

ORACLE

网络数据库

管理与应用

贾代平 吴占鳌 © 编著
吴丽娟 刘 岩

石油工业出版社

ORACLE 网络数据库管理与应用

贾代平 吴占鳌 吴丽娟 刘 岩 编著

石油工业出版社

内 容 提 要

本书主要介绍了 ORACLE 数据库系统在日常管理和开发过程中涉及的主要知识,重点介绍了在工程应用中管理、配置和操纵 ORACLE 数据库系统必须具备的专业知识,包括信息系统和 ORACLE 的体系结构、ORACLE 数据库的详细创建和配置过程、DBA 数据库维护的主要方面、数据库事务处理、数据备份和完全恢复方法、开发人员数据库端编程要点等内容。

本书可作为高等院校相关专业学生的教材或参考书,也可供从事数据库管理的工程技术人员参考。

图书在版编目 (CIP) 数据

ORACLE 网络数据库管理与应用/贾代平等编著.

北京:石油工业出版社,2010.10

ISBN 978-7-5021-8033-1

I. O…

II. 贾…

III. 关系数据库-数据库管理系统, Oracle

IV. TP311.138

中国版本图书馆 CIP 数据核字 (2010) 第 181902 号

出版发行:石油工业出版社

(北京安定门外安华里 2 区 1 号 100011)

网 址: www.petropub.com.cn

编辑部: (010) 64523697 发行部: (010) 64523620

经 销: 全国新华书店

印 刷: 石油工业出版社印刷厂

2010 年 10 月第 1 版 2010 年 10 月第 1 次印刷

787×1092 毫米 开本: 1/16 印张: 19.75

字数: 474 千字

定价: 48.00 元

(如出现印装质量问题, 我社发行部负责调换)

版权所有, 翻印必究

前 言

ORACLE 公司有这样一句宣传语：ORACLE Software Powers The Internet（ORACLE 软件驱动互联网）；在 IT 领域还有另外一个词语：ORACLE 帝国。这两种说法都反映了这样一个事实：如今的网络服务和各类信息系统，越来越多地依赖于像 ORACLE 这样的大型数据库系统，同时越来越多的 IT 从业人员的工作与 ORACLE 密切相关。一是随着整个社会信息化水平的提高，越来越多的企事业单位建立了自己的信息系统，并且信息系统的规模有日益扩大的趋势，这给 ORACLE 数据库提供了广阔的施展空间；二是随着用户应用水平的提高，各类信息系统的复杂性迅速增加，原有的中小型数据库难以适应这种复杂的应用环境，越来越多的信息系统需要升级换代。ORACLE 数据库系统在面对这种用户需求时，其产品的安全性、稳定性和可扩展性具有明显的优势。

然而，ORACLE 数据库系统是一个庞大而复杂的计算机软件系统，难以驾驭，若要比全面地掌握这种系统，在工程实践中运用自如，同样是一项艰巨的系统工程。人们一直在思考，如何将 ORACLE 这样庞杂的系统整理出一条清晰的思路，循序渐进、深入浅出地展现给读者。本书是作者多年来思考 ORACLE 数据库系统学习方法的一次总结。历经多年的 ORACLE 专项培训和用户现场的深入讲解，在收集大量学员和用户反馈的基础上，我们总结出了一条清晰而富有层次的学习路径，具体体现在本书的内容安排上。当然，作为一本教材，本书不可能涉及 ORACLE 数据库系统管理和应用开发的每个方面，而是对学习 ORACLE 需要掌握的众多知识点进行了轻重缓急的取舍，重点向读者展现了一条学习 ORACLE 数据库系统的主干线，这是读者学习 ORACLE 需要掌握的根基。掌握了这条主干线，进一步学习一些专题内容或解决工程实践中的各种复杂问题时就会有一个坚实的基础。

为了达到上述目标，我们对全书内容的组织做了精心的规划与设计。虽然目前业界实际投运的 ORACLE 数据库存在多个不同版本，但其核心机制和技术框架是相对稳定的，并且在不同平台下的原理与使用方法也是一致的，因此我们将这些不同版本、不同平台下的共性内容循序渐进地组织为十五章，各章的主要内容分述如下：

第一章首先向读者概要介绍现代网络数据库系统的学习层次、数据库应用环境的软件构架及其主流开发技术，展现 ORACLE 数据库及其应用系统的知识背景。

第二章介绍 ORACLE 数据库的体系结构，使读者对庞大而复杂的 ORACLE 系统从整体构架上有有一个全面的了解，特别是 Database 和 Instance 的区别和联系做了较为详细的说明。

第三章是在熟悉体系结构的基础上，介绍 ORACLE 数据库的动态运行机制，并向读者展示 ORACLE 数据库的参数配置及其详细的生成过程。

第四章介绍 ORACLE 数据库系统运行的网络环境 ORACLE NET，包括网络管理与配置、服务器端和客户端的各类接口需求，这是通过网络访问 ORACLE 数据库的前提。同时还介绍了共享服务器环境的配置和使用。

第五章主要阐述 ORACLE 数据库在物理存储及其管理方面的内容，包括空间分配与存储机制，重点介绍了逻辑存储结构和物理存储结构，涉及表空间、段（Segment）、回滚信息

