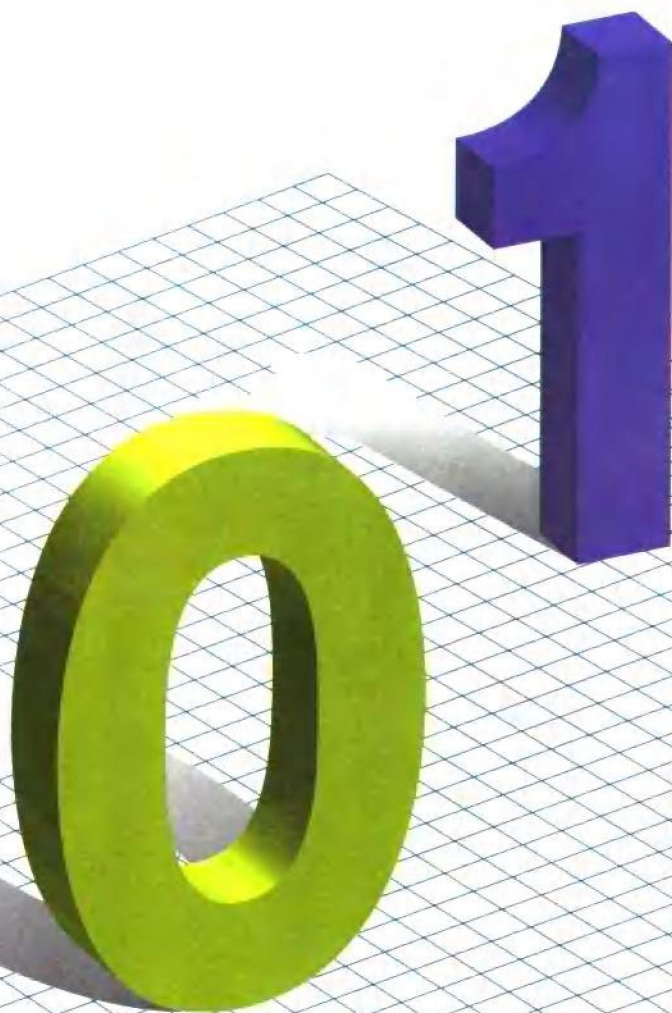


ORACLE 数据库系统 应用开发

顾 诚 编著



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

ORACLE 数据库系统应用开发

顾 诚 编著

電子工業出版社

Publishing House of Electronics Industry

内 容 提 要

据统计,有 80% 的数据库应用系统是在低效率、高风险和不合理的状态下运行的。造成这种现状的主要原因是作为全新的技术领域,设计和运行数据库应用系统缺少必要的行业性约束和规范,技术人员凭自身积累的有限经验工作,工作质量难以保证。

本书以 ORACLE 7 数据库管理系统为背景,以数据库应用设计者和数据库系统管理员为读者对象,讨论了数据库应用中要注意解决的问题和解决方法,尤其着重于如何利用数据库管理系统提供的功能来解决工程应用问题,兼有理论性和可操作性,对数据库应用工程有现实的指导意义,可以作为培训数据库应用人员的教材。

书 名:ORACLE 数据库系统应用开发

编 著 者:顾 诚

责任编辑:邓露林

特约编辑:徐波建

印 刷 者:

装 订 者:北京李史山胶印厂

出版发行:电子工业出版社出版、发行 URL:<http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036 发行部电话 68214070 邮购电话 66708599

经 销:各地新华书店经销

开 本:787×1092 1/16 印张:13 字数:332.8 千字

版 次:1998 年 4 月第一版 1998 年 4 月第一次印刷

书 号:ISBN 7-5053-4678-4
TP·2240

定 价 18.00 元

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

前 言

数据库应用系统是计算机应用的一个重要分支。特别是最近十年,计算机硬件和通信网络的发展推动着数据库应用迅速向大型化和网络化方向迈进。早期的以 xBASE 为代表的文件型数据库开发平台已不能适应当前大型数据库应用的需要,取而代之的是真正的关系数据库管理系统。其中尤以 ORACLE 产品的用户数最多,应用面最广,成为关系数据库管理系统的主要代表之一。

