

简单的

解决方案

高级的

使用技巧

Oracle 8TM

使用指南

David Austin 著

周生炳 译

que®



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

Oracle 8 使用指南

Using Oracle 8

David Austin 著

周生炳 译



電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

内 容 提 要

本书详尽地介绍了 Oracle 8 数据库所完成的全部功能,是一本很有实践指导意义的著作。全书共分九大部分 27 章,其内容包括:建立 Oracle 8 数据库、定制数据库、管理数据和索引段、管理用户和数据库安全、备份 Oracle 数据库、调整数据库及其应用程序,以及使用其他 Oracle 工具和选项等等。

Authorized translation from the English language edition published by Que Corporation Copyright
© 1998.

que

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from the publisher.

SIMPLIFIED CHINESE language edition published by Publishing House of Electronics Industry. Copyright
© 1999.

本书中文简体专有翻译出版权由美国 Que 公司授予电子工业出版社。未经许可,不得以任何手段和形式复制或抄袭本书内容。版权所有,侵权必究。

原 书 名: Using Oracle 8

书 名: **Oracle 8 使用指南**

著 者: David Austin

译 者: 周生炳

责任编辑: 陆伯雄 杨宝珍

印 刷 者: 北京市顺义县天竺颖华印刷厂印刷

出版发行: 电子工业出版社出版、发行

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070

URL: <http://www.phei.com.cn>

经 销: 各地新华书店经销

开 本: 787×1092 1/16 印张: 34.5 字数: 786 千字

版 次: 1999 年 3 月第 1 版 1999 年 3 月第 1 次印刷

印 数: 5000 册

书 号: ISBN 7-5053-5104-4
TP·2544

定 价: 50.00 元

著作权合同登记号 图字: 01-98-1676

凡购买电子工业出版社的图书,如有缺页、倒页、脱页者,本社发行部负责调换
版权所有·翻印必究

第 1 章 关系数据库和 Oracle 8 简介

什么是数据库管理系统?

数据库可以定义为以某种方式组织使之可以检索和利用的信息的集合。数据库管理系统(DBMS)可以进一步定义为可以用来管理与数据库相互作用的工具。

大多数 DBMS 完成下述功能:

- ◆ 存储数据
- ◆ 创建并维护数据结构
- ◆ 允许多个用户并发访问
- ◆ 加强安全性和保密性
- ◆ 允许提取和操作已存储数据
- ◆ 实现数据录入和数据加载
- ◆ 提供对指定数据快速提取的高效索引机制
- ◆ 提供不同记录的一致性
- ◆ 通过备份和恢复过程保护存储数据免遭丢失

已经开发出若干不同类型的 DBMS 支持这些要求。这些系统粗略分为以下几类:

- ◆ 层次 DBMS 以树形结构存储数据,它假定数据之间的父—子关系。树的顶端称为根,可能有很多后代。后代又有任意多个下一代,等等。层次数据库现在已经消失。
- ◆ 网络 DBMS 以记录和链接的形式存储数据。这个系统使多对多关系比层次 DBMS 更灵活。网络 DBMS 非常快而且存储效率很高。网络数据库管理系统允许复杂数据结构但非常不方便并要求繁重冗长的设计。航班订座系统就是这种 DBMS 系统的一个例子。
- ◆ 关系 DBMS(RDBMS)可能具备数据库可能有的最简单结构。RDBMS 中,数据组织成表,而表又由记录组成,记录由字段(域)组成。每个字段对应一个数据项。如果它们有一个或多个共同字段,两个以上的表可能链接(联结)在一起。RDBMS 易于使用并在过去十年中流行开来。它们一般在低档计算机上使用。但是,过去几年间,它们的使用已经扩展到更强大的计算机系统。Oracle, Informix 和 Sybase 就是市场上流行的一些 RDBMS。

Oracle 8 存储对象在关系表中

Oracle 8 是一个对象关系数据库管理系统,允许在表中存储对象,类似在 RDBMS 系统中存储数字和词。

- ◆ 面向对象的 DBMS 用于处理数字和词之类的数据。近几年,正在出现面向对象的 DBMS。这些系统可以处理视频、图像、图片等等之类的对象。

Oracle 数据库文件

Oracle 数据库物理上驻留在多个文件中。图 1.1 表示一个 Oracle 数据库的物理结构。

■ 初始化参数文件

参数文件,一般记为 INIT.ORA,包含初始化参数,它们控制数据库访问和数据库实例的行为和特性。你可以用你最喜欢的编辑器编辑这个文本文件。

更改参数文件

只能在启动期间由实例读取初始化参数文件。初始化文件中的所作的任何更改只能在关闭并重新启动实例后生效。

Oracle 在 \$ ORACLE-HOME/dbs 目录中提供一个样本 INIT.ORA 文件。\$ ORACLE-HOME 是顶级目录,Oracle 软件安装在其下面,它不必是用户 Oracle 的主目录。一个实例的参数文件的缺省名是 initsID.ora,其中 SID(System Identifier)是一个字符串,唯一地标识系统的实例。

