织的表之前，必须在该表上启用行移动功能。可以使用“Edit Table（编辑表）”页上“Options（选项）”选项卡中的“Database Control（数据库控制）”来进行此操作。

## 自动段指导

### 自动段指导：

- 由设置为在默认维护窗口期间运行的调度程序作业启动：
  - 每个工作日的晚上，星期一至星期五，从晚上 10:00 至次日凌晨 2:00
  - 星期六和星期日，两个窗口都开始于早上 6:00，并持续 20 小时
- 检查数据库统计信息，对段数据采样，然后选择以下对象进行分析：
  - 超过了严重或警告阈值的表空间
  - 包含活动最多的段
  - 增长率最高的段

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 自动段指导

自动段指导是由配置为在默认维护窗口期间运行的调度程序作业启动的。默认维护窗口是在调度程序中指定的，其初始定义如下：

- 每个工作日的晚上，星期一至星期五，从晚上 10:00 至次日凌晨 2:00（每晚 4 小时）
- 周末，星期六和星期日早上 6:00，每天持续 20 小时

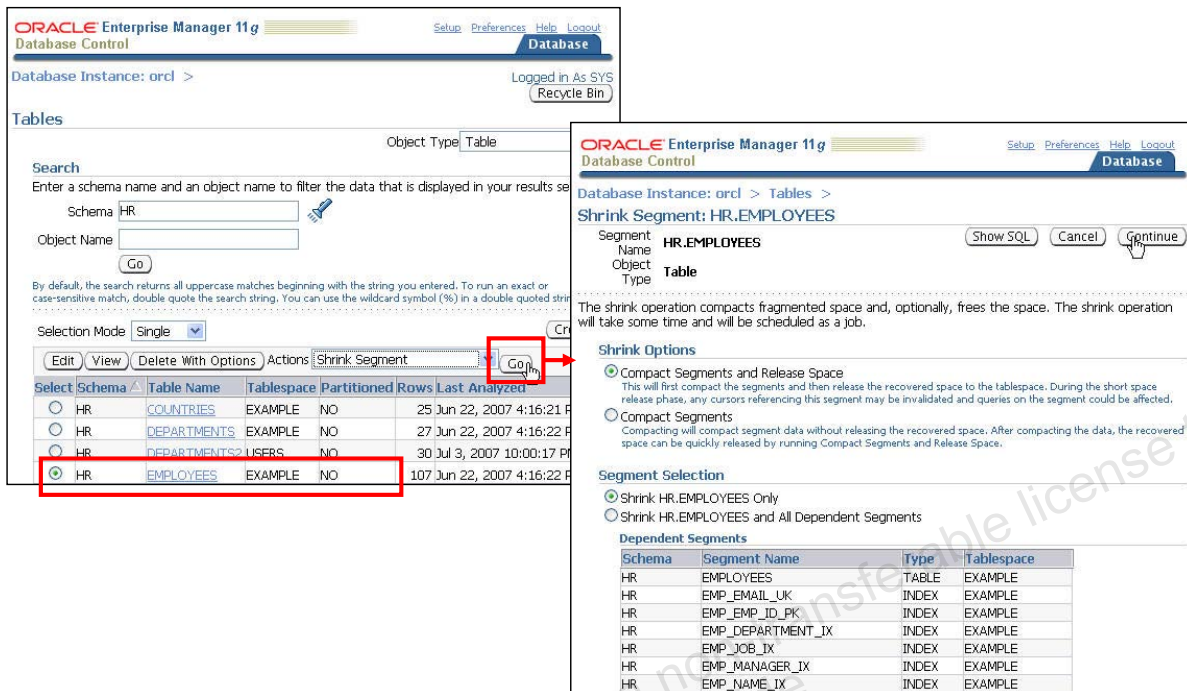
自动段指导不会分析每个数据库对象。而是检查数据库统计信息，对段数据采样，然后选择以下对象进行分析：

- 超过了严重或警告空间阈值的表空间
- 包含活动最多的段
- 增长率最高的段

如果选择了某个对象进行分析，但维护窗口在段指导能处理该对象之前失效了，则该对象将在下一次自动段指导运行中分析。不能更改自动段指导选择进行分析的那组表空间和段。但是，可以启用或禁用自动段指导作业，更改自动段指导的计划运行期间，或者调整自动段指导的系统资源利用率。



## 使用 EM 手动收缩段



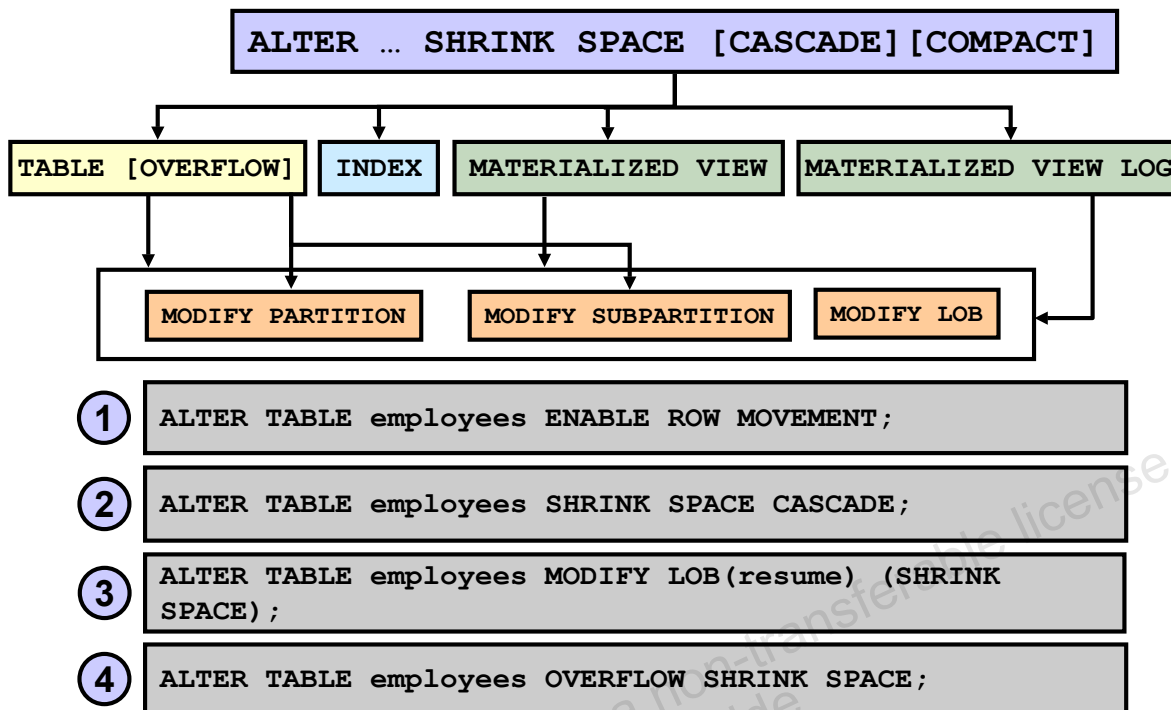
版权所有 © 2011, Oracle. 保留所有权利。

### 使用 EM 手动收缩段

此外（相对于实施段指导的建议），还可以收缩与特定数据库对象关联的各个段。例如，在“Database（数据库）”主页中选择“Schema（方案）”文件夹选项卡，然后单击“Database Objects（数据库对象）”部分中的“Tables（表）”链接。在“Tables（表）”页上选择表，接着在“Actions（操作）”下拉列表中选择“Shrink Segment（收缩段）”。然后单击“Go（执行）”按钮。这样将打开“Shrink Segment（收缩段）”页，您可以在其中选择要收缩的从属段。可以选择只压缩空间，或者选择压缩并释放空间。还可以选择 CASCADE 选项。

完成后，单击“Continue（继续）”链接。这样，收缩语句将作为调度作业提交。

## 使用 SQL 收缩段



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 SQL 收缩段

因为在按堆组织的段中，收缩操作可能会导致 ROWID 发生更改，所以在该段上执行收缩操作之前，必须在相应段上启用行移动。默认情况下，在段级别上行移动处于禁用状态。要启用行移动，需要使用 `CREATE TABLE` 或 `ALTER TABLE` 命令的 `ENABLE ROW MOVEMENT` 子句。幻灯片中的第一个示例对此进行了阐述。

使用 `ALTER` 命令可以针对对象调用段收缩。对象可以是下列类型之一：表（按堆组织的表或索引表）、分区、子分区、LOB（数据和索引段）、索引、实体化视图或实体化视图日志。

使用 `SHRINK SPACE` 子句可以收缩段中的空间。如果指定了 `CASCADE`，则收缩行为将级联到所有支持收缩操作的从属段，但实体化视图、LOB 索引和 IOT（索引表）映射表例外。第二个示例展示了 `SHRINK SPACE` 子句。

在索引段中，收缩操作将首先合并索引，然后压缩数据。

示例 3 显示了一个收缩 LOB 段的命令，并假设 `RESUME` 列为 CLOB。

示例 4 显示了一个收缩 IOT 溢出段的命令，此段属于 `EMPLOYEES` 表。

**注：**有关详细信息，请参阅《Oracle 数据库 SQL 参考》指南。



## 管理可恢复的空间分配

可恢复的语句有下列特性：

- 使用可恢复的语句，可以挂起大型操作，而不是收到错误
- 使用可恢复的语句，当操作挂起时，您可以解决问题，然后继续进行，无须重新开始
- 在下列条件下，可恢复的语句将挂起：
  - 空间不足
  - 达到了最大区数
  - 超出了空间限额
- 可恢复的语句可以多次挂起和恢复

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 管理可恢复的空间分配

Oracle DB Server 提供了一种方法，可以在空间分配失败时挂起大型数据库操作，稍后再恢复执行。使用这种方法，就有机会采取更正措施，而不是让 Oracle DB Server 向用户返回错误。更正了错误条件后，挂起的操作将自动恢复。这种功能称为“可恢复的空间分配”。受影响的语句称为“可恢复的语句”。仅当为系统或会话启用了可恢复的语句功能时，语句才能在可恢复模式下执行。

挂起语句时将自动挂起事务处理。因此在 SQL 语句的挂起和恢复过程中，将保留所有事务处理资源。错误条件不再存在时（例如在用户干预之后，或者很可能在其它查询释放了排序空间之后），挂起的语句将自动恢复执行。出现下列条件之一时，可恢复的语句将挂起：

- 空间不足条件
- 达到了最大区数条件
- 超出了空间限额条件

可恢复的语句有一个与之关联的挂起超时间隔。可恢复的语句的挂起时间超过超时间隔（默认为 2 小时）后将重新激活自身，并向用户返回异常错误。可恢复的语句可以多次挂起和恢复。