数据库应用的领域非常广阔,是信息产业的支柱之一。目前越来越多的人开始关心数据库应用,希望它能为自己的工作服务。数据库管理系统作为一种复杂的应用软件平台对应用系统的设计人员和运行人员均有较高的要求,成功的数据库应用系统是以优秀的应用设计人员和称职的运行人员为保障的。由于熟悉关系数据库原理及技术的人并不多,而数据库应用的需求却在高速增长,许多人正在介入与数据库应用相关的工作,为此作者特地结合自身从事数据库应用系统实践的经验,编著了本书。

本书以 ORACLE 数据库管理系统为背景,全面讨论了在设计和运行数据库应用系统时,如何最大限度地发挥数据库管理系统的功能。其中第二章讨论了数据存储空间,在对数据的物理存储方式进行分析比较的基础上,介绍了具有可操作性的数据存储空间设计方法;第三章和第四章讨论了数据表示与查询,查询是数据库的基础操作,在这两章中介绍了各种优化查询的方法和技术,并对其各自的适用环境进行了研究;第五章讨论了数据完整性,这是关系数据库管理系统区别于早期的文件型数据库平台的重要功能,对应用设计有重大影响;第六章和第七章分别讨论了进程管理和事务控制,介绍了数据库管理系统运行时的核心技术,可帮助运行人员正确操作和管理系统;第八章讨论了数据并发性,揭示了多用户系统并发工作的原理,设计人员和运行人员均可据此优化自己的工作;第九章和第十章分别讨论了数据库可靠性和系统安全性,前者是针对存储数据的物理介质故障,研究如何保障数据不因故障而丢失,后者是针对非法用户存取,研究如何通过权限划分保障数据安全;最后两章讨论了数据库的网络平台和硬件平台,应用系统设计人员可知道数据库应用系统对网络和硬件平台的基本要求。相信对这些问题的深入研讨可以帮助应用系统的设计人员和运行人员更加合理有效地开展工作。

由于时间仓促和水平所限,书中难免有疏漏和不妥之处,敬请读者批评指正。

作者

重庆大学 电气工程系

一九九七年十月

目 录

第一章 绪 论	(1)
1.1 关系数据库的由来	(1)
1.2 关系数据库系统的特点	(2)
1.3 关系数据库技术的新进展	(6)
1.4 主要厂商及数据库产品特点	(6)
1.4.1 几家主导厂商简介	(6)
1.4.2 ORACLE 的产品线	(8)
第二章 数据库的存储方式	(11)
2.1 数据库的逻辑载体:表空间	(11)
2.1.1 表空间与数据文件的关系	(11)
2.1.2 设计表空间和数据文件考虑的因素	(12)
2.1.3 表空间的操作	(13)
2.2 表空间的逻辑构成:段、区间和数据块	(15)
2.2.1 段、区间和数据块之间的关系	(15)
2.2.2 数据块	(16)
2.2.3 区间和段	(17)
2.3 存储参数的选择	(17)
2.3.1 存储参数	(17)
2.3.2 一次到位原则	(18)
第三章 数据表示与查询(表和视图)	(21)
3.1 数据类型	(21)
3.1.1 数据库中的主要数据类型	(21)
3.1.2 关于数据类型的 ANSI 标准	(24)
3.1.3 数据类型的转换	(25)
3.2 表	(26)
3.2.1 表的存储方式	(26)
3.2.2 空值和缺省值	(27)
3.2.3 管理表的 SQL 语句	(27)
3.3 查询语句及表达式、函数	(29)
3.3.1 SQL 的查询语句概述	(29)
3.3.2 运算符和表达式	(30)
3.3.3 函数	(32)
3.4 查询的基本技术与条件	(36)
3.4.1 查询三要素	(36)
3.4.2 表达式的组织和应用	(36)
3.4.3 查询输出顺序	(37)
3.4.4 聚组函数的计算过程	(38)