可以用 Server Manager 的启动命令的 PFILE 参数取代缺省名。该文件中的 IFILE 参数允许对同一个实例嵌套多个初始化文件。

■ 控制文件

控制文件包含关于数据库的物理结构和状态的有关信息,包括数据文件总数,日志文件,重演日志组,重演日志成员,数据库正写入的当前重演日志,每个数据文件和联机重演日志文件的名称和位置,已存档日志历史,等等。从 Oracle 8 开始,控制文件还包括数据库备份的有关信息。

Oracle 更新控制文件

Oracle 在控制文件中自动记录数据库的任何结构变化——例如,在该控制文件中数据文件的添加/删除。Oracle 实例利用操作期间的各种状态信息更新控制文件。

control_file 初始化参数指定数据库控制文件的名称和位置,我们强烈建议在 control_file 初始化参数中指定多个文件将控制文件镜像到多个位置。V \$ CONTROL_FILE 数据字典视图包含数据库控制文件的有关信息。

V \$ CONTROL_RECORD_SECTION 动态性能视图包含控制文件的详细结构信息,这个视图给出包含在控制文件中的所有记录的有关信息。

■ 数据文件

Oracle 数据库把用户信息存储在物理数据文件中。数据文件包括表、索引、簇、序列、数据字典、回滚段、临时段等。在逻辑级,Oracle 利用表空间(一个或多个数据文件组)管理空

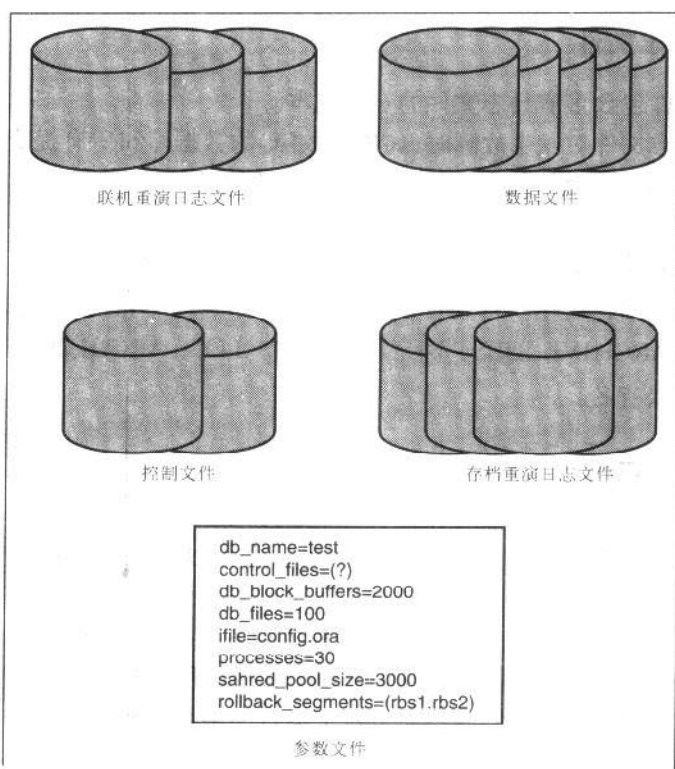


图 1.1 Oracle 数据库系统

间。创建一个 Oracle 数据库时,它只有一个表空间:SYSTEM。必要时,以后添加其他表空间和相关数据文件。

创建数据文件所属的表空间时,你可以指定数据文件的名称,位置和长度。Oracle 利用控制文件存储数据文件的名称和地址。使用数据字典视图 V \$ DATAFILE 和 DBA_ DATA_ FILES 检索数据文件的有关信息。

■ 重演日志文件

Oracle 在重演日志文件中记录数据库的所有变化,并在发生故障时,利用重演日志文件的内容再次产生事务变化。Oracle 数据库拥有两个或多个重演日志文件。Oracle 允许镜像重演日志文件,这样,一个重演日志组包含一个或多个文件(成员)。Oracle 同时写入一个重演日志组的所有成员。Oracle 实例以循环次序写入一个重演日志组的所有成员,即写入一个重演日志组并在前一个满时写入下一下。当最后一个重演日志组满时,转向第一个。

参见 关于重演日志文件的详细信息内容见第 6 章“数据库管理结构”。

创建数据库期间,可以指定重演日志文件的名称、位置和长度。V \$ LOGFILE 数据字典视图包括重演日志文件的有关信息。还可以使用 ALTER DATABASE 命令来添加,删除和重新定位重演日志文件。

■ 存档重演日志文件

存档日志文件包含重演日志文件的一份副本。

存档重演日志文件对发生故障时(如磁盘故障)的数据库和所有提交事务的恢复非常有用。当 Oracle 数据库在存档日志方式下运行时,必须在其重新使用之前存档最近填写的重演日志文件。

可以设置初始化参数 LOG_ARCHIVE_START TO TRUE 或在实例启动后执行存档日志启动命令使存档自动化。自动存档时,ARCH(存档程序)过程将已填写的重演日志文件复制到 LOG_ARCHIVE_DEST 确定的目录中。LOG_ARCHIVE_FORMAT 参数定义存档日志文件的缺省名。