(UNDO)、闪回 (FLASHBACK) 恢复区等基本概念以及主要操作方法, 并对磁盘阵列 (RAID) 和自动存储管理 (ASM) 等特性做了简要说明。

第六章介绍数据库用户管理及其权限控制的内容, 重点阐述了用户、权限、角色、数据库审计 (AUDIT) 等与数据库安全控制相关的内容。

第七章向读者展示在 ORACLE 数据库中最重要两类对象——表和索引的若干高级用法, 包括表和索引的分类、表和索引的分区处理、物化视图、表复制等内容。

第八章介绍数据库的访问控制机制, 重点阐述了会话 (Session) 许可、事务 (Transaction) 处理、检查点 (Checkpoint)、并发访问的锁 (Lock) 机制等内容, 这一章的内容虽然较为抽象, 但对数据管理和数据处理都是至关重要的。

第九章概要介绍 ORACLE 数据库系统的备份与恢复的主要面貌, 以便读者对数据库备份与恢复技术及其实施策略有一个全貌性的了解。

第十章介绍 ORACLE 数据库的逻辑备份 (导出 Export) 与恢复 (导入 Import), 包括逻辑备份与恢复的升级工具数据泵 (Data Pump) 的使用。

第十一章介绍日志管理和数据库备份的主要内容。首先介绍了日志与数据之间的密切关系, 并展示了日志用于数据恢复的核心机制, 接着详细阐述 ORACLE 数据库的归档运行模式以及各种有效的用户管理的数据备份方法。

第十二章详细阐述数据库在故障情况下的数据库恢复, 包括完全数据恢复 (不丢失数据) 和不完全数据恢复的原理与方法, 并展示了各种典型故障的处理思想和操作步骤。

第十三章是在用户管理的数据库备份与恢复的基础上, 介绍 ORACLE 恢复管理器 (RMAN) 的配置与使用, 这是 ORACLE 物理备份与恢复的主要工具。

第十四章对数据库端的编程知识做出介绍, 主要是 PL/SQL 语言及其编程单元。这部分内容属于数据库应用开发部分, 是对开发人员数据库端编程知识的总结。

第十五章作为数据库管理与应用开发的共同基础, 对工程应用中的数据库建模与设计做出概要介绍, 并介绍了 CDM 和 PDM 模型以及 PowerDesigner 建模工具。

本书由贾代平统稿, 其中第一章、第十五章由吴占鳌编写, 第二章、第三章由刘岩编写, 第四章至第九章由吴丽娟编写, 第十章至第十四章由贾代平编写, 景瑞林、姚剑负责全书的审校。

通过对本书的学习, 相信读者一定会领略到 ORACLE 数据库技术的主线。读者在使用本教材的过程中, 若有任何疑问或技术问题, 敬请与编者联系, 我们将为广大学员提供适时的技术支持。联系邮箱: jiadp@163169.net。

编者
2010 年 7 月

目 录

第一章 现代数据库系统与应用概述	1
第一节 关系模型与数据库回顾.....	1
第二节 现代数据库系统的技术层次.....	3
第三节 数据库应用系统结构.....	6
第四节 企业级应用系统的核心技术.....	8
第五节 数据库技术内容体系.....	9
第二章 ORACLE 数据库系统体系结构	12
第一节 ORACLE 与操作系统平台	12
第二节 数据库和实例	19
第三节 实例的内存结构	20
第四节 后台进程 (BACKGROUND PROCESSES)	26
第五节 前台进程 (FOREGROUND PROCESSES)	28
第六节 ORACLE 数据库的外部结构	29
第七节 体系结构小结	31
第三章 定义实例与创建数据库	33
第一节 实例的四种状态	33
第二节 实例启动和关闭选项	38
第三节 参数文件和口令文件	40
第四节 手工创建数据库的核心过程	42
第五节 使用 DBCA 创建数据库	48
第六节 克隆 ORACLE 数据库	58
第四章 ORACLE 网络的配置与管理	68
第一节 ORACLE 网络服务结构	68
第二节 服务器端的网络配置	71
第三节 客户端的网络配置	75
第四节 配置共享服务器环境	80
第五章 ORACLE 的物理存储及其管理	85
第一节 表空间与存储空间	85
第二节 系统表空间与数据字典	85
第三节 临时信息与临时表空间	87
第四节 回滚信息与回滚表空间	88
第五节 配置数据库表空间	91
第六节 数据库文件管理	96
第七节 其他辅助文件的管理.....	111