3.4.5 暂存段.....	(4 0)
3.5 视图.....	(4 1)
3.5.1 视图与表的关系.....	(4 1)
3.5.2 视图的操作和特点.....	(4 1)
3.5.3 视图上的查询.....	(4 2)
第四章 数据表示与查询(索引和聚簇)	(4 4)
4.1 索引.....	(4 4)
4.1.1 索引的概念.....	(4 4)
4.1.2 索引的操作.....	(4 5)
4.1.3 应用索引提高效率.....	(4 8)
4.2 连接.....	(5 2)
4.2.1 表连接的概念与分类.....	(5 2)
4.2.2 连接优化原则.....	(5 4)
4.2.3 连接模型分析.....	(5 6)
4.3 聚簇.....	(5 7)
4.3.1 聚簇的概念.....	(5 7)
4.3.2 聚簇空间需求计算.....	(5 8)
4.3.3 聚簇操作.....	(6 0)
4.4 HASH 簇存储	(6 1)
4.4.1 HASH 簇的概念	(6 1)
4.4.2 HASH 簇操作	(6 2)
4.4.3 HASH 函数	(6 4)
第五章 数据完整性	(6 6)
5.1 数据的完整性概念.....	(6 6)
5.1.1 数据完整性及基本实现方法.....	(6 6)
5.1.2 单一表上的完整性模型.....	(6 7)
5.1.3 参照完整性模型.....	(7 0)
5.2 数据完整性的定义方法.....	(7 3)
5.2.1 相关的 SQL 语句	(7 3)
5.2.2 增加和删除约束定义.....	(7 6)
5.2.3 对数据完整性约束的控制.....	(7 8)
5.3 数据输入设计.....	(8 1)
5.3.1 减少错误数据输入.....	(8 1)
5.3.2 减少输入操作量.....	(8 3)
第六章 动态存储与进程管理	(8 5)
6.1 动态存储.....	(8 5)
6.1.1 动态存储结构.....	(8 5)
6.1.2 系统全局区和程序全局区.....	(8 6)
6.2 进程管理.....	(8 8)
6.2.1 例程与进程结构.....	(8 8)

6.2.2	后台进程	(91)
6.2.3	多线索结构	(93)
6.2.4	程序接口	(94)
6.3	SQL 语句处理	(95)
6.3.1	SQL 指针	(95)
6.3.2	SQL 语句的处理阶段	(96)
6.4	数据库与例程管理	(98)
6.4.1	启动与关闭的概念	(98)
6.4.2	启动例程和数据库	(99)
6.4.3	初始化参数文件	(100)
6.4.4	关闭例程和数据库	(101)
第七章	事务控制	(103)
7.1	回退段结构	(103)
7.1.1	回退段的作用	(103)
7.1.2	回退段的区间分配	(104)
7.1.3	回退段的状态	(105)
7.1.4	创建回退段的原则	(106)
7.2	提交与回退	(109)
7.2.1	事务定义	(109)
7.2.2	事务控制语句	(110)
7.3	两阶段提交	(112)
7.3.1	两阶段提交的概念	(112)
7.3.2	会话树	(114)
7.3.3	事务级容错分析	(117)
第八章	数据并发性	(121)
8.1	数据并发性和一致性	(121)
8.1.1	基本概念	(121)
8.1.2	基本控制方法	(123)
8.1.3	用回退段保证读一致性	(124)
8.2	数据锁	(126)
8.2.1	数据锁的类型	(126)
8.2.2	数据锁的操作特点	(128)
8.3	字典锁和内部锁	(129)
8.3.1	字典锁	(129)
8.3.2	内部锁	(130)
8.4	改善数据并发性	(131)
8.4.1	按约定顺序封锁表	(131)
8.4.2	行级封锁中的问题	(133)
8.4.3	减少封锁时间的方法	(135)
第九章	数据库系统的可靠性	(137)