理解数据库实例

Oracle 数据库把数据存储在物理数据文件中,并通过一组操作系统进程控制用户访问这些文件。在实例启动期间启动这些进程。因为它们悄悄地工作,不与用户直接交互,因此称之为后台进程。为了实现高效的数据操作和各种进程间的通信,Oracle 使用共享内存,称为共享全局区(Shared Global Area, SGA)。这些后台进程和共享内存段一道称为一个 Oracle 实例。在并行服务器环境中,在不同机器上运行的多个实例可以访问一个数据库。

一个 Oracle 实例由下述后台进程组成:

- ◆ 进程监控进程(PMON)。在一个用户进程异常中止后,这个后台进程完成,返回到留在后面的未提交事务,并释放不再存在进程占用的资源。
- ◆ 数据库写入程序进程(DBWR)。为保证高效和并发的数据操作,Oracle 不允许用户进程直接修改磁盘上的数据块。需要修改或在其中插入数据的块首先被取到一个称为高速缓冲区的缓冲池中。这些块由 DBWR 后台进程批量写入磁盘中,这样, DBWR 是写访问 Oracle 数据文件的唯一进程。
- ◆ 日志写入程序进程(LGWR)。只要一个 Oracle 进程修改一个 Oracle 数据块,则也把这些修改写入重演日志缓冲区。把重演日志缓冲区写入联机重演日志文件是 LGWR 进程的责任。这个进程批量读取重演日志缓冲区的内容,并顺序地把它们写入联机重演日志文件。注意, LGWR 是写入联机重演日志文件的唯一进程。Oracle 的事务提交算法保证事务被提交时重演日志缓冲区的内容注入联机重演日志文件。
- ◆ 系统监控进程(SMON)。这个后台进程处理诸如释放排序空间及在一个大范围内合并相邻自由范围。SMON 还在实例恢复(异常停机后的实例启动)期间完成事务恢复。在并行服务器环境中,它还探测和完成另一个失败实例的实例恢复。
- ◆ 存档程序进程(ARCH)。数据库处于存档日志方式并实现自动存档时,启动这个进程。它把最近填入的联机重演日志文件复制到指定的备份目的地。

总是启动的 Oracle 后台进程

LMON, PMON, DBWR 和 LGWR 进程对于实例总是存在的,其他进程必须通过设置相关的初始化参数启动。

- ◆ 检测点进程(CKPT)。在检测点期间, DBWR 进程把所有修改块写入磁盘。DBWR 还通知 LGWR 用检测点信息更新所有数据文件的头信息。因为包含大量数据文件的数据库对 LGWR 来说是耗时的任务, 所以在实例启动期间启动 CKPT 进程帮助 LGWR 更新检测点期间的文件头。只有 CHECKPOINT_PROCESS 参数置于 TRUE 或数据库数据文件个数超过某个数时, 这个进程才启动。
- ◆ 恢复程序进程(RECO)。这个进程恢复分布式数据库环境中可疑的事务。初始化参数 DISTRIBUTED_TRANSACTION 设置大于 0 时, 启动这个进程。
- ◆ 并行查询从属进程($P \times \times \times$)。在有利的条件下, Oracle 可以把若干个特定进程间的操作分开而减少某个 SQL 操作的执行时间。这个进程用于 SQL 语句的并行执行, 称为并行查询从属。
- ◆ 快照进程(SNPn)。参数 JOB_QUEUE_PROCESSES 大于 0 时, 启动快照或作业队列进程。这些进程执行作业队列中的作业, 刷新为自动刷新而配置的快照, 等等。
- ◆ 调度程序进程($D \times \times \times$)。Oracle 支持某些操作系统中的多线程服务器。这些进程接收用户请求并把它放在请求执行的队列中, 它们还收集调度程序队列的执行结果, 并把它们传递给用户。
- ◆ 共享服务器进程($S \times \times \times$)。在多线程服务器环境中, 专用服务器进程执行来自请求队列的 SQL 操作, 并把结果放回相应的调度程序队列。

如果运行 Oracle 的并行服务器选项, 你还可以在每个实例中看到下面的后台进程:

- ◆ 锁进程(LCKn)。该 Oracle 并行服务器进程协调本地和远程实例的所有锁请求。它与用户进程和锁监督程序通信。
- ◆ 锁监控程序进程(LMON)。它负责 Oracle 并行服务器环境下的实例启动和关闭期间重新配置 Integrated Distributed Lock Manager(IDLM)。它还在用户进程非正常结束之后, 进行锁清理。
- ◆ 锁监督程序进程(LMDBO)。它是 IDLM 的一部分, 处理远程实例请求本地实例拥有锁的所有锁请求。

图 1.2 表示 Oracle 实例的成分:SGA 和后台进程。

■ 启动和终止实例

Oracle 数据库只有被一个 Oracle 实例打开时才对用户是可访问的。在 Oracle 并行服务器环境中, 一个 Oracle 数据库可接受多个实例访问。每个实例都有自己的后台进程和 SGA。实例启动操作包括启动所有后台进程并分配共享内存区(见图 1.3)。

谁可启动和关闭 Oracle 实例?