注：达到最大区数错误仅发生在字典管理的表空间中。

## 使用可恢复的空间分配

- 如果查询、DML 操作和特定 DDL 操作遇到空间不足错误，则可恢复这些操作。
- 可以通过 SQL、PL/SQL、SQL\*Loader 和数据泵实用程序或 Oracle 调用接口 (OCI) 来发出可恢复的语句。
- 仅当某条语句的会话已由下列操作之一启用时，该语句才可以在可恢复模式下执行：
  - **RESUMABLE\_TIMEOUT** 初始化参数被设置为非零值。
  - 发出了 **ALTER SESSION ENABLE RESUMABLE** 语句：

```
ALTER SESSION ENABLE RESUMABLE;
INSERT INTO sales_new SELECT * FROM sh.sales;
ALTER SESSION DISABLE RESUMABLE;
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用可恢复的空间分配

仅当在启用了可恢复模式的会话中执行语句时，才能使用可恢复的空间分配。有两种方法可以启用和禁用可恢复的空间分配：

- 发出 **ALTER SESSION ENABLE RESUMABLE** 命令。
- 使用 **ALTER SESSION** 或 **ALTER SYSTEM** 语句，将 **RESUMABLE\_TIMEOUT** 初始化参数设置为非零值。

在为会话或数据库启用可恢复模式时，可以指定一个超时时段，在此时段后，如果未发生任何干预，则挂起的语句将出错。**RESUMABLE\_TIMEOUT** 初始化参数指示超时发生之前经过的秒数。也可以用以下命令来指定超时时段：

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600;
```

**TIMEOUT** 值一直有效，直到另一个 **ALTER SESSION ENABLE RESUMABLE** 语句更改该值，或使用另一种方式更改该值，或直到会话结束。使用 **ENABLE RESUMABLE TIMEOUT** 子句启用可恢复模式时的默认超时间隔是 7,200 秒（2 小时）。

还可以为可恢复的语句命名。例如：

```
ALTER SESSION ENABLE RESUMABLE TIMEOUT 3600
NAME 'multitab insert';
```

**使用可恢复的空间分配（续）**

语句的名称用于在 DBA\_RESUMABLE 和 USER\_RESUMABLE 视图中标识可恢复的语句。

例如：

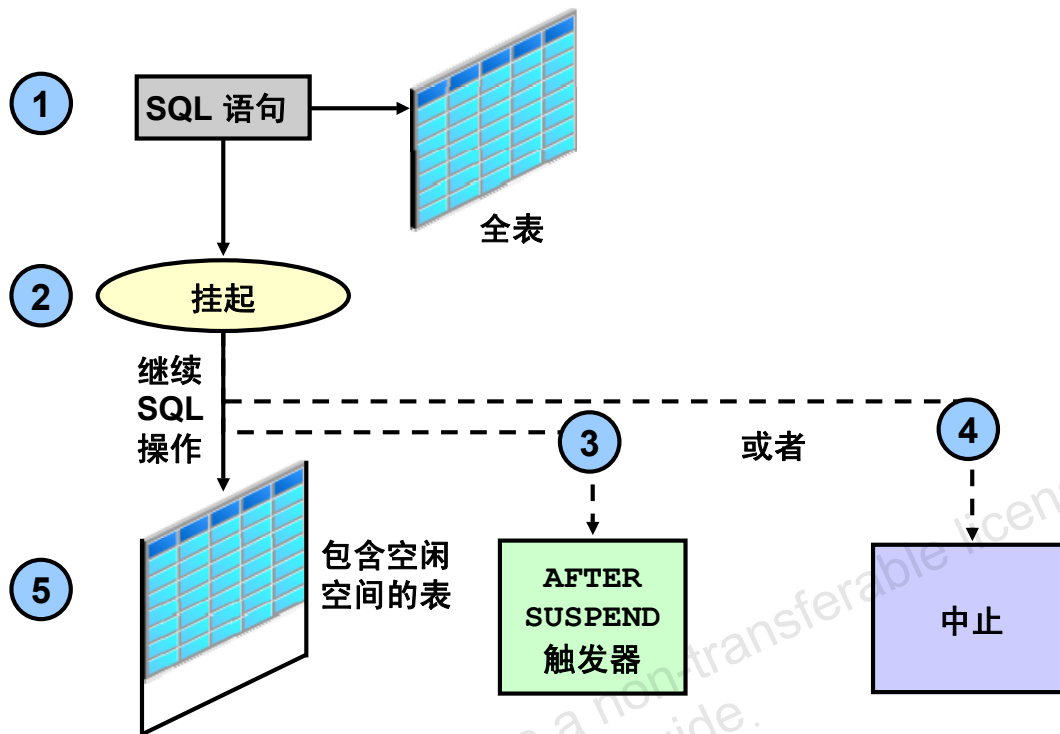
```
SELECT name, sql_text FROM user_resumable;
```

```
NAME                SQL_TEXT
-----
multitab insert INSERT INTO oldsales SELECT * FROM sh.sales;
```

要自动为各个会话配置可恢复的语句设置，可以创建并注册一个数据库级别的 LOGON 触发器，以更改用户的会话。此触发器可以发出命令来为会话启用可恢复的语句，指定超时时段，以及将会话发出的可恢复语句与某个名称关联。

因为挂起的语句会保留一些系统资源，所以必须先向用户授予 RESUMABLE 系统权限，然后他们才能启用可恢复的空间分配并执行可恢复的语句。

## 恢复挂起的语句



版权所有 © 2011, Oracle。保留所有权利。

### 恢复挂起的语句

#### 示例

1. INSERT 语句遇到错误，声称表已满。
2. 挂起 INSERT 语句，并且不向客户机传递任何错误。
3. 或者，执行一个 AFTER SUSPEND 触发器。
4. 或者，激活 SQLERROR 异常错误以中止语句。
5. 如果语句未中止，并将空闲空间成功添加到表中，则 INSERT 语句将恢复执行。

### 检测挂起的语句

可恢复的语句挂起时，不会向客户机发出错误。为了方便采取更正措施，Oracle DB Server 提供了无需向用户通知错误，也无需提供有关情况的信息的替代方法。

## 恢复挂起的语句（续）

### 在挂起期间可能执行的操作

可恢复的语句遇到可更正的错误时，系统将在内部生成 AFTER SUSPEND 系统事件。用户可以在数据库级别和方案级别同时为此事件注册触发器。如果用户注册一个触发器来处理此系统事件，则在 SQL 语句挂起后将执行该触发器。在 AFTER SUSPEND 触发器内执行的 SQL 语句始终不可恢复，并且始终是独立运行的。在触发器内启动的事务处理使用 SYSTEM 回退段。限定这些条件是为了克服死锁，并降低触发器与语句遇到相同错误条件的概率。

在触发器代码中，可以使用 USER\_RESUMABLE 视图或 DBA\_RESUMABLE 视图，或者使用 DBMS\_RESUMABLE.SPACE\_ERROR\_INFO 函数来获得有关可恢复语句的信息。

当可恢复的语句挂起时：

- 调用语句的会话将被置于等待状态。对于 EVENT 列包含 “statement suspended, wait error to be cleared” 的会话，系统会将一个行插入 V\$SESSION\_WAIT。
- 针对需要添加资源才能完成挂起语句的对象发出操作挂起的预警。

### 结束挂起的语句

解决了错误条件后（例如在 DBA 干预之后，或者很可能在其它查询释放了排序空间之后），挂起的语句将自动恢复执行，并且清除 “可恢复会话已挂起” 预警。

可以使用 DBMS\_RESUMABLE.ABORT() 过程强制挂起的语句激活 SERVERERROR 异常错误。此过程可以由 DBA 调用，也可以由发出语句的用户调用。如果达到了与可恢复语句关联的挂起超时间隔，则语句将自动中止并且系统会向用户返回错误。

## 哪些操作是可恢复的

下列操作是可恢复的操作：

- 查询：运行时临时空间不足（排序空间不足）的 SELECT 语句
- DML：INSERT、UPDATE 和 DELETE 语句
- 下列 DDL 语句：

- CREATE TABLE ... AS SELECT
- CREATE INDEX
- ALTER INDEX ... REBUILD
- ALTER TABLE ... MOVE PARTITION
- ALTER TABLE ... SPLIT PARTITION
- ALTER INDEX ... REBUILD PARTITION
- ALTER INDEX ... SPLIT PARTITION
- CREATE MATERIALIZED VIEW

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 哪些操作是可恢复的

下列操作是可恢复的操作：

- **查询：**运行时临时空间不足（排序空间不足）的 SELECT 语句是可恢复执行的语句。使用 OCI 时，OCIStmtExecute() 和 OCIStmtFetch() 调用也可恢复执行。
- **DML：**INSERT、UPDATE 和 DELETE 语句是可恢复执行的语句。用于执行这些语句的接口没有影响；接口可以是 OCI、SQL、PL/SQL 或其它接口。此外，外部表中的 INSERT INTO ... SELECT 也是可恢复的。
- **DDL：**下列语句是可恢复执行的操作：
  - CREATE TABLE ... AS SELECT
  - CREATE INDEX
  - ALTER INDEX ... REBUILD
  - ALTER TABLE ... MOVE PARTITION
  - ALTER TABLE ... SPLIT PARTITION
  - ALTER INDEX ... REBUILD PARTITION
  - ALTER INDEX ... SPLIT PARTITION
  - CREATE MATERIALIZED VIEW

## 测验

选择关于空间管理的正确陈述：

1. 在 Oracle Database 11g 发行版 2 中可为所有表延迟创建段。没有例外情况。
2. 创建的所有 UNUSABLE 索引和索引分区都没有段。
3. 收缩段空间是不可恢复的操作。
4. 您可按表空间设置阈值。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、4

## 小结

在本课中，您应该已经学会：

- 介绍 Oracle DB Server 如何自动管理空间
- 使用压缩节省空间
- 主动监视和管理表空间的空间使用量
- 介绍 Oracle DB 中的段的创建
- 控制延迟创建段
- 使用“段指导”
- 使用段收缩功能从表和索引中回收浪费的空间
- 管理可恢复的空间分配

ORACLE

版权所有 © 2011, Oracle。保留所有权利。



## 练习 18 概览：管理存储

本练习包含以下主题：

- 使用阈值预警主动管理表空间
- 使用段指导来收缩空间
- 在 SQL\*Plus 和 Oracle Enterprise Manager 中查看预警和预警历史记录

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

华周 (286991486@qq.com) has a non-transferable license to use  
this Student Guide.

# 19

## 管理数据库的空间

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

华周 (286991486@qq.com) has a non-transferable license to use this Student Guide.