第八节	闪回恢复区 FRA	114
第九节	RAID 磁盘阵列存储	117
第十节	自动存储管理 ASM	118
第六章	用户权限控制与访问	119
第一节	用户管理与身份验证	119
第二节	ORACLE 用户的分类	123
第三节	数据库系统的权限	126
第四节	基于角色的权限	128
第五节	用户权限与角色的授予和收回	130
第六节	会话许可与资源限制	133
第七节	数据库审计概述	138
第七章	数据的高可用性存储对象	141
第一节	表与索引概述	141
第二节	大型表的分区	145
第三节	关联表的聚簇	151
第四节	结合索引组织存储数据	154
第五节	临时数据的存储	155
第六节	分布式数据存储	156
第七节	基于表的复制功能	159
第八章	事务处理机制与并发访问控制	162
第一节	事务处理的基本概念	162
第二节	检查点机制	164
第三节	典型的事务处理过程	166
第四节	锁与并发数据处理	167
第九章	ORACLE 数据库备份与恢复概述	171
第一节	备份与恢复基础	171
第二节	备份与恢复的主要数据结构	172
第三节	数据库的备份策略	174
第四节	数据库的恢复策略	177
第十章	导出导入与数据泵	180
第一节	导出和导入	180
第二节	数据泵体系结构	185
第三节	数据泵的使用	187
第四节	可传输表空间	195
第十一章	日志管理与数据库备份	207
第一节	备份与恢复的基本概念	207
第二节	基于事务日志的数据备份与恢复	207
第三节	联机日志的配置与管理	209

第四节	归档日志与运行模式	211
第五节	制作备份的方法	215
第十二章	事务处理与数据库恢复	220
第一节	基于日志恢复一般原理	220
第二节	数据库的故障与介质恢复	222
第三节	归档模式下的介质恢复	223
第四节	不完全介质恢复	225
第五节	典型故障恢复方法	226
第六节	备份与恢复小结	234
第十三章	基于 RMAN 的备份与恢复	237
第一节	恢复管理器概述	237
第二节	RMAN 环境的配置	239
第三节	RMAN 执行备份	244
第四节	RMAN 执行恢复	252
第五节	恢复目录的维护	258
第十四章	PL/SQL 语言与 ORACLE 编程	266
第一节	数据库访问与 SQL 语言	266
第二节	基本数据类型与对象	267
第三节	PL/SQL 的基本语法	276
第四节	PL/SQL 程序性对象	280
第五节	构建 JAVA 环境与对象	286
第十五章	数据库设计与建模基础	294
第一节	数据库设计的基本概念	294
第二节	数据库建模技术	295
附录	ORACLE 专业术语中英文对照	298
参考文献		305

第一章 现代数据库系统与应用概述

数据库理论与技术是现代信息社会的重要技术基础，也是现代信息技术教育的核心内容之一。可以毫不夸张地说，没有数据库技术的推广应用，就没有今天的信息化社会。IT 行业的数据库教育就是要培养熟悉数据库理论与技术，掌握数据库应用系统的设计、开发与维护，满足 IT 产业需求的信息技术人才。

回顾数据库技术的发展历程，数据库应用系统经历了从单机到网络，从网络到 C/S 结构，再从 C/S 结构到多层（MULTI-TIER）结构、分布式结构等多个发展阶段，数据库技术在体系结构、应用环境、系统开发等方面已经发生了巨大的变化。在新的技术背景下，如何循序渐进地学习和掌握数据库技术及其应用，本章从应用技术的角度给出了一个比较全面的概述和一个总体的轮廓。

第一节 关系模型与数据库回顾

一、关系数据库之父

现代数据库理论与系统起源于 20 世纪 60 年代末到 70 年代初的关系模型。1970 年，IBM 的研究员 EDGAR FRANK CODD 发表了业界第一篇关于关系数据库理论的论文《A RELATIONAL MODEL OF DATA FOR LARGE SHARED DATA BANKS》，首次提出了关系模型的概念。后来 EDGAR FRANK CODD 又陆续发表多篇相关论文，奠定了关系数据库的基础。为了促进对问题的理解，1974 年 ACM（ASSOCIATION FOR COMPUTING MACHINERY，美国计算机协会）牵头组织了一次研讨会，会上开展了一场分别以 CODD 和 BACHMAN 为首的支持和反对关系数据库两派之间的辩论，这次著名的辩论推动了关系模型及其关系数据库的发展。

从 EDGAR FRANK CODD 的论文开始，拉开了关系型数据库系统软件革命的序幕。由于关系模型简单明了、具有坚实的数学理论基础，所以经过短暂的分歧之后很快就受到了学术界和产业界的高度重视和广泛响应，并很快成为数据库市场的主流。20 世纪 80 年代以来，计算机厂商推出的数据库管理系统几乎都支持关系模型，由此 EDGAR FRANK CODD 被誉为“关系数据库之父”，并因为在数据库管理系统的理论和实践方面的杰出贡献于 1981 年获图灵奖。图灵奖是计算机界最负盛名的奖项，有“计算机界诺贝尔奖”之称。图灵奖（TURING AWARD）是由 ACM 于 1966 年设立的，专门奖励那些对计算机事业做出重要贡献的个人，其名称取自计算机科学的先驱、英国科学家艾伦·麦席森·图灵（ALAN MATHISON TURING），这个奖设立的目的之一是纪念这位科学家，获奖者必须在计算机领域具有重大且持久的理论或技术贡献。

二、关系模型概要

关系模型是以集合论中的关系概念为基础发展起来的，关系模型中无论是实体还是实体间的联系均由单一的结构类型——关系来表示。在实际的关系数据库中关系也称为表，一个关系数据库就是由若干个表组成的。关系模型提供了关系操作的特点和功能要求，但不给 DBMS 的语言给出具体的语法要求。对关系数据库的操作是高度非过程化的，用户不需要指出特殊的存取路径，路径的选择由 DBMS 的优化机制来完成。