只有具备必要的 OS 权限或指派担任 OSOPER 或 OSDBA 的用户才能打开一个 Oracle 实例。

怎样启动实例

1. Oracle 启动所有后台进程并分配 SGA。在此期间, Oracle 读取初始化参数文件。
2. Oracle 读取控制文件并关联控制与实例。它从上次停机/故障中探测数据库的状况。

3. Oracle 读取所有数据文件和重演日志文件的文件头。它保证所有数据文件的一致性。如果在一次失败或异常停机之后启动实例,那么 Oracle 还应用上次成功的检测点以来的所有重演日志。这一步结束后,Oracle 数据库才对用户是可访问的。

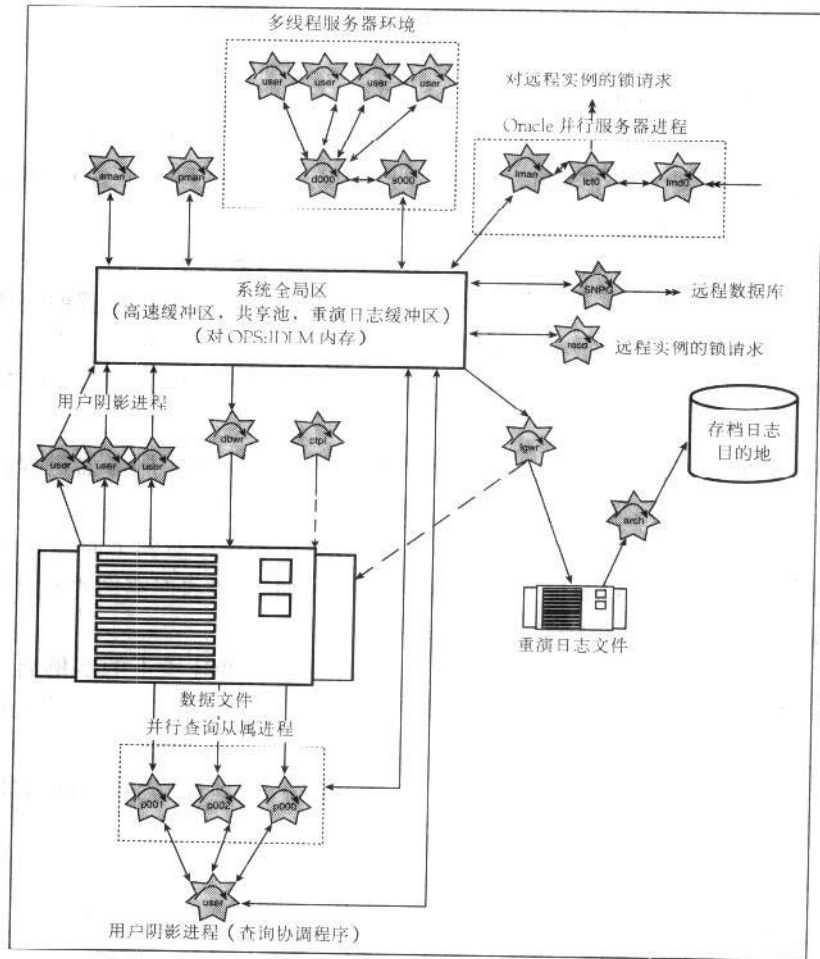


图 1.2 Oracle 实例由 SGA 和后台进程组成

如果在一次失败/异常关闭之后启动实例, Oracle 必须完成未提交事务的回滚操作。在数据库是打开并对用户可用时, 由后台进程 SMON 执行这个操作。

一个 Oracle 实例结束关闭数据库, 卸载它, 然后删除 SGA 和后台进程。结束提供三种方式: 正常、直接和异常。正常关闭和直接结束最常见, 而用异常结束就应小心。在正常结束时, Oracle 等待所有用户断开, 把所有修改过的数据写入数据文件, 然后更新文件头, 联机重演日志文件和控制文件。直接结束断开所有用户, 然后继续直接关闭。异常结束删除所有后台进程和 SGA, 清理工作在下次启动时进行。

表 1.1 列出启动和终止一个 Oracle 实例的 Server Manager 命令

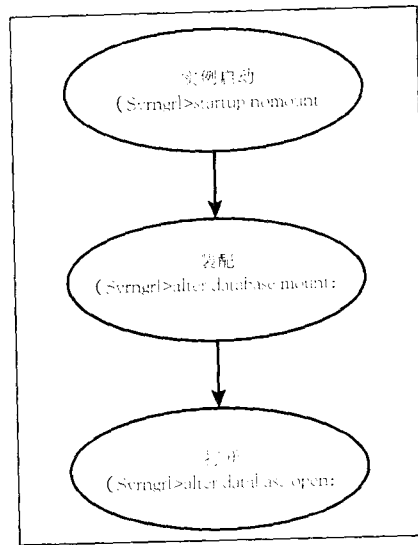


图 1.3 Oracle 实例启动包括三步

表 1.1 Server Manager 启动和关闭命令

命令	描述
Startup 和 Startup open	使用缺省参数文件启动实例, 安装数据库并打开它。
Startup pfile = file	使用指定的参数文件启动实例
Startup nomount	分配 SGA 并启动后台进程; 不装配和打开数据库
Startup mount	分配 SGA 并启动后台进程, 装配数据库; 但不打开数据库
alter database mount	用启动 nomount 命令启动实例后, 装配数据库
alter database open	用启动 mount 命令装配数据库以后打开它。
shutdown	所有用户断开以后关闭实例(正常结束)
shut down immediate	不允许启动新事务, 回滚到未提交事务并关闭实例
shutdown abort	立即删除 SGA 和后台进程