9.1 故障类型与基本对策.....	(137)
9.1.1 故障类型.....	(137)
9.1.2 前卷与回退.....	(138)
9.1.3 故障的基本对策.....	(139)
9.2 日志文件.....	(140)
9.2.1 日志文件工作方式.....	(140)
9.2.2 检查点.....	(141)
9.2.3 日志文件与检查点的优化.....	(142)
9.2.4 日志文件操作.....	(143)
9.3 数据冗余保存.....	(144)
9.3.1 在归档方式下的日志数据.....	(144)
9.3.2 影像备份.....	(145)
9.3.3 卸出和装入.....	(147)
9.4 介质故障恢复过程.....	(148)
9.4.1 恢复 SYSTEM 以外的表空间	(148)
9.4.2 日志文件和控制文件的恢复	(150)
9.4.3 共享磁盘系统的备份和恢复.....	(151)
第十章 数据库系统的安全性	(153)
10.1 用户管理	(153)
10.1.1 用户分类	(153)
10.1.2 用户的权限管理	(155)
10.1.3 用户管理	(156)
10.1.4 系统权限和对象权限	(157)
10.1.5 角色管理	(158)
10.2 资源管理	(159)
10.2.1 资源限制文件	(159)
10.2.2 资源限制文件的用途	(160)
10.3 数据库访问的安全性	(162)
10.3.1 三种基本访问角色	(162)
10.3.2 数据库用户的创建与删除	(165)
10.3.3 数据库访问的审计	(166)
10.4 数据库对象的安全性	(167)
10.4.1 访问表中的数据	(167)
10.4.2 审计数据库对象	(168)
10.4.3 统计数据库中的数据安全性	(169)
第十一章 数据库系统的网络平台	(172)
11.1 网络设备与硬件	(172)
11.1.1 基本网络设备:网卡.....	(172)
11.1.2 基本拓扑结构	(174)
11.1.3 扩充网络的范围	(176)

11.1.4	平衡网络传送能力	(177)
11.1.5	高速网络平台	(180)
11.2	网络设计	(180)
11.2.1	网络性能需求	(180)
11.2.2	合理划分网段	(181)
11.3	影响网络平台的因素	(183)
11.3.1	运行管理的影响	(183)
11.3.2	应用设计的影响	(184)
11.3.3	数据库系统软件的影响	(186)
第十二章	数据库服务器的硬件平台	(188)
12.1	中央处理器	(188)
12.1.1	比较标准	(188)
12.1.2	技术特征	(189)
12.2	体系结构	(190)
12.2.1	总线	(190)
12.2.2	多机并行结构	(191)
12.3	内存	(192)
12.3.1	容错能力	(192)
12.3.2	容量	(193)
12.4	硬盘子系统	(195)
12.4.1	硬盘配置	(195)
12.4.2	硬盘冗余	(196)

第一章 绪 论

1.1 关系数据库的由来

数据库是现代计算机系统，特别是软件系统的重要组成部分，是有效地进行数据存储、共享及处理的环境和工具。计算机作为信息处理工具，信息的表达和存储及其管理正在成为日益重要的工作，在办公自动化、管理信息系统、计算机辅助技术、甚至传统的科学计算等各种计算机应用领域，信息量的增加客观地要求有一种专门的技术来解决大量信息的存储、共享及管理等问题，这就是数据库技术。现代的大型计算机应用系统有很大一类是以数据库为核心的，如航空公司的订票系统、银行的业务系统、企业的管理系统等。可以预见，随着硬件与支撑技术的发展、高速网络的普及，应用系统中对数据库技术的依赖必将有增无减。

回顾数据库技术的发展，可以简单地归纳成两个阶段：

1. 准备阶段

这个阶段的基本特征是：文件系统已经十分成熟，大量计算机从科学计算转向到办公系统的应用，繁重的数据信息处理任务使旧式的文件管理模式面临挑战。计算机技术的发展使人们不得不寻找新方法、新概念和新技术以适应数据信息处理的需求，由此产生了今天我们所谈论的数据库。其中有代表性的工作有：

(1) 1963 年 IDS (Integrated Data Store) 系统投入运行，它为多个 COBOL 语言程序提供共享机制。这种以应用需求为驱动而设计的系统，严格地讲是对 COBOL 语言作了必要的拓展以满足数据信息处理的需要。这种思路在早期应用中大兴其道，也解决了相当大的问题。我国的一些企业（包括银行）直到 90 年代初仍未淘汰这种数据处理模式。