关系模型由以下几方面构成。

(1) 关系数据结构：单一的数据结构——关系。现实世界的实体以及实体间的各种联系均用关系来表示，从用户角度看，关系模型中数据的逻辑结构是一张二维表。

(2) 关系操作集合：常用的关系操作包括查询操作和插入、删除、修改操作两大部分。其中查询操作的表达能力最重要，包括选择、投影、连接、除、并、交、差等。关系模型中的关系操作能力早期通常是用代数方法或逻辑方法来表示，分别称为关系代数和关系演算。关系代数是使用对关系的代数运算来表达查询要求的方式；关系演算是用谓词来表达查询要求的方式。

(3) 关系的完整性约束：包括实体完整性、参照完整性和用户定义的完整性三类。实体完整性 (ENTITY INTEGRITY) 规则——若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值。参照完整性规则——若属性或属性组 F 是基本关系 R 的外键，它与基本关系 S 的主键 KS 相对应 (关系 R 和 S 不一定是不同的关系)，则对于关系 R 中每个元组在关系 F 上的值必须取空值 (F 中的每个属性值均为空) 或者等于 S 中某个元组的主键值。实体完整性和参照完整性是关系模型中必须满足的完整性约束条件，只要是关系数据库系统就应该支持实体完整性和参照完整性。除此之外，不同的关系数据库系统根据其应用环境不同，往往还需要一些特殊的约束条件，用户定义的完整性就是对某些具体关系数据库的约束条件。

三、第一个关系数据库系统 ORACLE

关系数据库系统是支持关系模型的数据库系统。虽然早在 1970 年就诞生了关系模型理论，但是市场上迟迟不见关系型数据库管理软件的推出，主要原因是很多反对者认为关系型数据库速度太慢，其访问效率不会优于层次型数据库。值得一提的是，IBM 虽然在 1973 年就启动了 SYSTEM R 的项目来研究关系模型及其软件实现的可行性，但一直没有及时推出具体的产品，主要原因是当时 IBM 已经拥有相对成熟的层次型数据库产品 IMS。

直到 1977 年 6 月，LARRY ELLISON、BOB MINER 和 ED OATES 在硅谷共同创办了一家名为软件开发实验室 (SOFTWARE DEVELOPMENT LABORATORIES, SDL) 的计算机公司 (ORACLE 系统公司的前身)。ED OATES 最先看到了 EDGAR FRANK CODD 那篇著名的论文连同其他几篇相关的论文并推荐给 ELLISON 和 MINER。受到关系模型思想的启发，ELLISON 和 MINER 敏锐地预见到数据库软件的巨大潜力，于是 SDL 开始策划构建可商用的通用关系型数据库管理系统 (RDBMS)。根据 ELLISON 和 MINER 在前一家公司从事的一个由中央情报局投资的项目代码，他们把这个产品命名为 ORACLE (该词在英文中的原意有“神谕，预言”之意)。1978 年，软件开发实验室更名为关系软件公司 (RELATIONAL SOFT-

WARE INC.，简称 RSI)。1979 年，RSI 开发出第一款商用 SQL 数据库——V2 (VERSION 2.0，简称 V2，下同。事实上，V1 根本就未推出过)。1982 年 RSI 为了突出公司的核心产品 ORACLE 数据库系统，更名为 ORACLE 系统公司 (ORACLE SYSTEM CORPORATION)，用产品名称为公司命名，帮助公司赢得了业界的认同。1983 年，ORACLE 系统公司决定开发便携式 RDBMS，命名为 V3，这是第一款在 PC、小型机及大型机上运行的便携式关系数据库系统。

随着全球信息化浪潮的推进，ORACLE 系统公司踏上了快速发展的道路，在数据库方面，先后推出了基于互联网 (I 系列)、基于网络 (G 系列) 的 DBMS 系统软件及其应用平台。今天的 ORACLE 系统公司已经发展为全球最大的信息管理软件及服务供应商，被誉为 IT 行业的“红色巨人” (红色为 ORACLE 公司的标志色，与之对应的，IBM 公司常以蓝色为基调，被誉为“蓝色巨人”)，产品涉及数据库、应用服务器及中间件、电子商务套件、开发工具、主机及操作系统等多个领域。

第二节 现代数据库系统的技术层次

一、启蒙：关系模型和 ANSI/ISO SQL

以关系模型为基础，讲授关系数据库原理和 ANSI/ISO SQL 的基础知识，是目前大多数高等学校的典型做法，也是数据库课程的传统教学/培训内容。进入 20 世纪 90 年代，在计算机软件技术和网络技术的推动下，关系模型和数据存储方式在悄然发生变化，数据库应用系统在性能和体系结构方面也发生了多次质的飞跃，由此，学习关系模型和 SQL，仅仅是走向数据库应用技术殿堂的启蒙课。

二、奠基：网络环境下的 DBMS

在今天的网络环境下，仅仅掌握数据库启蒙知识是远远不够的，因为数据库的主流应用早已从桌面环境迁移到基于服务器的网络环境，运行于数据库服务器上的 DBMS 处于信息系统的核心地位，因此人们认为，网络环境下的 DBMS 是掌握工程数据库应用的一块奠基石。