## 课程目标

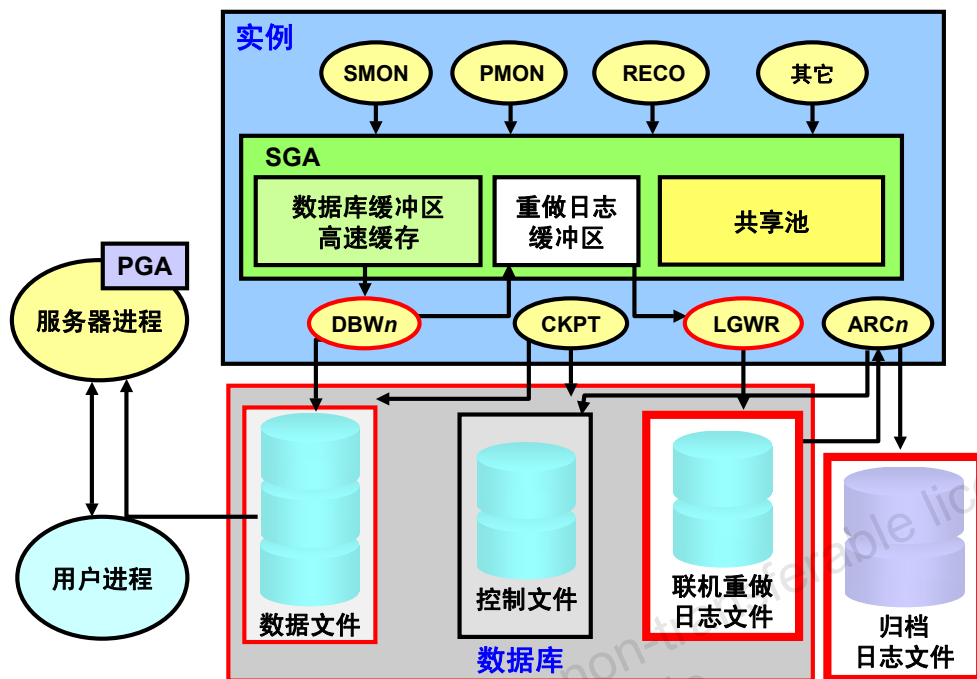
学完本课后，应能完成以下工作：

- 描述 4 KB 扇区磁盘的概念及使用
- 使用可移动表空间
- 描述可移动表空间的概念

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 数据库存储



版权所有 © 2011, Oracle. 保留所有权利。

### 数据库存储

数据库包括物理结构和逻辑结构。由于物理结构和逻辑结构是分开的，因此管理数据的物理存储时不会影响对逻辑存储结构的访问。

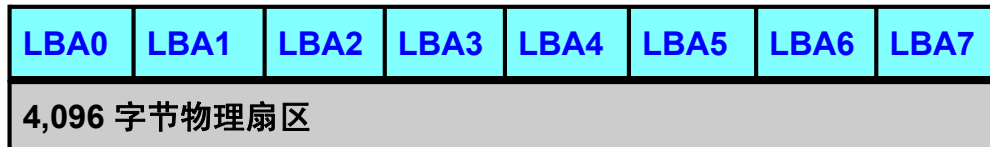
磁盘是数据库的主要存储介质，目前主要有 512 字节的扇区，而拥有 4 KB 大扇区的磁盘即将上市，它们可以用较低的开销提供更高的存储容量。Oracle DB 可以通过特定于平台的设备驱动程序访问硬盘。（数据库写进程 [和 ASM 进程] 可直接向磁盘进行写入，无需通过操作系统。）

Oracle Database 11g 发行版 2 检测磁盘扇区大小并使用高容量磁盘且不会导致性能下降

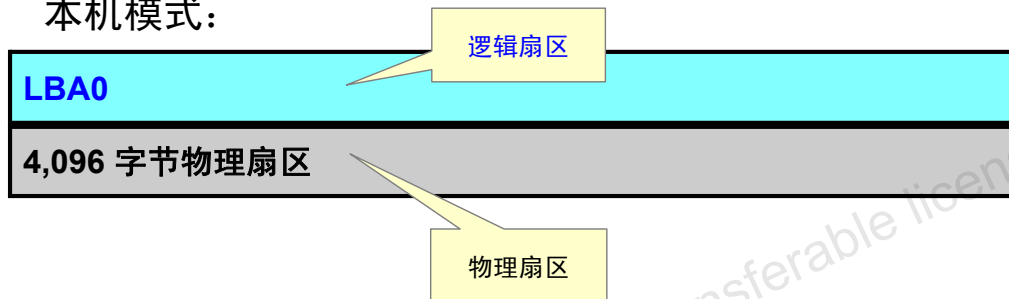
（原因之一是内部优化减少了可能会发生的重做空间浪费，具有许多短事务处理的应用程序，如邮件系统，可能会发生重做空间浪费）。

## 支持 4 KB 扇区磁盘

- 仿真模式：



- 本机模式：



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 支持 4 KB 扇区磁盘

4 KB 扇区磁盘有物理扇区（以灰色显示）和逻辑扇区（以蓝色显示）。有二种类型的 4 KB 扇区磁盘：仿真模式和本机模式。

- 仿真模式下的 4 KB 扇区磁盘每一个物理扇区具有八个逻辑扇区，如幻灯片所示。这类磁盘维护有一个到 4 KB 物理扇区的 512 字节接口 — 即逻辑块地址 (LBA) 引用磁盘上的 512 字节。

在仿真模式中性能可能会降低，因为磁盘驱动器将 4 KB 扇区读取到磁盘高速缓存内存中，更改 512 字节部分，并将整个 4 KB 扇区回写到磁盘。

- 本机模式下的 4 KB 扇区磁盘每一个物理扇区具备一个逻辑扇区，如幻灯片中所示。因此，只有 4 KB 接口。也就是说，LBA 引用磁盘上的 4,096 字节。

## 使用 4 KB 扇区磁盘

### 仿真模式：

- 建议为日志使用 4 KB 块大小
- 建议为数据文件使用 4 KB 块大小（或更大）

### 不受影响：

- 控件文件块大小：16 KB

### 本机模式：

- 日志必须使用 4 KB 块大小
- 数据文件必须使用 4 KB 块大小（或更大）



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 使用 4 KB 扇区磁盘

在 Oracle Database 11gR2 中，4 KB 扇区磁盘主要影响重做日志文件。这包括联机重做日志，备用重做日志和归档日志。Oracle 建议在 4 KB 仿真模式磁盘上创建 4 KB 块大小的日志。在 4 KB 本机模式磁盘上，必须创建 4 KB 块大小的日志。

也就是说，重做块大小必须与物理磁盘扇区大小相匹配（对于 512 字节和 4 KB 本机模式磁盘）。否则，您会收到 ORA-1378 错误。对于 4 KB 仿真模式磁盘，重做块大小可以是 512 或 4,096 字节。4 KB 是首选块大小。在 4 KB 仿真磁盘上创建大小为 512 字节的块时，预警日志中会写入一条警告，指明块大小不匹配将导致性能下降。ASM 磁盘组也是如此。

4 KB 扇区磁盘还会影响 Oracle 数据文件。Oracle DB 允许您在 512 字节扇区磁盘上创建 2 KB 块大小的数据文件。使用 4 KB 扇区磁盘时，Oracle 建议您在 4 KB 仿真模式磁盘上创建 4 KB（或更大）块大小的数据文件。在 4 KB 本机模式磁盘上，必须创建 4 KB（或更大）块大小的日志。

控制文件块大小已经为 16 KB。因此，4 KB 扇区磁盘不影响控制文件。

## 指定磁盘扇区大小

使用以下命令的 `SECTOR_SIZE` 和 `BLOCKSIZE` 子句：

- `CREATE DISKGROUP`
- `ALTER DATABASE`
- `CREATE DATABASE`
- `CREATE CONTROL FILE`

基于硬件的默认扇区大小  
(不是之前的 512 字节扇区)

```
CREATE DATABASE sample NORESETLOGS FORCE LOGGING
ARCHIVELOG
LOGFILE
GROUP 1 '$ORACLE_BASE/oradata/sample/redo01.log'
        SIZE 100M BLOCKSIZE 4096,
GROUP 2 '$ORACLE_BASE/oradata/sample/redo02.log'
        SIZE 100M BLOCKSIZE 4096
DATAFILE
...
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 指定磁盘扇区大小

在自动存储管理 (ASM) 环境下，您可以为磁盘组设置 `SECTOR_SIZE` 属性。该属性只能在创建磁盘组时设置（通过使用 `CREATE DISKGROUP` 命令）。

您可以使用以下命令的新增 `BLOCKSIZE` 子句指定日志文件的大小：

- `ALTER DATABASE`
- `CREATE DATABASE`
- `CREATE CONTROL FILE`

与在 512 字节磁盘上新建数据库相比，在 4 KB 扇区磁盘上新建数据库没有额外的工作。在 GUI 环境中没有更改。

您可以选择在 `CREATE DATABASE` 命令中使用 `BLOCKSIZE` 子句，如幻灯片所示。如果未指定块大小，Oracle DB 将检查底层磁盘扇区大小，并在创建重做日志时使用该磁盘扇区大小作为块大小。因此，默认情况下，重做日志块大小是磁盘扇区大小，而不是之前的 512 字节扇区大小。



## 测验

在 4 KB 本机模式磁盘上必须使用 4 KB 日志文件。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 测验

出于性能方面的考虑，Oracle 建议您在 4 KB 仿真磁盘上创建 512 字节的块。

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2

## 测验

控件文件不受 4 KB 扇区磁盘影响（因为它们已经比 4 KB 大。）

1. 对
2. 错

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：1

## 移动表空间

- 可跨平台移动的表空间：
  - 简化数据仓库和数据集市之间的数据移动
  - 允许从一个平台向另一个平台迁移数据库
- 支持的平台包括：

Solaris[tm] OE (32 位)	HP-UX (64 位)	Microsoft Windows IA (64 位)
Solaris[tm] OE (64 位)	HP Tru64 UNIX	基于 IBM zSeries 的 Linux
Microsoft Windows IA (32 位)	HP-UX IA (64 位)	适用于 AMD 的 64 位 Linux
Linux IA (32 位)	Linux IA (64 位)	Apple Mac OS
基于 AIX 的系统 (64 位)	HP Open VMS	Microsoft Windows 64-bit AMD
基于 IBM Power 的 Linux	HP IA Open VMS	Solaris x86 和 AMD64

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 移动表空间

使用可移动表空间是在两个 Oracle DB 之间移动大量数据的最快方式。使用可移动表空间，可以将 Oracle 数据文件（包含表数据、索引和几乎其它所有 Oracle DB 对象）从一个数据库移动到另一个数据库。而且，与导入和导出类似，可移动表空间除了移动数据以外，还提供了一种移动元数据的机制。

可以使用可移动表空间功能跨平台移动数据。此功能简化了从数据仓库环境向数据集市（数据集市通常在较小的平台上运行）分发数据的过程。此功能还允许通过重建字典并移动用户表空间的方式，将数据库从一个平台迁移到另一个平台。

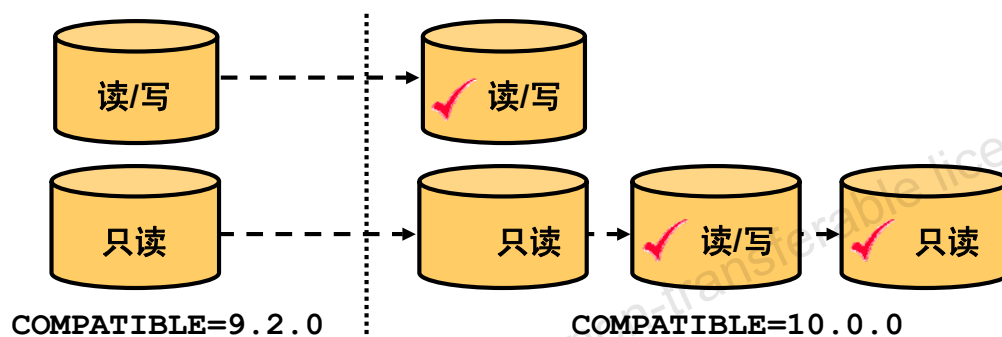
对于相同数据而言，使用可移动表空间移动数据比执行导入/导出或卸载/加载的速度要快得多。这是因为，包含所有实际数据的数据文件只是被复制到目标位置，只有表空间对象的元数据通过数据泵传送到新数据库。

为了能够将数据文件从一个平台移动到另一个平台，必须确保源系统和目标系统都在支持的平台（请参阅幻灯片）上运行。

**注：**可跨平台移动的表空间功能要求两个平台使用相同的字符集。

## 概念：最低兼容级别

- 源数据库和目标数据库都必须将 COMPATIBLE 设置为 10.0.0 或更高。
- 数据文件头是可识别平台的。
- 在移动前，请确保所有只读文件和脱机文件是可识别平台的。



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 概念：最低兼容级别

源数据库和目标数据库都需要先将其数据库 COMPATIBLE 初始化参数提高到 10.0.0 或更高，才能使用可跨平台移动的表空间功能。

COMPATIBLE 设置为 10.0.0（或更高）时，在 Oracle Database 10g 或 11g 下首次打开数据文件时，数据文件将成为可识别平台的。这在图表中体现为复选标记。每个文件都可以确定其所属的平台。这些文件在磁盘上有相同的文件头块格式，这些格式用于文件标识和验证。只读文件和脱机文件只有在变为读/写状态或变为联机之后才能提高其兼容性。这意味着，在 Oracle Database 10g 版本之前的数据库中，处于只读状态的表空间必须至少变为读/写状态一次才能使用可跨平台移动的功能。

## 最低兼容级别

	最低兼容性设置	
	源数据库	目标数据库
<b>移动方案</b>		
相同平台上的数据库	8.0	8.0
表空间的数据库块大小与目标数据库不同	9.0	9.0
不同平台上的数据库	10.0	10.0

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

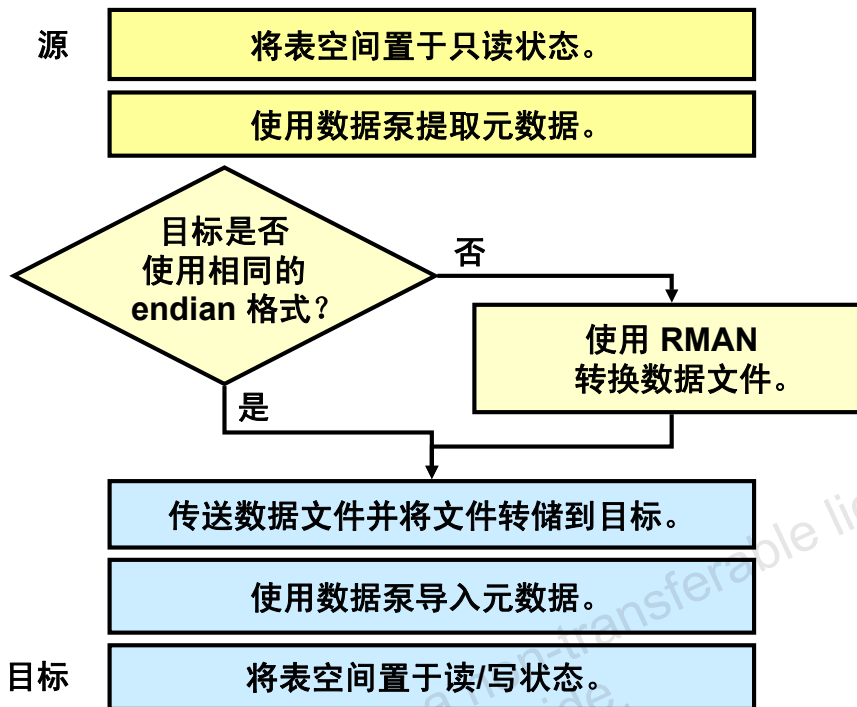
### 最低兼容级别

创建可移动表空间集时，Oracle DB 会计算运行目标数据库必须满足的最低兼容级别。这称为可移动集的兼容级别。从 Oracle Database 11g 开始，始终可将某个表空间移动到具有相同或更高兼容性设置的数据库，而不管目标数据库是基于相同的平台，还是基于不同的平台。如果可移动集的兼容级别高于目标数据库的兼容级别，则数据库将发出错误消息。

下表显示了不同情形下源表空间和目标表空间的最低兼容性要求。源数据库和目标数据库不需要具有相同的兼容性设置。

在第一次打开数据文件后再打开文件时，每个文件都可以确定其所属的平台。这些文件在磁盘上有相同的文件头块格式，这些格式用于文件标识和验证。只读文件和脱机文件只有在变为读/写状态或变为联机之后才能提高其兼容性。

## 可移动表空间过程



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 可移动表空间过程

要将表空间从一个平台移动到另一个平台（源到目标），必须将表空间集内的数据文件转换为目标数据库可以理解的格式。虽然使用 Oracle DB 时磁盘结构符合通用格式，但源平台和目标平台仍有可能使用不同的 endian 格式（字节顺序）。转至不同的 endian 平台时，必须使用 RMAN 实用程序的 CONVERT 命令来转换字节顺序。可以在源平台或目标平台上执行此操作。对于 endian 格式相同的平台，则不需要转换。

幻灯片中的图形描述了从源平台向目标平台移动表空间可能用到的步骤。但是，也可以在将文件传送到目标平台后执行转换。必须在目标平台上执行最后两步。

一般情况下，此过程与使用 Oracle DB Server 的旧发行版时的过程相同，除非两个平台使用不同的 endian 格式。此处假定两个平台都支持跨平台移动。

**注：**读写数据时，字节顺序会影响结果。例如，双字节整数值 1 在 big-endian 系统（如 Sun SPARC Solaris）上写为 0x0001，而在 little-endian 系统（如兼容 Intel 的 PC）上写为 0x0100。

## 确定平台的 Endian 格式

