目　录

第I部分 SQL语言与PL/SQL语言

第 1 章　数据库的概念与Oracle 9i的安装

本章的学习目标：

- 了解实体-关系模型的基本概念和方法
- 了解当今流行的关系数据库以及 Oracle 数据库的优势
- 熟悉 Oracle 9i 数据库服务器的安装过程

1.1 实体-关系模型

模型是对过程和对象的抽象化，经过模型可以深入了解复杂系统的主要特征。

所谓实体，是指客观存在的事物，可通过它的若干属性的值来描述。属性是事物某方面的特征，所谓关系，是指实体集之间的联系。

关系有 3 种不同的类型：

- 1:1 型，即实体集合之间形成一对一的关系
- 1:m 型，即实体集合之间形成一对多的关系
- m:n 型，即实体集合的多对多的关系

参加一个联系的实体集可以只有一个，也可以是多个，实体间发生联系后，可能产生某些属性。

在实体、属性和关系三要素的基础上作 E-R 图的步骤是：

⑴用长方形框表示实体集，在框内写上实体名。

⑵用菱形框表示实体集之间的联系，在菱形框内写上联系的名称，用弧或线段连接菱形框与有关长方形框（实体），并注明 1:1、1:m 或 m:n 的函数关系。

⑶用椭圆框表示实体的属性，在椭圆框中标上属性名，用线段连接实体和它的属性。

1.2 关系数据库系统概述

所谓数据库就是为满足某部门各种用户的应用要求，在计算机系统中按照一定的数据模型组织、存储和使用的相互关联的数据集合。它具有“一少三性”：

- 一少，即冗余数据少。
- 三性，即数据的共享性、数据的独立性和数据的完整性。

数据的共享性：数据库中的数据能为多个用户服务。

数据的独立性：用户的应用程序与数据的物理存储方式无关，数据的逻辑组织与数据的物理存储方式无关。

数据的完整性：数据在修改更新过程中保持正确性。主要有实体完整性、参照完整性和用户定义的完整性。

（1）实体完整性：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。

（2）参照完整性：若属性（属性组）F 是基本关系 R 的外码，它与基本关系 S 的主码 Ks 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每个元组在 F

上的值必须为：或者取空值（F 的每个属性值均为空值）；或者等于 S 中某个元组的主码值。

（3）用户定义的完整性就是针对某一具体关系数据库的约束条件，它反映某一具体应用所涉及的数据必须满足的语义要求。

数据库是保存数据的地方，数据库可以存储 3 种不同的数据类型：

- 普通数据：包括数值、日期和字符串，例如姓名和出生日期等。
- 复杂和大型的对象：一个数据库可以存储和管理很多数据类型（例如声音），地理数据（例如地图、图片、图形和视频）。
- 新的用户数据：大多数数据系统还允许用户存储他们自定义的数据类型。

数据库必须为用户提供以下功能：

- 向数据库写入信息
- 修改信息
- 检索数据库中的信息

1.3 Oracle数据库历史与Oracle 9i

作为关系型数据库的先驱和基于标准 SQL 数据库语言的产品，Oracle 数据库自 1979 年推出以后，受到社会广泛的注意。二十多年来，Oracle 数据库融汇了先进的技术，并极有预见性地领导着全球数据库技术的发展。它从 1979 年的第 2 版开始，经历了可移植的第 3 版，可造的第 4 版，客户/服务器和协同服务器的第 5 版，高可靠联机事务处理（OLTP，OnLine Transaction Process）的第 6 版，具有分布式处理能力的第 7 版，功能强大的 Oracle 8 和世界上第一个 Internet 数据库 Oracle 8i 直到当今世界最为先进的 Oracle 9i 和 Oracle 10G。

这种数据库技高一筹之处在于其新软件群---- 一种叫做 Oracle 9i Real Application Clusters 的独立产品，编码为 Cache Fusion。Cache Fusion 可以至少将性能提高 2 倍，把应用软件的性能推进 10 倍！该软件群可以应用于 Unix、Windows、Linux 等多种操作系统中。

作为新一代 Internet 电子商务基础架构 Oracle 9i 的关键功能组件，Oracle 9i 协作内容管理是专门为提供完全的协作内容管理功能而设计的。它包括在 Oracle 9i 的两个核心产品--Oracle 9i 数据库和 Oracle 9i 应用服务器中，使用户与合作伙伴可以直接在 Oracle 9i 中利用内容管理功能，快速、简单地接触到以前难以访问到的内容。同时，通过直接在 Oracle 9i 中提供全面的协作内容管理功能，降低用户部署 IT 基础设施的成本。

Oracle 9i 协作内容管理是业界惟一完整、集成的基础架构，能够管理所有的内容，包括文档、电子表格、演讲稿和 PDF 文件等工作文件，以及 HTML 文件、多媒体、电子邮件、XML 文件等，能够帮助用户快捷地根据文件内容设立单一文件库。由于文件内容直接保存在 Oracle 9i 数据库中，因此用户在管理文档时能够直接利用 Oracle 9i 无限的可伸缩性、高可用性和安全性。另一方面，它强大的企业内部自动搜索功能，使用户能够不受地域和语言的限制，方便地搜索所需内容。Oracle 9i 协作内容管理支持 XML 数据类型和大多数文件网络协议，主要包括 Oracle Internet 文件管理系统、Oracle 文本、Oracle 超级搜索和 Oracle 多媒体 4 项功能。

Oracle Internet 文件管理系统(Oracle Internet File System)是 Oracle 9i 的重要功能，它通过在 Oracle 9i 数据库中提供 XML 支持，使用户能够容易地搜索和管理 Word 文档、电子表格、PPT 文件和其他业务数据；Oracle 文本(Oracle Text)功能是 Oracle 9i 数据库提供的功能，允许用户以 57 种语言、对 150 多种文件格式进行基于内容的查询，还可以查询 XML 文件以及根据内容主题对文档进行快速分类管理；Oracle 超级搜索(Oracle

Ultra Search)功能为用户提供了基于 Web 的用户界面，对存储在数据库、文件系统、邮件服务器和 Web 网站中的内容数据提供统一的检索目录；Oracle 多媒体(Oracle Multimedia)功能支持所有类型的多媒体数据，提供一套标准的管理工具，使用户能够对图片、音频、视频的安全，备份和恢复都能够进行控制管理。

Oracle 9i 也提高了联机分析处理和数据开发的功能。Oracle 还计划推出第二版的 9i 应用软件服务器，和 9i 数据库一起在网络上保存个人化信息。业内分析家认为，通过这款新式数据库，在综合数据库分析功能方面，Oracle 会把竞争对手甩得更远。

1.4 Oracle数据库的安装

我们将以 Windows NT 环境为例，介绍 Oracle 数据库的安装。UNIX 系统下的安装也大致类似，请读者自行参考有关书籍。

⑴首先插入 Oracle 安装盘，安装盘会自动启动。

⑵单击“开始安装”后，可以看到 Universal Installer，稍等一会，会出现欢迎界面。

- 单击“关于 Oracle Universal Installer”按钮，可以查看正在使用的 Oracle Universal Installer 版本信息；
- 要查看并卸装机器中所有 Oracle 主目录下安装的组件，请单击“卸装产品”按钮；
- 要查看机器中所有 Oracle 主目录下已安装的组件，请单击“已安装产品”按钮；
- 单击“帮助”可以查看 Oracle Universal Installer 帮助。

⑶单击“下一步”按钮，出现“文件定位”对话框

- “来源···”下的“路径”下拉列表框：输入将要安装产品的 products.jar 文件的完整路径. 也可以使用“浏览”按钮查找 products.jar 文件。
- “目标···”下的“名称”下拉列表框：输入 Oracle 主目录名或从其下拉列表中选择。

如果用户现在尚未在自己的机器上创建主目录，则在安装时会自动创建一个主目录。

主目录名可以是在“名称”下拉列表框中输入的任何名称。Oracle 主目录通过名称进行标识，Oracle 主目录的名称标识和特定 Oracle 主目录相关联的程序组以及 Oracle 服务安装在相关的主目录中。Oracle 主目录名的长度必须在 1 到 16 个字符之间。并且只能包含字母、数字字符和下划线。Oracle 主目录名中不允许有空格。

注意：

“名称”下拉列表框只在 Windows 平台上出现。

- “路径“：输入带有完整路径的 Oracle 主目录或从现有的 Oracle 主目录的下拉列表中选择。

Oracle Universal Installer 在 Windows 平台上维护的 Oracle 主目录列表如下：

- 已使用 Oracle Universal Installer 创建的所有 Oracle 主目录
- 使用早期的 Oracle Installer（基于 ORCA）创建的所有主目录
- Oracle_HOME 环境变量所指向的主目录

Oracle Universal Installer 在 UNIX 上维护的 Oracle 主目录列表如下：

- 已使用 Oracle Universal Installer 创建的所有 Oracle 主目录

- 在/var/opt/oratab 文件中定义的所有主目录
- Oracle_HOME 环境变量所指向的主目录

如果这些主目录都不存在，系统就会根据具有最大空闲的磁盘卷计算默认主目录。也可以使用“浏览”按钮选择一个目录来安装产品。此位置是要安装产品的目标目录。

⑷输入适当的信息后，单击“下一步”按钮，出现“可用产品”对话框

- Oracle 9i Database：为 Oracle 数据库服务器安装启动程序数据库及其他 Oracle 数据库软件。
- Oracle Client：只安装基本的客户机管理维护和用来支持开发商软件。
- Oracle 9i Management and Integration：安装管理工具，Oracle 集成服务器和客户机软件。

在 Oracle 9i 中，Oracle 公司把数据库软件、中间软件和应用软件的管理都由一个统一的软件来实现。

⑸每个安装类型中含有一系列的单独组件。做出适当选择后单击“下一步”按钮。

安装类型包含以下几种：

- 企业版 为高端应用程序提供数据管理，例如大容量的联机事务处理（OLTP）环境，查询密集型的数据仓库和要求较高的 Internet 应用程序。所提供的工具和功能可以满足以任务为第一的应用程序的可用性和可伸缩性需求。
- 标准版 其目标为工作组或部门级应用程序，包括一组综合性管理工具，完全的分发、复制、Web 功能，以及构建以业务为第一的应用程序的产品和服务。
- 自定义 允许用户有选择地安装以上安装类型中的组件。请注意，某些附加组件只在通过“自定义”安装类型进行安装时可用。
- 个人版 主要为开发者提供开发测试平台。在 9.0.1.1.1 版本中，安装大小和标准版基本相同。

如果选择客户端安装，安装类型包含以下几种：

- 管理员 安装 Oracle Enterprise Manager Console（包括企业管理工具）、网络服务、实用程序以及基本的客户软件。
- 运行环境 为数据库应用程序用户提供了连接 Oracle 9i 数据库并进行交互的网络连接服务和支持文件。
- 自定义 允许用户有选择地安装以上安装类型中的组件。请注意，某些组件只在通过“自定义”安装类型进行安装时可用。定制可以选择以下组件：
  - Oracle Management Server 安装 Oracle Management Server 、Oracle Enterprise Management Console（企业管理工具）、网络服务、实用程序以及基本的客户软件。
  - Oracle Internet Directory 安装适当组件，使 Oracle Internet Directory 目录服务器成为 Oracle 9i 数据库上的一个应用程序。
  - Oracle Integration Server 安装使用高级队列，Oracle JVM 以及 Oracle Workflow 配置的数据库。

⑹单击“下一步”按钮，可以看到“数据库配置”对话框。

- 通用 选择“通用”数据库配置类型适合于安装多用途的预配置数据库，从简单的事务处理到复杂的查询均可。“通用”配置类型既可在这两种情况下支持大量并发用户对数据的快速访问，即事务处理环境的典型引用，也可满足决策支持查询（DSS）的性能需求，对复杂的历史数据进行数据扫描。
- 事务处理 选择“事务处理”数据库配置类型适合于安装大量并发用户执行简单事务处理的预配置数据库。事务处理数据库的典型应用包括银行事务处理或 Internet 商务的数据库。对于需要较高的可用性和事务处理性能，大量用户并发访问相同数据以及

较高恢复性能的数据库环境，事务处理类型的配置可以提供最佳支持。

- 数据仓库 选择“数据仓库”数据库配置类型将安装适合于对主题进行复杂查询的预配置数据库。“数据仓库”的典型应用包括用于客户订单研究、支持呼叫、销售预测、购物和采购模式以及其他战略性业务问题的历史数据库。对于需要对大量数据进行快速访问，以及使用类似联机分析处理（OLAP）等应用程序的数据库环境，“数据仓库”配置类型可以提供最佳支持。
- 自定义 选择“自定义”数据库配置类型，可以选择需要安装和配置的组件。如果用户对 Oracle 安装非常有经验并准备提供复杂的系统和产品配置信息，或者需要安装 Oracle 9i 的特定组件，例如 Transparent Gateway，“自定义”配置比“通用”、“事务处理”或“数据仓库”数据库配置需要更多的时间和用户交互。
- 只安装软件 如果需要安装 Oracle 9i 软件，但又不需要安装子数据库，请选择“只安装软件”配置类型。如果选择此配置类型，完成安装后不会启动 Oracle Configuration Assistant。

可以根据数据库的不同用途，选择“事务处理”或“数据仓库”等。事务处理主要是用来进行联机事务处理，数据库尺寸一般比较小，但性能要求很高，支持瞬时响应。“数据仓库”是用来进行大量数据处理，比如数据挖掘等，一般数据库尺寸非常大，响应时间较长。

⑺在此我们选择“通用”，单击“下一步”按钮，弹出“数据库标识”对话框。在其中输入合适的全局数据库名及 SID。SID 为本机上区别于其他数据库例程的标志。全局数据库名 SID 可以相同也可以不同，一般设为相同。完成后，继续单击“下一步”按钮，弹出“数据库文件位置”对话框。

⑻在 “数据库文件目录”里输入数据库文件的目录，此目录为用户数据库文件所要创建的位置。数据库文件主要包括数据库文件、控制文件和重做日志文件，其相关信息将在后面的章节中介绍。

⑼接下来是字符设置。有关字符设置的信息请参阅“帮助”，选择“使用默认字符集”单选按钮，单击“下一步”按钮。

⑽现在我们可以看到“摘要”对话框。

“摘要”对话框将显示所选选项的概要信息。这些信息与安装无关，可能包括下列内容：

- 全局设置

来源：是为安装而集中存放组件的位置。要更改“来源”，必须回到“文件定位”对话框。

目标：安装产品的 Oracle 主目录位置。要更改“目标”，必须回到“文件定位”对话框。

安装类型：一个预定义的组件集合，它自动选择了要安装的组件和相关的组件组。要更改“安装类型”，必须回到“安装类型”对话框。

- 产品语言：选择用于运行 Oracle 产品的语言。要更改“产品语言”设置，必须回到“可用产品组件”对话框。
- 空间需求：安装产品所需要的磁盘空间。如果可用磁盘空间少于所需要的空间，“空间需求”将以其他颜色显示。
- 新安装组件：第一次要在机器上安装的产品。
- 升级组件：要升级到较高版本的产品。
- 重新安装：在开始安装之前，Oracle Universal Installer 将首先卸装产品的旧版本。
- 降级组件：要降级到较低版本的产品。

- 卸装组件：将作为该安装的一部分而被删除的产品，因为没有其他产品与其相关。
- 已安装组件：已安装的产品。Oracle Universal Installer 将不安装这些产品。

⑾检查所做的选择后，单击“安装”按钮，将进入安装阶段。

⑿系统开始运行安装程序。根据安装环境的不同，安装时间一般不超过 3 个小时。安装程序完成此过程以后，将自动启动 NET CONFIG ASSISTANT、HTTP、DATABASE CONFIG ASSISTANT 等程序，系统将自动完成相关参数的配置。最后出现“安装结束”对话框。

⒀到此 Oracle 9i 数据库软件安装结束。可以单击“退出”结束安装，打开 SQL*PLUS 工具来确认数据库安装成功。选择 Windows 的“开始”->“程序”->Oracle-OraHome90->Application Development->SQLPlus Worksheet 命令打开“登录”对话框。

⒁在“用户名”文本框中输入 system，“口令”文本框中输入 manager，“口令”文本框中输入 mis，单击“确定”按钮。

⒂然后可以看到连接成功的界面。至此，整个安装程序完成。

1.5 本章小结

本章主要介绍了 E-R 模型、数据库的简单原理、数据库的历史、Oracle 的历史以及在当今商务部门中的应用等，我们介绍了 Oracle 的最新版本 Oracle 9i 的一些新特征及 Oracle 9i 在 Windows NT 操作系统上的简单安装。

第 2 章 SQL语言基础

本章的学习目标：

- 掌握 SQL 语言的基本语法
- 熟悉 Oracle 的数据类型
- 熟悉 SQL*PLUS 数据库工具的使用及环境变量设置

2.1 Oracle的数据类型

数据类型是数据的基本属性，它反映了数据所属的种类，如日期类型数据记录的是事件发生的时间，字符串类型表示存储的数据为字符类型等。Oracle 的数据类型主要有：

1. 字符数据类型
   - CHAR（n）
   - VARCHAR2（n）
   - LONG
2. 数字数据类型
   - NUMBER（m，n）
3. 日期数据类型
4. 大对象数据类型

   Oracle 的大对象类型数据主要有 3 种，它们分别是 BLOB、CLOB 和 NCLOB。

   Bfile 数据类型
5. 其他数据类型
   - RAW（n）
   - LONG ROW

2.2 SQL基本语法

2.2.1 数字运算符和比较运算符

数字运算符及其说明如表 2-1 所示。

表 2-1 数字运算符及其说明

数字运算符	说明
+ -	表示数字的符号，如正数+10，负数-35.4 等
+	可以用来实现两个数字或表达式相加 如 1+1
-	可以用来实现两个数字或表达式差异计算 如 12-5
*	可以用来实现两个数字或表达式相乘 如 3*4
/	可以用来实现两个数字或表达式相除 如 9/3

比较运算符及其说明如表 2-2 所示。

表 2-2 比较运算符及其说明

比较运算符	说明
=	等于号
!=	不等于
<>	不等于

<	小于
>	大于
<=	小于等于
>=	大于等于

1. ANY/SOME

比较其中的任意值，他的前面的比较符号必须是=、!=、<、>、<=、>=。

2. ALL

比较所列出来的每一个值。它的前面的比较符号必须是=、!=、<、>、<=、>=。

3. [NOT]BETWWEN m AND n

如果值大于或等于 m，小于或等于 n，则返回 TRUE。

4. [NOT]EXISTS

如果子查询里至少返回一行，则 EXISTS 为 TRUE。

5. m[NOT]LIKE n

用来返回列内容根传送给函数的常量相似的所有行

6. IS[NOT] NULL

用来测试所要操作的值是否为空。

2.2.2 逻辑运算符

- NOT

用来对结果取反

- AND

用来限制条件，当两个条件同时成立时返回 TRUE，如果有一个不成立，返回 FALSE。

- OR

当两个条件有一个成立时返回 TRUE，如果两个条件都不成立，返回 FALSE。

2.2.3 简单的查询

- SELECT 语句

SELECT 语句用来从数据库的一个或多个表中选区记录，这是最经常使用的语句。它必须和 FROM 联用。FROM 用来指定记录源，表、视图等。

- WHERE 语句

WHERE 语句是用来限制结果输出的。用 WHERE 语句来限制结果及内容的方法有两种：一是用 WHERE 语句来限制单个结果及的内容;一是用 WHERE 语句来将两个或多个表联接到一个结果集中。

2.3 SQL*PLUS 工具和SQL*Plus工作单

SQL*PLUS 是 Oracle 最为主要的管理界面之一，数据库管理员可以通过它直接对数据库进行操作。

下面即是一些 SQL*PLUS 窗口环境设置命令：

- @
- 符号“/”
- APPEND
- CHANGE
- CLEAR
- CONNECT

- DEFINE
- DEL
- DESCRIBE
- EDIT
- EXECUTE
- EXIT
- GET
- INPUT
- LIST
- PROMPT
- RUN
- SET
- SHOW
- SPOOL

下面是 SQL*PLUS 里面的 SET 设置使用的命令：

- ARRAY
- AUTO
- AUTOTRACE
- STATISTICS
- DEF/DEFINE
- ECHO
- EDITF
- ESC
- OFF
- LONG
- PAGES/PAGESIZE
- PAU/PAUSE
- SERVEROUT
- TERM
- TIME
- TRIMS
- VER

在 Oracle 9i 中还提供了基本和 SQL*PLUS 功能相同的图形化操作界面，即 SQL*Plus 工作单。

2.4 函数

在对数据的操作过程中，经常需要改变数据输出形式或进行数据运算输出等，这时可以考虑使用函数。SQL 语言中的函数包括单行 SQL 字符函数、转换格式函数、多行函数以及 DECODE 函数等。

2.4.1 单行SQL字符函数

- ABS 函数

  格式：ABS(n)

  此函数返回 n 的绝对值。

- ACOS 函数

  格式：ACOS(n)

  此函数返回 n 的反余弦值。

- ASIN 函数

  格式：ASIN (n)

  此函数返回 n 的反正弦值。

- ATAN 函数

  格式：ATAN (n)

  此函数返回 n 的反正值。

- CELL 函数

  格式：CELL (n)

  此函数返回大于或等于 n 的最小整数值。

- COS 函数

  格式：COS (n)

  此函数返回 n 的余弦值。

- COSH 函数

  格式：COSH (n)

  此函数返回 n 的双曲线余弦函数。

- EXP 函数

  格式：EXP (n)

  此函数返回 e 的 n 次幂。

- FLOOR 函数

  格式：FLOOR (n)

  此函数返回小于或等于 n 的最大整数值。

- LN 函数

  格式：LN (n)

  此函数返回 n 的自然对数，即以 E 为底的对数。

- LOG 函数

  格式：LOG (m,n)

  此函数返回以 m 为底的 n 的对数。

- MOD 函数

  格式：MOD (m,n)

  此函数返回以 m 除以 n 的余数。

- POWER 函数

  格式：POWER (m,n)

  此函数返回 m 的 n 次幂。

- ROUND 函数

  格式：ROUND (m,n)

此函数返回舍入到小数点左边或右边 m 位的 n 的值。

- **SIGH 函数**

  格式：SIGH (n)

  此函数返回 n 的符号值。

- **SIN 函数**

  格式：SIN (n)

  此函数返回 n 的正弦值。

- **SINH 函数**

  格式：SINH (n)

  此函数返回 n 的双曲线正弦值。

- **SQRT 函数**

  格式：SQRT (n)

  此函数返回 n 的平方根。

- **TAN 函数**

  格式：TAN (n)

  此函数返回 n 的正切值。

- **TANH 函数**

  格式：TANH (n)

  此函数返回 n 的双曲正切值。

- **TRUNC 函数**

  格式：TRUNC (n)

  此函数功能类似 ROUND。

2.4.2 转换格式函数

格式函数转换规则如表 2-3 所示。

函数	功能
CHARTOROWID	将包含外部语法 ROWID 的 CHAR 或 VARCHAR2 数值转换为内部的二进制语法
CONVERT	将字符串 CHAR 中的字符，从 SOUCE_CHAR_SET 标识的字符集转换为由 dest_char_set 标识的字符集
HEXTORAW	将包含十六制的 CHAR 转换为一个 RAW 数值
RAWTOHEX	将 RAW 数值转换为一个包含十六进制的 CHAR 值
ROWIDTOCHAR	将一个 ROWID 数值转换为 VARCHAR2 数值类型
TO_CHAR(日期转换)	将日期数据类型转换为一个在日期语法中指定语法的 VARCHAR2 数据类型的字符串
TO_CHAR(数据转换)	将 NUMBER 数据类型 N 转换为一个 VARCHAR2 数据类型
TO_DATE	该函数将 CHAR 或 VARCHAR2 数据类型的值转换为 DATE 数据类型
TO_MULTI_BYTE	该函数用来将 CHAR 中的所有单字节字符转换为等价的多字节字符
TO_NUMBER	该函数将 CHAR 或 VARCHAR2 数据类型的字符串（CHAR）按照指定的数值语法（FMT）转换为 NUMBER 数据类型值
TO_SINGLE_BYTE	将 CHAR 中所有的多字节的字符转换为它们等价的单字节字符
TRANSLATE USING	将文本 TEXT 按照指定的转换方式转换为数据库字符集和民族字符集

- **DECODE 函数**

语法：DECODE（<a>,<m1>,<n1>[,<m2>,<n2>…][,<b>]）

上式中 a 为一个表达式，如果 m1 和 a 相等，则返回 n1；接着是 m2 和 n2，这样一直继续下去，返回 n2 n3…如果条件都不满足，则返回 b。

2.4.3 多行函数

- **AVG 函数**

用来返回某一列中所有数值的平均值。

- **COUNT 函数**

返回列数并经常被用于获取某一视图中的总行数。COUNT 是惟一一个能用于非数值列的标准 SQL 集合函数。

- **MAX 函数**

返回某一列中的最大数值。

- **MIN 函数**

返回某一列中的最小数值。

- **STDDEV 函数**

返回某一列中的标准偏差。

- **SUM 函数**

返回某一列中的所有数值的总和。

- **VARIANCE 函数**

返回某一列中的所有数值的方差。

- **GROUP BY 函数**

GROUP BY 用来把数据分组。我们可以使用它把数据库返回的数据按要求分类。在 GROUP BY 函数当中，GROUP BY 所指定的列必须在 SELECT 所指定的列中存在。

- **HAVING 函数**

由于 GROUP 函数与 WHERE 函数不能同时使用。

一个 SQL 语句可以同时存在 WHERE 函数和 HAVING 函数。WHERE 函数在分组以前过滤数据，HAVING 函数在分组后过滤数据。

- **ORDER BY 函数**

在一般使用 SELECT 语句时，结果集中的记录是按照它们在基础表中被找到的顺序返回的，如果想把输出结果按一定的要求进行排序，可以使用 ORDER BY 函数。

此函数并不象 GROUP BY 那样所指定的列必须在 SELECT 语句当中出现。可以通过在 ORDER BY 函数的每个字段后面放置 ASC（升序）和 DESC（降序）来指定想以什么样的顺序排列数据。

2.5 本章小结

本章主要介绍了 Oracle 的数据类型、SQL 语言的语法基础及 SQL*PLUS 工具的环境变量设置等。Oracle 的数据类型主要包括字符数据类型，数字数据类型，日期数据类型和大对象数据类型等。SQL 语言共分为数据操作语言（DML—Data Manipulation Language）、数据定义语言（DDL—Data Define Language）、事务控制语句、会话控制语句和系统控制语句，对一些常用的函数读者应该熟练掌握。SQL*PLUS 工具的环境变量决定了 SQL*PLUS 工具的输出格式，可以根据具体使用环境不同来设置不同的环境。

第 3 章 数据操作和数据库对象

本章学习目标：

- 掌握多表查询的概念和方法
- 熟悉数据库数据的插入、修改、删除等操作命令，并能执行简单操作
- 掌握数据库对象的定义、用途以及实现方法

3.1 多表查询

单表查询比较简单，但是实际应用最多的还是多表查询，多表查询即是把多个表按照一定的关系连接起来，在用户看来好像是查询一个表一样。

3.1.1 简单的两表查询

在关系数据库管理系统中，数据存储在不同的表里，这些表都是相关的，用户可以使用 SQL 语句对这些相关信息进行查询。SELECT 语句必须和 FROM 语句联用。

在多表查询当中必须指定各列所属的表，两个表的关系在 WHERE 子句中列出。

3.1.2 三表查询和多表查询

要执行 3 个或更多表格联合查询，必须遵循以下步骤：

⑴先把两个表格按一定的条件组合到一起。

⑵把组合的结果同第 3 个表格组合到一起。

⑶重复第 2 步，直到把所有表格组合到一起。

3.1.3 一些连接操作设置符号

通过使用四个集合操作符 UNION、UNION ALL、INTERSECT 和 MINUS，Oracle 提供了将两个或者多个 SQL 查询结合进一个单独的语句的能力。

使用集合操作符的查询称为复合查询（compound query），Oracle 为复合查询的编写制定了需要遵循的指南：

- 在构成复合查询的各个单独的查询中，SELECT 表中值的数量和数据类型相匹配
- 用户不许在复合查询所包含的任何单独的查询中规定 ORDER BY 子句
- 用户不许在 BLOB、LONG 等大数据对象上使用集合操作符
- 用户不许在集合操作符 SELECT 列表中使用嵌套表或者数组集合。

1.UNION

UNION 语句将第一个查询中的所有行与第二个查询的所有行相加，消除重复行并且返回结果。

2.UNION ALL

UNION ALL 语句与标准的 UNION 语句工作方式基本相同，唯一不同的是 UNION ALL

不会从列表中滤除重复行。

3.INTERSECT

INTERSECT 操作会获取两个查询，对值进行汇总，并且返回同时存在于两个结果集中的记录。只由第一个查询、或者第二个查询返回的那些行不会包含在结果集中。

4.MINUS

MINUS 集合操作会返回所有从第一个查询中返回，但是没有从第二个查询中返回的那些记录。

3.2 数据修改

在数据库应用中，经常要对存放的数据进行更新操作，以满足不断变化的需求。因此对数据进行插入、修改、删除就成为必不可少的一项工作。

3.2.1 数据的插入

在插入数据时，应首先确认基表已经创建，然后确定基表的结构，基表的各列顺序、类型以及是否是非空（NOT NULL），可以通过 DESC 命令来查看，以保证插入数据的类型与基表列的类型匹配。若插入字符型和日期型数据，要用单引号括起来。具体语法如下：

insert into 数据表(字段名 1,字段名 2,……) values(字段名 1 的值, 字段名 2 的值,……)。

由于字段的类型不同，在书写字段值的时候要注意格式。

数值型字段，可以直接写值。

字符型字段，其值上要加上单引号。

日期型字段，其值上要加上单引号，同时还要注意年、月、日的排列次序。

在数据的插入语句中，插入列排序和插入值要一一对应。字符型和日期型字段要加上单引号，非空列必须有值。

3.2.2 数据的修改

在 Oracle 中，对数据的修改是通过使用 UPDATE 命令来实现的。具体语法如下：

update 数据表

set 字段名 1=新的赋值,字段名 2=新的赋值,……

where 条件

3.2.3 数据的删除

对表中数据进行删除是使用 DELETE 命令来实现的。

3.3 事务控制命令

1. 事务的概念

事务是指一个或多个 SQL 语句所组成的序列，是对数据库操作的逻辑单位。连续地执行提交操作（COMMIT）或回退操作（ROLLBACK）之间的操作称为一个事务。Oracle 的 RDBMS 要确保数据库的一致性是基于事务而不是基于单条 SQL 语句的。

对事务的控制包含事务提交、事务回滚（或撤消）和设立检查点。

2. 事务的提交

数据库数据的更新提交以后，这些更新操作就不能再撤消。事务的提交有 3 种方式：

- 显式提交

在 SQLPLUS 中，使用 COMMIT 命令来提交所有未提交的更新操作就是显式提交。

- 隐式提交

有些命令，如 ALTER、AUDIT、COMMENT、CONNECT、CREATE、DISCONNECT、DROP、EXIT、GRANT、NOAUDIT、REVOKE、RENAME 命令，以及退出 SQL*PLUS 都隐含 COMMIT 操作，而无须指明该操作。只要使用这些命令，系统就会提交以前的更新操作，就像使用了 COMMIT 命令一样。