从满足工程实践的需要看，DBMS 的教学应该包括以下两个方面的基本内容：网络数据库原理和后台数据库编程。网络数据库原理属于认知部分，在关系数据库原理的基础上，讲授包括网络数据库系统的体系结构、数据存储管理及查询策略、用户及其安全架构、事务处理机制、并发控制、备份与恢复、参数调整与性能优化、对象—关系模型等内容；后台数据库编程属于开发部分，其核心是掌握一种扩展的 SQL 语言，如 ORACLE 支持的 PL/SQL、SYBASE 支持的 TRANSACT-SQL 等，它们通常是一种扩展的 SQL 语言，主要表现在数据表达能力和可编程能力等方面得到加强，可以进行复杂的数据处理和函数调用，如开发存储过程 (STORED PROCEDURE)、触发器 (TRIGGER)、程序包 (PACKAGE)、函数库 (LIBRARY) 等后台程序，为进一步开发前端应用做准备。需要指出的是，虽然不同的数据库系统在表现形式、组织体系上会有所差别，但它们具有较强的共性，因此在教与学的过程中结合一种典型的 NDBMS 产品开展培训或教学会起到触类旁通的效果。

三、起步：业务规划、数据库设计与建模

数据库的业务规划与建模是创建数据库应用系统的前提，一个好的规划和设计方案可以使整个系统的开发达到事半功倍的效果。这其中有三个主要的环节：

(1) 通过对系统的需求分析，以数据流图 (DATA FLOW DIAGRAM, 简称 DFD) 的方式建立系统业务模型。

(2) 实体关系建模，通过业务模型，抽象出关系数据库的概念数据模型 CDM (CONCEPT DATA MODEL, 又称为实体—关系图或 E-R 图)。

(3) 数据库的物理实现，由概念数据模型建立物理数据模型 PDM (PHYSICAL DATA MODEL), 并结合具体的数据库系统加以调整和优化，如添加数据完整性约束、引入对象—关系特性等。

工程实践中建模过程通常可借助于第三方的 CASE 工具，如 UML、POWERDESIGNER 等。

“建模”是一项技术性很强的工作，规划和设计的结果集中地体现了开发人员对数据库理论、业务模型以及具体 DBMS 的掌握和理解程度，模型的优劣对建立整个应用系统往往起到举足轻重的作用，因此数据库建模是数据库应用技术教育与培训的必备内容之一。

四、应用：C/S 结构下的信息系统

网络技术的发展促进了数据库技术的进步，20 世纪 80 年代中后期，网络环境下的信息共享和互操作成为普遍的需求，此时诞生了具有重要意义的 CLIENT/SERVER 结构 (C/S 结构)。C/S 结构改变了数据库的应用模式，它将数据处理任务在客户端和数据库服务器之间进行分配，其中客户机用来运行提供用户接口和前端处理的应用程序，服务器为客户机的使用提供数据服务资源。

C/S 系统将应用程序从数据库服务器中分离出来，形成相对独立的客户端，其软件设计与开发技术经过多年的发展，已经变得相当成熟，掌握它需要两方面的知识：一般的软件开发技术 (包括结构化、对象化的编程)、数据库的功能调用与事务处理技术。前者是软件技术的通用基础，数据库应用开发也不例外；后者是 DBMS 的数据处理技术在客户端的延伸，其核心是数据库接口与数据操纵，通常前端程序通过如 ODBC、JDBC、ADO 等标准接口调用和处理数据，实现与远程数据库服务器的分工协作。

目前采用 C/S 结构的信息系统早已成为数据库应用的主流，C/S 结构下的软件开发技术成为掌握数据库应用技术所必需的基本技能，因此认为它应该成为培养数据库应用技术人才的重点教学/培训内容之一。

五、进阶：信息系统的多层体系结构

在 C/S 结构的信息系统获得广泛应用的同时，基于 INTERNET 和 WEB SERVICE 的数据库应用也逐渐成熟起来，由此构造出 BROWSER/WEB SERVER/DATABASE SERVER 的 B/S 应用模式，为数据库应用系统的开发提供了新的技术途径。不仅如此，B/S 应用模式还引发了信息系统体系结构的变革。

理论上，任何数据库应用系统都可以逻辑地划分为表示层、业务逻辑层 (也称商业规

则层)、数据层三个层次。随着应用规模的扩大,两层的 C/S 结构逐渐显露出它的不足,主要体现在部署企业级应用的灵活性和适应性上受到制约,于是三层体系结构应运而生,其中业务逻辑层位于表示层和数据层之间,是两者沟通的桥梁,因此又叫中间层。这种三层结构有利于部署大型分布式应用,实践中根据系统柔性结构的需要又可将中间层拆分成多级协同作业的结构,如在中间层增加实现负载均衡的代理服务、根据业务需要设置不同功能的应用服务等,这样理论上的三层结构在企业级应用中,往往表现为一个多样化的 N 层体系结构,如前述的 B/S 结构就可以演化成 BROWSER/WEB SERVER/APPLICATION SERVER/DATABASE SERVER 这样一种四层结构。