```
SELECT tp.endian_format
FROM   v$transportable_platform tp, v$database d
WHERE  tp.platform_name = d.platform_name;
```

源

目标

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 确定平台的 Endian 格式

可以查询 V\$TRANSPORTABLE\_PLATFORM 来确定两个平台的 endian 顺序是否相同。V\$DATABASE 有两个列可以用于确定您自己的平台名称和平台标识符。运行下面的查询可获得受支持的平台及其 endian 格式的完整列表：

```
SQL> SELECT * FROM V$TRANSPORTABLE_PLATFORM;
```

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
1	Solaris[tm] OE (32-bit)	Big
2	Solaris[tm] OE (64-bit)	Big
7	Microsoft Windows IA (32-bit)	Little
10	Linux IA (32-bit)	Little
6	AIX-Based Systems (64-bit)	Big
3	HP-UX (64-bit)	Big
5	HP Tru64 UNIX	Little
4	HP-UX IA (64-bit)	Big
11	Linux IA (64-bit)	Little
15	HP Open VMS	Little



## 确定平台的 Endian 格式 (续)

PLATFORM_ID	PLATFORM_NAME	ENDIAN_FORMAT
8	Microsoft Windows IA (64-bit)	Little
9	IBM zSeries Based Linux	Big
13	Linux 64-bit for AMD	Little
16	Apple Mac OS	Big
12	Microsoft Windows 64-bit for AMD	Little
17	Solaris Operating System (x86)	Little
18	IBM Power Based Linux	Big
19	HP IA Open VMS	Little
20	Solaris Operating System (AMD64)	Little

## 使用 RMAN 的 CONVERT 命令

RMAN:

- 将表空间、数据文件或数据库转换为目标平台的格式
- 不更改输入文件
- 将转换的文件写入到输出目标

```
CONNECT TARGET SYS@orcl
RMAN>
SQL 'ALTER TABLESPACE hr READ ONLY';
CONVERT TABLESPACE hr
      TO PLATFORM 'Solaris[tm] OE (64-bit)'
      FORMAT '/tmp/transport_to_solaris/%U';;
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 RMAN 的 CONVERT 命令

在执行跨平台传输的准备工作期间，使用 RMAN CONVERT 命令，将表空间、数据文件或数据库转换为目标平台的格式。CONVERT 不改变输入文件，因为转换不是原地执行的。RMAN 将转换的文件写入到指定的输出目标。

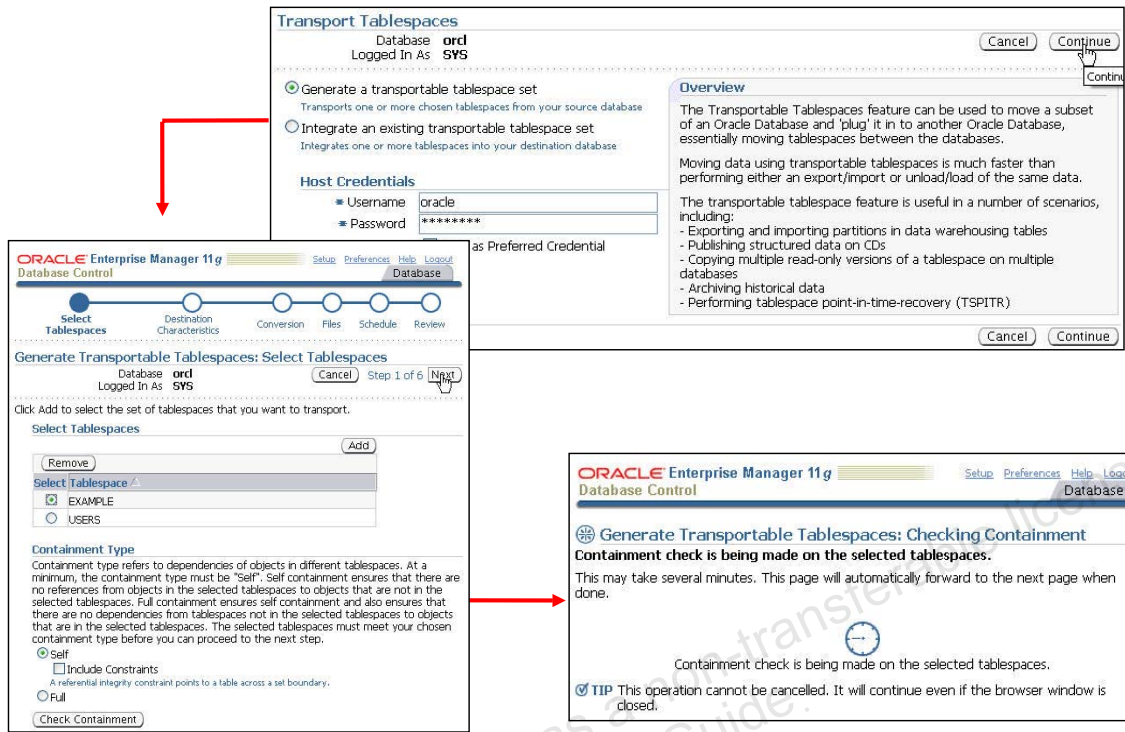
CONVERT TABLESPACE 示例：

- 假设您在 Linux 32 位平台上有一个 ORCL 数据库，您想将其传输到 Solaris 64 位平台上。
- 作为 TARGET 连接至源数据库（已装载或已打开的）。
- 转换时，表空间必须是只读的。
- 结果是 /tmp/transport\_to\_solaris/ 目录中的一组转换的数据文件，其中的数据以适合 Solaris 64 位平台的正确 endian 顺序排列。

限制：CONVERT 命令不处理需要 endian 转换的用户数据类型。如果数据库建立在底层类型（这些类型以平台特定的格式存储数据）之上，要在数据库之间传输对象，可以使用数据泵导入和导出实用程序。

要查看详细的先决条件、用法、限制及语法，请参阅《Oracle Database Backup and Recovery Reference》。

## 使用 Oracle Enterprise Manager 实施可移动表空间



版权所有 © 2011, Oracle。保留所有权利。

### 使用 Oracle Enterprise Manager 实施可移动表空间

可使用 Oracle Enterprise Manager 来实施可移动表空间。在“Database（数据库）”主页中，单击“Data Movement（数据移动）”文件夹选项卡，然后单击“Move Database Files（移动数据库文件）”部分下的“Transport Tablespaces（移动表空间）”。选择“Generate a transportable table set（生成一个可移动表集）”，并提供 oracle 用户的登录身份证明，然后单击“Continue（继续）”。在“Select Tablespaces（选择表空间）”页上，单击“Tablespace（表空间）”按钮从显示的列表中添加要移动的表空间。在接近页面底部的位置，选择要在处理表空间之前完成的封闭性检查的级别（必须选择一项）。封闭性检查会检查表空间中的对象依赖关系。完成后，单击“Next（下一步）”。在封闭性检查运行过程中，等待一会儿。如果该项检查发现任何问题，请先解决问题然后再继续。

## 使用 Oracle Enterprise Manager 实施可移动表空间

**ORACLE Enterprise Manager 11g** Database Control

Generate Transportable Tablespaces: Destination Characteristics

Database: ord  
Logged In As: SYS  
Source Platform: Linux IA (32-bit)

Destination Database Platform

Destination Platform: Linux IA (32-bit)  
The Conversion step will be skipped if the destination platform is compatible.