- 自动提交

用户可以使用 SET 命令来设置自动提交环境。经过设置后，SQL*PLUS 会自动提交用户的更新工作。也就是说，一旦设置了自动提交，用户每次执行 INSERT、UPDATE 或 DELETE 命令，系统就会立即自动进行提交。

3．事务回滚

尚未提交的 INSERT、UPDATE 或 DELETE 更新操作可以使用 ROLLBACK 命令进行撤消。

3.4 表的创建与修改

数据以表的形式存储在数据库中。方案或用户是数据库表的所有者，表的每一列定义为一个数据类型，存储数据必须满足该数据所在列的属性。

3.4.1 生成一个简单表

表的生成可以使用 CREAT TABLE 命令来实现。

一个新表可以从已有的表格或视图中来创建，新表的列必须已经在原图或视图中存在，列属性和数据类型不能改变，但列名可以改。另外，用户也可以指定新表的存储参数。

ALTER TABLE 用来修改表定义。

1．增加列

ALTER TABLE 有多种形式，对表增加新列的语法如下：

```
ALTER TABLE < table_name >
  ADD (<column define>[,< column define>]...);
```

<table name>指出要增加列的表，ADD 后的<column define>指出新增加列的名字及其数据类型，其格式同于 CREATE TABLE 语句中的定义，只有表中无数据时，新增加的列才可选 NOT NULL；当不作选择时，隐含为 NULL。新增加的列在表尾部，该列初值均为 NULL。另当只增加一列时，ADD 后的“()”可省略。

2．修改列

修改列的语法为：

```
ALTER TABLE <table_name>
  MODIFY (<column_name>[<data_type>][NULL | NOT NULL]
```

[,<column name>[<data type>][NULL | NOT NULL]]...);

<table_name>：指出要对列作修改的表

<column_name>：指出要作修改的列名

[data_type]：可取 CHAR(n), NUMBER, NUMBER(m, n), NUMBER(n), DATE 等中的一种, 如不作选择, 表示该列数据类型不作修改

[NULL | NOT NULL]：一个列要改为 NOT NULL，要求该列当前不含空值；如不选此项, 则缺省而保持原有值。

修改列必须遵循以下规则：

- 可以增加列的宽度或数字的精度。
- 减少列宽度时，列值必须为空。
- 当数据类型被修改时，列值必须为空。

3. 删除列

删除列的语法为：

ALTER TABLE <table_name> DROP COLUMN <column_name>

如果此列存在索引或约束，必须使用附加功能 CASCADE CONSTRAINTS。

这个操作会将表重新写入磁盘，并且移走被删除列的数据。对于更大的表，用户可以使用另一个操作来避免对表进行整体重写：

ALTER TABLE <table_name> SET UNUSED COLUMN (<column_name> [,<column_name>] [,<column_name>]……)

此命令不会对表进行重写，也不会回收空间。在语句执行之后，列只是会被简单忽略。如果需要将这些列删除，回收存储空间，可使用以下语句：

ALTER TABLE <table_name> DROP UNUSED COLUMNS;

3.4.2 表的重命名与删除

1. 删除表

当表或视图不需要时，可以用 DROP 语句删除，其格式为：

DROP TABLE <table_name> [CASCADE CONSTRAINTS];

当表被删除时，系统自动地删除表中的数据和在此表上建立的各种索引, 也删除了在该表上授予的操作权限和触发器。删除表并未删除在该表上定义的视图和别名, 但这些视图和别名已被标志为非法并不能应用, 用户应使用其它方法删除它们。当此表格存在约束时，必须使用附加功能 CASCADE CONSTRAINTS。

2. 重命名表

使用 RENAME 命令可以对表进行重命名。此命令对视图和私有同义词也有效。当对一个表格进行重命名后，Oracle 将自动更新相应的约束、索引和与此表相关的权限，而且 Oracle 将标志那些以此表为参考的视图、同义词、存储过程和函数为非法。

用 RENAME 来重命名的语法为：

RENAME < old_name > TO < new_name >;

3.5 视图

视图（View）是 SQL 语言提供的一种特殊类型的表，它可以由一个基表中选取的某些行

和列组成，也可以由几个表中满足一定条件的数据组成。视图就像是基表的窗口，它反映了一个或多个基表的局部数据。但视图本身并不实际存放数据，数据来自于基表，视图仅是对应基表的部分数据，因此视图只是一个虚表。

生成一个视图可使用 CREATE VIEW 命令。其格式为：

CREATE [OR REPLACE][FORCE] VIEW <view_name> [(<List of view column name>)]
AS <SELECT Statement> [WITH CHECK OPTION];

<view_name>：定义新的视图名

<SELECT Statement>：查询语句, 它从有关表中为视图选取满足一定条件的数据

[(<List of view column name>)]：是可选项, 当不选时, 新视图的列名与 SELECT 所选数据的名称相同; 当选择时, 则为 SELECT 所选数据重新取了个新的列名, 它们按顺序对应。

WITH CHECK OPTION：也是可选项, 当选择时, 向视图插入数据时, 用户必需保证数据要满足所选的检查条件。

[OR REPLACE]：通过使用此选项，可以重新定义视图，而且不损失以前授予的任何访问特权（如果删除后重新创建视图，所有相关联的授权也将被删除）。

[FORCE]：用户也可以使用 FORCE 来生成带有错误的视图，通常状况下如果视图存在错误，它将无法生成。但是，如果需要生成一些带错误的视图，如所参考的表格不存在，可以使用这种方法。这种方法生成的视图为非法，以后用户可以修改此错误，即生成相应的表格，随后此视图被重新编译。

注意：如果用户使用*来选择所有的列生成视图，当修改表格或增加列时，必须重新生成此视图。

删除视图使用命令 DROP VIEW。此视图的定义从数据字典中被删除，相应的权限也被删除，其他一些视图和存储程序需要参考此视图的，也被标志为非法。

3.6 其他数据库对象和数据字典

数据库对象除了表格和索引以外还有很多，如约束、触发器和同义词等。它们也是数据库所必需的对象。数据字典也是一系列的视图，它通过不同的视图呈现给不同的用户。用来对数据库的描述、管理及维护。

3.6.1 索引（Index）

1. 平衡树索引（B-tree index）

平衡树，又叫 B 树，是现代关系型数据库中最为普通的索引。B 树索引用由底向上的顺序来对表中的列数据进行排序。B 树索引不但存储了相应列的数据，还存储了 ROWID，此 ROWID 用来标志表格中相应行的剩余数据。索引以树形结构的形式来存储这些值，在检索时，Oracle 将先检索列数据。

2. 位图索引

位图索引（Bitmap Index）并不重复存取索引列的值，每一个值被看作一个键，相应的行的 ID 置为一个位（BIT）。位图索引适合于仅有几个固定值的列，如职员表中的性别列，性别只有男和女两个固定类型。不能用位图索引来生成具有惟一性的索引（UNIQUE KEY）和反键索引（REVERSE KEY）。

3.6.2 约束

Oracle 利用完整性约束机制以防止无效的数据进入数据库的基表，如果任何 DML 执行结果破坏完整性约束，该语句被回滚并返回一个错误。利用完整性约束实施数据完整性规则有下列优点：

- 定义或更改表时，不需要程序设计便很容易地编写程序并可消除程序错误，其功能由 Oracle 控制。所以说明完整性约束优于应用代码和数据库触发器。
- 对表所定义的完整性约束存储在数据字典中，所以由任何应用进入的数据都必须遵守与表相关的完整性约束。
- 具有最大的开发能力。当由完整性约束所实施的事务规则改变时，管理员只需改变完整性约束的定义，所有应用自动地遵守所修改的约束。
- 由于完整性约束存储在数据字典中，数据库应用可利用这些信息，在 SQL 语句执行之前或由 Oracle 检查之前就可立即反馈信息。
- 完整性约束说明的语义是清楚地定义，故对每一说明规则可实现性能优化。
- 由于完整性约束可临时地被屏蔽（DISABLE），以致在装入大量数据时可避免约束检索的开销。当数据库装入完成时，完整性约束可容易地使其重新启动，任何破坏完整性约束的新数据在例外表中列出。

Oracle 的 DBA 和应用程序开发者对列值输入可使用的完整性约束有下列类型。

- NOT NULL 约束
- UNIQUE 约束
- PRIMARY KEY 约束
- FOREIGN KEY 约束
- CHECK 约束

3.6.3 同义词

同义词（Synonyms）是指向其他数据库表的数据库指针。当创建一个同义词时就指定了一个同义词名字和同义词所引用的对象。当引用同义词名字时，Oracle 服务器会自动地用同义词定义的对象名字来代替同义词的名字。

同义词有两种类型：私有（PRIVATE）和公有（PUBLIC）。私有同义词是在指定的模式中创建的，并且只允许拥有它的模式访问。公共同义词由 PUBLIC 模式所拥有，所有的数据库模式都可以引用它们。

3.6.4 过程、函数和包

包、过程和函数与它们的源代码一起存在数据字典中。一个存储过程是一个操作的代码单元，可以被传递参数，并能够返回值。一个存储函数是一个代码单元，可以被传递参数，并能够返回一个值。一个包是一个过程、变量和函数的集合，包按照功能被逻辑地分组。

可以通过 DBA_OBJECTS 和 DBA_SOURCE 视图查看包、过程和函数的有关信息。

3.6.5 触发器

触发器是存储过程，当针对一个表发生特定的动作时，就会激活它。触发器可以被编码。当针对一个表进行插入、更新、删除或 3 种操作的结合时激活触发器，也可以在某行被影响或某条语句出现时被激活。触发器经常用于加强数据完整性约束和业务规则，这些业务规则对于内置的 Oracle 引用完整性约束来说实在是太复杂了。关于数据触发器的数据可以在 DBA_TRIGGERS 视图中找到。

3.6.6 数据字典

数据字典(Data Dictionary)通过不同的视图呈现给不同的用户，这些视图拥有带有不同前缀的相同名称。这些数据可以被划分为以下几类：

- 对象的存储信息，例如表空间、段和区间
- 表、列、簇和索引
- 视图、同义词和序列
- 过程、包、触发器和它们的源代码
- 用户、角色、环境资源文件和权限
- 性能信息
- 锁和审计信息

对于一个授权的数据库系统管理员用户，最适合使用的视图是加了前缀 DBA_的视图，如 DBA_TABLES、DBA_INDEXES、DBA_SEGMENTS 等。这些视图提供了所有模式中的每个数据库对象的有关信息，通常由模式的拥有者管理。对于其他用户，可以使用 ALL_视图和 USE_视图。另外，性能视图是公共可用的，是以 V$开头的，如 V$SGASTAT、V$SESSION、V$SYSSTAT、V$STATNAME 等。

3.7 本章小结

本章主要讲述了多表查询、数据库数据操作和数据库对象，其中数据操作主要包括插入、修改、删除等，数据库对象主要有视图、约束、索引、同义词、数据字典、触发器、包、过程和函数等。

第 4 章 PL/SQL语言

本章的学习目标：

- 了解 PL/SQL 语言的用途与结构
- 掌握 PL/SQL 语言的基本语法
- 熟悉 PL/SQL 在数据库中的应用

4.1 PL/SQL简介

PL/SQL 即模块式的过程化 SQL，它是 Oracle 公司对 SQL 语言功能的拓展，因此它具有许多 SQL 所没有的优点：

- 模块化结构
- 定义标识符
- 用过程化语言控制结构进行程序设计
- 错误处理
- 提高操作性能

如果不使用 PL/SQL，Oracle 一次只能处理一条 SQL 语句。每条 SQL 语句都导致客户（Client）向服务器（Server）调用，从而在性能上产生很大的开销，尤其是在网络操作中。如果使用 PL/SQL，一个块中的语言作为一个组，导致客户向服务器的一次调用，减少网络传输。

4.2 PL/SQL块结构与用途

一个基本的 PL/SQL 块由三部分组成：定义部分、可执行部分以及例外处理部分：

- 定义部分

定义将在可执行部分中调用的所有变量、常量、游标和用户自定义的例外处理。这部分可以没有。

- 可执行部分

包括对数据库中进行操作的 SQL 语句，以及对块中语句进行组织、控制的 PL/SQL 语句。这部分在 PL/SQL 块中必须存在。

- 例外处理部分

对可执行部分中的语句，在执行过程中出错或出现非正常现象时所做的相应处理。在 PL/SQL 块中可以没有该部分。

- 无名块

也就是没有命名的 PL/SQL 块，它可以是嵌入某一个应用之中的一个 PL/SQL 块。无名块在所有 PL/SQL 环境中都适用。

PL/SQL 主要用于以下几个方面：

- 存储过程/函数（Procedure/Function）

也就是指命名了的 PL/SQL 块，它可以接收参数，并且可以重复地被调用。

- 包（Package）

命名了的 PL/SQL 模块，由一组相关的过程、函数和标识符组成。

- 数据库触发器（Triggers）

数据库触发器是与一个具体数据库表相关联的存储 PL/SQL 程序。每当一个 SQL 操作影响到该数据库表时，系统就自动执行相应的数据库触发器。每个表最多可以有 12 个触发器。

数据库触发器由三部分构成：触发事件、用户预先设定好的触发器约束条件以及触发器动作。当该事件发生时，相应的触发器被激发并由某个无名 PL/SQL 块执行相应的动作。

4.3 常量与变量

4.3.1 变量声名

在 PL/SQL 块中引用的所有标识符都必须在 PL/SQL 块的定义部分声明。变量的声明语法如下：

variable_name type [CONSTANT][NOT NULL] [:=value]|[DEFAULT value];

- variable_name：变量的名字。变量名都必须以字母开头，不能有空格，不能超过 30 个字符长度，同时不能和保留字同名，常（变）量名称不区分大小写，在字母后面可以带数字或特殊字符。
- type：变量的数据类型
- [CONSTANT]：声明该变量是常量变量，若使用此项，则变量必须在声明时进行初始化
- [NOT NULL]：声明该变量是非空的，若使用此项，则变量必须在声明时进行初始化
- [:=value]|[DEFAULT value]：声明变量的初始值。如果不使用此项，变量的值默认为 NULL

在声明部分，每一行只能声明一个变量。下面的声明部分是错误的：

4.3.2 常量

常量和变量类似，但是它的值在定义时已经被指定，不能被改变。它的声明方式和变量一样，但必须指定关键字 CONSTANT。

PL/SQL 标量数据类型可分为 4 类：

- 数值类型
- 字符类型
- 日期类型
- 布尔类型
- 大数据类型

4.3.3 单字符分界符和双符分界符

1. 单字符分界符
2. 双字符分界符

说明：

单行注释可从一行中任何位置以“--”开始，直到行尾。多行注释以“/*”开始，以“*/”结束，注释可跨越多行且注释不能嵌套。

4.3.3 标识符

标识符是用户自己定义的符号串，用来命名常量、变量、例外、游标、子程序和包等。

标识符必须以字母打头，由字母、数字、$、下划线后#构成的字符串，长度不能超过30个字符，不能是PL/SQL的保留字，并且在其作用域内必须惟一。PL/SQL不区分标识符的大小写，如EMP和emp在PL/SQL中是相同的。

4.4 执行一个PL/SQL块

你可以对一个PL/SQL块进行命名，然后存储到数据库里面，当执行此块时，可以使用关键字EXECUTE。

如果在一个PL/SQL块中调用另一个命名的PL/SQL块时，直接引用其名称即可，不需使用关键字EXECUTE

4.5 条件语句 IF…THEN

其语法如下：

```
IF boolean_expression1 THEN
   Statements1;
    [ELSIF boolean_expression2 THEN
        Statements2;]
        …
 [ELSE
    Statementsn;]
 END IF;
```

4.6 循环

循环控制即重复多次地执行一组语句。循环的种类有下面3种：

- 简单循环：无限制地重复执行一组语句。
- FOR循环：有限制地重复执行一组语句。
- WHILE循环：重复执行语句，直到控制条件不再为真。

1. 简单循环

在简单循环中没有循环条件限制，循环体内的语句被无限次地重复执行。语法如下：

```
LOOP
    statements;
END LOOP;
```

在此循环中，可以将EXIT语句添加一个终止条件，语法如下：

```
EXIT [WHEN condition]
```

2. FOR循环

FOR 循环是对循环体内的语句重复执行指定的次数，其循环次数由计数器来指定。 采用

for..in..loop..end 循环控制的语法如下。

for 循环变量 in [reverse] 循环下界..循环上界 loop

循环处理语句段;

end loop;

3．WHILE 循环

WHILE 循环是当控制条件为真时，重复执行循环体内的语句：当控制条件不再为真时，终止循环体内语句的执行。采用 loop..exit..when..end loop 循环控制的语法如下。

while 条件 loop

执行语句段;

end loop;

4。GOTO 语句

跳转控制是指不按正常的语句执行顺序，而是执行指定的语句。用 GOTO 语句来实现。其语法如下：

GOTO label;

Label 是在 PL/SQL 中定义的标号。标号用双箭头括号括起来。当执行 GOTO 语句时会立即转到有标号标识的语句。

需要注意的是，GOTO 语句中的关键字 GOTO 是一个整体，不能把它们拆开来。而且，在进行 PL/SQL 编程时，尽量避免或少用 GOTO 语句，因为这种无条件的跳转语句打破了程序的逻辑性，有悖于自上而下的编程风格。

跳转语句的使用，可以是对于同一个块中语句之间的跳转,也可以是从子块中跳到父块。但需注意以下不能实现正常跳转的情况：

- 不能从父块跳转到子块中
- 不能从 IF 语句之外跳转到 IF 语句体中
- 不能从循环外跳转到循环体内
- 不能从子程序外部跳转到子程序中
- 不能从一个异常处理语句块内部跳转到当前语句块

4.7 游标

1.定义游标

游标是在 PL/SQL 块的定义部分定义的，语法格式为：

```
CURSOR cursor_name IS select_statement;
```

2.打开游标

```
OPEN cursor_name;
```

3.取值

```
FETCH cursor_name INTO variable[,variable]…;
```

4.关闭游标

```
CLOSE cursor_name;
```

5.游标属性

每个显式游标有 4 个属性：%FOUND、%NOTFOUND、%ROWCOUNT 和%ISOPEN。

这些属性附在游标名后面使用，即可得到有关多行查询执行中的有用信息。这些属性只能用在过程性语句 PL/SQL 中，而不能用在 SQL 语句中。

- 属性%FOUND
- 属性%NOTFOUND

- 属性%ROWCOUNT
- 属性%ISOPEN

4.8 出错处理

在 PL/SQL 中，将程序执行过程中的一个警告或错误称为一个例外（EXCEPTION）。例外情况的种类有 3 种：

1．预定义的 Oracle 错误

Oracle 预定义的例外情况大约有 24 种。对这种例外情况的处理，无须在程序中定义，由 Oracle 自动地将其引发。已经命名的部分内部例外如下：

- CURSOR_ALREADY_OPEN 表示企图打开一个已经打开的游标
- DUP_VAL_ON_INDEX 表示企图保存重复值到惟一索引约束的表格中
- LOGIN_DENIED 使用非法的用户名和口令登陆 ORACLE
- NOT_LOGGED_ON 表示登录 ORACLE 之前使用数据库调用

2．非预定义的 Oracle 错误

即其他标准的 Oracle 错误。这种例外情况，需用户在定义部分定义，然后由 Oracle 自动地进将其引发。

3．用户定义的错误

用户定义的错误是程序执行中出现的编程人员认为的非正常情况。对这种例外情况，用户要在程序中定义，而且显式地在程序中将其引发。

命名格式为：EXCEPTION_NAME EXCEPTION;

在 PL/SQL 块中，对于例外处理的语法为：

```
  EXECEPTION

WHEN 例外情况 1[OR 例外情况 2……] THEN
语句一；
语句二；
  ……

[WHEN 例外情况 3[OR 例外情况 4……] THEN
语句三；
语句四；
  ……
  [WHEN OTHERS THEN
语句五；
语句六；
……
```

其中：

例外情况	预定义的例外情况名或用户在定义部分定义的例外情况名
OTHERS	没有列例外处理部分中的所有其他例外情况。

4.9 本章小结

本章主要介绍 PL/SQL 语言的语法、构成、用途以及使用情况。PL/SQL 语言是一种过程化的语言，它可以使得经过多次处理的多条 SQL 语句一次执行，减轻了网络负担。PL/SQL 语言的语法和其他 3GL 类似，其结构体主要由定义部分、可执行部分和例外处理部分组成。定义部分定义了将在可执行部分中调用的所有变量、常量、游标和用户自定义的例外处理；可执行部分包括对数据库进行操作的 SQL 语句，以及对块中语句进行组织、控制的 PL/SQL 语句，这部分在 PL/SQL 块中必须存在；例外处理部分是指对可执行部分中的语句，在执行过程中出错或出现非正常现象时，所做的相应处理，在 PL/SQL 块中可以没有该部分。使用 PL/SQL 语言，DBA 可以建立功能强大的数据库存储过程、函数、包、触发器以及出错调用等。

第II部分 Oracle数据库结构与管理

第 5 章 Oracle的管理界面

本章的学习目标：

- 熟悉 Oracle 企业管理器管理界面
- 熟悉常用的管理工具
- 熟悉两种登录方式
- 熟悉数据库服务器的启动和关闭

5.1 企业管理器

最主要的几个组件有：

- Management Server
- Performance Manager
- Oracle Net Manager

展开“网络”根目录，你可以看到事件、作业、报告定义、HTTP 服务器、数据库、监听程序、组和节点 8 个子目录。下面就此作一些简单介绍。

- 事件
- 作业
- 报告定义
- HTTP 服务器
- 数据库

主要用来管理数据库的各种日常操作。可以分为例程、方案、安全性、存储、复制和工作空间等。下面分别叙述：

⑴例程主要用来管理全程的配置以及资源使用情况等。

⑵方案主要用来管理数据库对象，如表、索引、视图等。

⑶安全性用来管理用户、角色及概要文件。

⑷存储主要用来管理数据库，如控制文件、数据文件、重做日志文件以及归档日志文件等。

⑸复制主要是用来复制各节点之间的信息。

⑹工作空间是一种环境，一个或多个用户可在其中进行共享，更改数据库中的数据。工作空间管理是指管理这个由许多用户共享的工作空间的任务。

- 监听程序
- 组
- 节点

5.2 Oracle Net Manager

Oracle Net Manager 是 Oracle 配置网络连接的最主要图形界面。你可以选择多种网络连接方式并可以对数据实行加密。

- 概要文件：主要用来选择客户端与服务器端的连接方式。
- 服务命名：主要是配置数据库的服务信息。
- 监听程序：主要是用来配置系统监听器。
- Oracle Names Server：用来配置 Oracle 命名服务器。

5.3 登录方式

数据库的登录方式主要有两种，即操作系统确认方式和密码文件确认方式。

- 操作系统确认方式
- 密码文件确认方式

此登录方式为数据库默认登录方式，当用户使用此方式登录时，必须指定一个用户名和密码。为了使用此种确认方式，必须进行以下步骤的操作：

(1) 使用 ORAPWD 生成一个密码文件，此密码文件用来存放 SYS 的密码。

(2) 设置 REMOTE_LOGIN_PASSWORDFILE 参数

(3) 授予 SYSDBA 或者 SYSOPER 权限给用户

5.4 启动和关闭数据库服务器

1.启动数据库

数据库启动过程可以分为 3 个过程：数据库相关的例程（INSTANCE）启动，例程配置（MOUNT）数据库和数据库被打开。

可以有以下几种启动方式：

- 正常启动 STARTUP 或者 STARTUP OPEN。
- 启动到 NOMOUNT 状态，使用命令 STARTUP NOMOUNT，此时与数据库相关的例程被启动。
- 启动到 MOUNT 状态，使用命令 STARTUP MOUNT，此时例程读取控制文件信息，但数据库没有被打开。只可以查询某些 V$视图，但不能查询任何以 DBA 为起首的视图。
- 启动数据库到 RESTRICTED 状态，可以使用命令 STARTUP RESTRICT。启动到此状态后，只有 RESTRICTED SESSION 系统权限的用户才可以连接到数据库。当你进行数据库维护或进行其他操作时，将使用此种打开方式。当已经使用 STARTUP 启动数据库时，可以通过命令 ALTER SYSTEM ENABLE RESTRICTED SESSION 使数据库处于 RESTRICTED 状态。使数据库返回正常状态，使用命令 ALTER SYSTEM DISABLE RESTRICTED SESSION。
- 启动数据库到 READONLY 状态，使用命令 STARTUP READ ONLY。默认方式为 READ WRITE 状态。STARTUP RECOVER 用来启动数据库进行数据库恢复。

除了上面介绍的一些方法外，还有一些其他的启动方式，有时启动例程时可能会出现问题，这种情况下用 START FORCE 命令，它将重启数据库例程。

2. 关闭数据库

数据库的关闭和启动大致类似，也分为 3 个过程：关闭数据库、例程卸载数据库和例程关闭。

- Shutdown Normal 或 Shutdown

它是一种默认方式，主要过程为：

(1) 不允许新的用户连接

(2) 等待所有用户脱离数据库，所有用户可以继续工作，

(3) 数据库关闭，所有用户脱离数据库

此方法关闭较慢，一般不采用。

- Shutdown Immediate
  (1) 不允许新用户进行链接
  (2) 中止所有用户的链接
  (3) 回滚所有未提交的事务
  (4) 数据库关闭，所有用户脱离数据库
- Shutdown Transaction
  (1) 不允许新用户进行连接。
  (2) 禁止所有新事务发生，当用户试图建立新事务时，会话被中止。
  (3) 等待用户回滚或提交任何未提交的事务。
  (4) 数据库关闭，所有用户脱离数据库。
- Shutdown Abort
  (1) 终止所有的当前 SQL 语句。
  (2) 中断所有用户链接。
  (3) 立即结束例程。
  (4) 将要回滚所有未提交的信息，例程重启后需要恢复，此过程一般由 Oracle 自动完成。

一般情况下不要使用此命令，除非当使用以上 3 种命令均无法关闭全程时才考虑使用。你也可以通过 Oracle 企业管理器来实现数据服务器的启动与关闭。

5.5 配置系统初始化参数

配置系统初始化参数是系统启动时各个参数的配置情况，它的配置情况可以通过查询 INIT.ORA 文件得到，此文件位于 ORACLE_HOME/ADMIN/ORACLE_SID/PFILE 目录下。也可以通过 Oracle 企业管理器来查看系统初始化参数。

如果需要修改哪些参数，直接单击所对应的区域，然后输入新值，单击“应用”按钮完成初始参数的修改。

5.6 本章小结

本章主要介绍了 Oracle 的主要的图形化界面管理工具，用户登录方式以及数据库服务器的启动与关闭。Oracle 主要的图形化界面管理工具主要包括 Oracle 企业管理器、性能调整管理器以及 Net Manager 等。Oracle 企业管理器是最为主要的图形管理界面，通过它，DBA 可以方便地管理各种 Oracle 服务器的资源。用户登录有两种方式，根据不同情况可以选择不同的密码确认方式。数据库服务器的启动和关闭也有多种方式，每种方式对应它的执行过程，读者应该明白这些具体的执行过程。

第六章 Oracle服务器的例程

本章的学习目标：

- 熟悉 Oracle 服务器的例程结构
- 掌握 Oracle 服务器例程结构各个部分的功能
- 掌握 Oracle 例程中的进程

6.1 系统全局区

系统全局区（System Global Area）为一组由 Oracle 分配的共享的内存结构，可包含一个数据库例程的数据或控制信息。

SGA 区的各组成部分如下：

- 数据库高速缓冲区（Database Buffer Cache）
- 共享存储区（Shared Pool）
- 重做日志缓冲区（Redo Log Buffer）
- Java 存储区（Java Pool）
- 大型存储区（Large Pool）
- 空池（The “Null” Pool）

SGA 总容量 = 共享池 + 缓冲区高速缓存 + 大型存储区 + Java 存储区 + 日志缓冲区+空池

注意：

在 Oracle 9i 中，可以调整 SGA 的大小，而无须重新启动数据库（动态）。这是与以前版本所不同的。

6.1.1 数据库高速缓冲区

Oracle 用两种机制来管理高速缓冲区：

- 写列表 用来管理那些已经修改但还没有写到磁盘的内存。
- LRU 机制 管理下述 3 种类型的缓存。它是一些数据块的排序，最近被使用的数据将放置在最前面，最少被使用的数据块将放置在最后面，当新表进入并被加到列表时，最少使用的数据块列表就被清除。

Oracle 的数据库高速缓冲区主要可分为 3 种类型。

- “脏”缓存（Dirty Buffers）
- 自由缓存（Free Buffers）
- “被钉住的”缓存(Pinned Buffers)

提示:

数据库高速缓冲区的大小由初始化参数 DB_CACHE_SIZE 来决定，这是 Oracle 9i 与以前版本所不同的地方。在 Oracle 8i 以及以前的版本中则是由 DB_BLOCK_SIZE 和 DB_BLOCK_BUFFERS 来决定。这两者的乘积即为它的实际大小。

6.1.2 共享存储区

共享存储区包括共享游标、存储的过程、控制结构、并执行消息缓冲区以及其他内容。该值越大，多用户系统的性能就越好；值越小，使用的内存就越少。

共享存储区主要用来存放 SQL、PL/SQL、过程和包、数据字典锁和字符设置信息、安全属性等。它主要包括库高速缓存（Library Cache）和数据字典高速缓存（Data Dictionary Cache)。

- 数据字典高速缓存用于存储分析 SQL 语句的数据字典行。
- 库高速缓存用来存储已经提交给 Oracle 的 SQL 语句、分析过的格式和执行计划，以及被执行过的 PL/SQL 包头与过程 Java 的类。

注意:

这里所说的被使用过，是指先前执行的 SQL 语句与本次执行的语句完全相同，即包括字母大小写相同。

共享存储区的大小主要由参数 SHARED_POOL_SIZE 来决定。必须把它设置的足够大，以确保有足够的空间来存储 SQL 语句和 PL/SQL 块。

6.1.3 重做日志缓冲区

重做日志缓冲区用于在内存中存储已经被修改的数据库信息。它是被循环使用的，也就是说 LGWR 进程写重做日志缓冲区从始端到末端，然后再返回缓冲区的始端。当整个重做日志缓冲区被填满时，将它的全部内容写入联机重做日志文件。

重做日志缓冲区的大小由 LOG_BUFFER 初始化参数决定。它决定了在内存中保留多大空间存储重做日志信息。如果重做日志缓冲区设置的太小，进程之间的竞争将会增加，进而造成系统性能的下降。

6.1.4 Java存储区

在 Oracle 9i 中，系统明显加强了对 Java 的支持。在出现 Oracle 9i 版本以前，Java_POOL_SIZE 为可选项，可以把此项设置为 0，但在 Oracle 9i 中，此为必选项。

6.1.5 大型存储区

共享服务器将大型存储区的分配堆用作会话内存，通过并行执行将它用作消息缓冲区，通过备份将它用作磁盘 I/O 缓冲区。该值被指定为初始化文件参数 LARGE_POOL_SIZE。