在多层结构中,中间层通过一系列封装了企业业务逻辑的软件模块来实现表示层和数据层之间的信息交互和事务处理,这种处理方式极大地提高了部署企业级应用的灵活性,如当企业的业务逻辑发生变化时,原则上只需修改中间层即可。中间件技术将这类独立的软件模块推向标准化,目前具有代表性的技术规范有: SUN 的 J2EE/EJB、OMG 的 CORBA、微软的 COM/DCOM/COM+ 等,越来越多的信息系统采用这种技术实现多层分布式系统结构。

由此,数据库教育衍生出另外一个重要分支——多层体系结构与中间件技术(特别是事务处理中间件),这是数据库应用系统的技术骨架。掌握这种技术骨架及其内涵,可以使教学/培训内容更加贴近数据库系统的工程实践,更好地把握当今数据库应用技术的发展趋势——由集中应用走向分布式数据处理。

六、深化:分析型数据环境及其应用

典型地,数据库应用的任务是用来建立基于事务处理 (TRANSACTION PROCESSING) 的信息系统。但随着技术进步、应用水平的提高以及各种业务数据的积累,人们不再满足于,越来越多的企事业单位希望从大量的事务数据、历史数据中获得隐藏在数据背后的有价值的信息,于是产生了数据分析 (DATA ANALYSIS) 的应用需求,这种需求直接导致了传统数据库的另外一种表现形式——数据仓库 (DATA WAREHOUSE) 的诞生。它的出现促进了数据库应用系统由原来单一的联机事务处理 (OLTP) 向综合数据处理 (包括 OLAP 和决策分析) 扩展,决策支持系统 (DSS) 就是这种扩展的结果。目前 DSS 主要在两个方面展开应用: 联机分析处理 (OLAP) 和数据挖掘 (DATA MINING)。前者是针对特定问题的联机数据访问和分析,它主要面向系统的决策分析及其高层业务管理;后者则是在联机分析处理的基础之上更进一步,采用基于人工智能 (AI)、机器推理、逻辑和数理统计等技术来分析和归纳已有数据,从中发掘出蕴藏在大量数据中的潜在模式 (PATTERN),从而为业务管理与决策提供参考。当前在产业界各类分析型的应用正在快速起步,已经形成了数据库应用的一个重要分支。

数据库技术推进到这里,信息系统已经进入到了一个更高的层次,由此引出了一项新的教学/培训内容——分析型数据环境及其应用技术。这部分内容是现代数据库理论与技术体系中不可或缺的一个有机组成部分,在现代数据库教育中应该占有一席之地,并且随着应用水平的提高和技术的日渐成熟,这类教学/培训内容应该得到逐步加强,这是产业实践对高级数据库应用技术人才的必然要求。

第三节 数据库应用系统结构

一、客户/服务器体系结构

ORACLE 是一种建立在客户/服务器体系结构上的对象关系数据库管理系统。需要指出的是,虽然目前数据库应用出现多层体系结构(N-TIER),但仍然以两层的 C/S 结构为前提。因此 C/S 结构是基础性的。

什么是客户/服务器(C/S)体系结构?

从硬件角度看,客户/服务器体系结构是指将某项任务在两台或多台机器之间进行分配。从功能上看,客户机(CLIENT)用来运行提供用户接口和前端处理的应用程序,服务器机(SERVER)提供客户机使用的各种资源和服务。

从软件角度看,客户/服务器体系结构是把某项应用或软件系统按逻辑功能划分为客户软件部分和服务器软件部分。客户软件部分一般负责数据的表示和应用,处理用户界面,用以接收用户的数据处理请求并将之转换为对服务器的请求,要求服务器为其提供数据的存储和检索服务;服务器软件部分负责接收客户软件发来的请求并提供相应服务。

客户/服务器体系结构融合了大型机的强大功能和中央控制以及 PC 的低成本和较好的处理平衡,它为任务的集中/局部分布提供了一种新的方法,这种体系结构能够使用户对数据完整性、管理和安全性进行集中控制。在缓解网络交通和主机负荷以及满足用户需要方面,客户/服务器体系结构提供了良好的解决方案。

总之,客户/服务器体系结构的工作模式是:客户与服务器之间采用网络协议(如 TCP/IP、IPX/SPX)进行连接和通信,由客户端向服务器端发出请求,服务器端响应请求,并进行相应服务。

典型数据库应用的客户/服务器体系结构及其软件层次如图 1-1 所示。

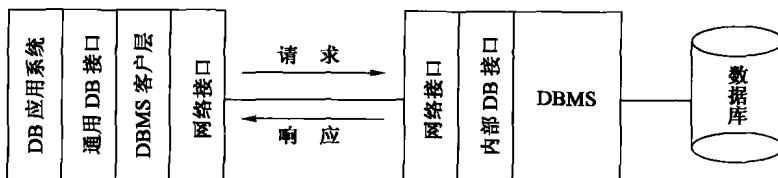


图 1-1 典型客户/服务器体系结构及其软件层次

二、数据库应用的多层体系结构

数据库应用经历了从单机到网络、从网络到 C/S 结构、再从 C/S 结构到多层(MULTI-TIER)分布式结构这样几个发展阶段。目前 C/S 结构如火如荼,多层数据库应用正处于推广过程中。