(2) 1968 年网状数据库系统 TOTAL 发表，1969 年层次数据库系统 IMS 发表，数据库技术的理论研究形成高潮。早期实现的数据库系统是网状模型和层次模型，上述系统仅是其中的代表，尤其是网状数据库系统得到了十分广泛的应用，客观上形成和巩固了许多重要概念，奠定了数据库作为一门独立技术的地位。但是早产儿多少都有些先天不足，而网状及层次数据库的先天不足是致命性的，所以今天我们在数据库市场上已看不到它们的踪迹了。

(3) 1970 年 IBM 公司的 E.F.Code 发表了论文《大型共享数据库数据的关系模型》，开创了数据库的关系方法和关系规范化理论的研究。这篇著名论文倡导的关系模型理论完备，结构简单，使之成为数据库发展中里程碑式的文献，十年之后，1981 年在历史充分证实其价值的基础上，Code 获得了计算机界的最高奖——图灵奖。

(4) 整个 70 年代是关系数据库由弱到强，最后得胜称王的十年，加州大学伯克利分校在 VAX 上实现了 Ingres 关系数据库管理系统，IBM 在 370 系列上开发了 SystemR 关系数据库管理系统。这些研究性的系统在赢得用户的同时，也统一了关系数据库的实现原则、性能尺度等技术规程。1978 年美国标准化组织发表了关于关系数据库系统的工作报告，从而规定了关系数据库系统的结构和特征。

2. 成熟阶段

这个阶段的特征是数据库系统的商品化得以实现，关系数据库一统天下，高额的商业利润使数据库厂商有实力投入巨资进行研发，数据库技术与数据库市场一起在竞争中发展，大公司成为这一高技术领域的主角。其中有代表性的工作为：

(1) 1979 年 ORACLE 公司推出第一个商品化关系数据库系统 ORACLE V2，如同第一台商用计算机并无明显的商业效果一样，ORACLE V2 的市场收效亦不理想。以现在的观点看，许多重要的功能都没有实现，如与硬件/操作系统无关的功能是 1983 年的 ORACLE V3 实现的，1986 年的 ORACLE V5 开始支持分布式查询，1989 年的 ORACLE V6 基本具备了联机事务处理能力，但第一个商品化的关系数据库系统为 ORACLE 公司赢得了市场优势。

(2) 1986 年美国国家标准化组织 ANSI 正式将 SQL 语言宣布为美国国家工业标准。SQL 语言最先是研究性的关系数据库查询语言，其后为 IBM、ORACLE 等公司在产品中采用并进一步发展，在 ANSI 确认为工业标准之后，所有数据库厂商均以其作为产品的基本支撑，从而为异种数据库的沟通和共享创造了条件。

(3) 八十年代后随着计算机应用范围的不断扩展，应用系统的物理场址从一间办公室到多间办公室、整栋办公楼、厂域、城域直至通信系统可达到的每一个地方，于是分布计算、分布处理等概念日益引人注目，其方法也日益成熟。分布式数据库技术是为满足网络环境中的数据库应用不断发展起来的，以期解决在大型应用系统中不同级别、不同地方、不同分工的多个数据库服务器上实现数据共享和协同操作，80 年代末解决了分布查询问题，九十年代初期分布更新的技术逐渐实用化，由于分布更新在硬件上尤其依赖网络通信系统，所以分布式数据库管理系统不仅与计算机技术一起发展，也与网络通信技术一起发展。

(4) 随着应用规模的不断升级，用户数量越来越多，应用的实时要求日益突出，如大型银行系统可能会要求拥有每秒处理数十乃至数百笔业务的高峰能力，这就要求数据库管理系统不仅能实现应用功能，而且要按时完成预定功能，于是 90 年代后具有联机事务处理能力的数据库系统得到了重视和发展，主要的数据库管理系统均通过各种方法优化算法和结构，注重发挥硬件系统的潜在能力以提供优良的联机事务处理能力。虽然联机事务处理能力还与硬件平台、应用需求等因素有关，但数据库管理系统作为应用平台，其影响举足轻重，所以发展联机事务处理能力成为 90 年代中后期数据库技术的热点。