6.1.6 空池

这个池不真正拥有名字。它是专门用于块缓冲区（高速缓存的数据库块）、重做日志缓冲区以及“固定 SGA”盘区的内存。

6.2 进程全局区

进程全局区（Process Global Area）是一个单一的操作系统进程或线程的特定内存区，包含单个进程的数据和控制信息，这个内存不可以被系统中其他的进程或线程所访问。它主要包括排序区、用户私人会话信息、堆栈空间等几部分。PGA 不会在 Oracle 的 SGA 之外分配，它总是由进程或线程在本地进行分配。

对 PGA 的大小影响最大的因素是 init.ora 或会话级参数 SORT_AREA_SIZE 和 SQRT_AREA_RETAINED_SIZE。这两个参数控制在写盘之前 Oracle 用来进行数据排序的空间数量以及在排序完成之后将保留多少内存段。可以通过查询 v$statname 和 v$mystat 来查询 PGA 的大小。

6.3 用户全局区

UGA 实际上是会话的状态。它是会话必须能够得到的内存。UGA 的位置总体上依赖于 Oracle 为了接受连接而实施的配置。如果已经配置了 MTS（多线程服务器），那么 UGA 必须在每个人都获得访问的 SGA 内存结构中。会话可以使用任何一个共享服务器，因为它们中的任何一个都可以读取或写入会话数据。如果正在使用专用服务器连接，对会话状态的统一访问的需求就会消失，UGA 本质上包含在 PGA 中。

对 UGA 的大小影响因素与 PGA 相同，也可以通过查询来查询 UGA 的大小。

6.4 Oracle进程

进程是操作系统中的一种机制，它可执行一系列的操作步骤。在操作系统中使用作业（Job）术语。一个进程通常有它自己的专用存储区。Oracle 进程的体系结构设计可以使 Oracle 的性能最大化。

在 Oracle 例程中有三类进程：

- 服务器进程（Server Process）：这些进程基于客户端的请求来执行工作
- 后台进程（Background Process）：这些进程伴随着数据库的启动而启动，并执行各种维护工作
- 从属进程（Slave Process）：这些进程类似于后台进程，但它们代表后台进程或服务器进程执行额外工作的进程

6.4.1 服务器进程

服务器进程用于处理连接到该例程的用户进程的请求。当应用和 Oracle 在同一台机器上运行，而不是通过网络时，它一般将用户进程和他相应的服务器进程组合成单个的进程，以降低系统开销。然而，当应用和 Oracle 运行在不同的机器上时，用户进程经过一个分离服务器进程与 Oracle 通信。

服务器进程主要执行下列任务：

- 对应用所发出的 SQL 语句进行语法分析和执行。
- 从磁盘（数据文件）中读入必要的数据块到 SGA 的共享数据缓冲区（该块不在缓

冲区时）。

- 将结果返回给应用程序处理。

服务器进程有两种类型：专用进程（Dedicated Process）和多线程进程（Multithread Process）。

专用进程（又称单用户 Oracle 进程）是一种数据库系统，一个进程之执行一个用户的程序。当用户从终端断开时此进程被终止。

下面是一个专用服务器进程相应用户程序的大致过程：

1．客户端应用程序向 Oracle 例程发出一个连接请求。

2．服务器上的监听程序探测到此用户请求并生产一个专门服务器进程来对用户登录信息加以确认。

3．用户执行查询操作。

4．专用进程执行用户查询操作中的所有源代码程序。

服务器进程执行 SQL 语句并决定此 SQL 语句的动作是什么，比如 SQL 语句含 SELECT 则执行查询操作，如含 INSERT 则首先改变高速缓冲区的内容并决定何时启动 DBWn 进程对数据文件进行写操作。专用进程适合于查询数据量比较大的用户连接方式。

在多线程系统中，多个用户可共享一个进程，这大大降低了系统资源的消耗，在 Oracle 多线程环境中，Oracle 使用调度（Dispatcher）来处理用户的请求。一个调度之可以处理一个用户进程，且只能处理一种通信协议。当用户向服务器发出一个请求时调度首先对用户请求进行排队，然后服务器进程按顺序执行用户请求，执行完毕后将其放入一个专有的位置。调度可以访问此位置并把处理的结果返回用户端。一个访问周期全部结束。

6.4.2 后台进程

系统为了优化性能并且协调多个用户，服务器进程在执行用户请求过程中，将调用后台进程来实现对数据库的操作。大部分后台进程在例程启动时自动建立。一个 Oracle 例程可以有许多后台进程，但它们不是一直存在。后台进程可以分为以下几种：

- DBMR 进程

该进程将缓冲区写入数据文件，是负责缓冲存储区管理的一个 Oracle 后台进程。

Oracle 采用 LRU（Least Recently Used）算法（即最近最少使用算法）保持内存中的数据块石最近使用的，使 I/O 使用最小化。在下列情况下 DBWR 要将弄脏的缓冲区写入磁盘：

  - 当一个服务器进程将一缓冲区移入“弄脏”表，该弄脏表到临界长度时，该服务进程将通知 DBWR 进行写。
  - 当一个服务器进程在 LRU 表中查找时间太长时，如果没有查到可用的缓冲区，它停止查找并通知 DBWR 进行写。
  - 出现超时（每次 3s），DBWR 将通知本身。
  - 当出现检查点时。

- LGWR 进程

该进程将日志缓冲区写入磁盘上的一个日志文件，它是负责管理体制缓冲区的一个 Oracle 后台进程。LGWR 进程将自上次写入磁盘以来的全部日志项输出。LGWR 输出如下内容：

  - 当用户进程提交一事务时写入一个提交记录。
  - 每 3s 将日志缓冲区输出。
  - 当日志缓冲区的 1/3 已满时，将日志缓冲区输出。

- 当 DBWR 将修改缓冲区写入磁盘时将日志缓冲区输出。

注意

当需要跟多的日志缓冲区时，LWGR 在一个事务提交前就将日志项写出，而这些日志项仅当在以后事务提交后采永久化。

- CKPT 进程

该进程在检查点出现时，对区布数据文件的头信息进行修改，指示该检查点。

- SMON 进程

该进程例程启动时执行恢复，还负责清理不再使用的临时段。

- PMON 进程

该进程在用户进程出现故障时执行进程恢复，负责清理内存储区和释放该进程所使用的资源。

- RECO 进程

该进程是分布式应用当中所使用的一个进程，自动的解决在分布式事务中的故障。

- ARCH 进程

当数据库运行在归档状态下，在线重做日志文件在被重写之前被 COPY 到另一路径。这些归档日志主要是用来恢复数据库。ARCH 进程最大数据为 10。

只有当初始化参数 LOG_ARCHIVE_START=TRUE，ARCH 才会自动启动。

- LCKn 进程

是在具有并行服务器选件环境下使用，可多至 10 个进程（LCK0，LCK1，…，LCK9），用于例程间的封锁。

- Dnnn 进程

该进程允许用户进程共享有限的服务器进程。

6.4.3 从属进程

从属进程分为 I/O 从属进程和并行查询从属进程两种。

- I/O 从属进程

I/O 从属进程用来为不支持异步 I/O 的系统或设备模拟异步 I/O。

控制 I/O 从属进程的 init.ora 参数有两个：

  - BACKUP_TAPE_IO_SLAVES：指定 I/O 从属进程是否由 RMAN 应用来备份数据、复制数据或把数据恢复到磁带中。此参数为布尔值。
  - DBWn_IO_SLAVES：它描述由 DBWn 进程使用的 I/O 从属进程的数量。DBWn 进程和它的从属进程总是向缓冲区高速缓存中的脏块的磁盘上进行写操作。

- 并行查询从属进程

Oracle 7.1 在数据库中引入了并行查询功能。这种功能使用 SQL 语句，并创建了一个执行计划，计划中包含许多可以同时完成的执行计划，其中每一个计划的输出都合并为一个大的结果。这样，并行执行查询的时间只是串行执行所用总时间的一小部分。

6.5 本章小结

本章主要介绍了 Oracle 的内存结构以及进程，内存结构主要有系统全局区和程序全局区。系统全局区由共享存储区、数据库高速缓存、Java 存储区、大型存储区和重做日志缓冲区组成，程序全局区主要由排序区、堆栈空间和用户私有会话区组成。Oracle 的进程可

分为专用进程和多线程进程，专有进程适合于少量用户执行大量操作，而当有大量用户连接到 Oracle 时，如果每个用户只执行简单的查询等操作，可以考虑使用 Oracle 多线程服务器。

第 7 章 Oracle数据库的物理结构

本章的学习目标：

- 熟悉控制文件的功能
- 掌握多控制文件系统的配置方法
- 熟悉重做日志文件的功能和作用
- 掌握重做日志文件的日常管理方法
- 学会查询日志文件信息
- 掌握归档日志模式的设置

7.1 数据文件

数据文件(Data Files)包含表或索引形式逻辑表示的实际数据库数据。一个 Oracle 表空间由一个或多个数据文件组成，一个数据文件不能与一个以上的表空间相关联。有关数据文件的详细内容，将在下一章介绍，本章只着重讨论控制文件和重做日志文件的内容。

7.2 控制文件

控制文件(Control files)用来存储数据库的物理结构----数据文件和重做日志文件。控制文件为一个二进制文件，在数据库生成时生成。当数据库的任何物理改变或是增加、重命名一个数据文件时，控制文件都将被更新。控制文件是数据库的重要组件，不能对它进行编辑，只有 Oracle 进程才能更新其内容。当启动数据库时，Oracle 使用控制文件来辨别数据文件和重做日志文件，然后将其打开。控制文件内容包括：

- 控制文件所属的数据库名，一个控制文件只能属于一个数据库
- 数据库生成的时间
- 数据文件的名称，位置，联机/脱机状态信息
- 重做日志文件的名称和路径
- 表空间名称
- 当前日志序列号
- 最近检查点信息
- 恢复管理器的备份信息

控制文件的大小在数据库生成时由参数 MAXLOGFILES，MAXLOGMEMBERS，MAXLOGHISTORY，MAXDATAFILES 和 MAXINSTANCES 来决定。Oracle 为控制文件预留了最大的存储空间，因此当增加或重命名一个数据文件时，控制文件大小从不改变。可以通过企业管理器来查询或管理控制文件信息。

7.2.1 多路控制文件

控制文件对数据库的操作十分重要，Oracle 建议最少有两个控制文件，通过多路技术或操作系统镜像技术，可以分散控制文件到不同的磁盘。多路可解释为保持拷贝文件到不同的路径。拷贝控制文件到不同的路径，必须修改初始化参数 control_files。

通过控制文件的多路化，可以避免单点错误的危险。在多路状态下，Oracle 更新数据文件时间会稍微变长，但极为可行。多路化的最大数目为 8 即每个数据库最多只能有 8 个。可通过下列步骤来增加控制文件：

（1） 关闭数据库 shut down immediate

（2） 使用操作系统命令拷贝控制文件到新的位置

（3） 编辑初始化参数 control_files（此参数在 init.ora 文件中可以找到，此参数文件为将要启动使用的参数文件）

（4） 启动数据库 startup

7.2.2 控制文件的生成

使用 CREATE CONTROLFILE 命令可以生成一个新的控制文件。只有在丢失所有的控制文件或者想改变此数据库的 MAX 参数时才考虑使用此命令。必须知道数据文件和重做日志文件的名称。下面是控制文件生成的步骤：

（1） 准备 CREATE CONTROLFILE 命令，应该包括所有数据库文件，如果忽略了其中的一个，数据库将永远丢失此文件！
保存此文件在 C 根目录下，并命名为 create_ctl.sql。

（2） 关闭数据库 shutdown immediate

（3） 启动数据库，并置数据库为 NOMOUNT 状态。记住，MOUNT 状态需要打开控制文件。Startup nomount

（4） 执行步骤（1）的语句，用以生成控制文件。@c:\create_ctl.sql

（5） 使用 ALTER DATABASE OPEN 命令打开数据库。

（6） 关闭数据库并进行备份。

控制文件生成以后，应该查询字典当中是否有数据文件丢失。可以通过查询视图 V$DATAFILE，丢失的文件将被命名为 MISSINGnnnn。在生成控制文件时，如果使用了 ALTER DATABASE OPEN RESETLOGS 命令，丢失的数据文件再也不能加回数据库，可以使用 NORESETLOGS 功能来生成控制文件，这样丢失的数据文件可以通过介质恢复来恢复丢失的数据文件。

7.2.3 查询控制文件信息

除了通过企业管理器来管理控制文件外，还可以通过视图 V$CONTROLFILE 来查看控制文件信息。

7.3 重做日志文件

重做日志文件 (Redo Log Files)用来记录对数据库的所有更改，LGWR 周期性写入重做日志缓冲区，例如有 3 个重做日志文件，LGWR 首先写第一个，当第一个满时，开始一写第二个，然后是第三个，最后返回第一个。联机重做日志文件包含了所有的重做记录。

7.3.1 管理重做日志文件

每个数据库有它自己的联机重做日志组，一个日志组可以包含一个或多个日志文件。当数据库创建或生成控制文件时，可以指定重做日志文件组的数目。

7.3.2 日志转换(Log Switch)

LGWR 进程周期性地写入重做日志文件。当 Oracle 写完一个文件，开始写下一个时，日志转换发生。你可以手动启动日志转换程序。

（1） 查询视图 V$LOGFILE

（2） 执行日志转换命令 alter system switch logfile

（3） 再查询视图 V$LOGFILE

发现日志文件 2、3 同时处于被写状态，这说明进程 LGWR 执行效率不够，日志转换速度过慢。

当日志转换发生时，Oracle 指派一个新的序列号码到新的重做日志文件，这个序列号码称为日志序列号。如果一个数据库的事务增加，日志转换将频繁发生。恰当地确定重做日志文件的大小可以避免这个现象。日志转换的信息将保留到 ALERT LOG 文件中。

7.3.3 检查点

当检查点事件发生时，所有已经修改的数据将从高速缓冲区写到磁盘，同时更新控制文件和数据文件。

手工触发检查点事件发生。使用命令 ALTER SYSTEM CHECKPOINT

可以设置 LOG_CHECKPOINT_TIMEOUT 、 LOG_CHECK_POINT_INTERVAL 和参数 FAST_START_IO_TARGET 来设定检查点发生的频率。

- LOG_CHECK_POINT_INTERVAL

此参数用来指定重做日志文件每写多少操作系统块发生一次检查点事件。

- LOG_CHECKPOINT_TIMEOUT

此参数用来指定每隔多少时间发生一次检查点。默认状态下，Oracle 的企业版设置为 1800，标准版设置为 900，即每个 1800 秒/900 秒系统将发生一次检查点事件。如果此参数设置为 0，即为忽略此参数。

- FAST_START_IO_TARGET

此参数用来限制数据库高速缓冲区中脏表中数据块的个数。在数据库例程恢复过程中，Oracle 从联机重做日志文件读取已经改变的数据块信息，并保存在内存中。通过设定此参数，可以限制例程在恢复过程中读取数据块的个数，从而减少了例程恢复时间。默认状态下，此参数值为数据库缓冲区的个数。

7.3.4 多路日志文件

可以保持多个联机重做日志文件的拷贝，以防止某些日志文件的损坏。在多路日志文件下 LGWR 同时写入相同的重做日志信息到多路联机重做日志文件，因此避免了因一个重做日

志文件损坏而不能写入的情况发生。所有的拷贝作为一个组，且大小相同。

当多路日志文件，最好放置每个组的成员到不同的磁盘。所以一个磁盘损坏并不影响数据库的连续操作。

在数据库生成时，可以生成联机重做日志文件的多个备份。

- 生成新的组
- 添加组成员
- 重命名组成员

重命名重做日志文件和把重做日志文件移动到另一个路径和磁盘有些麻烦，可以通过以下步骤来实现：

- 删除重做日志组
- 删除重做日志组员

增加或删除重做日志文件可以使用 Oracle 企业管理器，使用鼠标单击所要修改的部分，然后输入新的值，也可以利用它来查看或验证所做的修改，也可以通过使用图形化的企业管理器来实现对重做日志文件的创建与管理

7.4 归档日志文件

归档日志文件(Archive Log Files)也叫脱机日志文件。它是联机重做日志文件的拷贝。当数据库处于 ARCHIVELOG 状态下时，归档进程将拷贝重做日志文件到另一路径，归档日志文件主要用做数据库的恢复。

7.4.1 设置归档路径

归档路径由初始化参数来决定，可以使用#￥来改变归档路径。下面是一些归档日志路径的参数：

- LOG_ARCHIVE_DEST
- LOG_ARCHIVE_DUPLEX_DEST
- LOG_ARCHIVE_DEST_N
- LOG_ARCHIVE_MIN_SUCCEED_DEST
- LOG_ARCHIVE_MAX_PROCESSES

7.4.2 设置ARCHIVELOG/NOARCHIVELOG模式

（1）关闭数据库(SHUTDOWN IMMEDIATE)并编辑初始化参数
编辑初始化参数使 LOG_ARCHIVE_START=TRUE
（2）启动数据库于 MOUNT 状态 STARTUP MOUNT
（3）使用 ALTER DATABASE ARCHIVELOG 命令
（4）使用 ALTER DATABASE OPEN 命令打开数据库

如果使已经处于归档日志状态下的数据库工作于非归档日志状态可使用如下步骤：

（1） 关闭数据库(SHUTDOWN IMMEDIATE)并编辑初始化参数
编辑初始化参数使 LOG_ARCHIVE_START=FALSE

（2） 启动数据库于 MOUNT 状态
（3） 使用 ALTER DATABASE NOARCHIVELOG 命令
（4） 使用 ALTER DATABASE OPEN 命令打开数据库

使数据库工作于 ARCHIVELOG 状态，可用以下步骤来实现：
（1） 关闭数据库 SHUTDOWN IMMEDIATE。必须以 SYS 身份登录才有权执行此命令。
（2） 启动数据库于 MOUNT 状态 STARTUP MOUNT
（3）使用 ALTER DATABASE ARCHIVELOG 命令
（4）使用 ALTER DATABASE OPEN 命令打开数据库
可以手工进行归档全部日志文件(ARCHIVE LOG ALL)或当前的日志文件(ARCHIVE LOG NEXT)。

7.4.3 查询日志和归档信息

可以通过 ARCHIVE LOG LIST 命令来查询数据库工作于哪种模式（归档或非归档模式）。
ARCHIVE LOG LIST
注意：
使用此命令，必须以 SYS 身份登录。
通过查询视图 V$LOGFILE，可以查看数据库日志文件分组情况、状态、类型以及组员信息。
查询视图 V$THREAD:
查询视图 V$ARCHIVE_PROCESSES，此视图可以显示归档日志进程的运行情况。
查询视图 V$LOG_HISTORY，其中 RECID 为日志转换次数，FIRST_CHANGE#记录了日志文件开始写时的 SCN 号码，FIRST_TIME 记录了日志文件开始写时的时间，NEXT_CHANGE#记录了日志文件转换时的最大 SCN 号码。

7.5 本章小结

本章主要讲述了数据库控制文件和重做日志文件。对控制文件主要介绍了控制文件的概念、内容、创建过程、信息查看以及管理情况等；重做日志文件主要介绍了重做日志文件的概念、管理方法以及信息查询等。

第八章 Oracle数据库的逻辑结构

本章的学习目标：

- 掌握表空间与数据文件的概念及其相互关系
- 掌握表空间的日常管理内容和方法
- 掌握数据块和扩展区的概念
- 掌握回滚段的分类、作用与管理

8.1 表空间

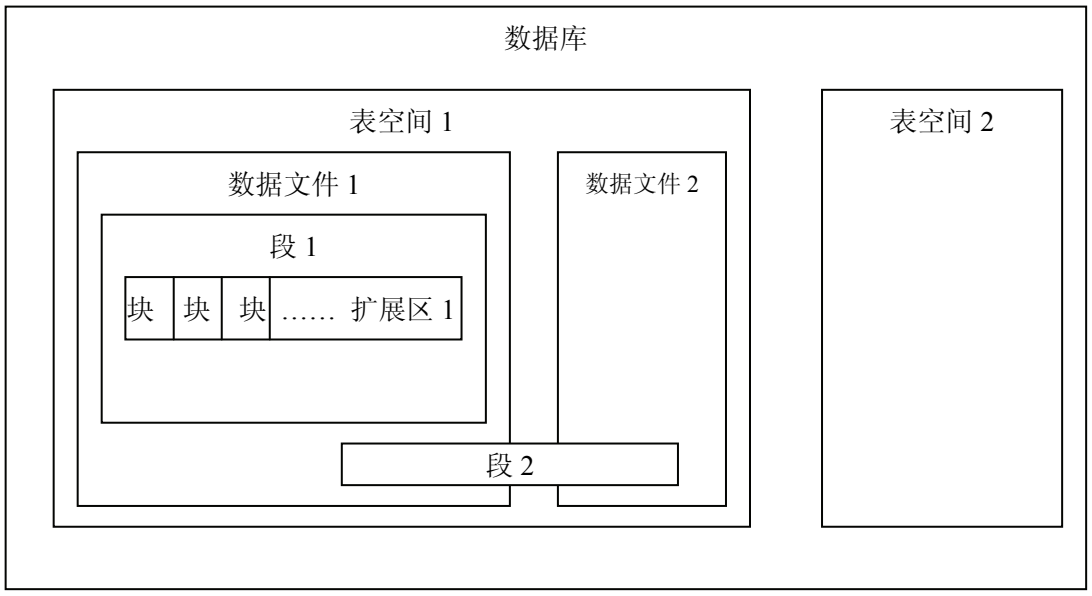

图 8-1 数据库、表空间、数据文件和数据块对象关系图

8.1.1 表空间管理

Oracle 向对象（表格、索引等）分配空间是通过分配一系列连续的数据库块即扩展区来实现的。每一个对象分配一个段，它包括一个或多个扩展区（如果对象被分区，如表分区等，每一个分区可分配一个段）。

1．表空间创建（Create Tablespace）

使用 Create TABLESPACE 可以生成表空间。表空间生成时可以指定扩展区，如果没有指定扩展区，Oracle 自动生成一个字典管理空间（Dictionary-Managed Tablespaces）。

字典管理表空间的创建的各个参数：

- INITIAL：指定表空间的第一扩展区的大小为 512K；
- NEXT：用来指定下一扩展区的大小，本例中设定为 512；
- PCTINCREASE：指定第三扩展区及随后扩展区增长的参数，默认值是 50。即下一扩展比前一扩展区大 50%，此例中也为 50%，即第三扩展区的大小为 512K*1.5=768K；

- MINEXTENTS：指定在段生成时向段指派的扩展区的最小数目，本例中为 2；
- MAXEXTENTS：指定在段生成时向段指派的扩展区的最大数目，本例中为 2048；
- LOGGING：指定在 DDL 操作和直接装载插入状态下在重做日志下保存记录；
- ONLINE：指定表空间生成以后立即处于联机状态
- PERMANENT：指定表空间是否用来生成永久性对象，如表、索引等；
- EXTENT MANAGEMENT：此参数用来指定表空间是由数据字典管理还是由本地管理（LOCALLY MANAGED TABLESPACE）。默认值为数据字典管理表空间。

本地管理表空间（Locally-Managed Tablespace）

使用 CREATE TABLESPACE 命令和 EXTENT MANAGEMENT LOCAL 可以生成一个本地的管理表空间。它可以更高效的管理表空间，减少磁盘碎片。

临时表空间（Temporary Tablespace）

临时表空间主要是用来提高排序操作的效率，一个临时表空间可以被只用作排序段。可只由一个段组成。

2．修改表空间

某些情况下你可能需要对表空间进行修改，如使它处于脱机状态或是进行数据库的联机备份等操作时。你可以使用 ALTER TABLESPACE 命令对表空间进行修改。

开始和结束备份名为 USERS 的表空间：

ALTER TABLESPACE USERS BEGIN BACKUP；

ALTER TABLESPACE USERS END BACKUP；

对表空间进行重命名时，必须先使它处于脱机状态，然后使用操作系统命令把与表空间相应的数据文件拷贝到相应路径，并重命名，最后使它重新回到联机状态。

1） 表空间处于脱机

2） 使用操作系统命力拷贝原有数据文件到目标路径。并重命名；

3） 修改数据文件名称

4） 使表空间处于联机状态

3．增加表空间

当发现原来的表空间不能满足数据库增长的需要时，作为数据库管理员必须知道如何增加表空间大小来满足不断增长的数据库空间需求。可以通过增加已有数据文件的大小来增加表空间的大小：

你也可以通过增加数据文件的个数来增加表空间：

4．删除表空间

使用 DROP TABLESPACE 命令可以删除一个表空间。如果表空间在删除时为非空，应该使用 INCLUDING CONTENTS 语句，步骤如下：

（1）生成表空间

（2）向表空间添加一个表

（3）执行命令 DROP TABLESPACE AZURE，提示错误，因为表空间非空

（4）执行命令 DROP TABLESPACE AZURE INCLUDING CONTENTS，系统显示操作成功。

当删除表空间时，只有控制文件被更新，而实际的数据文件并没有从操作系统删除，应该使用操作系统的命令来释放磁盘空间。

也可以通过 Oracle 企业管理器来查询、增加和删除表空间。

要想修改各个参数，只需双击相应的单元格，然后输入新的参数，按“应用”按钮即可。

5．查询表空间信息

- 视图 DBA_TABLESPACES

- 视图 V$TABLESPACE
- 视图 DBA_DATA_FILES
- 视图 DBA_FREE_SPACE

8.1.2 管理数据文件

在管理数据文件时，如果你想让此数据文件大小一定，以便有助于对表空间管理的话，你可以指定文件大小，这样，数据库数据的增长就不会超出预期范围。你也可以指定数据文件根据需求自动增长，它节约 DBA 不少的时间。使数据文件根据需求自动增长通过以下命令来实现：

ALTER DATABASE DATAFILE ‘f:\orcl\user001.dbf’ AUTOEXTEND ON;

要取消数据文件自动增长，使用以下命令：

ALTER DATABASE DATAFILE ‘f:\orcl\user001.dbf’ AUTOEXTEND OFF;

当已指派的扩展区被写满时，向扩展区插入数据的命令将终止，而且 Oracle 将返回一个错误信息，这时你可以手动指派扩展区。

8.1.3 数据文件信息查询

- V$DATAFILE，
- DBA_DATA_FILES
- DBA_TEMP_FILES

8.2 数据块

数据块（Blocks）是 Oracle 存储体系中最小的逻辑单元。数据块的大小在数据库生成时被指定，且不可更改。所有的数据块都是完全相同的，不管它是用来做存储表，索引还是聚族。

- 数据块头：主要包含块的类型和地址信息，通常占 24 个字节。块的类型可以是数据，索引或回滚类型。
- 表路径：此部分包含本数据块所对应的表信息。
- 行数据：存储在块中实际的行。
- 自由空间：用来存放新的行或用来更新已存在的行。

块存储参数：

- 当对象例如表或索引的、生成时，可以指定块存储的选项：PCTFREE 和 PCTUSED。
- 当向表插入数据时，如果表格的行的长度大于块的大小，行的信息无法完全存放在一个块中，行链接发生，Oracle 将使用多个块来存放行信息。
- 当表数据被更新时，如果更新后的数据长度大于块的长度，Oracle 将把整行的数据从原数据块中迁移到新的数据块中，只在原数据块中留下一个指针指向新数据块，这就是行迁移。行迁移和行链接都将影响系统查询性能。因为查询此行时，Oracle 将读去多个数据块。

PCTFREE：当数据块的自由空间百分率底于 PCTFREE 的值时，此数据块将被标志为 USED，

其相关信息被移处 FREELIST。此参数默认值为 10。

PCTUSED：当数据块的使用空间低于 PCTUSED 的值时，此数据块将标志为 FREE，其相关信息被加入到 FREELIST 当中。此参数默认值为 10。

INTRANS 用来指定当外部事务进入数据块时，可同时对此数据块进行 DML 操作的事务个数。表数据块的此参数通常默认值为 1，索引块默认值为 2。

MAXTRANS 同 INTRANS 类似，它用来指定当外部事务进入数据块时，可同时对此数据块进行 DML 操作的最多事务个数，最大可达 255。

INTRANS 和 MAXTRANS 通常在创建表、索引或是聚族时使用，如果你的系统用来进行联机事务处理，通常为 INTRANS 设置一个较高的值。

ROWID 用来指定每一行的物理存储位置,它由 18 位 64 进制的数字组成。数字包括 0--9，a--z,A--Z 以及符号+、-共 64 个。它包含了行所在的数据块地址信息、数据文件地址信息和行地址信息。每一行的 ROWID 在此数据库中惟一。

ROWID 以列的形式存在于每个表中。由于它是伪例，我们在通常的查询中是看不到的。可以选用 SELECT ROWID,[COLUMN]FROM[TABLE]来查看它。COLUMN 可为此表中任意一列,TABLE 为某一表格。

8.3 扩展区

扩展区（Extents）有一系列连续的数据块组成，当一个段生成时，首先指派扩展区，段与扩展区、数据块的关系如图 8-5 所示，当此扩展区数据已满时，新的扩展区将指派到此段，可用以下参数实现对扩展区的控制。

INTIAL：制定表空间的第一扩展区的大小。

NEXT：用来指定下一扩展区的大小。

PCTINCREASE：制定第三扩展区及随后扩展区增长的参数，默认值是 50。即下一扩展区比前一扩展区大 50%。

MINEXTENTS：指定在段生成时向段指派的扩展区的最小数目。

MAZEXTENTS：指定在段生成时向段指派的扩展区的最大数目。

8.4 段

段（Segment）是由一个或多个扩展区组成的逻辑存储单元，每个数据库对象指派一个段来存储数据，如一个未分区的表就是一个段，段中所有扩展区的大小的总和即是此段的大小。一个段只能从属于一个表空间，它可以覆盖多个数据文件，共有 4 种类型的段：数据和索引段(Data Segment & Index Segments)，临时段(Temporary Segments)，回滚段(Rollback Segments)。

8.4.1 数据段和索引段

数据段用来存储表、聚簇的数据，数据段包含：未分区的表（表的分区将在下一章中介绍），一个表的分区，一个表的聚簇。索引段存储 ROWID 和索引键，在索引生成时你可以指定存储参数。

8.4.2 临时段

当用户进行排序查询时，如果在指定的内存无法完成排序，Oracle 将自动从用户默认的临时空间中指派空间进行排序。指派的段称为临时段，当会话结束时，数据将从临时表中自动删除，临时段信息可通过视图 DBA_SEGMENTS 来查询。

8.4.3 回滚段

回滚段用于存放数据修改之前的值（包括数据修改之前的位置和值）。回滚段的头部包含正在使用的该回滚段事务的信息。一个事务只能使用一个回滚段来存放它的回滚信息，而一个回滚段可以存放多个事务的回滚信息。

像任何其他段一样，回滚段被定为在表空间中并且包含区间。每个数据库有一个 SYSTEM 回滚段，它位于系统表空间中并在数据库建立时建立。普通用户不允许使用它并且它不能被脱机使用或者删除。应该为专用于回滚段的表空间中数据库的正常工作建立附加回滚段。回滚段的扩展和压缩是很普通的。回滚段对数据块的作用主要有：

事务回滚

事务恢复

读一致性

1. 回滚段的种类

系统回滚段

非系统回滚段：

2. 回滚段的写操作

当外部查询对回滚段进行写现在时，如果有大量应用程序要使用回滚段，则回滚段将实行循环写操作，而如果此时回滚段的其他扩展还有活跃的事务时，系统将自动增加新的扩展区到回滚段，我们称之为回滚段的动态扩展或动态指派。