(一) 为什么需要多层结构

客户/服务器结构推动了数据库应用的普及,但随着数据库应用规模的扩大,特别是今

天的应用范围由 LAN 扩展到 WAN, 甚至扩展到 INTERNET 时, 传统的客户/服务器两层结构日益显示出它的弊端来。主要表现在以下几个方面:

(1) 执行效率无法满足日益膨胀的客户需求。客户/服务器结构中的客户数量受到一定的限制, 一般而言, 当客户数量由数十个增加到数百个时, 系统的执行效率便开始显著下降, 主要原因是大量的数据库并发连接消耗了宝贵的数据库资源, 而这种连接往往使用率非常低。

(2) 客户/服务器结构的维护成本较大。首先是硬件成本, 在数据库服务器端相对不变的情况下, 要提高系统的执行效率, 往往需要对客户机的硬件进行升级; 其次是软件成本, 包括应用软件的维护成本和系统的扩充成本, 在客户/服务器的体系结构下, 随着系统规模的扩大这两种成本会显著地增加。另外, 随着应用需求的变化, 整个系统缺乏灵活性。

(3) 大规模的应用系统在负载平衡能力上显得力不从心。虽然目前多数的数据库系统都支持分布式应用, 但仍然需要客户端的应用来实现负载的均衡。当数据库连接成为系统的负担时, 负载平衡能力就显得非常必要了。

上述客户/服务器结构的缺点在多层应用系统中都会得到比较好的解决, 除此之外, 两者相比较, 后者还具有如下几个明显的优势:

(1) 多层应用的客户端属于瘦客户模式。从数据库的应用角度来看, 瘦客户是一种总的趋势, 它代表着系统应用逐渐走向成熟、简洁、灵活。

(2) 多层应用可以实现更好的安全性。对于大多数数据库应用来说, 系统的安全性都被放在很重要的位置上。多层结构采用安全验证多级划分的分布式安全管理机制, 不仅使系统的数据得到更加安全的保护, 也使系统的应用软件得到保护。

(3) 多层结构可以更好地支持分布式计算环境。从应用的发展趋势来看, 分布式的业务促使软件采用分布式的体系结构, 而数据库的多层应用在这种体系结构下可以做到游刃有余地满足应用需求。

(4) 多层结构可以减少网络开销。在两层结构中, 同一次交易, 客户机与数据库往往要进行多次交互。在多层结构中, 可以通过中间层, 客户机的数据交互只需要一次, 数据的多次交互可以限定在中间层与数据库之间, 这样网络上的数据流量就会大大减少。

(5) 多层结构可以部分地消除数据库瓶颈。虽然数据库的并发度不能无限制地提高, 但多层结构中可以通过增加中间层的并发度、分解数据库承担的任务、共享数据库连接等措施来降低数据库的负荷, 从而提高数据库系统的运行效率。

(二) 数据库应用系统的任务分割

任何数据应用从逻辑上可以概括地分为用户界面、商业规则和访问控制、数据存储这三个基本任务。这三种不同的任务可以在一个程序里实现, 也可以分割在几个程序里实现。在数据库的多层应用中, 三层结构是应用的基本方式。如果将上述三种任务分别用不同的软件来实现的话, 这就是典型的三层应用结构, 这三种软件都是为整个应用系统服务的, 称之为系统的表示层、商业规则层、数据层。

表示层是一个人机交互的接口, 它提供给用户一个可视化的界面, 借此使用者可以下达应用软件操作的指令, 如数据的获取、数据的输入、数据的修改、数据的有效性验证等一系列操作。

数据层是一个相对独立的层面，它负责执行数据的存储管理、数据的完整性、数据的安全性等任务，另外它还需要响应数据操作的请求，产生数据操作的结果。这一层大多数的应用系统采用成熟的数据库管理系统来实现，如 ORACLE、SYBASE、MS SQL SERVER 等。

商业规则层位于表示层和数据层之间，是两者沟通的桥梁，因此又叫中间层。从表示层看，它是服务器，从数据层看，它又是客户端，因此它是两套客户/服务器结构的交汇点。从功能上看，它控制数据的访问、传递表示层和数据层进行信息交互所需要的指令和数据。从物理上看，它把用户和数据隔离开来，通过其内部封装的企业对象来完成表示层和数据层之间的数据交易，从而实现应用系统更好的安全性和灵活性。为了减轻表示层和数据层的负担，在多层结构中，可以把原先两层结构中存在于客户端或数据库端的数据处理和业务逻辑迁移到商业规则层（比如将原先数据库端的存储过程移向中间层），这样在整个应用系统中，数据的处理趋向集中，非常有利于后期的软件维护和系统升级。在一个大型分布式应用中，根据系统功能和应用需求又可将中间层的计算任务划分成多级分工运算的结构，如在中间层增加实现负载平衡的代理服务器等，这样理论上的三层结构在实际应用中，往往表现为一个多样化的 N 层体系结构。典型的三层结构如图 1-2 所示。