现在我们可以理解为什么有不只一家公司声称自己是关系数据库的鼻祖。因为 IBM 最先发表了模型，而 Ingres 最早实现了模型，ORACLE 则在商品化上领先一步。

1.2 关系数据库系统的特点

数据库系统可以简单地定义为能长久和可靠地管理大量共享数据的工具。从管理过程和管理对象看有如下特征：第一，数据量大，不能放在常规主存储器中处理，需要大容量外存支持；第二，数据的有效期长，不是单一地为特定应用服务，而需要长期保留，以备查访；第三，系统具有容错能力及故障预防处理机制，以保障数据的可靠性；第四，数据可以让多个用户并发存取，系统一方面能有效管理多个用户，另一方面能适应多个用户同时使用。早期的文件系统在一定程度上也具有这些特征，与文件系统相比数据库系统的区别在哪里，成为需要首先搞清楚的问题，归纳起来，数据库系统的显著特点有：

(1) 数据库中的数据是高度结构化的。在文件系统中，从整体上讲，数据是无结构的，即文件中的各个记录之间没有联系，它只关心记录内部数据项之间的联系。而数据库系统不仅

要考虑数据项之间的关系，更要考虑每个记录之间的关系，因为在一个信息系统中，存在于不同“文件”中的数据之间仍然有各种各样的联系。例如在企业管理信息系统中通常都会有人事系统、财务系统、库存系统、销售系统等，如果采用文件系统，它们的数据都必须存放在相互分离的文件中，但显然这些系统之间是有客观联系的，使用文件系统就人为地将它们分隔开来，它们的联系只能通过应用程序处理。相比之下，这种联系在数据库中反映得更加直观和易于处理，一般用“路径”来描述两个系统之间的联系，路径不仅指明了两组不同类型的记录有联系，而且规定了用于联系的数据项。例如在学生选课管理中，需要知道学生的基本情况、课程的基本情况和选课情况，它们尽管也可以分别放在不同的文件系统中，但由于这三种信息实体实际上是有着密切联系的，如图 1-1。显然，如果查找“张三的学习成绩”、“获满分的学生姓名及课程名称”等信息，就必须按图中的两条存取路径，从一组记录类型到另一组记录类型。通过存取路径来表示自然的数据联系，是数据库系统与文件系统的根本区别。用数据库的术语来讲，建立这种“路径”就是“连接”两个表。

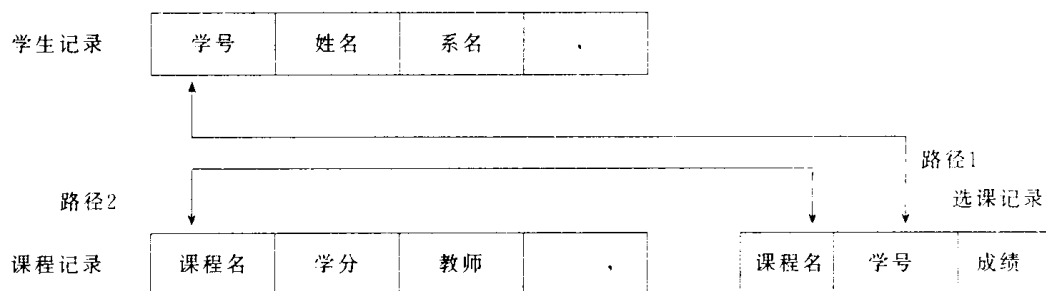


图 1-1 选课管理中的路径

(2) 数据库中的数据是面向系统的，而不是面向具体的应用，因此数据库的数据共享程度比文件系统更高。实现数据共享是数据库的重要特征，文件作为数据的组织方式基本上还是面向具体应用的，而在数据库中，数据不再是为具体应用准备的了，而是从整个信息系统出发，考察整个系统中的各种信息需求，统一地进行数据的组织、定义和存储。数据的定义和应用程序是分开的，因此数据库可以为信息系统中的各种应用所使用，达到数据共享的目的，如图 1-2 所示。因此数据库是面向系统的。数据面向系统的另一个优点是信息结构比较稳定，而且易于扩充。所以信息系统中数据库的设计可以较为独立地进行。