参见 关于用 Oracle Enterprise Manager 启动和终止数据库实例的内容见第 4 章“使用 OEM 的基本管理任务”。

Oracle 8 的工具

Oracle 提供许多工具用于应用程序开发并实现管理功能:

- ◆ Oracle Enterprise Manager(OEM)
- ◆ SQL* Plus
- ◆ PL/SQL
- ◆ Net8
- ◆ Developer 2000
- ◆ Precompilers

■ Oracle Enterprise Manager(OEM)

Oracle Enterprise Manager 是一个图形系统管理工具,允许你在复杂的数据库环境中执行多项任务。OEM 有多个组件,其中一些组件,如 Oracle Expert 和 Performance Manager,是分开计价的。第 4 章解释怎样使用这些组件。OEM 的主要组件如下(见图 1.4):

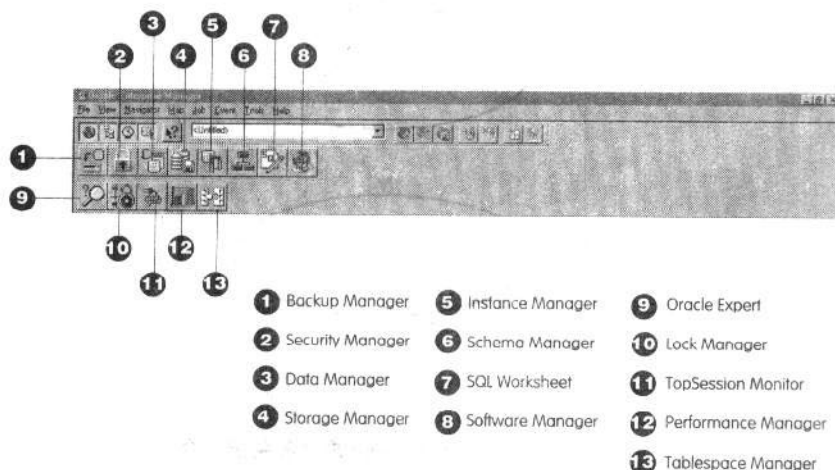


图 1.4 Oracle Enterprise Manager 由几个模块组成

- ◆ Backup Manager 负责与数据库有关的备份和恢复操作,允许与 Oracle 8 的高级备份和恢复实用程序 Recovery Manager 交互。
- ◆ Security Manager 允许执行用户管理任务,如添加、更改和撤消用户、任务和配置文件。
- ◆ Data Manager 是一个数据传输和加载工具,允许调用卸出、导入和加载实用程序。利用 Export 实用程序在 Oracle 中提取数据,这些数据与操作系统无关。卸出的数据以后又可装入另一个 Oracle 数据库或同一个数据库。还可以使用 Export 作为数据库的逻辑备份。Loader 实用程序用于从文本文件把数据插入 Oracle 数据库。
- ◆ Storage Manager 允许执行数据库空间管理任务,如创建、更改和撤消表空间。还可以创建联机/脱机并撤消回滚段。
- ◆ Instance Manager 实例管理,用户会话和可疑事务。可以用它启动和关闭 Oracle 实例。在 Oracle 并行服务器环境中,可以管理多个实例,还可在实例启动时管理初始化参数文件。
- ◆ Schema Manager 允许执行 Data Definition Language(DDL)操作,创建、更改、撤消和查看数据库对象,如表、索引、簇、触发器及序列。
- ◆ SQL Worksheet 的行为非常象 SQL* Plus 会话,可以用它输入并执行 SQL 命令。
- ◆ Software Manager 允许在分布环境中进行管理并使数据库管理任务自动进行。
- ◆ Oracle Expert 用于调整实例和数据库性能。它产生一个可自动实现改进性能的建议清单。
- ◆ Lock Manager 查看实例中的锁。对于分析挂起会话和其他类似情况,它是有益的工具。

- ◆ TopSession Monitor 允许监控活动的用户会话并查看用户资源利用。这个信息还可用于查找低速性能。
- ◆ Performance Manager 监控 Oracle 实例性能,提供各种性能统计的图形表示。
- ◆ Tablespace Manager 允许在对象级查看表空间中的空间利用,还可得到数据库中未用空间的有关信息。
- ◆ Oracle Trace 执行 SQL 语句以改进系统性能。

■ SQL* Plus

SQL* Plus 不能启动或终止一个实例

数据库管理员不能利用 SQL* Plus 启动或关闭 Oracle 实例。

终端用户和 RDBMS 之间唯一的接口是 Structured Query Language (SQL)。用户用来与 RDBMS 交互的所有其他应用程序和工具都起转换器/解释器的作用。这些工具应用户请求产生 SQL 命令并把已生成的 SQL 命令传送到 RDBMS。