Destination Character Set

For the tablespaces to be successfully integrated at the destination database, the source destination database must have compatible character sets. Specify the character set of the destination database to check for compatibility.

☒ Verify Compatibility of Destination Character Sets

Destination Database Character Set: AL32UTF8  
Destination National Character Set: AL16UTF16

**ORACLE Enterprise Manager 11g** Database Control

Generate Transportable Tablespaces: Files

Database: ord  
Logged In As: SYS

Dump File

Specify the location where the dump file associated with the transportable tablespace can be generated.

\* Dump File Directory: /u01/app/oracle/oradata/ord/  
\* Dump File Name: EXPDAT\_GENERATETTS000041.DMP

Datfiles

A copy of the datfiles associated with the selected tablespaces is needed. Selected tablespaces will be made read-only as part of Generate Transportable Tablespaces operation.

☒ Copy Datfiles Automatically (Recommended)  
The datfiles will be copied to the above specified dump file directory. This operation will require 105 MB of free disk space. The selected tablespaces will be made read-only for the duration of the job.

☐ Manually Copy Datfiles  
You are responsible for copying the datfiles. Tablespaces remain in read-only mode after the job is completed to maintain datfile consistency with the dump file. You must manually put tablespaces back into read-write mode after copying datfiles.

版权所有 © 2011, Oracle. 保留所有权利。

### 使用 Oracle Enterprise Manager 实施可移动表空间（续）

在“Destination Characteristics（目标特征）”页上，必须提供目标平台和字符集。在“Destination Database Platform（目标数据库平台）”部分下，从下拉列表中选择目标计算机的操作系统。如果目标平台与源平台不同，Oracle Enterprise Manager 会执行数据转换。继续转到该页的“Destination Character Set（目标字符集）”部分，从下拉列表中选择目标字符集和国家字符集。这些字符集必须与源字符集兼容。单击“Next（下一步）”继续时，Oracle Enterprise Manager 会检查字符集的兼容性。如果选择的字符集被标记为不兼容，则会返回到“Destination Characteristics（目标特征）”页以便您更正所做的选择。

## 使用 Oracle Enterprise Manager 实施可移动表空间

**Generate Transportable Tablespaces: Schedule**

Database: **orcl**  
Logged In As: **sys**

Specify a name and description for the export job. Specify a date to start the job.

**Job Parameters**

Job Name: **GENERATETTS000041**  
Description: **Transport Tablespace(s)**

**Job Schedule**

Start: **Immediately**

**Generate Transportable Tablespaces: Review**

Database: **orcl**  
Logged In As: **sys**

**Summary**

Job Name: **GENERATETTS000041**  
Transport Tablespaces: **EXAMPLE,USERS**  
Containment Type: **SELF**  
Include Constraints: **false**  
Source Platform: **Linux IA (32-bit)**  
Destination Platform: **Linux IA (32-bit)**  
Disk Space Required (MB): **105**  
Copy the datafiles: **AUTO**  
Dump File Format: **Datapump**

**Dump File**

Dump File Directory: **/u01/app/oracle/oradata/orcl/**  
Dump File Name: **EXPDAT\_GENERATETTS000041.DMP**

**Datafiles**

Datafile Name	New Datafile Name	Size (MB)
/u01/app/oracle/oradata/orcl/users01.dbf	/u01/app/oracle/oradata/orcl/users01.dbf	5
/u01/app/oracle/oradata/orcl/example01.dbf	/u01/app/oracle/oradata/orcl/example01.dbf	100

版权所有 © 2011, Oracle。保留所有权利。

### 使用 Oracle Enterprise Manager 实施可移动表空间（续）

在“Schedule（调度）”页上，为默认作业名称提供有意义的说明。可以选择立即启用作业，也可以调度作业以后执行。完成了选择后，单击“Next（下一步）”按钮继续操作。在复查页上，可以验证您的选择，然后提交作业以便执行。如果各个条目都正确，则单击“Submit Job（提交作业）”按钮。单击“Back（上一步）”按钮以更正任何不正确的条目。

## 移动数据库

- 概述可移动表空间功能。
- 可以轻松地从数据仓库环境向数据集市（数据集市通常在较小的平台上运行）分发数据子集。
- 数据库可以非常迅速地从一個平台迁移到另一个平台。



ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 移动数据库

可以使用可移动表空间功能将一个数据库迁移到另一个平台，方法是在目标平台上创建一个新数据库，然后移动所有用户表空间。不能移动 SYSTEM 表空间。因此，诸如序列、PL/SQL 程序包以及依赖于 SYSTEM 表空间的其它对象均不会被移动。必须在目标数据库上手动创建这些对象，或者使用数据泵移动不能通过可移动表空间移动的对象。

要从一个平台向另一个平台移动数据库，必须确保源系统和目标系统都运行于 V\$TRANSPORTABLE\_PLATFORM 中列出的平台上，并且具有相同的 endian 格式。例如，可以将 Linux IA (32 位) 上运行的数据库移动到一个 Windows 平台上。

如果两个数据库中至少有一个使用自动存储管理 (ASM)，则可能需要使用 DBMS\_FILE\_TRANSFER 程序包来将文件上传到 FTP。

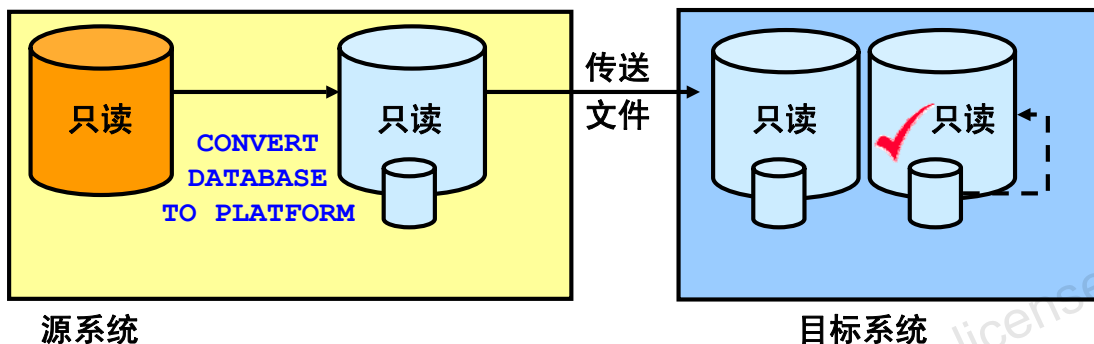
此功能与可移动表空间不同，可移动表空间有一个可以插入数据的目标数据库，而此功能将在目标平台上创建新的数据库。新创建的数据库中包含的数据与源数据库中相同。除了数据库名、实例名和文件位置等，新数据库与源数据库的设置也相同。

**注：**移动数据库比使用数据泵移动数据的速度快。



## 数据库移动过程：源系统转换

在只读模式下打开数据库，  
并且 COMPATIBLE=10.0.0 或更高。



### 数据库移动过程：源系统转换

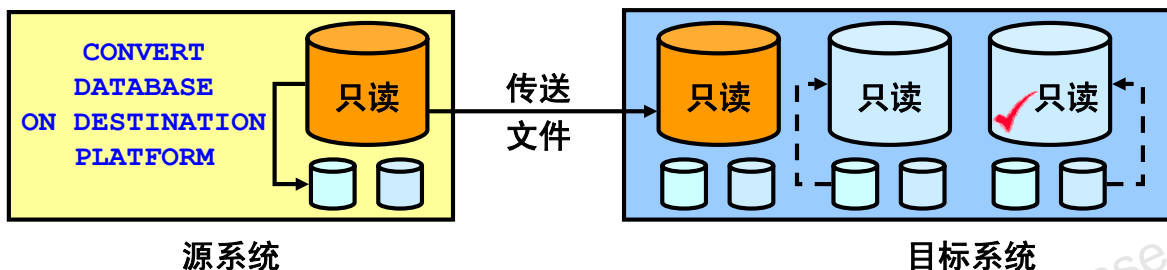
必须先以只读模式打开数据库，然后才能移动该数据库。然后使用 RMAN 转换数据库的必要数据文件。

在源平台上执行转换时，RMAN 命令 `CONVERT DATABASE` 将生成一个包含正确的 `CREATE CONTROLFILE RESETLOGS` 命令的脚本，此命令用于在目标系统上创建新数据库。然后 `CONVERT DATABASE` 命令将转换识别出的所有数据文件，以便这些文件可以在目标系统上使用。接着，您需要将已转换的数据文件和生成的脚本传送到目标平台。通过在目标平台上执行生成的脚本，创建数据库的新副本。

**注：**源数据库必须在 COMPATIBLE 初始化参数设为 10.0.0 或更高的条件下运行。自 COMPATIBLE 设置为 10.0.0 或更高版本以来，识别出的所有表空间必须至少已被读写过一次。

## 数据库移动过程：目标系统转换

在只读模式下打开数据库，  
并且 COMPATIBLE=10.0.0 或更高。



版权所有 © 2011, Oracle。保留所有权利。

### 数据库移动过程：目标系统转换

必须先以只读模式打开数据库，然后才能移动该数据库。然后使用 RMAN 转换数据库的必要数据文件。

在目标平台上执行转换时，（在源系统上执行的）CONVERT DATABASE 命令只生成两个脚本，用于在目标系统上转换数据文件以及重新创建新数据库的控制文件。然后，请将识别出的数据文件和两个脚本都传送到目标平台。完成后，请按正确的顺序执行这两个脚本。第一个脚本使用现有的 CONVERT DATAFILE RMAN 命令来进行转换，第二个脚本对已转换的数据文件发出 CREATE CONTROLFILE RESETLOGS SQL 命令来创建新数据库。

**注：**源数据库必须在 COMPATIBLE 初始化参数设为 10.0.0 或更高的条件下运行。自 COMPATIBLE 设置为 10.0.0 或更高版本后，识别出的所有表空间至少已被读写过一次。



## 数据库移动：注意事项

- 在目标平台上创建口令文件。
- 移动在源数据库中使用的 BFILE。
- 生成的 PFILE 和移动脚本使用 OMF。
- 使用 DBNEWID 更改 DBID。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 数据库移动：注意事项

重做日志、控制文件和临时文件不会移动。将在目标平台上为新数据库重新创建这些文件。因此，目标平台上的新数据库必须用 RESETLOGS 选项打开。

使用口令文件时，不会移动该文件，需要您在目标平台上创建该文件。这是因为口令文件支持的文件名类型是特定于操作系统的。但是，CONVERT DATABASE 命令的输出将列出所有用户名及其系统权限，并且建议在目标平台上重新创建口令文件并为这些用户添加条目。