3. 回滚段的创建、修改和删除

使用 CREATE ROLLBACK SEGMENT 命令可以生成回滚段。

也可以指定回滚段的存储参数：

修改回滚段用命令 ALTER ROLLBACK SEGMENT 可以修改回滚段信息。

当回滚段需要重建或由于其他原因需要删除时，可以使用命令 DROP 删除回滚段

注意：删除回滚段时必须使他脱机。

4. 常见回滚段问题----快照太久错误（Snapshot Too Old Error）

回滚段用来保持数据库的读写一致性，数据库的数据被修改时，Oracle 首先通过快照机制把原来的数据复制到回滚段，此时如果有其他用户需要查询此数据，Oracle 将从回滚段中读取，当用户执行 ROLLBACK（未提交前）命令时，Oracle 将从回滚段把数据返回原来的位置，即执行 UNDO 操作。如果一方面进行大量的数据修改操作，一方面进行长时间的查询工作，当回滚段太小时，快照太久错误就可能发生。

避免此种情况发生的最有效方法是增多回滚段空间。你也可以通过建立一个只被一个事务使用的回滚段来解决此冲突，具体方法，将在性能调整中介绍。

查询回滚段信息

回滚段信息可以通过查询视图 DBA_ROLLBACK_SEGS 和动态视图 V$ROLLNAME，V$ROLLSTAT。

8.5 本章小结

本章主要介绍了 Oracle 数据库的逻辑结构，即表空间、段、扩展区和数据块的概念以及使用方法。使用方法包括生成、管理及相关信息查询等。一系列连续的数据块构成了扩展区，一个或多个扩展区组成段，而一个表空间可能包含一个或多个段。Oracle 的段又分为数据段、索引段、临时段和回滚段。应该知道每种段的作用以及管理查询信息等。

第 9 章 表、索引与约束

本章的学习目标：

- 掌握表的创建方法
- 熟悉表分区的概念与方法
- 熟悉索引和约束的概念
- 熟悉索引和约束的功能与作用
- 熟悉索引和约束的创建方法

9.1 数据库表创建

数据以表的形式存储在数据库当中，它是数据库存储数据的基本形式，即所有的数据库数据都存储在表当中。方案或用户是数据库表格的所有者，表的每一列定义为一个数据类型，存储数据必须满足该数据所在列的属性。

9.1.1 生成一个简单表

- 表的生成可以使用 CREATE TABLE 命令来实现
- 如果在另外一个用户下生成表，必须指定表的所有者。
- 可以通过图形化界面 Oracle 企业管理器来创建表。

一个表包含许多列，每一列都有各自的属性。如果某一列的属性为数据类型，则可以指定此数据类型的精确度，如果这一列的列属性为字符，就可以指定其最大宽度。Oracle 为列提供了多种属性。如表 9-1 所示。

9.1.2 指定存储参数

在表生成时，如果没有指定表空间或存储参数，表将自动在当前用户默认的表空间生成。存储参数将采用此表空间的默认值。表生成时最好估计一下表的大小并指定确切的存储参数。一个表如果太大，考虑分区或存储到一个独立的表空间当中是十分必要的。

主要参数如下：

USERS：为表所在的表空间

STORAGE：指定块存储参数

INITRANS：指定可对表所有块进行同时更新操作的用户的数目

INITAIAL：指定表起始扩展区的大小

NEXT：指定下一扩展区的大小

MINEXTENTS：指定最小扩展区的个数

MAXEXTENTS：指定最大扩展区的个数

PCTINCREASE：指定扩展区递增百分率

FREELIST：指定空闲列表组中表数量，默认及最小值均为 1

FREELIST GROUPS：指定表空闲列表组的数量

BUFFER_POOL：指定表是否存储到数据库高速缓存当中，以及存储在数据库高速缓存的

具体位置。数据库高速缓存可分为 3 部分：KEEP，RECYCLE，DEFAULT，默认值为 DEFAULT。如果表较小且经常被查询，可把表存储到 KEEP 里面，如果表非常大很少被访问，可以把表存储到 RECYCLE 里。如果不是上述两种情况，将被存储到 DEFAULT 里。有关此方面的具体信息将在性能调整部分当中介绍。

9.1.3 为表分区

对表分区的方法有 3 种，即有序分区法、混乱分区法和混合分区法。下面我们主要以有序分区法为例加以介绍。

1． 有序分区法

你可以通过一个列值来创建一个有序分区，例如一个表可以通过日期来分区，可以把每一个月或每一季度来作为表的一个分区。

2． 混乱分区法

当你不知道每一个分区将有多少数据，或者不知道分区变化的大小时，将使用此种方法。混乱分区采用 HASH 算法。分区的数目必须指定为 2 的 N 次方（比如 2，4，8，16，32……）。

PARTITION BY HASH 用来指定此表是使用混乱分区法来分取的。

3． 混合分区法（Composite-Partitioned Table）

此分区法为以上两种方法的综合。有序分区法为主分区法，混乱分区法为子分区法。

9.2 表管理

表创建以后，应用程序不断地对表进行插入更改和删除数据，DBA 的很大一个任务是对表的管理，管理工作有为表指派扩展区、建立约束、对表进行分析和重新组织等。用户可以通过 ALTER TABLE 命令来改变表的存储位置，增加或删除列，修改列属性，如默认值，数据类型和长度等，也可以把表从一个表空间移到另一个表空间、屏蔽约束和触发器等。

9.2.1 指派与回收扩展区

由于数据不断地被插入到表里，表尺寸在不断地增加，当已经指派的最初扩展区被用完时，如果有新的数据插入，Oracle 将指派新的扩展区到此表，称此为扩展区的动态指派，也可以手动指派扩展区给表格。

9.2.2 表重组

表重组过程中指定大的扩展区可以减少扩展区的数目，或者设置合适的 PCTFREE 和 PCTUSED 参数来避免行迁移的发生，重组表使用命令 ALTER TABLE TABLE_NAME MOVE 命令。

9.3 表分析

表分区可以被用来查找行链接和行迁移，确认数据块是否损坏，收集表统计信息，也可以对表某一部分进行分析。对一个表进行分析主要有验证表结构、发现行迁移和收集统计信息等操作，下面分别来介绍：

1. 验证结构

由于硬件原因，包括磁盘错误或软件的 BUG，一些数据库块可能被损坏，当相应的行被访问时，Oracle 将返回一个错误信息。ANALYZE 可被用来确认数据块的完整性，如果 Oracle 发现块或行不可读将返回一个错误。损坏行的 ID 将被插入到一个表格当中。可以指定此表格的名字，默认表格为 INVALID_ROWS，可以通过运行 utlvalid 脚本来创建这个表。

如果 Oracle 碰到坏的行，它们将被插入到 UTLVALID_ROWS 表当中，指定一个具体的表可以使用 ANALYZE 命令。

2. 发现行迁移

可以使用 ANALYZE 来发现迁移或链接的行。

Oracle 将相当的行的 ID 插入到一个指定的表中，如果表未指定，Oracle 将查询 CHAINED_ROWS 表，该表可以用 utlchain 脚本来生成。

下面的方法将用来修复行迁移：

- 如果存在迁移，生成一个临时表用来存储迁移的行
- 从原表格中删除迁移行
- 把迁移行回插到原来的表格

3. 收集统计信息

使用 ANALYZE 命令可以收集统计信息并存放到字典表里，COMPUTE 参数将统计所有的信息，ESTIMATE 参数将统计表的部分信息。

9.4 创建索引

索引被用来快速访问数据库而非读取全部表，使用它可以大大减少磁盘 I/O 的读取次数。可以像表一样为索引指定存储参数，为索引分区，收集统计参数并进行分析，确认其结构等。

9.4.1 索引的分类与生成

可以使用 CREATE INDEX 命令来生成索引。索引的生成和删除对基表没有任何影响，Oracle 自动管理这些索引：当行被增加、更新或删除时，Oracle 自动更新相应的索引。可以生成下列几种类型的索引：

- 位图索引(Bitmap Index)
- 平衡树索引(B-tree Index)

此为默认方式，索引用平衡树的算法来生成。平衡树包含索引列的节点和行的 ID，ID 是用来区分表行信息的，下面是平衡树索引的几种类型：

非惟一索引

惟一索引

反键索引

使用命令 CREATE INDEX 命令可以生成一个非惟一索引，必须为索引指定一个名字，以及表的名称。

生成一个惟一索引你必须指定关键字；

生成一个位图索引必须指定关键字 BITMAP，位图索引不能惟一；

索引的存储参数以及分区状况大致与表格相同；

指定关键字 REVERSE 可以生成反键索引，在并行服务器下，反键索引可以提高特定 OLTP 应用程序的性能。

- 索引组织表格（IOT）：

IOT 是一个表的排序子集，其本身就是一个表，也是一个索引。有一点要记住，IOT 没有自己的 ROWID，这是它和其他普通表一个明显区别的地方。也正因为此，在 IOT 表格上是不能另外建立索引的。这一点可以从索引的原理得到。因为在 IOT 中索引和表的数据存储在一起，所以它适合于经常通过主键查询的数据表。一个 IOT 为一个平衡树索引，通过组建 IOT 可以提高系统的性能。

9.4.2 索引修改

表的数据经常被插入、删除可能会造成索引性能的下降，也可能是因为创建索引时参数设置不当，需要对索引做进一步的修改。使用命令 ALTER INDEX 可以改变索引的存储参数。

9.4.3 查询索引信息

视图 DBA_INDEXES 为管理索引的最为主要的视图之一，通过它可以查询大部分的索引信息，你可以用 DESC 命令来查看其视图数据类型。限于篇幅原因，我们只列出了视图 DBA_INDEXES 部分信息。

9.5 数据库的完整性约束

Oracle 利用完整性约束机制防止无效的数据进入数据库的基表当中，如果任何 DML 执行结果会破坏完整性约束，那么该语句被回滚并且返回一个错误，Oracle 实现的完整性约束完全遵守 ANSI X3.139-1989 和 ISO9079-1989 标准。

9.5.1 约束的分类

- Oracle 数据库的完整性约束类型有 NOT NULL 约束、UNIQUE 约束、PRIMARY KEY 约束、FOREIGN KEY 约束和 CHECK 约束。

9.5.2 约束的创建

约束可以在表格生成时创建，也可以通过 ALTER TABLE 命令给已经生成的表格追加约束，

当然也可以通过 Oracle 企业管理器来创建。选中所要追加约束的表格，在右边选项卡中选择“约束条件”即可对表格约束进行修改或添加。

9.6 本章小结

本章主要讲述了表、索引与约束的基本概念创建命令以及各自的使用方法。使用 CREATE TABLE 命令可以创建表，默认所有者为当前用户。可以指定表的存储参数，默认状态下为所在表空间的参数，INITIAL，NEXT，PCTINCREASE 用来指定扩展区的大小，表创建后 INITIAL 和 MINEXTENTS 参数不可更改！

表分区用来管理大型表，以便提高性能。有 3 种分区方法需要知道。分区的表也可以建立索引，索引的分区与表分区没有必然联系。

约束可以在表创建时生成，也可以使用命令 ALTER TABLE 命令生成。约束共有 5 种，非空约束，主键约束，外键约束，惟一性约束和检查约束，记住：每个表只能有一个主键约束！

第 10 章 概要文件、用户权限与角色

本章的学习目标：

- 熟悉概要文件的作用与管理内容
- 掌握数据库用户的创建与管理
- 熟悉数据库角色与权限的概念
- 掌握角色的创建与权限的授予
- 熟悉对象权限和系统权限的概念

10.1 概要文件

概要文件（Profiles）被用来控制用户对系统和数据库资源的使用，Oracle 提供了一系列的预定义资源参数，可以使用这些参数并设置所希望的值，概要文件还可以被用来管理密码。

资源管理功能，Oracle 允许通过概要文件控制以下几种类型的资源使用：

- 每个用户的当前会话数目
- CPU 使用时间
- 私有 SQL 和 PL/SQL 区的使用
- 逻辑读取的实现
- 用户连接到数据库的空闲时间

只有当设置了参数 RESOURCE_LIMIT 为 TRUE 时，才可以使用 ORACLE 资源限制。可以在初始化文件当中设置此参数，这样每次启动数据库时，就无须再改此参数了。也可以使用命令 ALTER SYSTEM 动态地更改设置。下面是一些用来控制资源使用的参数：

- SESSIONS_PRE_USER：用来限制一个用户当前会话的数量。当会话数量达到此参数时，用户不能使用更多的会话。
- CPU_PRE_SESSION：用来限制一个会话可以使用的 CPU 时间。
- CPU_PRE_CALL：用来限制一个 SQL 语句使用的 CPU 时间。
- LOGICAL_READS_PER_SESSION：用来限制一个会话所读取的数据库数据块的个数。它包括从内存和磁盘中读取的总和。
- LOGICAL_READS_PER_CALL：用来限制一个 SQL 语句所读取的数据库数据块的个数。它也包括从内存和磁盘当中读取的总和。
- PRIVATE_SGA：此参数用来限制 SGA 区中私有区域空间指派的大小。
- CONNECT_TIME：用来指定一个会话可以连接到数据库的最大时间数，当达到此参数时，用户被自动从数据库断开。所有未处理的事务得到回滚。
- IDLE_TIME:此参数用来指定一个会话可以连续空闲时间的最大时间数，单位为分钟。当达到此参数时，用户被自动从数据库断开，所有未处理的事物得到回滚。
- COMPOSITE_LIMIT：此参数用来设置用户对系统资源的综合消耗。此参数由 CPU_PER_SESSION、LOGICAL_READS_PER_SESSION、PRIVATE_SGA、CONNECT_TIME 这几个参数综合决定。

概要文件还可以用来管理用户密码。可以设置以下参数：

- FAILED_LOGIN_ATTEMPTS：此参数用来设置当用户登录时，错误登录的尝试总次数。
- PASSWORD_LOCK_TIME：用来指定当用户登录失败以后，用户账号被锁定的时间。
- PASSWORD_LIFE_TIME：指定用户可以使用同一个密码的天数。如果在设定的日期内

用户密码没有更改，用户将不能继续使用此密码进行登录。

- PASSWORD_GRACE_TIME：用来设定在用户密码被终止前，系统向用户发出警告，提醒用户修改密码的天数。
- PASSWORD_REUSE_TIME：用来指定当一个用户修改密码以后，用户可以必须经过多少时间才可以再次使用原来的密码。
- PASSWORD_REUSE_MAX：用来指定当一个密码被重新使用时，密码修改的次数。

可以通过 Oracle 企业管理器来查看 Profile 文件信息。

10.2 管理用户

用户也叫做方案，是一组逻辑对象的所有者。任何需要进入数据库的操作都需要在数据库中有一个合法的用户名。下面是一些与数据库用户相关的信息：

验证方式：每个用户必须通过一个密码连接到数据库，以便被确认，操作系统验证方式将在 10.3 节中介绍。

默认和临时表空间：当用户创建对象时，如果没有特殊指定另外一个表空间，将使用默认表空间。临时表空间用来创建临时段。

空间分配：在每个表空间当中，必须指派给用户一个空间配额，以便创建对象。

10.2.1 创建用户

使用命令 CREATE USER 可以生成用户，用户验证方式必须在用户生成时确定，通常用数据库来验证。用户被分配一个密码并以加密的方式存储在数据库中。当用户与数据库建立连接时，Oracle 对密码进行检验。

10.2.2 修改用户信息

使用 ALTER USER 命令可以修改所有的用户信息。可以指派或修改默认角色给用户，改变默认表空间。

10.2.3 删除用户

可用 DROP USER 命令来删除用户。

如果用户拥有对象，Oracle 将返回一个错误值。必须指定关键字 CASCADE，Oracle 将删除用户所有的对象，然后再删除用户。如果其他用户对象，比如过程、包或视图需要参考此用户的对象，它们将被标志为非法。当对象被删除后，空间立即释放给表空间。

10.3 用户验证

当用户连接到 Oracle 数据库时，Oracle 会验证用户输入的信息是否正确，如果正确，用户将被确认；如果不正确，用户登录被拒绝。用户验证机制有以下两种方式。

第一种是数据库验证。

第二种是操作系统验证。

注意：

必须十分小心使用此参数,因为任何具有此操作系统账号的计算机都可以通过网络登录到 Oracle 所在的服务器并连接到数据库。

10.4 查询用户信息

可以通过查询视图 ALL_USERS 来查看所有用户的用户名、USER_ID 和创建日期。

视图 DBA_USERS 为管理用户的最为主要的视图之一，通过它可以查看用户的 USER_ID、账号的状态、表空间分配以及相关概要文件设置等情况。

同样，也可以在 Oracle 企业管理器中查看并管理用户信息。

10.5 权限与角色

权限用来控制用户在数据库中所能进行的操作。通过权限，用户可以创建、删除或修改对象。权限可以通过两种方式授予用户

- 直接授予用户
- 先授予一个角色，再把角色指派给用户。
    1. 先生成一个角色。CREATE ROLE GRE
    2. 再把 JOBS 表的 SELECT 权限授给角色 GRE。GRANT SELECT ON JOBS TO GRE
    3. 最后把角色 GRE 授给用户 AZURE。GRANT GRE TO AZURE;

角色是一系列权限的集合，它是产生相同权限用户的一种简洁方式。可以把一系列权限分配给角色，再把角色分配给多个用户。Oracle 有两种类型的权限：对象权限和系统权限。

10.5.1 对象权限

对象权限（Objects Privilege）的授予必须指定一个具体的对象。对象所有者拥有对象的所有权限，它可以把此对象的权限授予其他用户，还可以决定其他用户是否有权限把权限授予另外的用户。

下面一共有 9 种权限可以授予用户：

SELECT 此权限可以读取表、视图、序列等的数据。

UPDATE 此权限可以更新表、列、视图等的数据。

DELETE 此权限可以删除表、视图等的数据。

INSERT 此权限可以向表、列、视图中插入数据。

EXECUTE 此权限可以执行一个存储的 PL/SQL 对象。

READ 此权限可以读字典中的数据。

INDEX 此权限可以生成索引。

REFFERENCES 此权限用来生成外键。

ALTER 此权限可以修改表或序列的结构。

下面是一些对象权限的补充：

SELECT 权限不能指定为列。当授予列的权限给用户时，先使用相应的列生成一个视图，然后把视图权限授予给用户。

拥有 DBA 权限，将一个用户的对象的权限授予给另一个用户时，DBA 也要被授予 WITH CRANT OPTION 权限。

10.5.2 系统权限

系统权限（System Privilege）可以像对象权限一样，授予用户角色和 PUBLIC。 Oracle 有许多的系统权限，其中 ANY 权限是非常“高级”的权限，必须很谨慎的使用，它可以对所有的数据库对象，包括系统所属的字典对象进行操作。

下面是一些系统权限的补充：

- 连接到数据库需要具有 CREATE SESSION 权限。
- 删除一个另外用户的表需要具有 DROP ANY TABLE 权限。
- SELECT、INSERT、UPDATE 和 DELETE 是对象权限；SELECT ANY、INSERT ANY、UPDATE ANY 和 DELETE ANY 是系统权限。

授予系统权限，系统权限可以授予用户角色和 PUBLIC。使用 WITH ADMIN OPTION 可以进行权限的传递。

可以通过 Oracle 企业管理器来查看、管理用户权限信息，

10.5.3 权限回收

对象权限和系统权限可以通过 REVOKE 语句加以回收。

权限回收时需要注意以下几点：

- 多个管理者授予同一个用户权限后，其中一个管理者回收其授予权限时，不影响其他管理者授予的权限。
- 为了回收 WEITH ADMIN OPTION 权限或者 WITH GRANT OPTION 权限，必须先回收其权限，然后再授予其相应的权限。
- 如果一个用户传递系统权限 WITH ADMIN OPTION 给其他用户后，如果用户的权限被回收，其他用户的权限不受影响。
- 如果一个用户传递权限 WITH GRANT OPTION 给其他用户后，如果此用户的权限被回收，其他用户的权限将自动回收。

10.5.4 角色管理

角色是一系列的权限的集合，他使权限的管理更加容易，使用 CREATE ROLE 命令可以生成一个角色。只有数据库的所有者才拥有此权限。角色生成时，没有任何权限与之相关联，必须指派权限给角色。

和用户一样，角色也有两种确认方法，即数据库验证和操作系统验证：

- 数据库验证方式
- 操作系统验证方式

使用 ALTER ROLE 可以修改角色密码或者验证方式，但不能对一个角色重命名。

删除一个角色使用 DROP ROLE。当一个角色被删除后，它将从用户的角色列表中立即删除。

1. 预定义角色

数据库生成时 Oracle 自动产生 6 个预定义角色，他们分别是：CONNECT、RESOURCE、DBA、SELECT_CATALOG_ROLE、EXECUTE_CATALOG_ROLE、DELETE_CATALOG_ROLE、EXP_FULL_DATABASE 和 IMP_FULL_DATABASE

2. 激活/屏蔽角色（Enabling/Disabling Roles）

如果一个角色不是用户的默认角色，当用户连接到数据库时，此角色无效，使用 ALTER USER 可以为用户设置默认角色。一共有 4 种方式设置默认角色。

- 设置某一角色为用户的默认角色
- 设置所有角色为用户的默认角色
- 设置除某一角色外均为用户的默认角色
- 设置用户没有默认角色

3. 查询角色信息

视图 DBA_ROLES 可以被用来查看具有 DBA 管理权限角色的信息。

视图 ROLE_ROLE_PRIVS 主要被用来查看角色和权限授予情况，以及是否有传递权限的权限。

视图 ROLE_SYS_PRIVS 可也用来产系统权限授予情况。

同样，通过 Oracle 企业管理器可以查看并修改角色信息。

单击选中左半窗口的角色，右半窗口显示出“一般信息”、“角色”等选项卡，如果要修改角色内容，直接选中它进行修改。

10．6 本章小结

本章主要介绍了 Oracle 的安全管理文件——概要文件（Profile 文件）、用户的创建与管理、用户权限的授予和回收、角色的创建与回收等，还介绍了一些查询用户及其权限的方法。

第 11 章 常用工具

本章的学习目标：

- 熟悉 SQL*Loader 的使用方法
- 熟悉数据导出导入工具 Export/Import

11.1 SQL*Loader

使用 SQL*Loader 可以把外部的文本文件导入到数据库的表中。当用户进行大量导入时，可以选择此工具。它可以加载多个表，且文件格式不必规则，还可以指定任何分隔符结束。

d:\sqlldr help=y

用 SQL*Loader 加载数据通常有两种方法，即一般装入法和直接装入法，一般装入法（Conventional Path）为 Oracle 默认方式。它适用于较小数据的导入。直接装入法(Direct-Path)加载数据时，系统将绕过数据库高速缓存，直接写数据块到数据文件，设置参数 Direct 可以使用此项功能。加载速度要快于一般装入法，它适合于大量数据快速装入。直接装入法可以通过指定参数 Direct=Y 来实现并行操作。

11.2 数据导入与导出

在日常管理过程中，经常要把数据从一个数据库移到另一个数据库，或者从一个用户移到另一个，这种情况下将考虑使用导入与导出工具。

11.2.1 用EXPORT导出数据

本小节中 EXPORT 工具将导出 Oracle 表中的数据到外部文件，此文件为二进制格式，默认名为 EXPDAT.DMP，并且只有 IMPORT 工具可以读取。在一个工作平台上导出的文件可以在不同的平台上进行数据导入。

数据导出可以有以下几种方式：

- 表模式
- 用户模式
- 数据库模式
- 表空间模式

DIRECT=Y 参数用于指定直接输出法，此方法之所以快速是因为它读取数据文件到数据库高速缓存而不经过 SQL 语句处理，用户也可以指定其他参数。

11.2.2 用IMPORT导入数据

IMPORT 工具将读取 EXPORT 工具导出的文件，并执行插入操作。可以选择导入模式为数据库，表或用户操作大致与 EXPORT 工具相同。IMP HELP=Y 可以看到在线帮助。

另外有一点需要介绍的是 IMPORT 工具对数据的重组功能，使用 EXPORT 和 IMPORT 工具

对表进行重组以消除行迁移，主要是使用了 COMPRESS 选项对导出导入扩展区进行了压缩。下面将分步加以介绍：

（1） 制定导出计划文件。

```
FILE=EMPLOYEES.DMP
TABLES=system.t_emp
COMPRESS=Y
DIRECR=Y
```

（2） 执行数据导出

```
EXP syetem/manager PARFILE=EMPLOYEES.PAR
```

（3） 将原表删除。

```
DROP TABLE t_emp;
```

（4） 生成导入参数文件。

```
FILE=EMPLOYEES.DMP
FROMUSER=system
TOUSER=manager
```

（5） 导入数据。

```
IMP system/manager PARFILE=EMPLOYEES.JAR
```

11.2.3 表空间传输

表空间传输(Transport Tablespace)用来实现本地数据库与远程数据库之间的数据传递。其基本操作是：将表和表空间导出，然后将导出的数据文件直接拷贝到目标路径下，然后再执行数据文件的导入。此种方法速度较快且适合大量数据的移动。下面两个重要参数必须加以注意：

- TRANSPORT_TABLESPACE
- TABLESPACES

执行数据导入，必须有以下两个参数：

- DATA_FILES
- TTS_OWNERS

表空间的传输有以下的限制：

- 块大小在两个数据库当中必须相同。
- 目标数据库不能存在于将要导入的表空间同名的表空间。
- 原数据库和目标数据库必须工作于相同的平台（硬件和操作系统平台）。
- 传输表空间当中，要么包含一个表的所有分区，要么一个分区也没有。

11.3 国家语言支持

国家语言支持（Native Language Support）可以存储国家语言和格式。Oracle 支持绝大多数语言和字符设置。Oracle 使用 UNICODE 来支持多语言。数据库的字符设置在数据库创建时使用子命令 CHARACTER SET 来指定，设置字符用来存储如 CHAR，VARCHAR2，CLOB 和 LONG 类型的数据，其他的如表名、列名、PL/SQL 变量等也适用。下面有几个重要的 NLS 参数信息：

NLS_DATE_FORMAT 用来设定日期默认格式

NLS_CURRENCY 用来设定货币符号

NLS_LANGUAGE 用来设定使用的语言

NLS_ISO_CURRENCY 用来设定 ISO 货币符号

其他参数可通过视图 NLS_DATABASE_PARAMETERS 来查看。

11.4 本章小结

本章主要介绍了 Oracle 的常用数据加载工具 SQL*Loader、输出输入工具 EXPORT 和 IMPORT 的使用方法及表空间传输等。把外部的大量文本信息加载到数据库最常用的方法是使用 SQL*Loader 工具，它可以方便地把其他数据源转化到 Oracle 数据库里面。导出导入工具 EXPORT 和 IMPORT 可以用来进行数据库的逻辑备份和恢复，也可以用它来进行数据的远距离传输。

第三部分 Oracle 数据库的备份与恢复

第 12 章 Oracle备份与恢复机制

本章的学习目标：

- 了解备份与恢复的重要性
- 理解数据库备份一些基本概念
- 理解冷备份与热备份的区别
- 理解归档备份与非归档备份对恢复的影响
- 理解几种不同的恢复机制

12.1 理解数据库备份

简单地说，备份是一个数据的拷贝，这个拷贝包括数据库的重要的部分，如控制文件和数据文件。备份的目的是为了防止意外的数据丢失和应用错误。

备份分为物理备份和逻辑备份。所谓物理备份就是复制物理的数据文件，而逻辑备份是指使用 Oracle Export 程序将数据从数据库中抽取出来存在一个二进制的文件中。可以将逻辑备份作为物理备份的补充，还可以使用 Recovery Manager 或操作系统命令来实现物理备份。

还原（Restore）一个备份指的是重新构造它并使其可以被 Oracle Server 使用。恢复（Recovery）一个已还原数据文件是指用重写日志记录（Redolog record）来更新它，例如数据库备份以后所修改的记录。

Oracle 在一个例程失败之后，自动进行例程恢复，例程恢复包括两个操作：前滚（Rolling Forward）和回滚（Rolling back）。

- 前滚：即执行联机重写日志来使备份更接近当前状态
- 回滚：即还原未提交事物中的修改

数据库可能发生以下几种失败：

- 语句错误：当用户发出一个错误的 SQL 语句或是中间软件与客户端软件不兼容，Oracle 返回一个错误。
- 进程失败：当网络发生意外错误，如路由器出现错误或用户不正常地结束会话，会发生进程失败。
- 用户错误：当用户错误或不小心地删除了某个数据库对象，如表，此时发生用户错误。
- 介质失败：当存储数据文件或控制文件的磁盘或磁盘控制器发生错误时，介质失败发生。

语句错误和进程失败不需要 DBA 介入，Oracle 将会自动加以恢复，而当用户错误和介质失败发生时，需要 DBA 制定恢复策略并执行恢复。

当用户设计、实现并使用其备份策略时，应记住以下几点：

- 设计备份与恢复策略；
- 定期测试你的备份与恢复策略（对于新的维护成员特别重要）；
- 确保操作系统的备份仍然脱离数据库服务器。数据库服务备份不能保护 INIT.ORA

文件、Oracle 设备或操作系统。

- 在重要的修改以前或以后执行适当的数据库备份。假如此修改包括删除段或表空间的话，执行适当的数据库备份就显得特别重要。
- 包含动态数据的表空间比多数静态表空间需要更为频繁地备份。
- 保存较老的备份。
- 定期导出数据库进行附加的保护。
- 考虑适用于高利用率应用程序的分布式数据库备份。

12.2 冷备份与热备份

冷备份即脱机备份（offline backup），是用操作系统命令把所有的数据库文件拷贝到另一个磁盘或磁带上面。它提供了最简单，最直接的方法保护数据库免遭因介质破坏而丢失。

冷备份一般涉及生成所有必要的数据库元素（配置文件、数据文件、控制文件、重做日志和归档日志）的一个文件拷贝。在备份过程开始以前，需要确保数据库能够足够长时间地停留在脱机状态以便得到完全备份。假如你没有足够的时间，应考虑使用热备份。

热备份即联机备份（online backup），它是在数据库处于开放状态下对数据库进行的备份。要进行热备份数据库必须处于归档日志状态下。使用热备份的好处在于当一个数据文件或表空间处于备份状态时，用户仍然可以访问其他部分。

热备份要求用户对 Oracle 和支持 Oracle 的操作系统具有较高的专业知识以及适应水平。

12.3 归档备份与非归档备份对恢复的影响

在归档日志状态下，数据库拷贝联机重做日志到归档日志，这就是说数据库保留了外部事务对数据的更改。归档备份的好处有：

- 可以进行完全恢复和不完全恢复。
- 可以使联机备份即热备份成为可能。
- 可以使备用数据库（standby database）成为可能。

在非归档日志状态下，没有归档日志文件产生，对数据库的更改也就没有被完整保存下来。这样可以节省磁盘空间，提高系统性能，但却使数据库处于一种危险状态。因为一旦介质错误发生将无法对数据库进行恢复，从而造成数据的丢失。

配置系统模式

归档日志状态可以在数据库创建时设定，也可以使用 ALTER DATABASE ARCHIVELOG 命令实现。使用此命令时，数据库必须处于 MOUNTED 状态。具体步骤如下：