SQL* Plus 是 SQL 的 Oracle 版,它是最常用的 Oracle 工具之一。SQL* Plus 使用户指示 Oracle 实例完成下述 SQL 功能:

- ◆ 数据定义或 DDL 操作,如创建、更改、撤消数据库对象
- ◆ 选择或检索存储数据的数据查询
- ◆ 插入、更新和删除数据的数据操纵或 DML 操作
- ◆ 在数据库之间存取和传输数据
- ◆ 允许用户交互地输入数据
- ◆ DBA 功能或数据库管理任务,如管理用户(创建、更改、撤消用户),管理空间(创建、更改、撤消表空间),以及备份与恢复。多这些基本 SQL 功能外,SQL* Plus 还提供几种编辑和格式功能,使用户能以报表格式打印查询结果。

建立 SQL* Plus 环境

SQL* Plus 有许多高级功能,可以用于以可视格式表达数据。可设置各种环境变量以控制 SQL* Plus 输出查询的方式。表 1.2 列出建立这种环境的一些最常用的命令,用户可在 SQL* PLUS>光标下输入命令。

表 1.2 SQL* Plus 环境命令

命令	描述
Set Pagesize	设置每一页的行数
Set linesize	设置一行中的字符个数
Set newpage	设置页面之间空白行的数目
Set pause	使 SQL* Plus 在每一页前暂停
Set array	设置一次被检索的行数
Set feedback	显示一个查询处理的记录个数
Set heading	打印报表开始处的标题
Set serveroutput	显示 DBMS_OUTPUT.PUT_LINE 存储过程的输出
Set time	显示定时统计
Set term	取消执行一条文件中的命令产生的输出

自动设置环境

启动 SQL* Plus 时,还可使用 LOGIN.SQL 和 GLOGIN.SQL 文件设置当前会话的环境。

■ PL/SQL

PL/SQL 是 Procedural Language/Structured Query Language. 允许用户利用类似于第三代语言,如 C, Fortran 和 COBOL 的结构程序设计构造。PL/SQL 增强了 SQL 的能力,它加入了以下功能:

- ◆ 定义和使用变量
- ◆ 控制程序流(IF, IF...THEN...ELSE 和 FOR LOOP 构造)
- ◆ 使用指针和数组
- ◆ 文件 I/O
- ◆ 函数和过程
- ◆ 移动大量数据的 PL/SQL 表

利用 PL/SQL, 你可以用 SQL 命令操纵 Oracle 数据库中的数据, 并使用结构程序设计构造处理数据。

Oracle 8 在工具中嵌入 PL/SQL

尽管 PL/SQL 可以作为程序设计语言,但它不是 Oracle Forms 和 Oracle Reports 这类 Oracle 工具的一部分。嵌入这些工具中的 PL/SQL 引擎可当成预处理器。

■ NET 8

Net8 以前叫 SQL* Net, 是 Oracle 的联网接口, 允许驻留在不同机器上的各种 Oracle 产品间的通信, 还能在分布式环境中的客户机、服务器和 Oracle 数据库之间通信。在客户机端, 客户机应用程序代码把消息传送到驻留本地的 Net8, 而本地 Net8 通过基本传输协议把消息传送到远程 Net8。这些消息在服务器端被 Net8 接收, 即将其发送给数据库服务器执行。服务器执行请求按照同样路径并响应客户机。图 1.5 表示客户机和服务器利用 Net8 的通信。

Net8 比它的前身 SQL* Net 增强了许多功能, 如连接轮询, 多路复用, 监听程序负载均衡和客户机端的高速缓存网络地址。Net8 向后兼容并可与 SQL* Net 第二版共处。

■ 预编译程序

第三代语言编译程序不识别与 RDBMS 交互所需的 SQL。因此, 如果需要像 C, C++, Fortran, COBOL 之类语言的强大功能和灵活性, 而且希望它与 Oracle 8 RDBMS 交互, 那么必须拥有一个工具, 能把 SQL 语句转换成一种语言编译程序能理解的调用。如图 1.6 所示, 一个预编译程序读取结构化源代码并生成一个语言编译程序能处理的源文件。Oracle