CONVERT DATABASE 命令列出源数据库中的所有目录对象以及使用 BFILE 数据类型或外部表的对象。可能需要用新的目录名和文件名来更新这些对象。如果在数据库中使用了 BFILE，则必须移动 BFILE。

生成的 PFILE 和移动脚本将 Oracle Managed Files (OMF) 用于数据库文件。如果不想使用 OMF，则必须修改 PFILE 和移动脚本。

移动的数据库与源数据库具有相同的 DBID。可以使用 DBNEWID 实用程序来更改 DBID。在移动脚本以及 CONVERT DATABASE 命令的输出中，系统将提示您使用 DBNEWID 实用程序来更改数据库 ID。

## 测验

选择表述正确的语句：

1. RMAN CONVERT 命令执行原地转换，所以您的输入文件在传输至目标前会被更改。
2. 读/写表空间在进行 endian 转换时需要处于只读模式。
3. 可以对表、表空间和数据库使用 RMAN CONVERT 命令。
4. 可以将数据库传输至数据仓库环境。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、4

## 小结

在本课中，您应该已经学会：

- 描述 4 KB 扇区磁盘的概念及使用
- 介绍可移动表空间和数据库的概念

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 19 概览：管理数据库的空间

本练习包含以下主题：

- 查看“使用 4 KB 扇区磁盘”的演示

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

# 20

## 复制数据库

ORACLE®

版权所有 © 2011, Oracle。保留所有权利。

## 课程目标

学完本课后，应能完成以下工作：

- 列出创建副本数据库的目的
- 选择用于复制数据库的方法
- 使用 RMAN 复制数据库
- 使用 RMAN 备份复制数据库
- 基于正在运行的实例复制数据库

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 使用副本数据库

- 使用副本数据库可执行以下任务：
  - 测试备份和恢复过程
  - 通过创建导出文件并将对象导入生产数据库来恢复对象
- 创建副本数据库：
  - 使用 RMAN 的 `DUPLICATE` 命令
  - 在同一主机上或在不同的主机上
  - 包含与源相同的内容或源的子集
  - 对于基于备份的复制，是通过辅助通道执行的
  - 对于活动的数据库复制，是通过目标通道执行的

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 使用副本数据库

副本数据库是您的目标数据库的副本，它具有一个新的唯一数据库标识符 (DBID)。通过独立运行副本数据库（不依赖目标数据库）可执行以下任务：

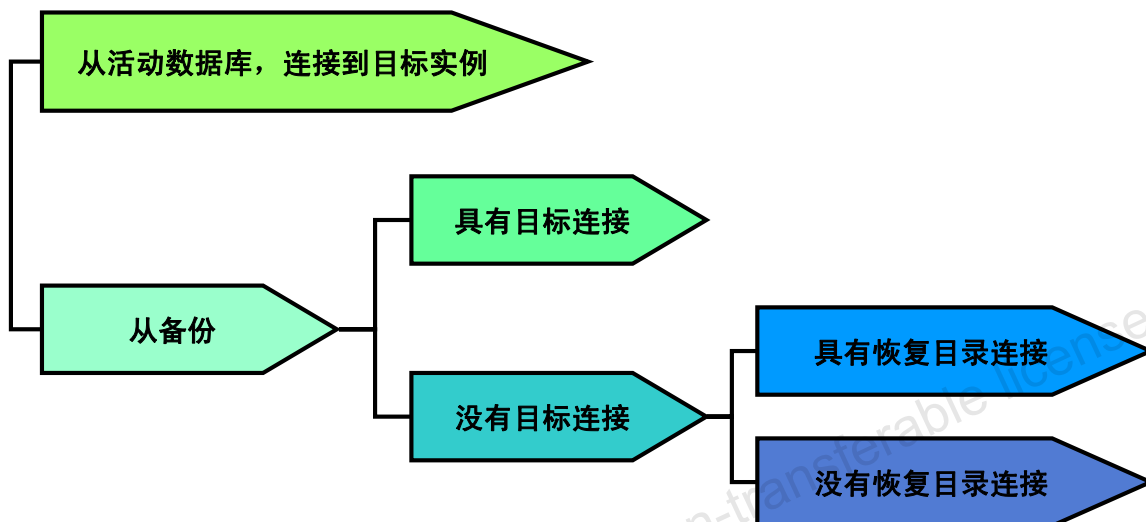
- 测试备份和恢复过程。
- 创建包含副本数据库中对象的导出文件并将这些对象导入生产数据库，恢复不小心从目标数据库中删除的对象。尽管您可能发现闪回查询、闪回删除和闪回表是用来恢复对象的更简单快速的解决方案。

创建副本数据库：

- 可以使用 RMAN 的 `DUPLICATE` 命令在同一主机上或不同的主机上创建副本数据库。
- 副本数据库可包含与源数据库相同的内容或仅包含源数据库的子集（本课程稍后会详细介绍）。
- 复制的主要工作是由辅助通道执行的。对于基于备份的复制，这些通道对应于目标主机上的辅助实例上的服务器会话。
- 对于活动数据库复制，由目标通道执行将数据文件副本推入到辅助实例的工作。

## 选择数据库复制方法

选择复制数据库的方法 — 始终连接到辅助实例：



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 数据库复制方法

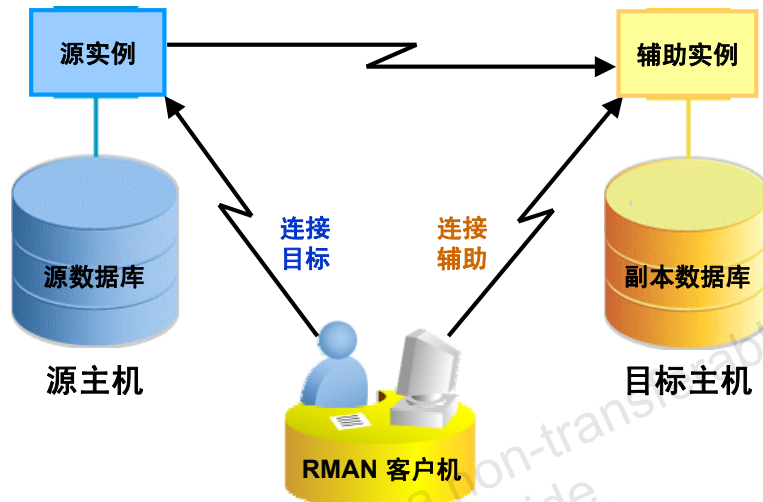
可将源数据库复制到位于同一计算机或其它计算机上的目标数据库。与副本数据库关联的数据库实例称为辅助实例。所有复制方法都需要连接到辅助实例。此图表显示了以下数据库复制方法：

- 从活动数据库，连接到目标实例和辅助实例
- 从备份，连接到目标实例和辅助实例
- 从备份，连接到辅助实例，不连接到目标实例，但具有恢复目录连接
- 从备份，连接到辅助实例，不连接到目标实例和恢复目录



## 复制活动数据库

- 使用网络（不需要备份）
- 包括自定义 SPFILE
- 通过 Oracle Enterprise Manager 或 RMAN 命令行



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

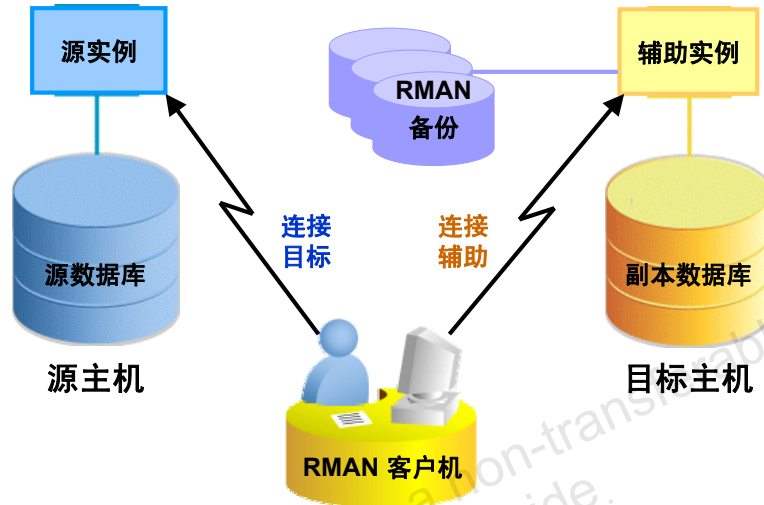
### 复制活动数据库

通过使用 Oracle Enterprise Manager 或 RMAN DUPLICATE 命令的 FROM ACTIVE DATABASE 子句，可以指示源数据库直接将映像副本和归档日志副本复制到辅助实例。此操作不需要使用备份。RMAN 作为 TARGET 连接到源数据库实例并作为 AUXILIARY 连接到辅助实例（如幻灯片中所示）。

通过实例间的网络连接将数据库文件从源数据库复制到目标数据库或辅助实例。RMAN 随后使用“内存脚本”（仅包含在内存中）完成恢复并打开数据库。

## 通过目标连接复制数据库

- 连接到目标数据库（源数据库）
- 连接到辅助实例
- 可选，连接到恢复目录（或使用目标控制文件）



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

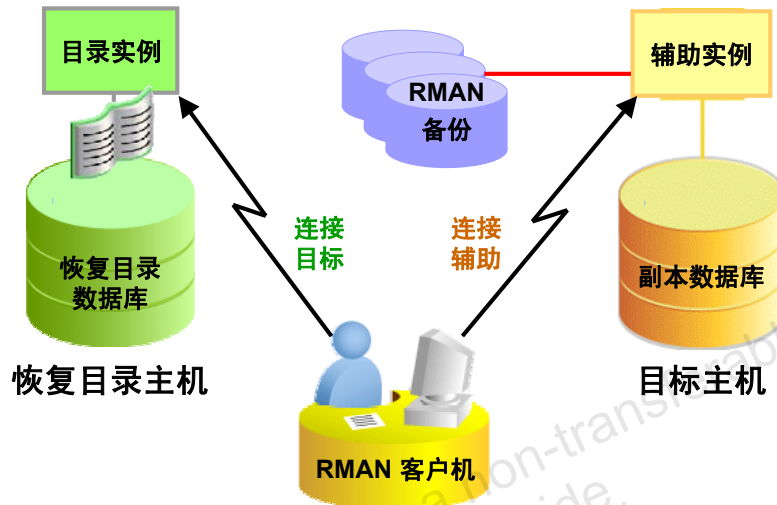
### 通过目标连接复制数据库

当通过目标数据库连接复制数据库时，RMAN 可从目标数据库控制文件或从恢复目录获取有关备份的元数据。

此图表描述了通过目标连接执行的基于备份的复制。RMAN 连接到源数据库实例和辅助实例。此外，RMAN 还可以连接到恢复目录数据库（图中未显示）。目标主机必须具有对创建副本数据库所需的 RMAN 备份的访问权限。