设置 LOG_ARCHIVE_START 参数为 TRUE，如果使用 Oracle 9i 版本，则此步骤可忽略。

关闭数据库并使用 STARTUP MOUNT 重启。

```
SHUTDOWN IMMEDIATE
STARTUP MOUNT
ALTER DATABASE ARCHIVELOG
ALTER DATABASE OPEN
```

此方式将使数据库工作在自动归档状态下。设置好以上参数以后，检查数据库是否运行在归档状态。下面是几种检查的方法：

- 查看初始化文件 INIT.ORA 的参数 LOG_ARCHIVE_START 是否为 TRUE。

- 在 SQL PLUS 下执行命令 ARCHIVE LOG LIST。
- 查看操作系统的进程。
- 查看视图 V$DATABASE
- 在 SQL*PLUS/SQLPlus 工作单下执行命令：ALTER SYSTEM SWITHLOGFILE

可以使用以下初始化参数来配置归档路径：

LOG_ARCHIVE_DEST_n

LOG_ARCHIVE_DUPLEX_DESTINATION

LOG_ARCHIVE_MIN_SUCCEED_DEST

LOG_ARCHIVE_DEST_STATE_n

初始化参数 LOG_ARCHIVE_DEST 用来指定归档日志文件的存储路径。可以手工编辑 INIT.ORA 文件来指定其路径。

12.4 理解几种不同的恢复机制

1. 完全恢复

完全恢复是从一个热备份或冷备份中恢复一个已丢失的数据文件的较老的(并且是有效的）拷贝，然后重新运用自恢复的数据文件得到备份以来发生的所有变化。

2. 不完全恢复

假如一个归档日志文件遗失，在用户执行完全恢复过程中，Oracle 将停止恢复过程并提示用户把遗失的文件放入归档日志目录中。因为没有可用的此日志文件，所以用户需要让 Oracle 使用到目前为止已经恢复的日志文件打开数据库。Oracle 不能跳过一个归档日志并用随后的日志继续恢复。如果是这样，那么只能执行不完全恢复。

3. 备份测试

我们在前面已经讲过了备份和恢复的有关内容了，但是在实际操作中每一个公司或站点都需要自己独特的备份与恢复操作。因为完全依赖于备份和恢复方案，所以必须肯定它们确实可靠。一个未经过测试的备份与恢复方案比没有方案还糟糕。

最理想的情况是有一台与要保护的机器完全一样的机器,用户可以用它来测试备份和恢复方案。通过把例程复制到测试系统上，来测试备份与恢复方案。

12.5 本章小结

本章主要介绍了数据库备份的重要性以及与备份相关的概念和设置。数据库的备份对每一个系统来说都是极其重要的，因此每个 DBA 都要理解数据库安全可靠性，并能制定相应的备份策略。另外，数据库运行模式的选择对数据库的备份有着决定性的影响，用户可以通过各种方式来查看数据库的运行状态。

第 13 章 非RMAN下物理备份与恢复实现

本章的学习目标：

- 掌握数据库的冷备份方法
- 掌握数据库的热备份方法
- 掌握控制文件的备份方法
- 掌握几种不同的数据库恢复方法

13.1 数据库的冷备份

用户可以通过以下步骤来实现冷备份：

（1） 关闭数据库。SHUTDOWN IMMEDIATE

（2） 复制所有数据库到目标路径。

（3） 打开数据库。STARTUP

注意：

数据库关闭时，必须保证各文件处于一致状态，即不能使用 SHUTDOWN ABORT 命令强行关闭数据库，只能使用 SHUTDOWN NORMAL 或 SHUTDOWN IMMEDIATE 来关闭数据库。

13.2 数据库的热备份

热备份即联机备份，它是在数据库处于开放状态下对数据库进行的备份。数据库的热备份也是通过操作系统复制命令来实现的。用户可以通过下列步骤来实现：

（1） 查询视图 V$DATAFILE 和视图 V$TABLESPACE 来决定需要备份的数据文件。

（2） 使表空间处于备份状态。ALTER TABLESPACE USERS BEGIN BACKUP

（3） 操作系统拷贝数据文件。注意，USERS 表空间有两个数据文件，不要丢失一个，否则数据库恢复时会有麻烦。

（4） 结束备份状态。ALTER TABLESPACE USERS END BACKUP

重复执行步骤（2）、（3）、（4）对表空间进行逐个备份。

13.3 控制文件的备份

控制文件的备份有 3 种方式，其中前两种可以用命令行来实现，第 3 种要使用 RMAN。我们这里只介绍前两种。

第一种是把控制文件备份为二进制文件，用户可以指定备份的路径。

第二种是把控制文件备份为 ASC 文件，此 ASC 文件为所生成的跟踪文件的拷贝。此文件存在于 USER_DUMP_DEST 所指定的目录下，USER_DUMP_DEST 参数可以在 INIT.ORA 文件中找到。在 Oracle 8i 当中，需要在 MANAGER SERVER 工具下实现，而在 Oracle 9i 当中，可以直接在 SQL*PLUS 中实现。

13.4 几种不同的恢复方式

数据库运行的模式不同，其恢复机制也有所不同，我们介绍其中常用的 3 种，即非归档日志下的数据库恢复、归档日志下对丢失部分数据文件的恢复和丢失整个数据库情况下的恢复。

13.4.1 非归档日志下的数据库恢复

当系统丢失了一个或多个数据文件时，需要对整个数据库进行还原（RESTORE），其步骤如下：

（1） 启动数据库，发现数据文件丢失的错误警告。
（2） 关闭数据库。
（3） 从冷备份中拷贝所有的数据库文件到原始位置。
（4） 启动数据库。

13.4.2 归档日志下对丢失部分数据文件的恢复

当系统运行于归档日志状态下，如果丢失了一个或多个数据文件，可以对数据文件进行逐个恢复。其步骤如下：

（1） 执行 ALTER TABLESPACE 命令准备表空间备份:ALTER TABLESPACE USERS BEGIN BACKUP
（2） 使用操作系统命令拷贝表空间 USERS 至另一路径
（3） 使表空间结束备份状态：ALTER TABLESPACE USERS END BACKUP
（4） 在 USERS 表空间中创建一个名为 t_test 的表格，模拟备份表空间后数据库数据的更改
（5） 使用操作系统命令把 USERS 所对应的数据文件删掉，模拟介质损坏。
（6） 关闭数据库服务器。SHUTDOWN IMMEDIATE
（7） 启动数据库，发现数据文件丢失的错误警告。STARTUP
（8） 警告显示系统无法识别数据文件，但提示用户参考 DBWR 跟踪文件。
查看 ALERT 文件，文件提示查看跟踪文件 orclDBW0.TRC
（9） 查看跟踪文件，文件显示只有文件 USERS01.DBF 丢失:
（10） 对丢失数据文件进行恢复。STARTUP
（11） 使 USERS01.DBF 数据文件处于脱机状态
（12） 行操作系统命令，从备份中拷贝所有的数据文件到原始位置。
（13） 结束备份状态。
（14） 打开数据库。ALTER DATABASE OPEN;
（15） 查询表 t_test，确认没有数据丢失
（16） 如果存在多个数据文件丢失，则重复执行步骤 11，12，13 可实现对多个数据文件的恢复。

13.4.3 丢失整个数据库情况下的恢复

如果需要恢复整个数据库，则使用以下步骤：

（1） 对整个数据库进行热备份
（2） 向表中插入数据，以模拟完全备份后数据库的更改。
（3） 然后重启数据库，发现错误信息。STARTUP
（4） 关闭数据库。SHUTDOWN ABORT
（5） 查看 ALERT 文件，文件提示查看追踪文件 orclDBWO.TRC
（6） 查看跟踪文件，显示所有数据文件丢失
（7） 启动例程到 MOUNT 状态:STARTUP MOUNT
（8） 把所有的数据文件拷贝到原来路径。
（9） 执行数据库恢复命令。RECOVER DATABASE
（10） 打开数据库 ALTER DATABASE OPEN, 确认没有丢失数据

13.5 非完全恢复

前面我们介绍的是把数据库完全恢复的例子,有时用户并不想或并不能把数据库完全恢复，这时将考虑使用不完全恢复机制。

13.5.1 基于Cancel的恢复

使用基于 Cancel 的恢复(Cancel-Based Recovery)机制可以把数据库恢复到错误发生前的某一状态。采用此方法，Oracle 执行恢复进程，直到发出 CANCEL 命令恢复结束。

（1） 对数据库进行一次完全的备份，包括数据文件、控制文件、重做日志文件和参数文件等。
（2） 遇到数据文件等其他文件丢失，先关闭数据库 SHUTDOWN IMMEDIATE 或 SHUTDOWN ABORT
（3） 使用操作系统命令把原来备份的文件拷贝到对应路径。
（4） 使用 STARTUP MOUNT 命令启动数据库
（5） 对数据库进行恢复 RECOVER DATABASE UNTIL CANCEL;
（6） 使用 RESETLOGS 或是 NORESETLOGS 打开数据库 ALTER DATABASE OPEN RESETLOGS /NORESETLOGS;

通常，执行完 RESETLOGS 或是 NORESETLOGS 都要对数据库进行备份，因为先前的数据库备份对系统已经不可用了。

13.5.2 基于Time的恢复

使用基于 Time 的恢复（Time-Based Recovery），此方法可使数据库恢复到某一特定时间。其具体做法大致与基于 Cancel 的恢复机制相同。

使用以下命令替代上述的步骤（5）的命令：

RECOVER DATABASE UNTIL TIME ’12-10-2005,19:23:18’;

ALTER DATABASE OPEN RESETLOGS

所不同的是，这里结束恢复不是使用 CANCEL 键，而是以某个时间点为终点。有一点要注意的是在输入时间时，时间格式可能有所不同，可以通过查询视图 NLS_DATABASE_PARAMETERS 来得到相关信息。

13.5.3 基于SCN的恢复

基于 SCN 的恢复（Change-Based Recovery）机制可使数据恢复到某一事务前。事务的具体信息可通过查询 V$LOG_HISTORY 视图获得。其具体做法大致也与基于 Cancel 的恢复机制相同，只需使用以下命令替代上述的步骤（5）的命令即可。

RECOVER DATABASE UNTIL CHANGE 87654321;

13.6 本章小结

本章主要介绍了在 SQL*PLUS 界面下几种不同的备份和恢复方法，用户可以执行物理冷备份、热备份、完全恢复和不完全恢复等。冷备份是在停机状态下使用操作系统命令把所有的数据库文件包括数据文件、控制文件、重做日志文件、参数文件等拷贝至另一路径。热备份即是在不停机状态下对数据库的备份，但此时数据库必须处于归档日志状态下。完全恢复是把数据库恢复到上次停机前的状态，只有在归档日志状态下才可以实现。有时，数据库不需要或者是不能执行完全恢复，此时只有考虑不完全恢复。使用不完全恢复有 3 种方式：基于 Cancel 的恢复、基于 Time 的恢复和基于 SCN 的恢复，可以根据实际情况来选择恢复方式。

第 14 章 逻辑备份与恢复

本章的学习目标：

- 熟悉数据库逻辑备份与恢复的概念
- 掌握使用 EXPORT/IMPORT 工具对数据库对象进行备份与恢复
- 掌握使用 Oracle 企业管理器对数据库进行备份与恢复

14.1 使用逻辑备份与恢复工具

在命令提示符下，键入 EXP HELP=Y 可以显示所有的数据导出关键字。

其中：

FULL：用来指定是否导出整个数据库。

BUFFER：用来指定在提交到导出文件以前，内存中装入多少行。

COMPRESS：当导入对象时，选择 Y，它将合并扩展区为一个大的扩展区。例如，一个有 5 个区间的表，导出前每个扩展区为 10M，当把表重新导入回数据库中，新表将有一个 5×10=50M 的扩展区。选择 N，新表仍旧为 5 个 10M 扩展区。

ROWS：选择 Y，将导出表及其所有数据；选择 N 仅导出重新创建该表的 SQL 代码。

INCTYPE：用来指定导出的类型。

在操作系统命令符提示下：键入 IMP HELP=Y 可以显示所有的关键字。

其中，FROMUSER 和 TOUSER 用来指定导出前用户的名字和导入对象的所有者的名字。TOUSER 必须在数据库中已经存在。

其他参数大致与 EXP 相同。

14.2 数据库逻辑备份与恢复的实现

掌握了备份与恢复工具的使用方法以后，下面通过一个实际的例子来模拟数据库数据的逻辑备份与恢复过程。

（1）使用 SYSTEM 帐号身份登录，创建一个名为 t_test 的表格，并插入数据

（2）执行表导出

（3）检查文件 test.dmp 显示文件已创建。接下来删除表 t_test，模拟数据丢失。

（4）执行数据导入，来模拟数据恢复

（5）验证数据导入成功

14.3 使用Oracle企业管理器

使用 Oracle 企业管理器对数据库进行备份与恢复的操作步骤如下:

1. 打开其交互界面，选中并运行它。

在出现的“导出向导”对话框，可以指定导出文件名与路径，如果想要修改它则可以双击它然后输入新文件名或路径即可

2. 在对话框中可以让你选择导出类型，可以选择整个数据库、一个或多个表格以及某个用户（方案）的所有信息。完成此设置后，单击“下一步”按钮，也可以直接单击“完成”

按钮完成设置。我们这里选择导出“表”并单击“下一步”按钮。

4．在 “高级选项”的对话框中“优化”选项卡中可以选择是否一致导出、是否压缩导出以及设定导出使用的缓冲区大小等。完成后单击“确定”按钮。

5．在接下来的对话框中，将是作业调度的设置，可以选择立即执行、只执行一次或是定期周期性的执行导出工作。这就是给 DBA 一个比较方便的管理方式，而从重复性的工作中解脱出来。

6．单击“下一步”按钮将出现有关作业信息的对话框。可以设置是否立即提交资源信息等，单击“完成”按钮。

7．在 “概要”对话框中可以浏览刚才所有的设定信息。

8．单击“确定”按钮将出现成功提交作业信息，确认操作成功，单击“确定”退出操作。

这个导出过程全部结束了，当熟悉此方法后，很觉得使用 Oracle 导出向导更方便。当然，如果读者使用命令行操作也会很方便，因为 Oracle 的图形操作和命令操作功能几乎完全相同，只是一个使用导航窗口，另一个使用语句操作罢了。

14.4 本章小结

本章主要介绍了 Oracle 数据库的逻辑备份和恢复方法。数据库的逻辑备份可以实现某一个表或者是用户模式的数据输出，这是在物理备份下所不能完成的。数据的逻辑输出还可以压缩扩展区以提高性能。而实现数据库的逻辑备份和恢复的方法有两种，即使用 EXPORT/IMPORT 工具和使用 Oracle 企业管理器，这两种方法都需要掌握。

第 15 章 Oracle恢复机制的补充

本章学习目标：

- 熟悉并行恢复的实现与设置
- 掌握控制文件的恢复方法
- 掌握只读表空间的恢复方法

15.1 并行恢复的实现

并行是一种加快恢复速度，减少停机时间的有效方法。特别是当数据库非常大时，此方法更为有效。在并行恢复过程中，Oracle 使用多个恢复线程，即读取多个恢复重做日志文件，而不是通过一个恢复进程读取一个重做日志文件，所以提高了恢复的速度。

为了使用并行恢复机制，必须为数据库配置并行查询进程。下面两个初始化参数必须正确设置：

PARALLEL_MIN_SERVERS：此参数用来指定最小的并行恢复线程。

PARALLEL_MAX_SERVERS：此参数用来指定最大的并行恢复线程。

并行查询线程不能开得过多，最佳的方法是 PARALLEL_MAX_SERVERS 应该等于或小于 CPU 总和的个数，或者是硬盘驱动的个数。

15.2 控制文件的重建

简单的方法就是复制一个已有的完好的控制文件备份到目标路径。此方法的前提是你的数据库由多个控制文件。

（1）启动数据库，发现 ORA-00205 号错误警告。

（2）查看日志文件

（3）此信息显示控制文件丢失。然后关掉例程。SHUTDOWN IMMEDIATE

（4）执行操作系统拷贝命令，把备份的控制文件拷到相应路径下。

（5）重新启动例程，发现启动成功 STARTUP

第二种方法是使用备份的 ASCII 控制文件，重建控制文件。如果你丢失了全部的控制文件时，可以考虑使用此方法。下面是用此方法进行重建的步骤：

（1）首先备份控制文件为 ASCII 控制文件。

（2）查看备份的追踪文件，遍及并执行，打开生成的跟踪文件

（3）一定要把头文件删除掉，否则无法执行。以符号“#”开头的行为提示文字，它不影响文件的执行。

（4）使用操作系统命令把控制文件移走，即模仿控制文件丢失。

（5）启动例程，发现错误 STARTUP

（6）关闭数据库。SHUTDOWN IMMEDIATE

（7）以 SYS 帐号登录，并执行已经编辑好的脚本。

（8）执行一些简单的查询，以验证数据库已经处于正常状态。

15.3 只读表空间的恢复

只读表空间就是只可以读取的表空间，即只能用 SELECT 语句来查询。使用 INSERT、UPDATE、DELETES 这些语句会显示错误，由于只读表空间处于只读状态，系统不能像更改其他表空间的数据文件那样更新只读表空间数据文件的文件头信息,所以从只读表空间恢复到只读表空间就比较容易。

恢复只读表空间比较简单，直接将从备份文件拷贝到目标路径，然后执行恢复就行了。下面试几个简单步骤：

(1) 查看表空间的状态

(2) 关闭数据库,然后使用操作系统命令把 USERS01。DBF 文件移走,模拟只读表空间的丢失,再启动例程

(3) 复制备份文件到目标路径。

(4) 再一次启动例程

(5) 查看表空间状态

至此,整个表空间的恢复就完成了。

15.4 本章小结

本章主要介绍了一些 Oracle 恢复机制的补充性方法,如并行恢复、控制文件的重建以及只读表空间的恢复等。并行恢复机制用来加速数据库恢复速度，只有使用并行恢复机制可以加速数据库的恢复速度。当你的系统得控制文件全部丢失后，要考虑进行控制文件的重建，重建控制文件要有相应的权限。只读表空间处于只读状态，系统不能像更改其他表空间的数据文件那样更新只读表空间数据文件的文件头信息,所以从只读表空间恢复到只读表空间就比较容易。当全部控制文件丢失后，才考虑使用控制文件重建的方法，只读表空间的恢复较容易。

第 16 章 Oracle数据库恢复管理器和待命服务器

本章的学习目标：

- 熟悉 RMAN 的用途
- 掌握恢复目录的创建和管理
- 掌握使用 RMAN 进行备份
- 掌握使用 RMAN 进行还原与恢复
- 熟悉待命数据库服务器的作用和概念
- 了解待命数据库服务器的配置

16.1 RMAN简介

RMAN 是 Oracle 数据库备份和恢复的主要管理工具之一，它可以方便快捷的对数据库实现备份和恢复，而其它还可以保存已经备份的信息以供查询。用户还可以不经过实际的还原即可检查已经备份的数据文件的可用性。你还可以通过用户图形界面和命令进行所有的备份工作。RMAN 恢复管理器的主要特点归纳如下：

- 可实现增量备份；
- 可对数据库表，控制文件，数据文件和归档日志文件进行备份；
- 可实现多线程备份；
- 可以存储备份信息；
- 可以检测备份是否可以成功还原。

可以在$ORACLE_HOME\BIN 路径下找到 RMAN 工具，也可以在操作系统命令下输入 RMAN 来运行。

16.1.1 Nocatalog 下连接RMAN

RMAN 允许不经过一个恢复目录数据库直接与目标数据库相连（即 Nocatalog 状态下）。目标数据库（Target Database）即所要执行备份和恢复的数据库。选择此方法，所有的目标数据库都将存储与目标数据库的控制文件中。初始化参数 INIT.ORA 决定了控制文件中目标数据库的恢复信息的有效时间，默认值为 7，即在 7 天内所要备份的数据库的信息有效。

在 DOS 操作符下，输入 RMAN HELP，可以查看 RMAN 的相关帮助。

使用 RMAN 连接目标数据库必须设置环境变量 Oracle_SID 为目标数据库。使用连接命令 CONNECT，必须使用 SYS 帐号或其它任何具有 DBA 权力的帐号。下面的例程中将启动 RMAN 工具，然后，输入登录名称和密码 connect target system/manager@mis

可以在输入 RMAN 命令提示下输入 HOST 命令查看操作系统版本信息。

16.1.2 创建恢复目录

恢复目录为可选项，如果选择选用恢复目录，它必须单独存放在另外一个数据库中，而其恢复目录数据库最好与目标数据库处于不同的服务器上。为了使用恢复目录功能，必须拥有 CONNECT，RESOURCE 和 RECOVER_CATALOG_OWNER 权限，用户必须使用 SYS 账号。

以下为建立一个恢复目录的过程。

1. 假设已经创建恢复目录数据库，设置环境变量为目标数据库。

2. 在 SQL*PLUS 里执行 CREATE USER 命令来创建 RMAN 用户，并授予其权限。不过，执行此命令必须要有一定的权限，我们可以使用账号 SYS 登录并创建用户 RMAN，密码为 RMAN，默认表空间为 USERS，默认临时表空间为 TEMP。

3. 为 RMAN 用户指派权限

4. 使用 RMAN 账号登录

5. 创建恢复目录

16.1.3 管理恢复目录

使用 RMAN 可以进行目标数据库的注册、验证注册结果、反注册数据库、重置恢复目录等操作。

1. 注册目标数据库命令
2. 验证注册结果
3. 反注册数据库
4. 重置恢复目录
5. 重置同步

任何 COPY，BACKUP，SWITH 或 RESTORE 命令都会引发这种同步。可是这种同步可能不是总包括重做日志开关信息和存档重做日志创建的信息。所以有必要定期对重做日志的信息进行更新恢复。两次重置同步之间的空隔根据备份频率不同而变化。如果每天执行一次数据库完全备份，备份中的一部分是存档日志，那么恢复目录也应该每天更新。这样的话，恢复目录就不会滞后于备份，并一直保持时间上的最近性。

16.1.4 LIST和REPORT命令

作为 DBA，日常工作中要经常对数据库进行备份，频繁的操作容易使人对所做过的事情产生混淆，很容易忘记昨天对哪个数据文件备份了，哪个没有，这种情况下，使用 LIST 和 REPORT 命令将非常有用。

1. 使用 LIST 命令可以显示已经备份过的表空间的信息

2. 使用 PERORT 命令可以查看执行数据库完全备份时所要备份的数据库结构。要执行此命令，必须首先连接到恢复管理器。

3. 输入 LIST INCARNATION OF DATABASE 命令。此命令显示通过 RMAN 备份过的数据库的信息及备份操作相关的历史信息。

16.1.5 生成存储恢复管理器语句

恢复管理器的命令语句可以存储到数据库当中，你可以象存储 PL/SQL 过程一样来存储它，当使用时可以直接调用，而无需重新编写。具体步骤如下：

1. 连接恢复目录
2. 创建语句
3. 命令生成后，将自动存储到数据库当中可以在 RMAN 下执行它

4. 验证已经备份的表空间是否还可用

16.1.6 操作系统命令备份

Oracle 备份管理器也可以用来管理存储那些操作系统备份的信息，在前面章节中曾介绍过的热备份，只进行表空间的备份是，在 SQL*PLUS 中执行 ALTER TABLESPACE BEGIN BACKUP 命令，随后使用操作系统命令把所要备份的表空间拷贝到另一路径。在备份管理器里，操作过程也是大致类似。

16.2 使用RMAN进行备份

使用 RMAN 对数据库备份与使用传统的操作命令和 SQL 命令备份有所不同。RMAN 使用备份集（Backup set）来存储备份数据。一个备份集文件可含有多个物理数据库文件，但同一个数据库文件不能分成若干部分分别存储在不同的备份集中。

16.2.1 备份的分类与实现

通过 RMAN 可以对数据库实现完全备份和增量备份

- 完全备份：完全备份就是对数据文件当中所有的数据块进行备份。
- 增量备份：就是在一个基准线备份的基础上实现增量备份。

为了实现增量备份，必须先进行一个 LEVER 0 的基准线备份，其操作步骤如下：

1.设置环境变量指向目标数据库，并使用 RMAN 帐号进行登录

2.进行 0 级增量备份，用来做为基准线备份。

3.备份结束后，进行一级增量备份

4.进行 2 级增量备份，此过程将备份从上次备份以来所有改变的数据块。如果没有 1 级备份，则将备份 0 级备份以来的所有改变的数据块。

16.2.2 备份操作的调整

首先是磁带读写装置处于最大的工作状态，因为在磁盘 I/O 和磁带体系当中，只有磁带写入装置会造成瓶颈问题。然后可以通过以下几种方法实现备份性能的改善：

- 改善同步和异步的 I/O 操作。
- 提高磁盘读出速度：设置 DISKRATIO 参数，当此参数设置为 40，你可以得到 4 倍的信息读出速度倒磁带装置，即假定每个磁盘读出速度为 1M/S，那么总的读出速度为 1M×4 = 4M/S。
- 采用并行通道指派：采用并行通道指派进行备份可以增加备份速度。

16.3 使用RMAN进行还原与恢复

使用 RMAN 进行还原（RESTORE）和恢复的方式将由用户对数据库、表空间、数据文件

等的备份的方式来决定。Oracle 的 RMAN 可以经由备份在磁盘和磁带装置上的数据来实现还原和恢复。

16.3.1 数据文件的恢复

数据库必须处于归档日志状态下，才能对数据文件进行恢复。但相应的表空间的恢复过程中必须处于脱机状态。

1. 设置环境变量 Oracle_SID 为你的目标数据库，在这里为 Oracle9i。
2. 启动 RMAN 并以 RMAN 身份登录恢复目录数据库。
3. 备份数据文件
4. 恢复数据文件
5. 打开数据库

16.3.2 表空间的恢复

表空间的恢复与数据文件的恢复大致相同，因为表空间是由一个或多个数据文件组成。

1. 设置环境变量 Oracle_SID 为你的目标数据库，在这里为 Oracle9i。
2. 启动 RMAN 并以 RMAN 身份登录恢复目录数据库。
3. 备份表空间
4. 恢复表空间
5. 打开数据库

16.3.3 非归档日志下数据库的还原

在非归档日志状态下，数据库无须恢复，你只需要把数据库还原即可。其操作步骤如下：

1. 确认数据库处于非归档日志状态。
2. 连接到目标数据库和 CATALOG 数据库
3. 备份数据库
4. 还原数据库
5. 打开数据库

从以上步骤可以看出来，数据库没有归档日志文件存在，也就是说当你执行完数据库还原，所有的从上次备份以来所有数据库的更改的数据将会全部丢失。

16.4 Oracle服务器的备用数据库（Standby Database）

不同类型服务器的应用对数据的依赖程度也不相同,在一些银行或者电信部门里一般对数据库服务器系统的长时间稳定运行要求很高，如果仅仅使用一台大型服务器，即使服务器制造商的技术很高，也难免出现系统服务中断甚至系统崩溃的情况，为了避免服务中断，我们可以采用 Oracle 服务器的备用数据库（Standby Database）。

16.4.1 考虑使用Standby Database

Oracle 服务器的备用数据库主要用在实现高可靠性（High Available）的环境，通常状况下为 24×7 的环境，

要使用 Oracle 服务器的备用数据库，必须有一个主数据库。这个数据库有 4 个待命数据库来共同实现高可靠性。在 oracle 服务器的备用数据库中，待命数据库一直处于恢复状态，它是通过不断的接收来自主服务器的归档日志文件来实现的，当备用数据库当中的主服务器因故障停止工作时，待命服务器可以立即接管主服务器的工作，而用户看来好像没有任何事件发生一样。

为了防止像断电、水灾、火灾等事故造成的停机现象，待命服务器可以和主服务器处于不同的物理位置，它们之间可以通过高速网络连接。

16.4.2 初始化参数的配置

要配置待命数据库服务器，主数据库和待命数据库的大部分参数应相同，有些必须不同，下面是一些不同的参数：

CONTROL_FILES:此参数用来区分主数据库与待命数据库的控制文件。

STANDBY_ARCHIVE_DEST:此参数用来在远程文件系统即 RSS，用来决定待命数据库服务器的归档日志所存放的路径。

LOG_FILE_NAME_CONVERT:此参数用来区分主数据库和待命数据库的重做日志文件的命名方式。

DB_FILE_NAME_CONVERT:此参数用来区分主数据库和待命数据库的数据文件的命名方式。

下面两个参数在主数据库和待命数据库中必须相同：

DB_FILES:此参数用来指定数据库的最大文件个数。

COMPATIBALE:此参数用来决定数据库服务器的兼容程度，如果此参数不同，主数据库可能不能和待命数据库一致地工作。

16.4.3 创建待命数据库

创建待命数据库即创建一个与主数据库相匹配的数据库。

下面是创建待命数据库的步骤：

1. 复制主数据库当中的 INIT.ORA 文件，并作相应的编辑（编辑内容如上一节所示）。
2. 确认主数据库中的数据文件（可通过查询视图 V$DATAFILE 来得到相关信息）。
3. 把主数据库关闭，然后进行完全备份。
4. 启动主数据库，并生成待命数据库控制文件。
5. 把所有联机日志文件归档。
6. 把所有必要的物理文件传输到待命数据库机上。
7. 使待命数据库处于 EXCLUSIVE 状态，且启动时使用 STARTUP MOUNT 命令。
8. 把主数据库中的最后一次归档日志文件传输到待命数据库当中。
9. 激活待命数据库。
10. 关闭待命数据库，并执行一个完全备份。
11. 使用 STARTUP MOUNT 命令启动待命数据库，然后使他处于 READ WRITE 状态。

12.配置主服务器当中的初始化文件当中的参数。
13.配置主数据库名的 TNSNAMES.ORA 文件。
14.配置待命数据库中的监听文件。
15.启动待命数据库到 NOMOUNT 状态。
16.使待命数据库处于 MOUNT 状态。
17.使待命数据库处于自动恢复状态。
至此，待命数据库的配置完毕了。

16.5 本章小结

本章主要介绍了恢复目录的创建，使用 RMAN 管理恢复目录，使用操作系统命令备份，使用 RMAN 对数据库进行还原与恢复，数据文件的恢复以及非归档日志下数据库的还原等。学习完本章，读者应该熟悉 RMAN 工具的作用，并能熟练使用 RMAN 对数据库进行一些常用操作以及 ORACLE 的待命数据库的创建方法。

第IV部分　　性能调整

第 17 章 性能调整概要

本章的学习目标：

- 熟悉调整目标与计划的制定
- 熟悉调整内容和调整工具的使用

17.1 调整目标与计划的制定

一些系统由于以下几种原因可能造成系统运行性能下降、反应过慢等令人不满意的状况出现：

- 数据库设计的缺陷
- 数据库尺寸和用户量的增长
- 商业需求对数据库结构及功能的变化

做任何工作都要制定一个工作目标，Oracle 数据库的性能调整也不例外。如果对一个系统进行没有目标的调整，很可能把这个系统搞得一塌糊涂，甚至崩溃。衡量系统性能的指标如下：