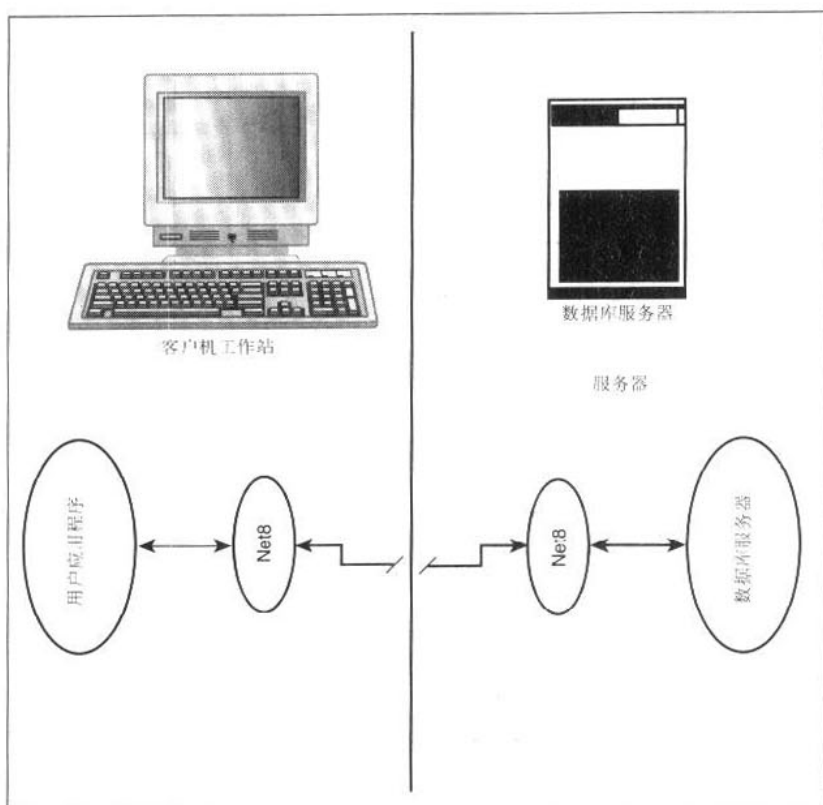


图 1.5 客户机和服务器通过 Net8 相互通信

提供若干个预编译程序, 如 Pro* C, Pro* Cobol, Pro* Fortran 和 Pro* Pascal。

在开发运行时间长的批程序或运行时间至关重要的程序时, 你可以使用预编译程序得到更好的性能。使用预编译程序可完成以下工作:

- ◆ 使用动态 SQL
- ◆ 更好地控制指针和程序流
- ◆ 开发程序库并应用于多个应用程序
- ◆ 并发存取多个数据库的数据
- ◆ 通过分叉进程编写多线程应用程序

■ Developer/2000

Developer/2000 提供完备的工具集开发访问 Oracle 数据库的应用程序。它由创建表单、报表、图表、查询和过程的工具组成, 还允许利用 Web 上现有的和新的应用程序。Developer/2000 包含下述组件:

- ◆ Project Builder。可以跟踪和控制应用程序文档、源代码文件、图表、表单、报表、查询等等。
- ◆ Forms Builder。表单是终端用户与数据库交互的最早和最流行的一种方法。终端用户总体上不熟悉数据库而 SQL 语言可以轻而易举地了解一个基于表单的应用程

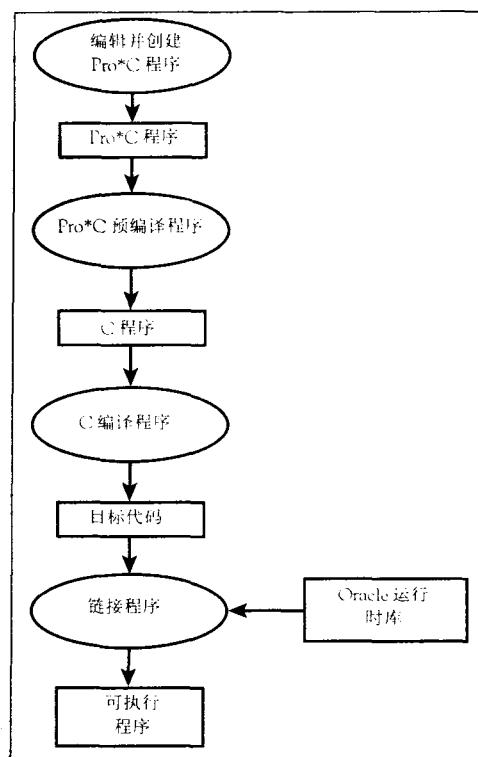


图 1.6 使用预编译程序开发程序

序来访问 Oracle 数据库。Forms Builder 是应用程序开发人员用于开发表单的工具。

- ◆ Report Builder。尽管表单可使用户与数据库进行联机/直接交互,但 Report Builder 可使用户在远程报表服务器上把数据提取请求在一个队列中排队。而报表服务器则与数据库交互产生报表,还可以把报表嵌入其他联机工具,如 Web 浏览器,图形和表单。
- ◆ Graphics Builder。数据的图形和可视表示比原始数据更有效。可以用 Graphics Builder 产生交互式图形显示,还可用它在表单和报表中插入图形。
- ◆ Query Builder。Query Builder 允许在屏幕上以表格形式与数据库交互。查询中涉及的表可在屏幕上看到,用户可用鼠标构造中意的查询。也显示格式化的查询结果。
- ◆ Schema Builder。Schema Builder 是一个图形 DDL(数据定义语言)。可以用它创建、更改和撤消数据库对象,如表,索引,簇及序列。
- ◆ Procedure Builder。Procedure Builder 帮助你交互地建立过程。可以使用它的图形界面创建、编辑、调试并编译 PL/SQL 程序。Procedure Builder 生成的 PL/SQL 程序单位包括包、触发器、函数和程序库。
- ◆ Translation Builder。Translation Builder 使你从 Oracle 或非 Oracle 资源提取可转换的串并完成希望的事务。例如,把 Microsoft Windows (.RC)的 HTML 文件转换成 Oracle 资源文件。