## 在具有恢复目录但没有目标连接的情况下复制数据库

- 连接到备份元数据的恢复目录
- 连接到辅助实例，该实例必须具有对 RMAN 备份的访问权限



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

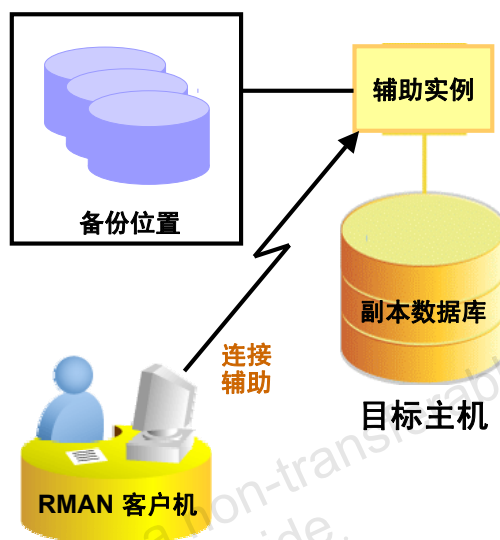
### 在没有目标连接的情况下复制数据库

如果在复制数据库时没有连接到目标数据库，但具有恢复目录，则 RMAN 将使用恢复目录获取有关备份的元数据。

此图表描述了没有目标连接时基于备份的复制。RMAN 连接到恢复目录数据库实例和辅助实例。目标主机必须具有对创建副本数据库所需的 RMAN 备份的访问权限。

## 在没有恢复目录或目标连接的情况下复制数据库

连接到辅助实例，该实例必须具有对磁盘备份位置的访问权限



ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 在没有恢复目录或目标连接的情况下复制数据库

如果在没有目标数据库连接且没有恢复目录的情况下复制数据库，则 RMAN 将使用所有必需的备份和副本所在的备份位置。

此图表描述了在没有连接到目标数据库实例或恢复目录数据库实例的情况下执行的基于备份的复制。包含复制所需的所有备份或副本的磁盘备份位置必须可供目标主机使用。

## 创建基于备份的副本数据库

1. 创建辅助实例的 Oracle 口令文件。
2. 建立到辅助实例的 Oracle Net 连接。
3. 创建辅助实例的初始化参数文件。
4. 在 NOMOUNT 模式下启动辅助实例。
5. 装载或打开目标数据库。
6. 确保备份和归档重做日志文件可用。
7. 根据需要分配辅助通道。
8. 执行 DUPLICATE 命令。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 创建基于备份的副本数据库

了解以上基本步骤和 RMAN 数据库复制过程非常重要。

如果您使用的是 Oracle Enterprise Manager 界面，向导可帮助您执行大部分步骤。如果您使用命令行界面创建副本数据库，则需要手动执行这些步骤。您还可将 EM 界面用作测试或示例，并以输出日志为基础编写自己的数据库复制脚本。

本幻灯片中概括了创建副本数据库的基本步骤。本课会详细介绍其中的一些步骤。

## 创建辅助实例的初始化参数文件

请按如下方式指定参数：

- **DB\_NAME**
  - 如果副本数据库和目标数据库在同一 Oracle 主目录中，则它们的名称必须不同。
  - 在 **DUPLICATE** 命令中使用相同的值。
- **DB\_BLOCK\_SIZE**
  - 指定的值应与为目标数据库设置的值相同。

**ORACLE**

版权所有 © 2011, Oracle。保留所有权利。

### 创建辅助实例的初始化参数文件

您必须为辅助实例创建文本初始化参数文件。文本初始化参数文件驻留的主机必须与用来执行 **DUPLICATE** 命令的 **RMAN** 客户机驻留的主机是同一主机。

请注意下列每个参数的要求：

- **DB\_NAME**：如果目标数据库和副本数据库在同一 Oracle 主目录中，则必须将 **DB\_NAME** 设置为不同的名称。如果它们在不同的 Oracle 主目录中，则必须确保副本数据库名称不同于其 Oracle 主目录中的其它名称。请确保执行 **DUPLICATE** 命令时使用的数据库名称与为此参数设置的名称相同。
- **DB\_BLOCK\_SIZE**：辅助数据库的块大小必须与目标数据库的块大小匹配。辅助数据库初始化参数文件中指定的值必须与目标数据库初始化参数文件中设置的值相同。如果没有在目标数据库的初始化参数文件中设置此参数，在辅助实例初始化参数文件中也不要设置它。

另外，请务必验证指定路径名的所有初始化参数的设置。请确认在副本数据库主机上可以访问所有指定的路径。

## 为目标指定新名称

可用的方法：

- SET NEWNAME 命令
- CONFIGURE AUXNAME 命令（对恢复集数据文件已弃用）
- 在 DUPLICATE 命令中指定 DB\_FILE\_NAME\_CONVERT 参数

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 为目标指定新名称

可以使用下列方法指定数据文件的新名称：

- 将 SET NEWNAME FOR DATAFILE 命令包含在 RUN 块中来指定数据文件的新名称。
- 使用 CONFIGURE AUXNAME 命令。CONFIGURE AUXNAME 是 SET NEWNAME 的一种替代方法。区别是首次配置辅助名称后，其它 DUPLICATE 命令将重用所配置的设置。相对而言，每次执行 DUPLICATE 命令时必须重新发出 SET NEWNAME。  
**注：**对于恢复集数据文件，使用 SET NEWNAME 替换 CONFIGURE AUXNAME。
- 随 DUPLICATE 命令指定 DB\_FILE\_NAME\_CONVERT 参数。

## 使用 SET NEWNAME 子句

- SET NEWNAME 子句用于为数据库或命名表空间中的所有文件指定默认名称格式。
- 默认名称用于 RUN 块中的 DUPLICATE、RESTORE 和 SWITCH 命令。
- 它使得您可以通过单个命令设置文件名称，而无需分别设置每个文件名称。

```
SET NEWNAME FOR DATABASE
TO {NEW| 'formatSpec'};
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 使用 SET NEWNAME 子句

您可以使用 SET NEWNAME 为命名表空间中的所有数据文件和数据库中的所有数据文件指定默认名称格式。

SET NEWNAME 命令的优先顺序如下所示：

1. SET NEWNAME FOR DATAFILE 和 SET NEWNAME FOR TEMPFILE
2. SET NEWNAME FOR TABLESPACE
3. SET NEWNAME FOR DATABASE

示例：

```
RUN
{
  SET NEWNAME FOR DATABASE TO '/u01/app/oracle/oradata/duplodb/%b';
  DUPLICATE TARGET DATABASE TO duplodb
  LOGFILE
  GROUP 1 ('/u01/app/oracle/oradata/duplodb/redo01a.log',
    '/u01/app/oracle/oradata/duplodb/redo01b.log') SIZE 50M REUSE,
  GROUP 2 ('/u01/app/oracle/oradata/duplodb/redo02a.log',
    '/u01/app/oracle/oradata/duplodb/redo02b.log') SIZE 50M REUSE,
  GROUP 3 ('/u01/app/oracle/oradata/duplodb/redo03a.log',
    '/u01/app/oracle/oradata/duplodb/redo03b.log') SIZE 50M REUSE;
}
```



## SET NEWNAME 的替代变量

语法元素	说明
%b	指定不带目录路径的文件名
%f	指定为其生成新名称的数据文件的绝对文件号
%I	指定 DBID
%N	指定表空间名称
%U	指定如下格式的系统生成文件名: data-D-%d_id-%I_TS-%N_FNO-%f

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### SET NEWNAME 的替代变量

发出 SET NEWNAME FOR DATABASE 或 SET NEWNAME FOR TABLESPACE 时, 必须在 TO <filename> 子句中指定替代变量以避免名称冲突。请至少指定以下替代变量中的一种: %b、%f 和 %U。%I 和 %N 是可选变量。

## 指定用于文件命名的参数

或者指定下列参数来显式控制辅助数据库的文件命名：

- CONTROL\_FILES
- DB\_FILE\_NAME\_CONVERT
- LOG\_FILE\_NAME\_CONVERT

```
CONTROL_FILES='/u01/app/oracle/oradata/aux/control01.ctl',
              '/u01/app/oracle/oradata/aux/control02.ctl',
              '/u01/app/oracle/oradata/aux/control03.ctl'
DB_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                    '/u01/app/oracle/oradata/aux'
LOG_FILE_NAME_CONVERT='/u01/app/oracle/oradata/orcl',
                    '/u01/app/oracle/oradata/aux'
```

ORACLE

版权所有 © 2011, Oracle. 保留所有权利。

### 指定用于文件命名的参数

执行 DUPLICATE 命令时，RMAN 会生成必需的数据库文件的名称。通过在辅助实例初始化参数文件中指定下列初始化参数，可以控制这些文件的命名：

- **CONTROL\_FILES:** 可以使用此参数指定控制文件的名称。如果未使用此参数设置名称，Oracle Server 会在默认控制目标中创建 Oracle 管理的控制文件。有关具体的信息，请参阅《SQL 参考手册》中的 SQL CREATE CONTROLFILE 命令。
- **DB\_FILE\_NAME\_CONVERT:** 此参数用于指定辅助数据库的数据文件名。其格式为 DB\_FILE\_NAME\_CONVERT = 'string1', 'string2'，其中 string1 是目标数据库文件名的模式，string2 是辅助数据库文件名的模式。还可以指定 DB\_FILE\_NAME\_CONVERT 参数作为 DUPLICATE DATABASE 命令的选项。
- **LOG\_FILE\_NAME\_CONVERT:** 此参数用于指定辅助数据库的重做日志文件名。其格式为 LOG\_FILE\_NAME\_CONVERT = 'string1', 'string2'，其中 string1 是目标数据库文件名的模式，string2 是辅助数据库文件名的模式。还可以使用 DUPLICATE DATABASE 命令的 LOGFILE 子句来指定重做日志文件名。

### 指定用于文件命名的参数（续）

除了使用初始化参数控制文件命名之外，还可以使用下列方法重命名重做日志文件：

- 使用 DUPLICATE 命令的 LOGFILE 子句。
- 设置 Oracle Managed Files 的初始化参数：DB\_CREATE\_FILE\_DEST、DB\_CREATE\_ONLINE\_DEST\_n 或 DB\_RECOVERY\_FILE\_DEST。

## 在 NOMOUNT 模式下启动实例

- 在 NOMOUNT 模式下启动辅助实例。
- 根据用来启动实例的文本初始化参数文件创建服务器参数文件 (SPFILE)。

```
SQL> startup nomount pfile='$HOME/auxinstance/initAUX.ora'
ORACLE instance started.