吞吐量：表示在单位时间内系统完成的工作量，通常以每秒钟所进行的事务处理量即 TPS 来表示。当然，吞吐量越大越好。

反应时间：即系统对用户的一个请求的反应时间。此反应时间为从接到用户命令到返回用户结果所需的时间。通常以毫秒为单位表示，反应时间越小越好。

在任何一个系统中，以吞吐量和反应时间作为调整目标是互相对立的。在系统可用资源不变的情况下，通常增加系统的吞吐量会造成反应时间的延长。同时，如果想减少反应时间，系统的吞吐量必然受到限制。这一点可以通过浏览某些网站就可以得到证实。当有一些爆炸性新闻出现时，浏览某些新闻网站，会发现网页打开速度明显降低，这是因为大量用户访问的结果，而网站的服务器资源的数量并没有减少。

17.2 调整内容

制定好调整目标以后，接下来就是要确定调整内容，Oracle 主要调整地内容有以下几个方面：

- 数据库的安装与设置
- 应用程序的调整
- 数据库数据访问方法与路径
- 调整内存
- 磁盘 I/O 的调整
- 系统资源的利用与分配
- 调整资源竞争

这些调整内容的顺序虽然没有固定，但我们还是建议按照以上的步骤对系统进行调整。假如数据库的安装与配置不恰当，将很难通过内存和其他方式提高系统的性能。

17.3 常用调整工具

1. Oracle 调整专家

Oracle 调整专家（Oracle Expert）是一个基于图形化的性能调整工具，它是 Oracle 企业管理器性能调整工具包中的一个重要组成部分，Oracle 调整专家为你指定了调整方向。Oracle 调整专家最终调整结果将生成建议脚本文件，检查此文件后，即可直接在 SQL*PLUS 当中执行以实现对性能的调整。

2. Oracle 性能管理器

Oracle 性能管理器（Performance Manager）是一个基于图形化的性能监视器，这个工具提供给数据库系统管理员实时性能监控和记录性能特性作为日后分析的依据。而该工具提供的一系列图表形式的预定义的度量规则可以让数据库系统管理员开始使用这个工具，而不需要预先的开发。如果性能管理器提供的标准图表不能满足要求，数据库系统管理员可以添加新的度量以满足特定的需求。

性能管理器拥有内嵌的图表用于收集和分析竞争。全局数据库统计、I/O、工作负载级别、内存和并行服务器等信息。使用一条 SQL 语句所收集的任何信息都是特定的分析和调整所需要的新图表的基础。

性能管理器虽然是一个综合的工具，但是易于使用。它能够很快创建以灵活的图解细目显示的性能信息集合。

3. Oracle 索引调整导航向导

Oracle 索引调整向导（Oracle Index Tuning Wizard）可以建议哪些地方该使用索引，它是基于 Oracle 调整专家所提供的建议，通过分析已处理过的 SQL 语句来搜集索引建立的信息，并提供建立索引的最佳方案。

4. 使用 REPORT.TXT 文件

Oracle 提供了一些常用的性能调整脚本，这些脚本位于$Oracle_HOME\RDBMS\ADMIN 目录下。主要包括：

- UTLBSTAT.SQL/UTLESTAT.SQL
- UTLCHAIN.SQL
- UTLXPLAN.SQL
- DBMSPOOL.SQL

注意：

运行 UTLBSTAT.SQL 脚本必须在数据库已经处于运行状态一定的时间后进行，这样采集到的统计信息才较有价值,Oracle 建议至少系统开启运作 30 分钟后再运行它,而且运行脚本 UTLESTAT.SQL 必须和运行脚本 UTLBSTAT.SQL 有一定的时间间隔,这样搜集的统计信息才较有价值。

在后面的调整部分当中会经常碰到 REPORT.TXT 文件。

17.4 本章小结

本章主要介绍了 Oracle 数据库性能调整的概念、调整目标、计划的制定、调整内容以及常用调整工具。学习完本章之后，读者应该懂得在什么情况下需要进行数据库的性能调整，以及如何调整，使用什么工具来监测性能等。在进行任何具体的性能调整之前应该制定详细的调整目标，这样才能保证把数据库服务器调整好。

第 18 章 Oracle内容调整

本章的学习目标：

- 熟悉 Oracle 内存调整的一些基本概念
- 掌握内存性能测试的基本方法
- 掌握内存优化的方法

18.1 共享存储器调整

18.1.1 调整库高速缓存与数据字典高速缓存

库高速缓存用来存储已经提交给 Oracle 的 SQL 语句、分析过的格式和执行计划，以及被执行过的 PL/SQL 包头与过程 JAVA 的类也被存储到这里。

使用库高速缓存可以提高 SQL 语句的执行效率。当一条 SQL 语句由应用程序发出时，服务器首先查找高速缓存，查看相同的语句是否已经被使用过，如果有，Oracle 将从相应的内容中直接读取已经分析的结果，这样就减少了重新执行这些语句所耗费的资源与实践。库高速缓存主要包括共享 SQL 区和私有 SQL 区。

数据字典高速缓存用于存储分析 SQL 语句的数据字典行。

注意：

这里所说的被使用过，是指先前执行的 SQL 语句与本次执行的语句完全相同，即包括字母大小相同。

18.1.2 共享存储区的“命中率”

共享存储区的“命中率”（Hit Ratio）可以解释为库高速缓存命中率和（所要得到的/所有执行的）×100%。它又分为库高速缓存命中率和数据字典高速缓存命中率。下面将分别加以介绍。

1. 库高速缓存命中率

对于几乎所有的高速缓存装置来说，保证高校性能的准则是 90%+命中率。命中率可以被定义为 100×（1.00 页号-（遗漏的数量/请求的数量）），其中请求的数量=遗漏的数量+命中的数量。高速缓存通常由一个最近最少使用的（LRU）算法来管理，该算法保证在任何给定的时刻，最近使用最多（MRU）的数据被保存在高速缓存中，而最近最少使用的数据被拿走。

当 Oracle 对一条 SQL 语句进行语法分析时，通过把一个数学公式应用到 SQL 语句中并使用该结果在高速缓存中存储（供以后查找和利用）它，Oracle 便可以在库高速缓存中为该 SQL 语句分配一个 SQL 区。换句话说，它使用一个哈希函数（HASH FUNCTION）。为了一条语句可以被重复使用，该语句必须是相同的。

你可以查看视图 V$LIBRARYCACHE 来查看相关信息。

描述库高速缓存性能主要有以下几个参数：

- GETHITRATIO
- PINHITRATIO

- RELOADS
- INVALIDATIONS

在库缓存性能调整过程中，Oracle 有几点建议：

- 在联机事务处理系统（OnLine Transaction Processes）中 GETHITRATIO 的值应大于 95%，PRINHITRATION 的值应大于 90%。
- 在决策支持系统（Decision Support System）中，根据情况而定，一般命中率较低。

也可以通过查看 REPORT.TXT 来查看上述信息。

2. 数据字典高速缓存命中率

在描述资料字典缓存效率的时候我们也同样使用命中率。它表示应用程序在内存中查找资料字典信息，而不是在磁盘读取。它可表示为 1-（sum（getmisses）/SUM（gets））。此值也是越大越好。

查询视图 V$ROWCHCHE，你可以得到数据字典高速缓存各部分的的命中率，通过一些 SQL 语言函数，我们可以得到数据字典高速缓存命中率。

我们建议此值大于 85%

18.1.3 提高共享存储区的性能

提高共享存储区最简单的方法就是增加共享存储区的容量，此方法也最有效。它的大小可以通过查看初始化参数 SHARE_POOL_SIZE 得到。如果配置中使用了 JAVA 类，则要首先配置系统初始化参数 JAVA_POOL_SIZE。在 Oracle9i 当中，JAVA_POOL_SIZE 从共享存储区中分离出来，并作为系统全局区（SGA）中的一个独立部分。

1. 把常用的代码保存到内存中
2. 为大型 PL/SQL 语句创建一个独立的空间
3. 代码重用——变量的绑定
4. 生成一个大存储区（LARGE POOL）

18.2 数据库缓冲调整

18.2.1 存取区缓存管理机制

和我们前面介绍一样，Oracle 管理高速缓冲区的机制有两种：

- 写列表——用来管理那些已经修改但还没有写到磁盘的内存。
- LRU 机制——它是一些数据块的排序，最近被使用的数据块将放置在最前面，最少被使用的数据块将放置在最后面，当新表进入并被加到列表时，最少使用的数据块列表就被清除。

当 Oracle 进程访问缓存时，它将被移到 MRU 端，系统经常别访问的资料被存储到缓存当中。

正如我们之前所说的，在读一个数据块到缓存之前，系统进程必须事先确定有自由空间存在。

主要有 3 种类型的缓存。

- “脏”缓存（Dirty）

- “空闲”缓存（Free）
- “钉住的”缓存（Pinned）

数据库高速缓冲区的大小由初始化参数 DB_CACHE_SIZE 来决定，这是 Oracle9i 与以前版本所不同的地方。在 Oracle 8i 以及以前的版本中则是由 DB_BLOCK_SIZE 和 DB_BLOCK_BUFFERS 来决定。这两者的乘积即为它的实际大小。

但是当整表扫描（FTS 即 FULL TABLE SCAN）事件发生时，存储表的内存部分被立即放到 LRU 末端，这样很快就被移出 LRU 表。这种机制保护了其他缓冲区的资料不被移出内存。

18.2.2 测量高速缓冲区的性能

查询视图 V$SYSSTAT，它包含以下几个计算行参数：

- PHYSICAL READS：从这次例程启动后，数据高速缓冲区从磁盘读取的数据块总个数。
- DBBLOCK GETS：用户发出资料查询，如果这些资料已经存储在缓冲区，我们就称它为 GET。
- CONSISTENT GETS：为了维护数据库读的一致性，原来的数据块信息将经过快照机制拷贝到回滚段。为了提高性能，这部分资料也被放到高速缓存中。我们把这段回滚段缓存里面读取的数据块的总个数称之为 CONSISTENT GETS。

使用视图 V$SYSSTAT 查询高速缓存命中率

查询视图 V$SESS_IO 和 V$SESSION 得到相同的结果。

也可以通过 REPORT.TXT 来的到以上信息。

也可以使用 Oracle Performance Manager 来查看数据库高速缓冲区的性能情况。

18.2.3 提高缓冲区的性能

当然提高缓冲区性能最直接、最有效的方法使增大内存容量。提高他们的性能就是提高它的命中率，可分为 4 个方面：

1. 为高速缓冲区分区

默认状态下，所有的数据库块信息可以平等地使用数据库缓冲区。在特定情况下，少量用户会频繁地访问那些以前不经常访问的表，这样就造成一个短期的访问峰值，这样 Oracle 的 RLU 机制很可能将那些平时经常访问的表挤出数据库高速缓存，当其他用户在访问这些被挤出的表时，Oracle 将重新从硬盘中读取大量信息，从而造成性能下降。

避免此情况发生的最好方法就是为高速缓冲区分区。通常将其分为 3 部分：

- KEEP POOL：用来放置那些最经常访问的段（SEGMENT）信息。
- RECYCLE POOL：用来放置那些最不经常访问的段信息。
- DEFAULT POOL：用来放置那些上述两种以外段的信息。

除非用户在系统启动参数里指定，默认状态只有 DEFAULLT POOL 产生。

通过联合查询视图 V$CACHE 和 DBA_USERS 可以查询哪些段应该存储到哪个 POOL 里面。

V$CACHE 视图可以由脚本 CATPARR.SQL 生成。要生成此视图，必须以 SYS 身份登录，在 Oracle 9i 中，在 UNIX 上可以通过以下路径找到此脚本：$Oracle_HOME/rdbms/admin；在 Windows NT 上可以通过以下路径找到此脚本：$Oracle_HOME\rdbms\admin。联合查询视图 V$BH 和 DBA_SEGMENTS 也可以得到相同的信息。

下面讲述以下这 3 个区的初始化参数设置规则：

- DB_BLOCK_BUFFERS：在例程启动时，数据库高速缓存的总缓存数目。

- DB_BLOCK_LRU_LATCHES：例程启动时，数据高速缓存的总闩（LATCH）个数。
- BUFFER_POOL_KEEP ： 分 配 给 KEEP POOL 的 DB_BLOCK_BUFFERS 、DB_BLOCK_LRU_LATCHES 的个数。
- BUFFER_POOL_RECYCLE ： 分 配 给 RECYCLE POOL 的 DB_BLOCK_BUFFERS 、DB_BLOCK_LRU_LATCHES 的个数。

LATCH 用来控制对共享结构的存取，它主要用来保护 Oracle 的共享内存结构不被外部程序因内存竞争而交换出内存。以上几个参数的关系如下：

- 每个 LRU 闩至少管理 50 个高速缓存缓冲区。
- 每个 BUFFER POOL 至少分配一个 LRU 闩。
- 指派给 KEEP POOL 和 RECYCLE POOL 的缓冲区的总数应不大于 DB_BLOCK_BUFFERS 的值。
- 指派给 KEEP POOL 和 RECYCLE POOL 的闩的个数应小于 DB_BLOCK_LRU_LATCHES 的值。

在为高速缓冲区分区以后，接下来你可以把段分配给各缓冲区。

- 把 t_emp 表分配给 KEEP POOL
  ALTER TABLE t_emp STORAGE(BUFFER_POOL KEEP);
- 把 t_dep 表分配给 RECYCLE POOL
  ALTER TABLE t_dep STORAGE(BUFFER_POOL RECYCLE);

通过查询视图 DBA_SEGMENTS 可以查看每个段所在的缓冲区类别（KEEP POOL、RECYCLE POOL、DEFAULT POOL）。

通过查询视图 V$BUFFER_POOL、V$BUFFER_POOL_STATISTICS 可以查看各类缓冲区的信息。后者主要是一些关于各分类缓冲区的命中率的统计。

2. 存储表格

不管缓冲区有多大，当发生 FTS（Full Table Scan，整表扫描）时，表的相应的块信息立即被放到 LRU 列表的最末端，因此很快被移出内存。当用户下次再进行整表扫描时，系统再次读取整表，然后其数据块信息立即被放到 LRU 列表的最末端，很快被移出内存。如果表很小，但是被高频率的访问，也就是说发生 FTS 频率会很高，它被移出的次数也就比较多。这样会造成系统资源的大量消耗。为了避免此种情况发生，可以把一些经常访问的小型表，存放到内存当中，这样表的数据信息就永远不会被交换出内存。从而提高了查询的速度。

18.3 重做日志缓冲区的调整

重做日志缓冲区用于在内存中存储已经被修改的数据库信息。它是被循环使用的，也就是说 LGWR 进程重做日志缓冲区从始端到末端，然后再返回缓冲区的始端。当整个重做日志缓冲区被填满时，将它的全部内容写入联机重做日志文件中。

重做日志缓冲区的大小由 LOG_BUFFER 初始化参数决定。

18.3.1 测试日志缓冲区的性能

可以通过查询相关视图来获取重做日志缓冲区性能的相关信息。如视图 V$SYSSTAT V$SESSION_WAIT、V$SYATEM_EVENT。REPORT.TXT 文件也可以用来获取重做日志缓冲区性能的相关信息。下面为视图 V$SYATEM_EVENT 的结构：

DESC V$SYSTEM_EVENT;

18.3.2 提高日志缓冲区的性能

同样道理，提高日志缓冲区性能的最好方法就是增大内存的容量。通过设置系统初始化参数 LOG_BUFFER 来指定其大小。

1. 通过提高检查点的效率来实现

检查点一旦发生，它就必须成功地中止，否则它会唤醒 DBWO 和 LGRW 清空数据库高速缓村和重做日志缓存，造成不必要的 I/O 使用。

用户可以设置 LOG_CHECKPOINT_TIMEOUT 和 LOG_CHECK_POINT_INTERVAL。

- LOG_CHECK_POINT_INTERVAL 参数用来指定重做日志文件每写多少操作系统块发生一次检查点。
- LOG_CHECKPOINT_TIMEOUT 用来指定每隔多少时间发生一次检查点。如果此参数设置为 0，即为忽略此参数。

2. 减少重做日志信息的生成

通过使用 NOLOGGING 功能可以忽略重做日志信息的生成，可以在表生成时指定它。

也可以使用 ALTER TABLE 命令来实现。

```
ALTER TABLE  t_emp NOLOGGING;
```

取消 NOLOGGING 可以使用以下命令：

```
ALTER TABLE t_emp LOGGING;
```

使用 CREATE TABLE AS SELECT、CREATE INDEX、ALTER INDEX…REBUILD 等 DDL 语句是，也可以使用 NOLOGGING 功能项。除非指定 NOLOGGING 功能，表在默认状态下是产生日志信息的，即为 LOGGING 状态。

以下两种情况例外：

- 使用 SQL*LOADER 进行直接装在（DIRECT PATH）。
- 使用/*+DIRECT*/附加项直接插入数据。

在这两种情况下不会产生日志信息。

18.4 本章小结

本章主要介绍了 Oracle 内存调整的内容、方法和步骤以及如何测量其性能，调整内容主要分为共享存储区调整、高速缓存调整和重做日志缓冲调整。针对每种内存结构 Oracle 提供了不同的调整方法和调整步骤。要记住，最为简单直接的方法就是增加内存容量。

第 19 章 结构查询语句与应用程序设计调整

本章的学习目标：

- 熟悉结构查询语句优化的概念
- 掌握结构查询语句的优化方法
- 掌握优化应用程序设计的两种方法

19.1 TKPROF工具

不作介绍。

19.2 解释计划

Oracle 的解释计划(Explain Plan)工具可以被用来查看一个具体 SQL 语句的执行过程。通过它可以知道哪些 SQL 语句消耗更少的系统资源，哪些 SQL 语句系统执行时间最少。

1. 使用 EXPLAIN PLAN FOR 命令生成解释计划

在系统运行过程中或是在系统开发的调试阶段，你可能发现某些 SQL 语句执行效率非常低，此时你可以使用 EXPLAIN PLAN FOR 命令来生成此语句的解释计划，下面便是生成过程：

（1）创建一个名为 PLAN_TABLE 的表，要想创建它，执行脚本 utlxplan.sql。

（2）执行 EXPLAIN PLAN FOR 命令

（3）EXPLAIN PLAN FOR 命令只是更新解释计划表 PLAN_TABLE，而实际的查询并没有执行，即只生成了执行查询的执行过程，你可以查询表 PLAN_TABLE 的内容。

2. 解释 EXPLAIN PLAN 的输出

19.3 使用AUTOTRACE工具选项

AUTOTRACE 工具选项可以实现 EXPLAIN PLAN 的功能。

使用 AUTOTRACE 工具选项前必须做一些准备工作：

（1）必须像使用 EXPLAIN PLAN 工具那样先生成 PLAN_TABLE 表，生成此表后，在当前用户下执行脚本 utlxplan.sql，此脚本在 Oracle_HOME\ora90\rdbms\admin 目录下；UNIX 操作系统上为 Oracle_HOME/ora90/rdbms/admin 执行此脚本。与 EXPLAIN PLAN 不同的是 AUTOTRACE 每次都会自动清除上次记录。

(2)生成一个名为 PLUSTRACE 的数据库角色，使用 plustrce.sql 脚本可以实现此操作。

（3）接下来，把此角色授予需要使用 AUTOTRACE 工具选项的用户。

下面介绍 AUTOTRACE 的使用。

AUTOTRACE 工具选项只可以在 SQLPLUS 窗口中使用。为了使用此项功能，你必须使用 SET AUTOTRACE ON 命令使 SQLPLUS 窗口处于自动跟踪状态。

如果你已经设置 AUTOTRACE 状态，想使系统返回非 AUTOTRACE 状态，可使用命令 SET AUTOTRACE OFF

验证设置结果使用 SHOW AUTOTRACE 命令

设置 AUTOTRACE 选项时，还有一些附加的选项功能：

- ON EXPLAIN

显示查询结果和解释计划（EXPLAIN PLAN 输出信息），但不进行统计；

- TRACEONLY STATISTICS

与上一个正好相反，只进行查询统计，但不显示查询结果和解释计划（EXPLAIN PLAN 输出信息）

- TRACEONLY

只显示解释计划和统计结果，不返回查询结果；

19.4 理解Oracle的最佳性能

在 Oracle 的优化机制里面，有两种模式，即规则（Rule）和代价(Cost)。

1.基于规则的优化

基于规则的优化（Rule-Based Optimization），使用一系列预定义的规则来决定哪一个是最佳方案。

2.基于系统消耗资源的优化

不像 RBO 那样只使用一些预定义的执行方式来执行查询,CBO(Cost-Based Optimization)会考虑不同的执行计划，然后选出系统消耗资源最低的那一种。系统资源的消耗主要是 CPU 和 I/O。为了决定哪一种执行计划消耗资源最少，CBO 使用了大量对表格和索引的当前统计，这些统计存储到数据字典里面，主要包括：

- 表和索引的大小
- 表和索引的行数
- 每个表或索引的使用数据块的个数
- 每个表的行的长度
- 索引列数据集的势(Cardinality)的多少

默认状态下，数据库中不存在以上的统计。只有你对表和索引进行分析时，这些统计信息才存在于数据库中。

可以使用下列命令来分析一个表或索引，从而得到统计信息。

此命令收集 west16 表中的每一行统计信息,它还会收集与此表相关的索引的统计信息。如果此表非常大，这种方法将花费较长的时间和较多的系统资源来分析此表。

使用 ESTIMATE 只分析表格当中的一部分行数据，而不是像 COMPUTE 那样分析整个表。ESTIMATE 的默认统计行数为 1064 行。多数情况下使用 ESTIMATE 分析的精度和 COMPUTE 的大致相当。

也可以指定分析的百分率或指定分析的行数。

使用命令可以删除已经分析的统计。

19.5 设置优化模式

现在我们已经明白了 RBO 和 CBO 原理及差别，接下来我们将介绍如何设置优化模式(Optimizer Mode)。优化模式可以设置为三个级别：

- 例程级(Instance Level)
- 会话级(Session Level)
- 语句级(Statement Level)

19.5.1 例程级优化模式

例程级的优化模式可以通过初始化参数 OPTIMIZER_MODE 来设定。此参数位于 INIT.ORA 文件当中，一旦设定，此例程当中所有的会话都工作在所设定的模式当中。此参数可以设置为以下几个值：

- RULE
- CHOOSE
- FIRST_ROWS
- ALL_ROWS

1.RULE 模式

当数据库当中不存在统计信息时，将使用 RULE 模式。要使用此模式，只需设置 INIT.ORA 文件的参数 OPTIMIZER_MODE=TRUE。

2.CHOOSE 模式

当数据库中统计信息存在时，将使用 CHOOSE 模式。要使用此模式，只需设置 INIT.ORA 文件的参数 OPTIMIZER_MODE=CHOOSE。如果你已设置了 OPTIMIZER_MODE=CHOOSE，但是数据库中不存在表或索引的统计信息时，例程将自动使用 RBO（RULE）模式来执行查询。默认状况下，系统运行在 CHOOSE 模式下。

3.FIRST_ROWS 模式

要使用此模式，只需设置 INIT.ORA 文件的参数 OPTIMIZER_MODE=FIRST_ROWS。在这种模式下，如果统计信息存在，例程使用 CBO 模式。如果统计信息不存在，将使用 RBO 模式。此模式用来缩短系统查询的响应时间。

4.ALL_ROWS 模式

要使用此模式，只需设置 INIT.ORA 文件的参数 OPTIMIZER_MODE=ALL_ROWS。在这种模式下，如果统计信息存在，例程使用 CBO 模式。如果统计信息不存在，将使用 RBO 模式。此模式用来提高查询的吞吐量。默认状态下，ALL_ROWS 使用 CBO 模式。

19.5.2 会话级优化模式

如果例程级的优化模式已经设定，所有连接到此例程的用户都在默认的优化模式状态下进行查询操作。如果用户想改变他的优化模式，可以使用命令 ALTER SESSION。

和例程级一样，会话级也有 RULE，CHOOSE，FIRST_ROWS，ALL_ROWS 四种模式。并且其使用规则也类似。此模式将不会改变，直到用户发出重置会话模式的 ALTER SESSION 命令或退出会话。此会话级设置方式优先于例程级。

19.5.3 语句级优化模式

语句级优化模式可以通过/*+和*/来设置

同样的，和例程级一样，语句级也有 RULE，CHOOSE，FIRST_ROWS，ALL_ROWS 四种模式。此语句级设置方式优先于会话级和例程级。

19.6 应用程序的性能

收集和分析应用程序信息只是理解提高应用程序性能的第一步。下面我们将介绍一下提高应用程序性能的各个选项。这些选项可以分为两类，即提高执行路径的效率和最小化 I/O 操作。执行路径效率的提高可以通过存储优化计划(Store Optimal Plan)和使用物化视图(Using Materialized Views)。最小化 I/O 的操作可以通过使用索引和聚簇(Indexes and Clusters)来实现。

1.执行计划稳定

对数据库的许多不经意的动作都可以造成对执行计划的影响。Oracle 9i 允许通过指定执行计划的稳定性（Plan Stability）来标址（即 ADDRESSING，非标志）此问题。稳定性计划通过在数据字典中存储一些首选的执行计划来实现。

存储概要（Stored Outlines）

Oracle 数据库以存储概要的形式把那些预定义计划保存在数据字典中，这些存储概要表示了优化的执行计划，即执行 SQL 语句时，与此存储概要相关的概要文件将被执行。这里所说的 SQL 语句必须与存储概要完全相同，包括大小写等。

2.物化视图

存储概要通过优化器如何对一个具体的 SQL 语句使用哪种执行方案来加速查询。物化视图通过存储实际的数据来加速查询。这些数据可以通过预联合(Pre-joined)和预概括（Pre-summarized）的查询得到。物化视图主要用来构建数据仓库和决策支持系统。在这些系统当中，通常要访问十分庞大的数据。当正确使用物化视图时，和存储概要一样，优化器可以动态地调整对相关视图的查询。但是不同的是，一个查询并不一定要完全与生成物化视图的 SQL 语句相同。事实上，优化器可以动态地重写事先定义的那个查询为另一个相近的查询。这样，物化视图就可以应用在任何的表格。

19.6.2 索引与聚簇来最小化I/O

存储概要和物化视图可以通过优化执行路径来提高查询性能。下面我们将介绍另一种加速执行查询的方法，即建立索引与聚簇。

1.索引

正确建立索引可能是DBA最有效的工具。这是因为索引可以大大降低不必要的I/O操作。你可以建立四种索引：B 树索引、位图索引、反键索引和索引组织表格（IOT）

- 平衡树索引（B-Tree Index）

平衡树，又叫 B 树，是现代关系型数据库中最普通的索引。B 树索引用由底向上的顺序来对表当中的列数据进行排序，B 树索引不但存储了相应列的数据，它还存储了 ROWID，这个 ROWID 用来标志表中相应行的剩余数据。索引以树形结构的形式来存储这些值。

通过解释计划或者检查跟踪文件后，确定在 emp_id 列上建立一个 B 树索引来提高查询性能。

当经由索引查询并通过 ROWID 访问相应表的数据时，整表扫描操作将可以避免，同时查询性能也得到了提高。特别是当执行查询只返回很少量的数据时，使用 B 树索引更为有效。Oracle 建议当查询返回的数据行数小于表格总行数的 5%时，考虑使用 B 数索引。还必须考虑一个情况，那就是必须有高的集的势(Cardinality)。

当表的数据被删除时，与其相对应的索引数据也被删除，因此频繁地对表格执行删除和插入操作会造成索引性能的下降，即索引的存储结构可以分为很多级。当 Oracle 服务器从

根部或者起始数据块遍历索引数据块，花费的时间也相应变长，因此造成了性能的下降。可以通过查询 DBA_INDEXES 来查看索引信息。

B 树索引为默认方式，索引用平衡树的算法来生成。平衡树包含索引列的节点和行的 ID，ID 是用来区分表的行信息的，下面是平衡树索引的几种类型：非惟一索引；惟一索引；反键索引。

注意：

*LEAF BLOCKS 是平衡树当中最低级的块。

- 位图索引

位图索引并不重复存取索引列的值，每一个值被看作一个键，相应的行的 ID 置为一个位（BIT），位图适合于仅有几个固定值的列，不能用位图索引来生成具有惟一性的索引和反键索引（UNIQUE KEY，REVERSE KEY）。

- 反键索引

反键索引是 B 树索引的一种特殊类型。如果某个表的某一列数据是一些序列号，那么在此列上建立 RKI 是比较有效的方法。一般的 B 树索引包括列数据，这样索引就会分成许多级。既然超过四级的 B 树索引会造成性能的下降，这种情况下使用 RKI 将更适合。

反键索引通过把索引列的数据取反，然后把这些新数据存储起来，从而来解决以上问题。

- 索引组织表格（IOT）

不像 B 树索引那样，通过存储列数据然后直接指向表中相应行的 ROWID，索引组织表格把数据完全存到索引当中。因此，读取索引然后再读取表所造成的额外 I/O 消耗被消除。数据存储在 IOT 当中的一个好处就是数据也就是存储 B 树结构里，如果你经常通过一个主键进入表的话，使用 IOT 返回行数据的速度远高于普通表的速度。

IOT 是一个表的排序子集，他本身就是一个表，也是一个索引。有一点要记住，IOT 没有自己的 ROWID，这是它和其他普通表一个明显区别的地方。也正因为此，在 IOT 表上是不能另外建立索引的。这一点可以从索引的原理得到。因为在 IOT 中索引和表数据存储到一起，所以它适合于经常通过主键查询的数据表。一个 IOT 为一个平衡树索引，通过组建 IOT 可以提高系统的性能。

2.聚簇

索引通过建立直接确定查询行的精确地址的方法以避免整表扫描，从而提高了查询性能。避免整表扫描，即降低 I/O 操作。降低 I/O 操作的另一种方法是使用聚簇。

聚簇就是把一个或多个数据存储在同一数据块中的表。通过这种方式查询一系列的行只需要读取一个数据块而不是两个。在 Oracle 8i 当中有两种聚簇可以使用，即索引聚簇和 HASH 聚簇。

- 索引聚簇

索引聚簇用来存储一个或多个表的数据到同一些物理数据块当中。使用索引聚簇还有一些限制，通常聚簇的表应该遵循下列原则：

这些表格必须经常地被联合查询，而不是只查询其中的一个；

当这些索引聚簇生成以后，很少或没有 DML 操作；

每个父键必须有大致相同的子记录数目。

- HASH 聚簇

不像检查普通索引那样，当返回一个行时，HASH 聚簇不经过索引，而是使用一个 HASH 算法来直接决定返回行数据被存储的位置，因为聚簇本身存储全部数据。也就是说 HASH 聚簇就是把列值作为输入，然后使用一个专用的 HASH 函数计算该行的物理地址（实际是 ROWID）作为输出。

19.7 OLTP和DSS系统的性能调整要求

在当前使用的数据库系统中主要有两种类型，即联机事务处理系统(OnLine Transaction Processing System)和决策支持系统(Decision Support System)。

1.OLTP 系统

联机事务处理系统通常有大量的用户执行简短的 DML 事务操作，像订单操作、订单目录管理、日程安排等都是联机事务处理的较典型例子。联机事务处理的速度一般比较快，用户最关心的是吞吐量（所支持的最大同时联机用户数目），下订单的总共时间，添加或删除订单目录的反应速度等。

OLTP 系统需要足够多的 B 树索引来提高性能。但像 INSERT，UPDATE，DELETE 这样的操作造成的性能下降比较少。表和索引的统计通常使用 CBO 模式。