传统上, Developer/2000 支持客户机/服务器体系结构, 其中客户机工具和应用程序驻留在一台机器上(一般是终端用户 PC), 而数据库服务器驻留在另一台机器上。然而, 由于 Web 的庞大丰富, Oracle 引入三级体系结构, 其中引入运行应用程序代码的其他服务器。

安装 Developer/2000 的客户机/服务器或三级体系结构受到高度推荐, 因为工作负载分布在这个结构中的客户机、数据库服务器和应用程序服务器上。此外, 应用程序、Developer/2000 和数据库软件相互独立, 这使数据库更易于维护。SQL*Net 或 Net8 安装在客户机上而数据库服务器则实现两者的连接。

Oracle 8 数据字典

Oracle 存储用户定义的所有对象的有关信息、数据库的结构信息以及它的内部表的有关信息。这些 Oracle 内部表和相关对象统称为数据字典。数据字典由用户 SYS 拥有并永远驻留在 SYSTEM 表空间。

用户可通过数据字典视图利用数据字典中存储的信息。数据库管理员或用户可以使用数据字典查看如下信息:

- ◆ 数据库对象的定义, 如表、分区、索引、簇、视图、快照、触发器、包、过程、函数、序列和同义词
- ◆ 数据库中定义的用户
- ◆ 数据库对象的存储分配和分配给每个用户的限额
- ◆ 完整性约束
- ◆ 数据库链接
- ◆ 权限及角色
- ◆ 被复制对象、快照及其刷新特性
- ◆ 审计信息, 如各种对象的访问模式
- ◆ 作业队列中的作业
- ◆ 数据库中容纳的锁和锁存器
- ◆ 警告和表队列(高级队列)
- ◆ 回滚段
- ◆ SQL*Loader 直接加载
- ◆ NLS 设置

创建数据库的同时创建数据字典表

Oracle 自动地更新不再需要表, 用户不能更新数据字典的任何表。若为 Oracle 和非 Oracle 工具还创建一些数据字典对象, 这些对象用于存储操作、引用和配置信息。

Oracle 数据字典视图可在以下类中定义:

- ◆ 带 DBA 前缀的视图。这些视图包括关于整个数据库的信息。例如, 视图 DBA_TABLES 给出数据库中所有表的信息。这些表缺省地只有 DBA 用户可访问。
- ◆ 带 USER 前缀的视图。USER 视图包含用户拥有的对象的信息, 例如, USER_TA-

BLES 给出用户拥有的对象的信息。

- ◆ 带 ALL 前缀的视图。这些视图包含用户可访问的所有对象的信息。用户可访问的对象包括用户创建的对象和他得到其他用户授权的对象。例如, ALL_TABLES 视图包含用户可访问的所有表的信息。

表 1.3 列出主要的 Oracle 8 数据字典视图。限于篇幅,带 DBA 和 ALL 前缀的类似视图未列出。

表 1.3 主要数据字典视图

视图名	描述
USER_ALL_TABLES	包含用户可用的所有表的描述
USER_CLUSTERS	包含用户创建的簇信息
USER_CONSTRAINTS	包含用户定义的约束信息
USER_DB_LINKS	包含用户创建的数据库链接信息
USER_ERRORS	给出为用户存储的所有对象的所有当前错误
USER_EXTENTS	列出用户拥有对象所使用的范围
USER_FREE_SPACE	列出用户拥有权限表空间中的所有范围
USER_INDEXES	给出用户创建的索引信息
USER_IND_COLUMNS	给出用户在其上创建索引的所有列的名称
USER_JOBS	给出用户拥有的作业队列中的所有作业
USER_RESOURCE_LIMITS	给出用户可以应用的资源限制
USER_SEGMENTS	给出用户拥有的所有段的信息
USER_SEQUENCES	列出用户拥有的所有序列的信息
USER_SNAPSHOTS	给出用户可查看的所有快照的信息
USER_SYNONYMS	给出用户的所有私有同义词的名字
USER_TAB_COLUMNS	给出用户拥有的所有表中所有列的名字
USER_TAB_PARTITIONS	给出用户拥有的所有表分区的信息
USER_TABLES	给出用户拥有的所有表的信息
USER_TRIGGERS	给出用户创建的所有触发器的信息

统计和数据字典

有多个数据字典视图包含带对象统计信息的列。例如, USER_TABLES 视图包含列 NUM_ROWS(表中行数), BLOCKS(表中所有数据块的个数), AVG_ROW_LEN(表中行的平均行长度), 等等。仅当使用 ANALYZE 命令分析对象时,才填充这些列。应定期分析对象以保持最近的统计信息。

动态性能表

Oracle 实例维持关于它的当前配置和活动的全面的信息。数据库管理员通过动态性能视图访问这些统计信息,这些视图中的大多数都建立在内存中象表一样的结构上,称为虚拟表(因为它们并不是真正的表)。它们大多有以 V\$ 开头的名字。这些虚拟表并不要求磁盘存储空间也不存储在任何表空间中。动态性能视图缺省地对 SYS 用户或具有 SYSDBA 角色的用户是可访问的,视图的内容在实例活动期间持续更新。

表 1.4 描述主要的动态性能视图。这些视图是 Oracle 8 提供,有些在 Oracle 7 中不存在。