Total System Global Area  285212672 bytes
Fixed Size                  1218992 bytes
Variable Size              92276304 bytes
Database Buffers          188743680 bytes
Redo Buffers               2973696 bytes
SQL> create spfile
      2  from pfile='$HOME/auxinstance/initAUX.ora';

File created.
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 在 NOMOUNT 模式下启动实例

创建文本初始化参数文件后，通过调用 SQL\*Plus 可在 NOMOUNT 模式下启动辅助实例。

调用 SQL\*Plus 后，就会根据文本初始化参数文件创建服务器参数文件 (SPFILE)。可以在启动实例前后执行 CREATE SPFILE。应该在默认位置创建 SPFILE，这样不需要在 DUPLICATE 命令中指定 PFILE 选项。RMAN 在复制过程中会关闭辅助实例，然后再重新启动它。因此，如果不使用 SPFILE，就必须指定 PFILE 选项。

## 确保备份和归档重做日志文件可用

- 在副本主机上必须能访问所有目标数据库数据文件的备份。
- 备份可以是完全备份和增量备份的组合。
- 在副本主机上必须能访问恢复副本数据库所需的归档重做日志文件。
- 归档重做日志文件可以是：
  - 介质管理器上的备份
  - 映像副本
  - 实际的归档重做日志文件

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 确保备份和归档重做日志文件可用

在副本主机上必须能访问还原数据文件所需的备份。您并不需要完整的数据库备份。

RMAN 在复制过程中可使用单个数据文件的完全备份和增量备份的组合。

另外，还必须能访问将副本数据库恢复到期望的时间点所需的归档重做日志。这些归档重做日志文件可以是备份、映像副本或实际的归档重做日志。备份或副本可传送到副本数据库节点的本地磁盘上，或者通过网络文件系统 (NFS) 之类的某种方式通过网络进行装载。

## 分配辅助通道

- 辅助通道指定 RMAN 和辅助数据库实例之间的连接。
- 如果未配置自动通道，请分配辅助通道：
  - 启动 RMAN 时连接到目标数据库实例、辅助实例和恢复目录（如果适用）。
  - 在 RUN 块中分配至少一个辅助通道。

```
$ rman target sys/oracle_4U@trgt auxiliary  
sys/oracle_4U@auxdb  
RMAN> RUN  
    {ALLOCATE AUXILIARY CHANNEL aux1 DEVICE TYPE DISK;  
    ALLOCATE AUXILIARY CHANNEL aux2 DEVICE TYPE DISK;  
    ...  
    DUPLICATE TARGET DATABASE to auxdb; . . .
```

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 分配辅助通道

如果未配置自动通道，请在发出 DUPLICATE 命令之前，至少手动分配一个辅助通道。ALLOCATE AUXILIARY CHANNEL 命令必须与 DUPLICATE 命令位于同一 RUN 块中。ALLOCATE AUXILIARY CHANNEL 命令中指定的通道类型必须与目标数据库备份所在的介质相匹配。

- 如果备份驻留在磁盘上，则可分配多个通道以减少复制过程所花费的时间。
- 对于磁带备份，可以指定与可用设备数相对应的通道数。

必须使用 NOMOUNT 选项启动辅助实例，且目标数据库必须已装载或已打开。

## 了解 RMAN 复制操作

当您执行 DUPLICATE 命令时，RMAN 会执行下列操作：

- 1A. 为辅助实例创建控制文件服务器参数文件（对于活动数据库和使用目标连接执行的基本备份的复制），或：
- 1B. 从备份中还原（对于备用数据库和没有目标连接的基于备份的复制）
2. 装载备份控制文件
3. 对于基于备份的复制：选择用于将数据文件还原到辅助实例的备份
4. 将目标数据文件还原到副本数据库
5. 使用所有可用的增量备份和归档重做日志文件执行不完全恢复

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 了解 RMAN 复制操作

执行 DUPLICATE 命令时，RMAN 将执行幻灯片中列出的操作。

1A. 如果满足以下条件，RMAN 将为辅助实例创建默认服务器参数文件：

- 复制不涉及备用数据库。
- 不对服务器参数文件进行复制。
- 辅助实例不是使用服务器参数文件启动的。

1B. RMAN 从备份中还原 — 对于备用数据库和没有目标连接的基于备份的复制，始终如此。

2. RMAN 从活动数据库装载已还原或复制的备份控制文件。

3. 对于基于备份的复制：RMAN 使用 RMAN 资料档案库选择用于将数据文件还原到辅助实例的备份。

4. RMAN 还原并复制数据文件副本。

5. RMAN 使用增量备份和归档重做日志文件将数据文件恢复到非当前时间点。RMAN 必须执行数据库时间点恢复，即使不为复制提供显式时间点也是如此。因为源数据库中的联机重做日志文件未备份且无法应用于副本数据库，所以时间点恢复是必需的。副本数据库的最远恢复点是由源数据库归档的最近重做日志文件。

## 了解 RMAN 复制操作

当您执行 `DUPLICATE` 命令时，RMAN 会执行下列操作：

6. 关闭辅助实例并以 `NOMOUNT` 模式重新启动它
7. 新建一个控制文件，然后控制文件在数据文件中创建并存储新的 `DBID`
8. 通过 `RESETLOGS` 选项打开副本数据库
9. 为副本数据库创建联机重做日志文件

注：数据库复制过程在重新执行时尝试从故障点继续。



版权所有 © 2011, Oracle。保留所有权利。

### 了解 RMAN 复制操作（续）

6. RMAN 关闭数据库实例并以 `NOMOUNT` 模式重新启动它。
7. RMAN 新建一个控制文件，该控制文件然后在副本数据库的数据文件中创建并存储新的唯一数据库标识符 `DBID`。
8. RMAN 通过 `RESETLOGS` 选项打开副本数据库。
9. RMAN 为副本数据库创建联机重做日志文件。

注：如果 `DUPLICATE DATABASE` 命令失败，可重新执行 `DUPLICATE DATABASE` 命令，复制过程尝试从故障点继续执行。



## 指定 DUPLICATE 命令的选项

可以在 DUPLICATE 命令中指定下列选项：

选项	用途
SKIP READONLY	排除只读表空间
SKIP TABLESPACE	排除命名表空间
TABLESPACE	包括命名表空间
NOFILENAMECHECK	防止检查文件名
OPEN RESTRICTED	自动启用 RESTRICTED SESSION

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

### 指定 DUPLICATE 命令的选项

在执行 DUPLICATE 命令时根据需要指定其它选项。

**SKIP READONLY:** 用于排除只读表空间数据文件。

**SKIP TABLESPACE:** 用于从目标数据库中排除表空间。不能排除 SYSTEM 表空间或包含还原段或回退段的表空间。

**TABLESPACE:** 用来包括目标数据库中的表空间。

**NOFILENAMECHECK:** 用于防止 RMAN 检查目标数据库数据文件是否与正在使用的副本数据库数据文件同名。当目标数据库和副本数据库的数据文件和重做日志文件使用相同的名称时，必须指定此选项。当创建副本数据库的主机具有与目标数据库主机一样的磁盘配置、目录结构和文件名时，通常使用此选项。如果这种情况下未指定 NOFILENAMECHECK，RMAN 会返回错误。

**OPEN RESTRICTED:** 用于在数据库打开之后自动启用 RESTRICTED SESSION。

## 使用其它 DUPLICATE 命令选项

选项	用途
NOREDO	向 RMAN 发出信号，指出在恢复过程中不应用重做日志 如果目标数据库在备份时处于 NOARCHIVELOG 模式，则该选项必须同无目标 DUPLICATE 一起使用 还可用于显式声明不应用任何归档重做日志文件
UNDO TABLESPACE	当目标数据库未打开且不存在恢复目录连接时，必须指定该选项以便 RMAN 不检查表空间来查找 SYS 拥有的对象

ORACLE

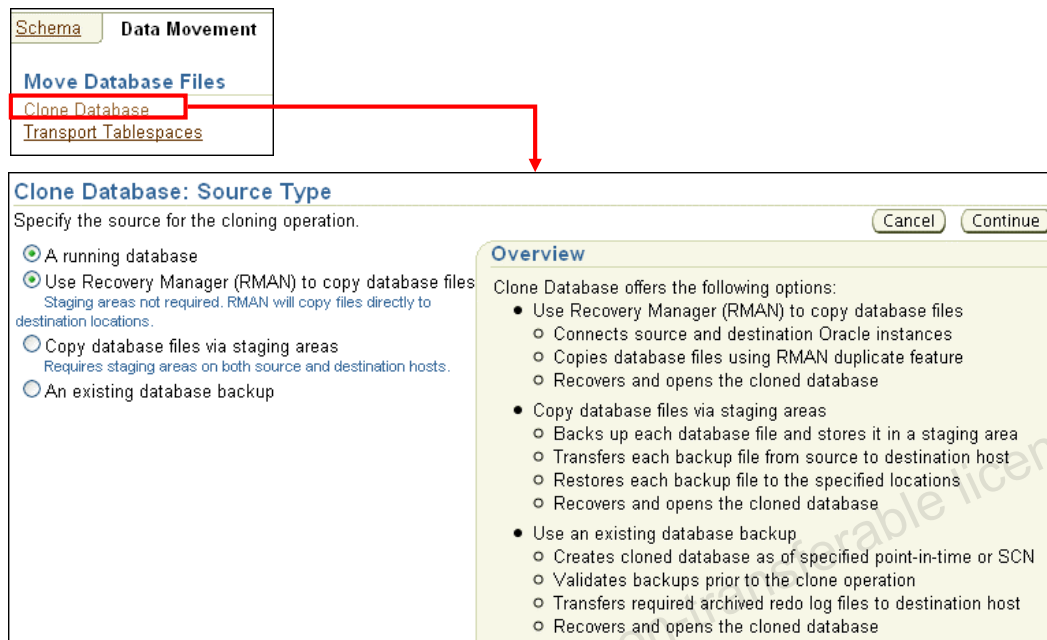
版权所有 © 2011, Oracle。保留所有权利。

### 使用其它 DUPLICATE 命令选项

Oracle Database 11g 发行版 2 中引入了 DUPLICATE 命令的以下其它选项：

- **NOREDO:** NOREDO 选项用于向 RMAN 发出信号，指出在复制操作的恢复阶段不应用重做日志。在下列情况下应指定此选项：在备份时数据库处于 NOARCHIVELOG 模式，或者在复制操作过程中归档重做日志文件不可用。如果当前处于 ARCHIVELOG 模式的数据库正被复制到它在 NOARCHIVELOG 模式时的一个时间点，使用此选项非常合适。  
如果正在执行无目标 DUPLICATE，且该数据库处于 NOARCHIVELOG 模式，则必须使用此选项将数据库模式告知 RMAN。如果未连接到目标数据库，RMAN 将无法确定模式。
- **UNDO TABLESPACE:** 在非整体数据库复制过程中，RMAN 会检查所复制的所有表空间以确认不存在属于 SYS 用户的对象。SYSTEM、SYSAUX 和还原段表空间不在此检查的范围内。但是，如果复制过程中未打开目标数据库，且未使用恢复目录，RMAN 将无法获取还原表空间的名称。所以必须使用 UNDO TABLESPACE 选项提供还原段表空间的名称。

## 使用 EM 克隆数据库



版权所有 © 2011, Oracle. 保留所有权利。

### 使用 EM 克隆数据库

还可以使用 Oracle Enterprise Manager (EM) 创建副本（克隆）数据库。从 EM 主页，导航到 Data Movement > Clone database（数据移动 > 克隆数据库）。屏幕快照将显示“Clone Database: Source Type（克隆数据库: 源类型）”页面。

可以选择下列位置来执行克隆操作：

- **正在运行的实例：**可以指定克隆一个正在运行的实例。
- **登台区：**在源主机和目标主机上指定的磁盘区。在此处创建并存储备份，然后将其放入目标登台区，并对该目标主机进行读取以创建克隆数据库。
- **现有备份：**如果已有备份反映了要克隆的数据库所处的状态，则可以使用该位置。

## 测验

选择关于数据库复制的所有正确语句：

1. 无论是否连接到辅助实例，都可复制数据库。
2. 无论是否连接到恢复目录，都可复制数据库。
3. 无论是否连接到目标数据库，都可复制数据库。
4. 仅当具有 RMAN 备份时才可复制数据库。
5. 您必须始终在辅助实例上手动重新创建控制文件。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

答案：2、3

## 小结

在本课中，您应该已经学会：

- 列出创建副本数据库的目的
- 选择用于复制数据库的方法
- 使用 RMAN 复制数据库
- 使用 RMAN 备份复制数据库
- 基于正在运行的实例复制数据库

ORACLE

版权所有 © 2011, Oracle。保留所有权利。

## 练习 20 概览：复制数据库

本练习的内容包括了克隆数据库以及使用实用程序完成功能正常的复制数据库的设置。

ORACLE

版权所有 © 2011, Oracle。保留所有权利。