2.DSS 系统

决策支持系统和数据仓库可以表示为另一种性能调整方法。这些系统通常执行很少的 INSERTA，UPDATE，DELETE 操作。这些系统用户最关心的是响应时间，即从查询执行到返回结果的时间。DSS 将使用非常庞大的整表扫描，所以正确地使用索引聚簇和 HASH 聚簇是非常重要的。索引组织表格也通常被用在比较庞大的 DSS 系统当中。位图索引在比较低的集的情况下也经常考虑。数据库块的大小和排序以及 Oracle 并行查询功能也必须考虑在内。

19.8 本章小结

本章主要介绍了 SQL 语句调整以及应用程序设计对性能的影响，调整 SQL 语句可以使用 EXPLAIN PLAN 解释计划也可以使用跟踪（TRACE）功能。在使用这两项功能前必须创建 PLAN_TABLE 表。在调整应用程序设计过程中，你可以通过提高执行路径的效率和最小化 I/O 使用的方法来实现，提高执行路径的效率有两种方法，即执行计划稳定和建立物化视图。同样，最小化 I/O 的使用也有两种方法，即使用索引与聚簇。

第 20 章 物理I/O调整

本章的学习目标：

- 熟悉 Oracle 物理 I/O 调整的概念
- 熟悉物理 I/O 调整的内容
- 掌握物理 I/O 性能测试的方法
- 掌握物理 I/O 调整的方法

20.1 数据文件I/O的调整

数据文件是数据库数据的存储“基地”，所有的用户数据都是从数据文件中读取的，数据文件 I/O 的性能直接影响了数据库的性能。因此，数据文件 I/O 性能调整在整个物理 I/O 调整当中也是十分重要。

要进行数据文件 I/O 性能调整，必须先知道数据文件 I/O 的使用情况，测量数据 I/O 使用情况，主要使用 V$FILESTAT 和 V$DATAFILE 视图。V$DATAFILE 视图我们在前面已经熟悉了，V$FILESTAT 视图主要记录了一些数据文件的统计，这些统计主要是数据文件的物理读，物理写，读写时间，最大、最小物理 I/O 时间，平均使用 I/O 时间和一些数据块的 I/O 信息等。

提高数据文件 I/O 性能的一般方法：

- 把临时段和回滚段表空间与系统表空间分离。
- 分离数据文件到不同的磁盘驱动器

把一个数据文件或其他数据库对象同时放到多个磁盘驱动装置上，当读取一个数据文件时，多个磁盘驱动装置将同时工作，完成原来只有一个驱动装置完成的工作。这样单个磁盘 I/O 的使用率将大大降低，因此性能得到提升。磁盘驱动装置可以为一个 RAID，这是最简单的方法。也可以手工将不同的扩展区分配到不同的磁盘驱动上。由于使用 RAID 完全自动控制的过程，它不需要 DBA 做一些特殊的工作，所以我们下面仅以手工操作加以说明：

- 本地管理表空间
- 修改 DB_FILE_MULTIBLOCK_READ_COUNT 参数

20.2 数据库写进程的调整

1. DBWR I/O 性能的检测

可以使用 V$SYSTEM_EVENT，V$SYSSTAT 视图来了解其使用情况。在 V$SYSTEM_EVENT 当中系统时间 BUFFER BUSY WAIT 和 DB FILE PARALLEL WAITE 均表示 DBWR 造成 I/O 问题。

也可以通过 REPORT.TXT 文件来查看 DBWR 的有关信息。其中 BUFFER BUSY WAITS 表示在高速缓存中，寻找存储区所经历的等待次数。DB FILE PARALLEL WRITE 表示 DBWR 并行写数据块所经历的等待次数。

Redo log SpaceRequest 表示在日志转换过程中，新的重做日志文件在可使用前所作的等待次数。因为日志转换发生时，DBWR 和 LGWR 都必须写内存数据到磁盘，这些动作必须在 LGWR 开始写下一个重做日志之前完成。如果系统发生等待重做日志文件的事件，则表明 DBWR 还没有完成。可以打开输出文件 REPORT.TXT 来查看相关信息。

也可以使用 V$SYSSTAT 视图来查看 REDO LOG SPACE REQUESTS，DBWR BUFFERS SCANNED AND DBWR LRUSCANS 等信息。

2. DBWR I/O 性能的提高

DB_WRITER_PROCESSES 用来指定例程启动时 DBWR 的进程个数。默认值为 1，最大值为 10。当 DBWR_IO_SLAVES 等于 0 时，此参数才有效。

使用 DBWR_IO_SLAVES，此系统初始化参数指定了例程在启动时数据库 SLAVE PROCESSES 的个数，他的作用大致与 DBWR 相同，只不过它不像 DBWR 那样，要进行 LRU 表的扫描。当因 LRU 扫描过慢而造成 DBWR 的效率下降时，使用 DBWR_IO_SLAVES 就可以大大提高数据文件写进程的效率。

DBWR I/O SLAVE 命名方式如 ora_i10n_<instance name>

20.3 段与数据块的调整

Oracle 的块是数据库最小的逻辑单元，一些连续的块组成扩展区，一些扩展区组成了段。因此只要他们的存取和增长管理的效率越高，系统地性能也就越好。

我们调整段和块的 I/O 的方法和目标是：

- 减少扩展区的动态指标
- 消除行迁移和行连接对性能的影响
- 调整 HWM（HIGH WATER MARK）

当行迁移和行连接发生时，外部查询需要读额外的 I/O 到内存，造成系统的性能下降，辨别行迁移和行连接的方法是使用 ALTER TABLE 命令。

此命令将把发生迁移和连接的行的信息插入到一个名为 CHAIN_ROWS 的表中，生成此表可以使用 Oracle 提供的 UTLCHAIN.SQL 脚本。此脚本在 UNIX 上位于 $Oracle_HOME/RDBMS/ADMIN 目录，在 NT 上位于$Oracle_HOME\RDBMS\ADMIN 目录。大概步骤如下：

1. 运行上述脚本
2. 分析表
3. 建立一个表用来存放发生行连接的数据
4. 删除原表中的迁移行
5. 把迁移行回插到原表
6. 重新分析此表

当 FTS（Full Table Sacn）发生时，系统扫描整个系统的表的数据块，一直到 HWM 为止。

当数据从一个段删除时，块的空间被释放，但 HWM 并不因此而降低，此时如果发生 FTS，额外的数据块将被扫描，但对外部的查询并没有什么帮助，而其，在 HWM 以上的表空间也被浪费掉。所以应该给此段生成一个合适的 HWM 值。

20.4 检查点进程的调整

检查点进程是和 LGWR 进程以及 DBWR 进程相关的。当检查点发生时，DBWR 和 LGWR 进程的写操作也同时发生，因此检查点发生的频率对 I/O 性能也有较大的影响。

可以通过查询 V$SYSSTAT 视图来查看检查点的相关信息。

下例中显示检查点开始和结束的次数：

BACKGROUND CHECKPOINTS STARTED：表示系统启动以来检查点开始的次数；

BACKGROUND CHECKPOINTS COMPLETED：表示系统启动以来检查点结束的次数；

这两个值应该相等。如果不等，则表明检查点还没有结束，两者相差的值不能超过 1。

SELECT * FROM v$sysstat WHERE name=’background checkpoints started’ OR name=’background checkpoints completed’;

控制检查点发生的频率可使用初始化参数：

- LOG_CHECKPOINT_TIMEOUT
- LOG_CHECKPOINT_INTERVAL

另外参数 FAST_START_IN_TARGET 和 DB_BLOCK_MAX_DIRTY_TARGET 可以用来调整检查点进程：

FAST_START_IN_TARGET 参数用来指定例程恢复时所使用的 I/O 的数量。默认值和 DB_BLOCK_BUFFERS 大小相等

DB_BLOCK_MAX_DIRTY_TARGET 参数用来指定 DBWR 写脏表到数据文件时，脏表的最大缓冲区个数。一般此值小于 FAST_START_IO_TARGET 的值。

20.5 归档日志进程的调整

如果日志转换发生的频率太高，也会造成 I/O 性能下降。主要原因是归档进程 ARCH 拷贝联机日志文件到归档日志文件效率不够高。可以通过设定初始化参数 LOG_ARCHIVE_MAX_PROCESSES 的值来设定最大归档日志进程数目。默认值为 1，最大值为 10。你也可以使用多个日志文件或增大每个日志文件的空间，这样 ARCH 就有足够的时间来执行归档操作。

与此相关的几个参数为：

- LOG_ARCHIVE_DEST_n
- LOG_ARCHIVE_ DUPLEX_DESTINATION
- LOG_ARCHIVE_MIN_SUCCEED_DEST
- LOG_ARCHIVE_DEST_STATE_n

以上几个具体参数的使用方法请参阅前面章节。

20.6 排序区的调整

当一个排序操作发生时，可以存在于两个位置：内存和磁盘。在内存排序中是最快的方法，在磁盘中排序会造成额外 I/O 的使用。因此，排序区的调整的目的也就是使排序事件尽可能发生在内存当中。

以下命令会造成数据库排序事件的发生：

- ORDER BY
- GROUP BY
- SELECT DISTINCT
- UNION
- INTERSECT
- MINUS
- ANALYZE

- CREATE INDEX

系统初始化参数 SORT_AREA_SIZE 用来指定排序区大小。必须注意，它表示每个用户进行排序操作时所使用的内存大小，而不是所有排序区占用的内存空间。他的最小值为 6 个数据块的大小，

使用视图 V$SYSTAT 可以查看排序区的效率。

可以通过如下几种方法来提高排序 I/O 的性能:

- 执行此操作的一个有效方法是增加内存的大小。可以设置参数 SORT_AREA_SIZE 的增加系统排序区的大小。
- 避免排序操作。可以通过避免排序操作来提高性能，主要有以下一些方法：
  使用 UNION ALL 代替 UNION;
  为 ORDER BY 或 GROUP BY 限制列建立索引，这样就避免了排序操作。
- 在分析表格和索引时使用 COMPUTE。
- 生成临时表空间（CREATE TEMPORARY TABLESPACE），当排序操作占用的空间非常大时，你可以为用户指定临时表空间。临时表空间的默认位置在系统表空间中，所以必须另外建立。

V$SORT_SEGMENT 视图可以用来管理临时表空间信息;

V$SORT_USAGE 视图用来显示所有用户的排序操作。

SORT_MULTIBLOCK_READ_COUNT 参数用来指定用户服务器进程通过每个排序段的 I/O 所读取的操作系统块的最大数量。默认值为 2，最小值为 1，最大值受限于操作系统。

20.7 回滚段的调整

回滚段用于存放数据修改之前的值（包括数据修改之前的位置和值），回滚段的头部包含正在使用的该回滚段事务的信息。一个事务只能使用一个回滚段来存放他的回滚信息，一个回滚段可以存放多个事务的回滚信息。

20.7.1 回滚段的作用

回滚段在数据库运行中扮演着主要角色，它是维护数据库一致性的主要方式。回滚段的作用主要有：

- 事务回滚
- 事务恢复
- 读一致性

20.7.2 回滚段的种类

回滚段可以简单的分为系统回滚段和非系统回滚段，它们的定义如下：

- 系统回滚段

当数据库创建后，将自动创建一个系统回滚段，该回滚段只用于存放系统表空间中对象的前影像（Before Imagine）。

- 非系统回滚段

拥有多个表空间的数据库至少应该有一个非系统回滚段，用于存放非系统表空间中对象

的数据前影像，非系统回滚段又分为私有回滚段和公有回滚段。私有回滚段应在参数文件的 ROLLBACK SEGMENTS 参数中列出，一边例程启动时自动联机。公有回滚段一般在 Oracle 并行服务器（OPS）中出现，将例程启动时自动联机。

20.7.3 回滚段I/O性能测试

1. 回滚段块头（HEADER BLOCK）的竞争

每个回滚段最前端的数据块称为块头。Oracle 把事务表存储在此块中，用来跟踪那些使用此回滚段存储它们前影像的事务。此块通常被存储在数据库高速缓存中，以便所有用户可以访问他。在一个繁忙的联机事务处理系统中，用户访问块头信息可能要排队，这就造成事务处理能力下降。

- 回滚段扩展区的竞争

当外部事务进行到回滚段的每个独立扩展区时，也可能发生等待。查询相关信息可以使用 V$WAITSTAT 和 V$STSTEM_EVENT 视图。

- 回滚段的动态指派

当一个外部事务所要使用的空间超过了指派给他的空间时，如果其他的扩展区正在被其他事务占用，回滚段将生成一个新的扩展区，扩展区的动态指派也会造成 I/O 的使用，应尽量避免。

可以通过查看 V$ROLLSTAT 视图和 V$STSTEM_EVENT 视图来获取回滚段的动态指派信息。

20.7.4 提高回滚段I/O性能

知道了如何查看回滚段 I/O 性能的方法以后，接下来就是该决定如何提高回滚段 I/O 的性能了，下面是提高回滚段 I/O 性能的一般使用的方法：

- 最直接办法就是增加更多的扩展区给回滚段；
- 有时，当大型事务发生时，只靠增加扩展区是不够的，还要确保每个扩展区的空间足够大；
- 对于一些特殊的、非常大的事务，如果这些事务经常发生，可以指派一些大型回滚段专门用来存储它们。

20.8 本章小结

本章主要介绍了物理 I/O 性能测试、查询以及调整的方法，它又分为数据文件 I/O 性能调整、数据库写进程 I/O 性能调整、数据块归档进程 I/O 性能调整、排序区 I/O 性能调整、回滚段 I/O 性能调整。

数据文件 I/O 调整主要通过把应用程序访问的数据对象同系统对象隔离开来实现。数据库写进程 I/O 调整通过增加数据库写进程和数据库写入器的 SLAVE 来实现。回滚段调整比较困难，要记住两个方法即防止其动态扩展和使其工作量最小化。

第 21 章 调整竞争

本章学习目标：

- 熟悉 Oracle 的锁和闩
- 掌握锁和闩的分类与用途
- 掌握消除 Freelist 列表的方法

21.1 锁

数据库争用可以被分为两大类：

- 锁----用于限制其他用户对数据的存取。
- 闩----Oracle 使用的一个内部机制，用来管理和维护内存锁。

Oracle 根据前后关系自动决定需要为给定的情况应用什么样的锁以便使用最小的约束提供最大的数据并行性。Oracle 有两种级别的锁：共享锁与专用锁。

- 共享锁(SHARE LOCK)通过数据存取的高并行性来实现。假如获得了一个共享锁，其他用户就可以共享相同的资源。许多事务可以获得相同资源上的共享锁，但是对于专用锁来说，情况就不是这样。例如，多个用户可以在相同的时间读取相同的数据。
- 专用锁(EXLUSIVE LOCK)防止同时共享相同的资源。例如，假如一个事务获得了某一资源上的一个专用锁，那么直到该锁被解除，其他事务才能够修改那个资源。还有，也允许对资源进行共享。例如，假如一张表被锁定在专用模式下，它并不能阻止其他用户从同一张表中进行选择。

Oracle 主要有两种不同类型的锁：

- 数据锁(DML LOCK)在表中获得并且保护数据，从根本上来说是保护数据的完整性。
- 字典锁（DDL LOCK）用于保护对象的结构如表格、视图和索引等的定义。在遇到 DDL 数据定义事务时，字典锁将自动获得。

21.1.1 数据锁

当用户对表中数据执行 INSERT、UPDATE 和 DELETE 操作时将要用到数据锁，它用来保护数据的一致性的。按级别分可分为行级锁和表级锁两种。

1. DML 锁的用途

DML 锁的用途是保护被多个用户并行存取的数据的完整性。DML 锁可以防止同时进行的 DML 操作的破坏性干扰。

DML 语句使用两种类型的锁结构：

- 事务获得表上的一个共享锁
- 事务获得它正在更改每一行上的专用锁

注意：

假如有 3 个用户视图同时对同一行进行修改，他们全都会得到共享的表锁，但是只有最先请求锁的用户得到行锁。

一个正在被修改的行总是被专门地锁定以使其他用户不能传输修改该行，直到包含该锁的事务被提交回滚为止。

2. DML 锁的模式

下面列举了用于 DML 的 Oracle 锁模式：

- 行共享（RowShare, RS）：行共享锁是一个共享的表锁。它是一个在被查询行上的专用锁。
- 行专用（Row Exclusive, RX）：行专用锁通常表明持有该锁的事务已经读一表中的行完成了一次或多次修改。
- 共享锁（Share, S）：假如一个事务持有一个共享表锁，那么它便防止其他事务在该表中执行 DML 操作。
- 共享行专用（Share Row Exclusive, SRX）：假如一个事务包含一个 SRX 表锁，那么它防止其他事务执行 DML 或 SELECT.....FOR UPDATE 语句。
- 专用（Exclusive, X）：专用锁是表锁的最具限制性的形式，它允许持有表锁的事务对该表进行独占访问。

21.1.2 字典锁

当用户创建、修改或者删除表时将要用到字典锁。它通常是表级锁，用来防止两个用户同时修改同一个表的结构。

1.DDL 锁的功能

- DDL 锁在过程中引用的对象在过程编译结束以前不能被改变或删除。
- DDL 锁仅影响在 DDL 操作期间被修改或引用的单个模式对象。它们不锁定整个数据字典。
- DDL 锁不能被显式地实现。

2.不同形式的 DDL 锁

- 专用 DDL 锁

当诸如 CREATE、ALTER 和 DROP 这样的语句用于一个对象时使用此锁。

专用 DDL 锁的特征：

假如另外一个用户保留了任何级别的锁的话，那么本用户不能得到表中的专用 DDL 锁。例如，假如另一个用户在该表上有一个未提交的事务，那么 ALTER TABLE 语句会失效。

- 共享 DDL 锁

当诸如 GRANT 和 CREATE PACKAGE 这样的语句用于一个对象时使用此锁。

共享 DDL 锁的特征：

一个共享 DDL 锁不能阻止类似的 DDL 语句或任何 DML 语句用于一个对象上，但是它能防止另一个用户改变或删除已引用的对象。

共享DDL锁的另一个特征是在DDL语句执行期间它一直维持，直到发生一个隐含的提交。

事务可以通过发出如 INSERT、UPDATE、DELETE 等的语句获得表锁（TM）。为了使事务能够保护表中的 DML 存取及防止表中产生冲突的 DDL 操作，Oracle 获得表锁。例如，假如某个事务在一张表上持有一个表锁，那么它会阻止任何其他事务获得该表中用于删除或改变该表的一个专用 DDL 锁。

3.手工锁定

假如你想要执行一张表中列值的全局更新并且希望事务对该表进行单独存取，以便该事务不必等待其他事务完成该表的操作，那么你可以通过人工锁定该表以防止其他事务获得该表中的锁，从而实现这一点。例如：

```
LOCK TABLE t_emp IN SHARE MODE
```

当用户执行如下的语句时获得共享模式锁：

LOCK TABLE t_emp IN SHARE MODE;

当用户执行如下的语句时获得共享行专用模式锁：

LOCK TABLE t_emp IN EXECLUSIVE MODE

现在可以在不必与其他事务发生争用的情况下修改列值。

实现上，必须是使用这些语句的应用很少；因此，同其他每日频繁发生的锁定类型相比，它们被认为不是很重要的。

21.1.3 死锁

死锁是指两个或多个用户都在等待被彼此相互锁定的同一数据而形成的一种局面。

解决办法有两种：

- DBA 可以通知 AZURE 提交事务，这样 HR 的操作就可以完成。然后 HR 再提交事务，则 WJC 的操作也可以继续。
- DBA 中止 HR 的会话，则 HR 对 emp_id=100 的锁将被释放，WJC 的操作将可以完成。

可以通过查询视图 V$SESSION 来查看用户的 SID 和 SERIAL#。当所有用户使用同一个 ORACLE 用户名登录应用程序时，还要增加 MACHINE 列来区分。

21.2 闩的调整

闩(Latch)是用来保护对 Oracle 内存结构的访问。闩也是一种特殊的锁，每个闩根据名字的不同作用也不同。在任何时间内，一个闩只允许一个进程对它进行访问。换句话说，Oracle 确保在任何时间内没有两个进程同时访问同一数据结构。

当一个进程需要进入一个闩时，如果此闩处于 BUSY 状态，进程就要经历一个等待。这种等待将随着所要进入闩的不同而不同：

如果是等待闩，即 Willing-to-Wait 闩，进程每隔一小段时间将会再次向闩发出请求，可能要请求多次，直到成功为止。

如果是立即闩，即 Immediate 闩，进程不经过等待而立即向闩再次发出请求，直到成功。

动态视图 V$LATCH 用来管理 Willing-to-Wait，Immediate 闩。

其中：

NAME：显示闩的名字；

GETS：不经过等待而获得 Willing-to-Wait 闩的次数；

MISSES：需要经过等待并且再次向 Willing-to-Wait 闩发出请求的次数；

SLEEPS：一个进程在获得 Willing-to-Wait，Immediate 闩之前所经历的时间；

IMMEDIATE_GETS：不经过等待而获得 Immediate 闩的次数；

IMMEDIATE_MISSES: 需要经过等待并且再次向 Immediate 闩发出请求的次数；

在性能调整中，我们要尽量降低闩等待的次数。通常我们可以通过增加闩的个数来解决此问题。在 Oracle 9i 当中，有 142 个闩(Willing-to-Wait 闩和 Immediate 闩)。只有 CACHE BUFFERS LRU CHAIN LATCH 比较重要。

当服务进程在数据库高速缓存的 LRU 列表里寻找空闲的块时，CACHE BUFFERS LRU CHAIN LATCH 将被使用。CACHE BUFFERS LRU CHAIN LATCH 的个数由初始化参数 DB_BLOCK_LRU_LATCHES 决定。可以通过查询视图 V$LATCH 和 REPORT.TXT 文件来查看相关信息。

当闩竞争出现时，可以增加闩的个数，即增大 DB_BLOCK_LRU_LATCHES 的值。通常状况

下，Oracle 建议此参数等于 6 倍的 CPU 个数，或者是 DB_BLOCK_BUFFERS 的 1/50。

21.3 Freelist的竞争

通常情况下，一个表只有一个 Freelist 列表，但是当许多用户进行高频率的插入操作时，应用程序的服务器进程在进入 Freelist 列表时要经过一些等待，这些等待称为 Freelist 的竞争。我们调整的目标是消除 Freelist 等待的出现。

可以通过查询 V$WAITSTAT、V$SYSTEM_EVENT 和 V$SESSION_WAIT 视图来查看相应信息。DBA_SEGMENTS 数据字典视图和性能调整器界面工具也可以用来显示相关信息。

如果有 buffer busy wait 发生，那还不一定意味着出现 Freelist 竞争。还必须通过查询视图 V$WAITSTAT 来确定是否有 Freelist 竞争出现。

使用 V$SESSION_WAIT 和 DBA_SEGMENTS 数据字典视图联合查询可以得到发生竞争的具体的数据库对象。读者可以自行组织查询一下。

调整 Freelist 列表竞争的方法是删除此数据库对象，然后重建 Freelist 列表。因为 Freelist 列表只有在段生成时被指定且不能动态更改，默认状态下每个段只有一个 Freelist。

21.4 本章小结

本章主要介绍了 Oracle 数据库的锁机制、闩机制以及空闲列表竞争管理。锁是用来保护数据库的写一致的，通过不同级别的锁，可以对用户修改数据事件排序，闩是 Oracle 用来保护内存的一种机制，空闲列表用来管理数据库段的插入操作。在实际应用当中，我们应当尽量避免以上竞争机制的出现，另外，我们还讲述了一些用于查询以上竞争的视图及工具，这些方法也要掌握。

第 22 章 Oracle资源管理

本章的学习目标：

- 熟悉 Oracle 资源管理的概念以及内容
- 掌握在图形和命令行两种方式下创建资源使用者组和资源计划

22.1 资源管理概况

Oracle 9i 资源管理使用一种新的方法来管理应用程序用户对资源的使用。可以创建“资源计划使用者组”并指派给他们具体的资源使用限制。这些资源计划使用者组可以按次序指派给应用程序用户。

Oracle 资源管理器分为 3 部分：

- 资源计划使用者组（Resource Consumer Group）,即具有相似的数据库资源消耗的用户群。
- 资源计划（Resource Plan），即一些用来执行资源管理的资源调度的集合。
- 资源计划调度（Resource Plan Derective），用来指定每一个资源使用者组的 CPU 和并行查询资源的数目。

资源管理器只可以管理两种资源：

- 每个资源计划使用者组的用户使用 CPU 的个数。
- 每个资源计划使用者组的用户执行的并行度（即并行查询级别，并行查询具有优先级，级别越高优先级越高）。

当向数据库用户指派 CPU 和并行度时，资源管理器允许按不同的优先级分为 8 个级别。例如当创建一个名为 RES_GRP 的资源使用者组，我们可以指定此组的用户在 RES_GRP_PLAN 资源计划中使用 CPU 和并行度。

Oracle 的 CPU 指派和并行度。

级：指定分配给一个组的 CPU 资源百分比。

CPU 资源分配：CPU 资源的指派使用百分率（%）来计算。

并行查询资源使用相对方法，即指定其并行值，如 Level 1:5,Level 2:3 等。其他设置与 CPU 资源分配相同。

22.2 资源管理配置

Oracle 的资源管理配置有两种方式，即命令行和图形界面方式。下面我们分别举例加以介绍。

我们现在要创建名为 RES_GRP 的资源计划使用者组。

（1） 创建资源使用者组。

（2） 创建资源计划。

（3） 激活资源计划。

22.3 资源管理器的管理

完成资源计划的创建以后，接下来就可以对资源计划进行管理了，资源使用者组创建完以后，如果发现需要更改，可以通过企业管理器很方便地更新现有的资源使用者组

22.4 使用SQL*PLUS创建资源计划和使用者组

默认状态下，拥有 DBA 权限的用户可以创建并管理资源使用者组，如果没有 DBA 权限，用户必须具有 ADMINISTER_RESOURCE_MANAGER 权限。不能像授予其他权限一样授予此权限给用户，即不能使用 GRANT…TO 命令来授予。要授予此权限，可以使用名为 DBMS_RESOURCE_MANAGER_PRIVS 的 PL/SQL 包。

使用具有 ADMINISTER_RESOURCE_MANAGER 权限的用户身份登录，创建资源管理对象，请依照以下步骤：

（1） 生成 PENDING 区域。
（2） 创建资源使用者组。
（3） 创建资源计划。
（4） 创建资源调度。
（5） 验证挂区。
（6） 现在保存设置。
（7） 把资源使用者组 res_grp1 指派给用户 hyben。
（8） 把资源使用者组 res_grp1 指派给用户 hyben 且 res_grp1 为用户 hyben 的默认资源使用者组。

22.5 本章小结

本章主要介绍 Oracle 资源管理内容及配置方法，管理内容主要是用户 CPU 的分配和并行度的使用，配置过程主要是创建资源计划使用者组、创建资源计划和使用资源计划调度。可以通过 Oracle 的命令管理界面 SQL*PLUS 来实现，也可以通过 Oracle 的企业管理器来实现。有一点要记住就是不要忘记分配资源使用者组 OTHER_GROUPS。

第 23 章 Oracle性能调整工具

本章学习目标：

- 熟悉 Oracle Expert 性能调整工具的作用
- 熟悉 Oracle Expert 对数据库性能调整的模式
- 掌握通过 Oracle Expert 对数据库性能调整的过程

23.1 考虑使用Oracle Expert

在 Oracle 服务器的性能调整过程中，就像我们在前面几个章节介绍的一样，对 Oracle 数据库进行性能调整需要执行很庞大的性能调整测试。要记住每一个数据统计以及所建议的调整目标和方法是一项很繁杂的事情，应用这些调整技术也是很困难的。我们引进了 Oracle Expert 用来辅助 DBA 进行重要工作的调整，这样，就可以把 DBA 从繁忙的调整中解脱出来，从而更好地去做其他更重要的事情。

打开 Oracle 企业管理器，找到 Oracle Expert 工具集，选中并运行它。

23.2 Oracle Expert的使用

在“优化会话界面”对话框可以选择下列单选按钮之一：

- “装载优化会话示例”单选按钮称为“人事信息会话”的优化会话示例是用于一个不存在的人事信息数据库德优化会话。其中包括 Oracle Expert 使用的数据示例，如数据库、例程、方案、表、列、同义词和 SQL 工作量信息。
- “新建一个优化会话”单选按钮使用 Oracle Expert 优化数据库环境时，Oracle Expert 会在优化会话的框架中收集并分析数据，并生成优化建议案、报告和实施脚本。初次使用 Oracle Expert 时需要创建一个新的优化会话。
- “打开一个现有的优化会话”单选按钮启动 Oracle Expert 时可以打开一个现有的优化会话。

在此对话框中，还可以决定是否希望在 Oracle Expert 启动时显示“优化会话向导”。

在这里我们选择“创建一个新的优化会话”单选按钮，然后单击“下一步”按钮，将可以执行以下操作：

- 选择要优化的数据库，如果通过 Oracle Enterprise Manager 已搜索到某数据库，该数据库将被列出。
- 为新的优化会话提供名称。新名称包括的字符最多不能超过 40 个。Oracle Expert 能够保留名称中字母字符的大小形式。

23.2.1 设定范围

在“优化会话”窗口的“范围”选项卡上，可以选择希望检查的优化范围。所选的优化范围将确定 Oracle Expert 所执行的评估类型。选择的优化范围也将决定“收集”选项卡中哪些选项可用。

1. 优化范围

- “检查例程优化”：使用此优化范围来确定是否设置了正确的优化参数，以及数据库例程是否能够有效利用系统资源。
- “检查 SQL 复用可能性”：使用此优化范围来确定优化会话是否包含性质相同而语法上稍有不同的 SQL 语句。这样的 SQL 语句必须被单独分析和高速缓存。如果语法上的差异已被排除,那么 Oracle Expert 将高速缓存该语句的单个版本,允许应用程序重复使用该高速缓存的 SQL 语句。
- “检查相应的空间管理”：使用此优化范围来评估数据库空间管理问题，如表空间结构、方案对象的大小调整和布局，以及数据库用户的表空间分配。
- “检查最佳的数据访问”：使用此优化会话来优化指定表的索引，并检查需要重建的索引，共有 3 种选项：

“对执行性能最差的 SQL 语句引用的表执行综合索引评估”：若选中此选项，Oracle Expert 将自动在执行性能最差的 SQL 语句（在您的优化会话工作量中标识）引用的表中集中进行数据访问优化。优化会话的 SQL 语句将根据每个语句的每次执行的物理读取比率来划分等级。Oracle Expert 也将自动检查目标表中现有索引上的索引碎片。

“对指定的表进行综合索引评估”：若选中此选项，Oracle Expert 将在您指定的特定方案或表中集中进行数据访问优化。Oracle Expert 也将自动检查目标表中现有索引上的索引碎片。

“对指定的表进行索引碎片评估”：如果只希望执行索引碎片检查，请使用此选项。若选中此选项，Oracle Expert 将只标识目标中的索引，该目标表必定是遇到了索引滞流，并且需要重建以提高性能。

2. 优化会话特性

在“范围”选项卡，还可以指定优化会话的某些特性。Oracle Expert 使用优化会话特性来促进优化评估。例如，数据库应用程序类型是正确的例程设置和索引策略的主要决定因素。