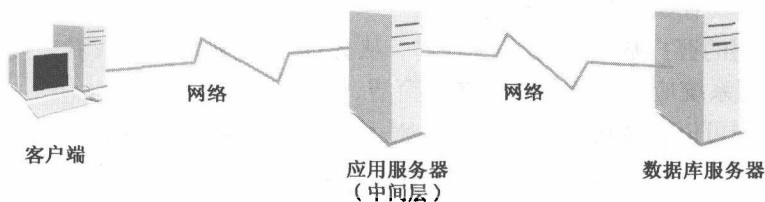


图 1-2 典型的三层结构框架

第四节 企业级应用系统的核心技术

一、J2EE 解决方案

JAVA 是一种跨平台的编程语言。大到 IBM 大型机 MAINFRAME，小到手持移动设备的手机都可以运行 JAVA 的程序。但 JAVA 又仅仅是一种编程语言，它无法提供企业应用所需要的许多功能。在前 SUN 公司（2009 年 4 月，被 ORACLE 系统公司并购）的领导下，现在有了 J2EE（JAVA 2 ENTERPRISE EDITION），这是专门为企业应用制定的标准，企业可以用它来编制企业级的应用，它为企业应用提供了像数据库存取、交易完整性、可靠消息传递等功能。从 2000 年开始，越来越多的企业把自己的关键应用开始构建在支持 J2EE 标准的 WEB 应用服务器之上。现在主流的一些 J2EE 应用服务器都可以支持企业应用所需的稳定性、可用性、安全性、靠性能、可扩展性等。

J2EE 应用服务器从企业的应用层次来说，是处于中间层，前边有客户端层，后面有企业的数据系统。一般情况下，此层包含支持客户端请求的表示逻辑层和业务逻辑层。表示逻辑层由显示动态 HTML/WML 页面的 JSP 和 SERVLETS 实现。它们的主要作用是接受并检验用户输入的数据，把数据传给后端的业务逻辑层，并把业务逻辑层处理的结果返回给客户端

层。而业务逻辑层用来满足金融、电信、零售等商务应用的需要，由 EJB (ENTERPRISE JAVA BEAN) 来实现。EJB 依靠 EJB CONTAINER 来实现事务处理、生命周期、状态管理、资源池和安全等问题。J2EE 目前有三种 EJB：

(1) 会话 (SESSION) BEAN，表示客户端程序和 J2EE 服务器的交互状态，当客户端程序执行完后，会话 BEAN 和相关数据就会被删除。

(2) 实体 (ENTITY) BEAN，表示数据库中的记录。当客户端程序中止或服务器关闭时，实体 BEAN 的数据仍然保存在数据库中，不会消失。

(3) 消息驱动 (MESSAGE-DRIVEN) BEAN，是一种以消息触发的 EJB，它会对消息进行监听，一旦消息到来，它就会去做相应的操作。

对于未来，企业级网络计算的核心是 WEB SERVICES，J2EE 将会在 WEB SERVICES 上下工夫。通过标准 WEB SERVICES 模型，提供一套标准的方法在 JAVA 中实现 WEBSERVICES。前 SUN 公司的 CEO SCOTT MCNEALY 称 WEB SERVICES 将是 SUN 公司事业发展的一个顶峰，比尔·盖茨认为它是微软 .NET 的核心。其实 WEB SERVICES 更广泛的寓意在于它是一场分布式计算模式的跃进，它真正要解决分布式计算的问题，利用庞大的、无处不在的互联网计算能力，实现广泛的计算和企业服务。一个事实已经确立，自 IT 有史以来唯有 ASCII 技术标准得到了几乎所有厂商的支持，并被应用到几乎所有的相关领域，现在 WEB SERVICES 极有可能再次成为这样的标准，真正成为企业与企业之间连接的桥梁，使得企业可以把自己的商业服务推广到全球，推广到互联网上，并真正使自己成为动态发展变化的商业价值链中的一个环节，为将来商业模式、商业服务的多样化，及跨企业、跨行业、跨地域的业务提供了技术的实现方法。

二、微软 .NET 架构

.NET 的前身是微软的 DNA (DISTRIBUTED NETWORK ARCHITECTURE)，在 2000 年中的时候被 .NET 所取代。.NET 在诞生之初就秉承了微软一贯的文化和战略：帮助客户加快对市场反应，提供完整、集成的解决方案和缩短客户投资回报的时间。这些完全可以在微软的产品特性中体现出来。比尔·盖茨作为首席软件架构师在把 .NET 的远景和框架规划完以后，就开始大力做市场宣传和研发的工作，大有后来居上的势头。.NET 以 WEB SERVICES 为核心，全面支持 SOAP、UDDI 和 WSDL，并在底层实现了类似 JAVA 虚拟机的 CLR (COMMON LANGUAGE RUNTIME) 和一套具有 3500 多个类的 .NET 基础类库，以支持之上的 VISUAL BASIC、C# 和 ASP 等，并且在后端支持其数据库 SQL SERVER 和 PASSPORT，加之完善的开发工具 .NET STUDIO 组成了完整的解决方案。

第五节 数据库技术内容体系

综上所述，本书以数据库应用技术为主线，在坚持理论与实践并举的原则下，将传统数据库课程的教学/培训内容由桌面环境延伸到网络环境、由事务型应用延伸到分析型应用，并根据技术的推演方式，将整个数据库的教学/培训内容划分为相互依赖、渐进提高的多个层次，由此构成了一个相对完整、面向数据库工程实践需要的教学/培训内容体系，如表 1-1