- “应用程序类型”

“应用程序类型”优化会话特性向 Oracle Expert 表明数据库环境中使用的工作量类型。这使得 Oracle Expert 可以根据工作量类型来优化数据库。可能的值为：

OLTP：OLTP 工作量通常对包含混合读写请求的表使用需要快速响应的简单查询。

“数据仓库”：数据仓库工作量通常对大型的，通常为只读的数据库表使用综合查询。

“多用途”：多用途工作量通常具有非常宽松的响应时间限制。其特征通常是一个或少数几个用户进行大量写密集型的事务处理。

- “关闭时间容差”

利用“关闭时间容差”优化会话特性，您可以确定系统的建议案将倾向于优化恢复还是优化性能。如果容差很大，Oracle Expert 将优化性能。如果容差很小，Oracle Expert 将优化恢复时间。可能的值为：

“无”：无容差。

“小容差”：表示允许由于意外失败而导致数据库在正常运行中出现短暂延迟。

“中等容差”：表示允许由于意外失败而导致数据库在运行中出现中等程度的延迟。

“大容差”：表示允许由于意外失败而导致数据库在运行中出现较长时间的延迟。如果必须在优化恢复时间与优化响应时间之间进行权衡，Oracle Expert 将会优化性能。

- 峰值逻辑写速率

在尚未收集到实际的事务处理统计信息时，“峰值逻辑写速率”优化会话特性向 Oracle Expert 表明最大写入事务处理量。该信息用来评估服务器是否已配置为支持预期的写入事

务处理速率。可能的值为：

“低”：低写入事务处理速率应用于每秒最多执行 5 个插入、删除或更新操作的环境。

“中等”：中等写入事务处理速率应用于每秒最多执行 50 个插入、删除或更新操作的环境。

“高”：高写入事务处理速率应用于每秒最多执行 500 个插入、删除或更新操作的环境。

“超高”：超高写入事务处理速率应用于每秒最多执行 500 个以上的插入、删除或更新操作的环境。

- “使用的表格应用程序”

“使用的表格应用程序”优化会话特性告知 Oracle Expert 是否在数据库环境中使用表格应用程序。Oracle Expert 包含专用于表格应用程序的规则，如为例程设置打开游标的最小数量。可能的值有：“是”和“否”。

- “综合分析”

“综合分析”优化会话特性告诉 Oracle Expert 在当前数据库中有完整的工作量。综合分析能够提供最为有效的优化建议案。如果选择不执行综合分析，分析将集中在当前正被数据库资料档案库访问的 SQL 语句的子集。

- “优化程序验证”

“优化程序验证”优化会话特性通知 Oracle Expert 在实施建议之前对建议案加以验证测试，以确保充分改善了性能。Oracle Expert 建议只实施那些确实能改善性能的建议案。要使用此功能，必须具有服务器的相应写权限。该选项仅适用 Oracle 8.1.5 或更高版本的数据库。

23.2.2 收集统计

下面我们分别介绍 3 个选项：“系统”、“数据库”和“例程”：

1. “系统收集选项”

“系统收集选项”对话框可用于指定 Oracle Expert 为优化会话收集系统类数据的方式。系统数据为 Oracle Expert 提供了一个计算系统资源的概要文件，以下选项允许指定将要收集的系统数据的源，以及如何收集该数据：

- “源”选项区域
- “选项”选项区域

2. “数据库收集选项”

“数据库收集选项”对话框可用于指定 Oracle Expert 为优化会话收集数据库类数据的方式，以下选项允许您指定将要收集的例程数据的源以及如何收集该数据：

- 源选项区域
- “选项”选项区域

3. “例程收集选项”

“例程收集选项”对话框可用于指定 Oracle Expert 收集优化会话的例程类数据的方式，以下选项允许您指定将要收集的例程数据的源以及如何收集该数据：

- “源”选项区域
- “选项”选项区域

在“系统收集”对话框中可以输入有关系统设置的特定数据。Expert 在生成建议案时需要使用此信息。单击“高级”按钮，可以查看高级数据参数。

“系统名”：为本计算机名，默认情况下，主机名将显示在此字段中。

“物理内存总量（RAM）”：输入 RAM 的总量（以 GB 或 MB 为单位）

“例程可用的 RAM 总量百分比”：输入用于此例程的 RAM 的百分比

“平均内存利用率”：指定一段时间内的系统内存使用率。

“内存最大利用率”：指定内存的最大利用率。

“CPU 平均利用率”：指定一段时间内的 CPU 利用率。

“CPU 最大利用率”：指定 CPU 最大利用率。

“操作系统页大小（字节）”：关于系统的页大小，请参考操作系统的相关文挡。

在完全设置好以上各项后，单击“确定”即可。系统将自动执行有关信息的收集工作。

23.2.3 复查

“复查”选项卡可用于在优化会话和 SQL 历史记录收集的分层视图中查看已收集的数据。可收集的对象类型在“复查“选项卡上用文件夹图标表示。为优化会话或 SQL 历史记录收集的对象可显示在“复查”选项卡层次中相应对象类型的文件夹下。而某特定类型的各个对象用图标表示，这些图标是此类型对象独有的。

除了使用“复查”选项卡查看已收集的数据对象外，还可以使用“复查”选项卡和“编辑”菜单对优化会话和 SQL 历史记录数据执行以下操作：

- 将新对象添加到数据中
- 修改数据（更改对象的属性值和规则值）
- 删除对象
- 验证对象
- 导出数据库、例程、方案、系统和工作量数据
- 查看并更改默认规则的值

1. 属性
2. 规则

按照以下步骤可实例化对象的规则而不更改规则的值：

（1）在“复查”选项卡上选择对象。

（2）选择“编辑”->“修改”命令。Oracle Expert 将为该对象显示“编辑”对话框。此对话框有“规则”选项卡。

（3）选择“规则”选项卡。此选项卡将列出该对象当前的规则值。

（4）选择要实例化的规则。

（5）单击“实例化”按钮来实例化规则。

（6）如果改变主意，可通过选择已实例化的规则然后单击“移去”按钮来移去实例化。

（7）单击“确定”按钮确认所做的更改，然后退出“规则”选项卡。

遵照以下步骤更改规则的值：

（1）在“复查”选项卡上选择对象。

（2）选择“编辑”->“修改”命令。Oracle Expert 将为该对象显示“编辑”对话框。此对话框有“规则”选项卡。

（3）选择“规则”选项卡。Oracle Expert 将显示“规则”选项卡，此选项卡将列出该对象当前的规则值。

（4）选择要更改的规则。

（5）单击“实例化”按钮。

（6）将光标置于要更改其值的规则的“值”列中。

（7）单击鼠标左键，然后为该规则提供一个新值。

（8）如果改变主意，可通过选择已实例化的规则，然后单击“移去”按钮来移去实例化。

（9）单击“确定”按钮确认所做的更改，然后退出“规则”选项卡。

注意：

实例化某个规则后，字母 R 将显示在“复查”选项卡中该对象的图标上。这表示已为该对象实例化某个规则。

默认情况下，“规则”选项卡仅显示对象的基本规则。单击“高级”按钮，即可看到该对象的高级规则。Oracle Expert 会将基本规则和高级规则一并显示。显示高级规则时，在每个高级规则的“高级”列中都将显示一个 X，并且“高级”按钮被替换为“基本”按钮。如果要只显示基本规则，请单击“基本”按钮。

注意：

如果实例化某个规则后，单击了“取消”按钮，已实例化的饿规则仍将被保存。要移去已实例化的规则，请选择该规则然后单击“移去”按钮。

23.2.4 生成建议案

完成复查工作后，我们单击“建议案”标签，“优化会话”窗口的“建议案”选项卡可用于复查 Oracle Expert 作为分析优化会话数据的一部分生成的建议案。特定类型的建议案被聚集成组，并放在具有描述性名称的文件夹下（像“复查”选项卡中的文件夹）。“建议案”选项卡按 Oracle Expert 生成建议案的顺序显示它们。

单击“建议案”选项卡山个文件夹旁的加号（+），可以查看特定类型的建议案。

要获取有关 Oracle Expert 为何生成某种特定建议案的详细信息，请选择该建议案并单击查看详细资料按钮。Oracle Expert 将显示该建议案的基本原理。

1. 拒绝例程建议案
2. 实现和禁用其他建议案

23.2.5 脚本的生成

完成“建议案”的生成后，单击“脚本”标签，利用“优化会话”窗口的“脚本”选项卡，可以生成文件和脚本，并利用它们来实施 Oracle Expert 建议案。

23.3 本章小结

本章主要介绍了 Oracle 性能调整工具 Oracle Expert 的调整原理与使用过程等。Oracle Expert 的调整原理是 Oracle 根据大量的性能统计参数，然后给出提示建议的。在使用 Oracle Expert 的过程中要按照一定的步骤执行，先要设定调整的范围，然后再执行收集统计，接着是设置复查，设置完以后，就可以生成建议案了，最后，Oracle Expert 以脚本的形式给出调整建议，经过简单修改后直接在 SQL*PLUS 工具中执行了。

第 5 部分 网络管理

第 24 章 Net Manager基本架构

本章学习目标：

- 熟悉 Oracle Net Manager 的作用和功能
- 熟悉 Oracle Net Manager 堆栈段的结构和作用
- 熟悉 Oracle 连接管理器及域的概念和功能

24.1 Oracle Net Manager功能简介

Oracle Net Manager 支持像 Oracle Application Server 等中间件产品，通过中间件可以建立多层网络架构是支持复杂的企业网络应用。Oracle Net Manager 主要有以下特征：

- 多协议支持
- 多平台操作
- Java 连接特性
- 开放的 API

24.2 Oracle 监听器

监听器为 Oracle 最主要的服务器端网络组件，它用来监测所有连接到数据库上的信息。监听程序是驻留在服务器上的一种进程，其职责是监听入网的客户机连接请求和管理到服务器的通信量。每次客户机请求与服务器进行网络会话，监听程序就会接受到实际请求。如果客户机的信息与监听程序的信息相匹配，监听程序就授权连接服务器。

监听程序控制实用程序：包含在 Oracle Net 中的实用程序，用来控制各种监听程序功能，如启动、停止以及获取监听程序的状态。监听程序控制实用程序以下列方式运行：

Lsnrctl command [listener_name]

listener_name 是在 listener.ora 文件定义的监听程序名。如果正在使用默认的监听程序名 listener，则不必标识监听程序名。

24.3 概要文件

所谓的概要文件是一系列参数的集合，用来指定客户机或服务器上启用和配置 Oracle Net 功能的首选项。通过概要文件所有配置信息存储在 sqlnet.ora 文件当中。

概要文件可以手工创建或修改，它是确定客户机如何连接到 Oracle 网络的参数的结合。可以配置命名方法、事件记录、跟踪、外部命令参数以及 Oracle Advanced Security 的客户机参数。

通过概要文件可以：

- 指定附加到未限定名后的客户机域
- 区分命名方法的优先次序

- 启用事件记录和跟踪功能
- 通过指定进程路由连接
- 为外部命名配置参数
- 配置 Oracle Advanced Security

24.4 网络服务命名

网络服务命名是客户机应用程序尝试连接到数据库服务时使用的一种解析方法，用来将连接标识符解析为连接描述符。

Oracle 命名服务器使用 Oracle Names 软件为服务存储网络地址信息及其简称的一种计算机，这样客户机应用程序可以使用简称而不是冗长的地址请求连接。Oracle Names Server 可以存储下列类型的信息：

- 数据库服务地址
- 网络服务名
- 全局数据库连接
- 任意已定义网络对象的别名
- 其他的 Oracle Names Server 地址信息

24.5 Oracle Net Manager网络协议堆栈段

分布式处理的概念依赖于设计上和物理位置上都分离的计算机之间相互通信和交互的能力。通过堆栈通信过程可以完成通信。堆栈通信可以引用开放系统互联（OSI）模型来解释。在 OSI 模型里，分开的计算机之间的通信在一个类似堆栈的方式里通过几层代码从一个节点到另一个节点传递信息。

信息在客户段为了能通过网络介质传输而打包，能通过各层下行，在这种方式下，他可以被对应的服务器端的各层解释和理解。

24.5.1 典型的OSI协议通信栈

一个典型的 OSI 协议通信栈包含七层：

- 客户端应用程序层：最接近用户的 OSI 层，并同样的依赖于用户请求的功能。例如，在数据库环境中，表单应用程序可能为了从服务器访问数据而试图发起一个通信。
- 表示层：确认数据是以一种应用层和会话层都能够接受的格式表示的。这包括明确在客户端和服务器之间传输的数据的语法和语义。如果需要，表示层可以使用一个公共的数据格式在多数据表示格式之间转化。
- 会话层：建立、管理和终止客户端和服务器端之间的网络会话。这是一个携带数据请求和相应的虚拟管道。会话层决定数据通信能否在同一时间在两个方向传输。或者只能在同一时间里在一个方向上传输。
- 传输层：实现数据传输，确保数据可靠地进行传输。
- 网络层：确认数据传输通过最优的路径和一系列的互联子网来发送。
- 链路层：提供通过物理连接的可靠数据传输。
- 物理层：为客户端和服务器之间建立、维护和释放物理连接定义一个电子的、机械

的和过程的规范。

24.5.2 Oracle Net Manager客户端/服务器中的堆栈

Oracle Net Manager 客户端/服务器中的堆栈的各层参照标准的 OSI 通信栈各层并与之类似。Oracle Net Manager 是 Oracle 通信堆栈结构的整体构建，它基于 TNS 层（Transparent Network Sustrate)。TNS 为一个复杂网络堆栈结构，可以处理多种网络通信协议，并负责客户端与服务器之间复杂的通信。

Oracle Net Manager 的通信堆栈层如下：

- 客户端应用程序层（Appliction）
- Oracle 调用接口（Oracle Call Interface，OCI）
- 双任务公共层（Two Task Common，TTC）
- TNS 层
- Oracle 网络协议转接层（Oracle Protocol Adapters，OPA）
- 具体网络协议

1. 客户端应用程序层

Oracle 客户端应用程序提供了所有用户定位的活动，如字符图形用户显示、屏幕控制、数据表示和应用程序流。应用程序标识数据库操作，把他们发送到服务器，并通过 Oracle 调用接口（OCI）来传递他们。

2. Oracle 调用接口（OCI）

Oracle 调用接口代码包含了在客户段和服务器之间初始化一个 SQL 对话所需的所有信息。它主要用来打开或关闭游标、服务器端变量的邦定等任务。

3. 双任务公共层（TTC）

双任务公共层是表示层的 Oracle 实现。TTC 提供了字符集和客户端与服务器之间不同字符集或格式的数据类型转换。该层可以在每个连接的基础上优化为只在需要时才执行转化。

在初始连接时，TTC 负责评估内部数据和字符集表示的差异，并决定为实现两个计算机之间通信是否需要进行转换。

4. TNS 层

TNS 层包含以下几个分层：

网络接口（NI)：该层为 Oracle 客户端、服务器或外部进程访问 NET8 函数提供一个通用的接口。NI 层处理连接的“中断”和“重启”请求。

网络命名（NN)：NI 使用网络命名将名称解析到连接描述符。

网络会话（NS)：该层从 NI 接受请求，并解析所有机器级的连接问题，如服务器或目的机器的位置（Open，Close 函数)；一个或更多的协议是否将在连接里涉及到（Open，Close 函数)；以及如何去处理基于服务器和客户端之一的中断（Send，Receive 函数)。

网络路由（NR)：NS 使用网络路由来路由网络会话到目的地。

网络认证（NA)：NS 使用 NA 来与目的地协商任何认证要求。

5. Oracle 网络协议转接层

Oracle 网络协议转接层直接和具体网络协议打交道。他负责 TNS 层和网络协议之间的转接。

6. 具体网络协议

所有客户端----服务器连接过程中的 Oracle 软件都需要一个现存的网络协议栈来为传

输层在两台机器之间产生机器级别的连接。网络协议仅负责从客户端取数据到服务器上，数据在这个点上传递到服务器端的 Oracle 协议上。

Oracle 编程接口（OPI）执行一个 OCI 的补充函数。它负责对ＯＣＩ发送的每一个消息进行响应。例如，一个取 30 行数据的 OCI 请求将产生一个 OPI 响应来返回 30 行数据，当然这 30 行数据必须已经取过来了。

Oracle9i 通过 Oracle9i JServer 支持 Java。Oracle 9i JServer 包括对 Java 存储过程、JDBC、SQLJ、公共对象请求代理体系结构（Common Object Request Broker Architecture）以及企业版 JavaBeans 的支持。

Oracle9i JServer 是 Java 虚拟机（Java Virtual Machine，Java VM）支持 GIOP 的表示。客户端用 GIOP 来访问企业版 Java Beans 和 Java VM 上的 CORBA 服务器。

EJB 和 CORBA 客户端使用不同于典型的 NET8 客户端栈的通信栈。如图 24-3 所示,不同之处在于：它使用 GIOP 作为表示层，而不是使用 NET8 通信栈的会话层作为表示层。

服务器端不需要典型的 NET8 连接里所需的许多 NET8 通信层。服务端只需要 TCP/IP 网络协议和 Oracle TCP/IP 协议或带 SSL 的 TCP/IP 协议。NET8 所需的惟一组件是网络会话。

24.6 Oracle连接管理器

Oracle 连接管理器是一个路由器。客户端的连接请求通过它可以发送到它的下一个节点或者直接发送到数据库服务器上。通过连接管理器路由连接请求的客户端可以利用连接管理器里配置的连接集中、NET8 访问控制或者多协议支持特性。

1. 连接集中

Oracle 连接管理器是通过单一的传输协议到多线程服务器目的地的连接能够多路复用或分散多客户端网络会话。

也就是说，多个客户端可以同时连接到一个 Oracle 连接管理器，再通过 Oracle 连接管理器连接到数据库服务器。这样就可以增加一个服务器能够处理的网络会话的数目，同时减少了系统资源。通过使用多个连接管理器，使成千上万的并发用户连接到一个服务器成为可能。

2. 访问控制

Oracle 连接管理器也包含可以用来控制客户端访问 TCP/IP 环境里的指定服务器的特性。通过指定某些过滤规则，能够允许或限制特定的客户端基于下列的标准来访问一个服务器。

- 源主机名称或客户端的 IP 地址
- 目的主机名称或服务器的 IP 地址
- 目的数据服务名称

这样，通过 NET8 访问控制，我们就可以实现对任何一个客户端的访问权限控制。可以禁止某些 IP 访问所有的数据库，或是允许某个 IP 访问某个数据库，还可以限制某些 IP 只能访问制定的数据库服务名称。

3. 多协议支持

Oracle 连接管理器同时也提供多协议支持，使客户端和服务器能够通过不同的网络协议互相通信。NET8 可以遍历所有可安装和支持的网络协议栈。实际上，网络协议支持的协议栈的数目仅受到特定节点的硬件、内存和操作系统的限制。

也就是说，如果客户端和 Orace 连接管理器之间是通过 SPX/IPX 协议来连接的，那么 Oracle 连接管理器可以通过另外一种协议（如 TCP/IP）来与数据库服务器进行连接。

24.7 域

域是机器和网络服务的逻辑组。在每个组里，所有的名称必须是惟一的，但是域之间的简单名称可以重复。

管理区域包括一个或更多域，用来分担管理责任。

网络域和许多操作系统使用的文件目录很相似，它们都是层次结构的，但是跟文件系统不一样的是，网络域可以对应可以不对应网络中的任何数据库或其他对象的物理排列。它们是简单的命名空间，是为避免命名空间冲突而设计的。

24.8 本章小结

本章主要讲述了 Oracle Net Manager 的基本特征、结构及组成、OSI 网络架构、Oracle Net Manager 网络堆栈段结构、Oracle 连接管理器以及域的概念和区域数据库等。Oracle Net Manager 是 Oracle 服务器进行网络连接的最为主要的管理和设置界面，通过它用户可以方便的配置复杂的网络连接，Oracle Net Manager 主要可以管理监听器，概要文件以及 Oracle 的网络连接方式等。Oracle Net Manager 网络堆栈段结构依照 OSI 网络架构，并对它进行了重新开发，主要分为客户端应用程序层、Oracle 调用接口、双任务公共层、TNS 层、Oracle 网络协议转接层和具体网络协议层，这使得它能更好的满足 Oracle 网络连接的需要。

第 25 章 Oracle网络服务配置

本章的学习目标：

- 掌握监听器的配置方法以及各个参数的意义
- 掌握客户机连接服务器的方法以及配置过程
- 熟悉多线程服务器的概念及设置方法
- 了解网络安全有关知识

25.1 配置监听器

监听器是一个驻留在服务器上的独立进程，用来监测所有连接到数据库上的信息。一个监听器使用一个或多个监听协议。所有的监听器协议信息都存放在 Listener.ora 文件中。监听器的默认名称为 LISTENER，默认网络协议为 TCP/IP，协议端口为 1521。在 Oracle 9i 中默认为 IPC 的 KEY 值为 EXTPROC0。

通常状况下，服务器需要有多个监听器来监听，这样可以大大平衡一个监听器的工作负载，降低了单个监听器失败对工作的影响，从而提高了服务器的可靠性。你可以通过 NET8 CONFIGURE ASSISTANT 来进行配置，你也可以通过 NET8 ASSISTANT 进行配置。Oracle 推荐你使用 NET8 CONFIGURE ASSISTANT 来进行配置。当然你也可以手工改动 LISTENER.ORA 文件来进行配置。我们不建议初学者这样做。因为任何符号错误甚至空格的存在都可能使配置失败。下面我们以 NT 平台为例介绍一下监听器的配置过程：

（1）启动 NET8 ASSISTANT。

（2）在 NET8 ASSISTANT 界面中单击 Oracle NET 配置，展开目录。

（3）打开“本地”目录，选中“监听程序”。

（4）单击左边工具条中的“+”号，弹出“选择监听程序名称”对话框。在“监听程序名称”处输入监听器名称，但要注意所输入的监听器名称不能与已经存在的监听器名称相同，这里选择默认值，即 LISTENER。单击“确定”按钮。

（5）单击“添加地址”按钮。

（6）在“协议”选项中选择 TCP/IP，此为默认选项，也可以选择其他协议，如“使用带 SSL 的 TCP/IP”、“IPC”、“NMP”。端口中填入端口值，但要注意所填的端口值不能与已经存在的端口相同，这里选择 1522。

（7）单击菜单“文件”->“保存网络配置”命令，这样一个新的监听器就配置完毕。LISTENER1 在 LISTENER.ORA 文件中的配置如下：

```
LISTENER1=
(DESCRIPTION=
(ADDRESS=(PROTOCOL=TCP)(HOST=azure)(PORT=1522))
)
```

25.2 本地命名服务器配置

本地命名服务器方法为 Oracle 网络连接中最为常用的方法，下面我们来介绍它的配置方法：

25.3 主机命名法

主机命名法（Host Names）一般适用用较小型网络。它配置简单方便，在实现应用中。通常用它来作为实验型数据库的配置。主机命名法必须遵循以下规则：

- 必须使用 TCP/IP 网络环境；
- 定义在服务器上的监听器的监听端口为 1521；
- 必须有一个外部命名服务，如 DNS（DOMAIN NAMES SERVER）或在客户机上存在 HOSTS 文件，如 NOVELL 目录服务（NDS）、网络信息服务（NIS）等。
- 监听器的 GLORAL_DBNAME 参数必须与服务器的计算机名相同

默认状态下，Oracle 在尝试使用本地命名法和 Oracle 命名服务器法以后，再尝试使用主机命名法。如果想修改它们的顺序，可以修改 SQLNET.ORA 文件当中的 NAMES.DIRECTORY_PATH 参数。也可以使用 NET8 ASSISTANT 来配置。

主机命名法不需要任何客户端的配置，只要客户机运行在 TCP/IP 协议环境下即可使用此方法连接到服务器。检查 TCP/IP 连接方法可使用 PING 命令。配置全局数据名为主机名。

25.4 Oracle命名服务器配置

DBA 在日常管理中会发现，管理繁多的服务命名是一项较为繁杂的事情，尤其在大型网络里，你可能会忘记 TNSNAMES.ORA 文件的拷贝具体对应哪一个网络服务名，更为糟糕的是，服务名在服务器端更改以后，每个客户机也要做出相应的更改，在一个大型网络里其繁杂程度可想而知。在这种情况下，我们可以考虑使用 Oracle 命名服务器，Oracle 命名服务器掩盖了繁杂的服务命名，客户机只需要知道一个和很少的 Oracle 命名服务器信息就可以连接到目标数据库，Oracle 命名服务器一般使用在比较大型的网络。

25.5 多线程服务器配置与网络安全

使用多线程服务器可以降低单个用户对系统资源的平均消耗，因此可以应用在一些查询量不太大且用户较多的网络里。Oracle 网络安全机制用来确保数据在网络传输中不被“偷看”或者修改，从而为企业数据安全提供可靠的保障。我们将在下面分别介绍他们的简单原理及设置。

25.5.1 多线程服务器配置

单进程 Oracle（又称单用户 Oracle）是一种数据库系统，一个进程只执行一个用户的程序，当用户从终端断开程序时，进程被中止。

下面是一个专用服务进程响应用户程序的大致过程：

（1） 客户端应用程序向 Oracle 例程发出一个连接请求。

服务器上的监听程序探测到此用户请求并生成一个专用服务进程来对用户登录信息加以确认。

（2） 用户执行查询操作。

（3） 专用进程执行用户查询操作中的所有源代码程序。

（4） 服务器进程执行 SQL 语句并决定此 SQL 语句的动作是什么，比如 SQL 语句含 SELECT 则执行查询操作。如含有 INSERT 则首先改变高速缓冲区的内容并决定何时启动 DBWN 进程对数据文件进行写操作。

由以上可知：专用进程适合于查询数据量比较大的用户连接方式。

1.考虑使用多线程服务器

2. 配置多线程服务器

25.5.2 高级网络安全

Oracle 高级安全机制提供了一套综合安全特性，来保护企业内部网络及与 INTERNET 连接的扩展联合网络。并且，还提供了一个集成网络加密和认证方法的单一源点、单一签名服务及安全协议。通过集成工业标准，Oracle 高级安全机制为 Oracle 网络及更高层网络提供了前所未有的安全保障。

1. 安全威胁

在分布环境下，随着分布数据量的增加，也带来了严重的安全威胁，包括下列各种威胁：数据篡改；数据偷听及盗窃；伪造用户身份；管理多个口令。

- 数据篡改
- 数据偷听及盗窃
- 伪造用户身份

2. 数据保密性设置

Oracle 高级安全机制保证在采用以下两种加密算法进行数据传输时，数据不被泄露。你可以选择两种加密方式----RSA 加密算法和 DES 加密算法。

- RSA 加密算法
- DES 加密算法

25.6 本章小结

本章主要介绍了 Oracle 网络服务服务器配置方法和过程，多线程服务器的配置和高级网络安全等。Oracle 网络服务服务器配置包括客户端配置和服务器端配置，配置方法有本地命名法、主机命名法和 Oracle 命名服务器方法。本地命名法最为常用，主机命名法配置最为简单，几乎不用在客户端做任何改动，Oracle 命名服务器方法适合大型网络应用，其配置过程也最为复杂。当系统运行于 OLTP 状态下，会有大量的用户对数据库进行访问，每个用户只是对他进行简单的操作，此时可考虑使用 Oracle 多线程服务器。Oracle 高级安全机制用来保护企业内部网络及与 INTERNET 连接时数据不被非法入侵，从而提高了数据的安全。

第 26 章 出错处理

本章的学习目标

- 熟悉网络连接中经常出现的连接失败
- 可以动手进行简单的连接故障排除
- 学会使用日志和跟踪来进行连接错误分析

26.1 服务器段异常处理

网络联接发生失败时，很可能是服务器有异常出现。当此情况发生时，可以按照以下几个常用的方法检查。

（1） 检查数据库是否已经启动

当客户机登录数据库时，发现不能连接，可能是数据库服务器还没有启动。因此确信连接时数据库服务器已经处于运行状态，且每个用户都可以登录。

（2） 检查监听器

监听器是运行在 Oracle 服务器端的。为确保监听器能够监听所有的服务，可以使用 LSNRCTL 工具来查看相应信息。

（3） 检查用户权限

当用户登录时，如果用户没有 CREATE_SEEEION 权限，用户登录将被拒绝。

（4） 检查服务器端协议转接器。

26.2 命名服务器异常出理

命名服务器出现异常时，可以在 NAMESCTL 工具中检查命令服务器的状态及设置情况：

```
c:\>namesctl
namesctl>status
```

26.3 客户机异常处理

客户机异常发生时,通常是客户机设置错误或是不能连接到监听器,下面即是两个例子。

（1） 检查客户端是否能连接到监听器，你也可以使用 TNSPING 工具来确认。

（2） 检查 NETWORK 文件位置

26.4 NET8 日志文件

在监听器的一般参数设置中，可找到事件记录和跟踪功能设置。

使用事件记录和跟踪

使用“事件记录和跟踪”选项卡可以启用监听程序的事件记录和跟踪。Oracle Net Manager 使您能够指定监听程序日志文件被写入的名称和位置。

禁用事件记录

选择此选项可以禁用监听程序上的事件记录。

启用事件记录

选择此选项可以启用监听程序上的事件记录。一旦启用，就可以指定写入日志信息的位置。日志文件输入将写入监听程序日志信息的目录路径和文件名。请注意该文件的扩展名总是.log。默认名称是 listener.log。在 UNIX 上，默认目录是$Oracle_HOME/network/log，在 Windows 平台上的默认目录是 Oracle_HOME/network/log。下面是记录的一段服务器端日志信息：

26.5 NET8 跟踪文件

通过 Oracle Net Manager，你可以在客户机和服务器上设置启用或禁用跟踪功能，默认状况下，系统为禁用跟踪。

- 禁用跟踪 选择此选项以禁用监听程序上的跟踪，此为默认设置。
- 启用跟踪

选择此选项以启用监听程序上的跟踪。一旦启用跟踪，您就可以指定需要的跟踪级别，并通过更改默认跟踪文件来指定写入跟踪信息的位置。

- 跟踪级别

从列表中，选择跟踪级别：OFF—未启用跟踪功能。OFF 为默认设置。USER—将跟踪设置为用户的相应级别。跟踪已标识用户导致的错误条件。ADMIN—将跟踪设置为数据库管理员的相应级别。跟踪已标识特定的安装问题。SUPPORT—将跟踪设置为客户支持人员的相应级别。跟踪文件可能会变的很大。Oracle Corporation 建议在不进行网络问题诊断时关闭跟踪功能。

跟踪文件 输入监听程序跟踪信息将写入的目录路径名和文件名。请注意该文件的扩展名总是.trc。默认名称是 listener.trc。在 UNIX 平台上，默认目录是$Oracle_HOME/network/trace，在 Windows 平台上的默认目录是 $ Oracle_HOME/network/trace。

通常情况下，使用其他方法不能排除网络故障时才使用 Oracle 的跟踪设置。

26.6 本章小结

本章主要讲述了在网络连接中可能遇到的网络连接失败,如果遇到了客户端不能连接服务器，你要根据 Oracle 的提示信息，确定是客户端错误、中间件错误还是服务器端错误，另外，你还可以设置日志和跟踪开帮助查找错误原因。