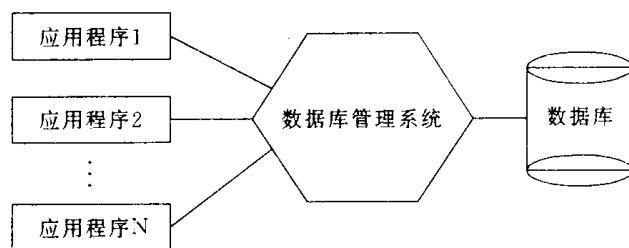


图 1-2 数据库中程序与数据的关系

(3) 数据库系统比文件系统具有更高的独立性，在使用数据库时，应用程序对存储物理结构有较高的独立性。这种独立性是由系统在存储物理结构和逻辑结构之间提供的映象获得的，这样，当存储结构或者说物理结构改变后，只要相应地改变系统的逻辑结构和物理结构之间的映象就可以使逻辑结构保持不变，建立在逻辑结构之上的应用程序也可以保持不变，这就是物理独立性。更进一步，一个数据库系统所拥有的数据比特定的应用所需要的数据要多得多，因此对每个应用还要提供局部的逻辑结构，这种局部逻辑结构只是总体逻辑结构的子

集，局部逻辑结构和全局逻辑结构之间使用映射进行联系。这样就可以做到当总体逻辑结构变化后，局部逻辑结构可以保持不变，在局部逻辑结构基础上编写的程序也可以不变，这就是所谓的逻辑独立性。提高数据的逻辑独立性是数据库管理系统追求的主要目标。

(4) 数据库系统具有较好的保护数据安全性的措施。由于数据库的数据是面向系统的，数据信息是整个管理系统或业务系统的基础和核心，非常重要，并不是任何人都可以查看。因此数据库系统都需要有一定的授权机制以保护数据，防止不合法的使用，只有那些被授权可以存取数据库的人或程序才能执行对数据库的存取。

(5) 数据库提供各种维护数据一致性的措施。在数据库中强调共享，在一个多用户系统中会出现许多用户同时使用数据库的情况，这种并发的存取动作如果不加控制必将造成严重的后果，如多个用户同时修改同一数据，可能造成修改的参照标准丢失，数据有可能失去准确性，数据库提供的控制并发存取的功能也称为数据库的操作一致性，这是最基本的一致性措施。另一方面，数据库中的数据是对客观世界实体性质的反映，因此它的存在应当符合一定的原则。例如，年龄这个数据是反映客观世界中人的年龄属性，因此数据库中年龄的值就不能超过 200 岁，又如在财务系统中收支应当平衡，这在会计数据库中就要求在任何时候收入之和减支出之和等于剩余数。在数据库中数据的这种正确性要求，称为数据的语义一致性，又被称为数据完整性。

(6) 数据库将存取数据的最小单元细化到数据项或字段上。在数据库中对记录的存取不一定像文件系统那样必须以整个记录为基本单位，如果记录很长，而仅仅只需要其中的少数字段，那么以记录为单位存取就显得浪费了，数据库可以有效地克服这种浪费，仅将需要的信息取出来。

上面通过与文件系统的比较更加突出了数据库系统的特点，综合起来，可以说数据库是管理应用系统中大量、持久、可靠和共享数据的工具，这些数据具有最小的冗余度和较高的独立性，而且数据库应能保持数据安全性，维护数据一致性和完整性。

数据库管理系统有多种模型，其中关系模型已占主导地位，为了后面讨论方便，先简单介绍关系数据库中的几个基本概念。在关系模型中，信息被组织成一些二维表结构，每一个二维表称为一个关系 (RELATION) 或表 (TABLE)。每个表中的信息只能用于描述客体世界中的一件事情，如表 1-1 所示的员工信息即表 EMP 和表 1-2 所示的部门信息即表 DEPT (在后面章节的讨论中要反复使用这两个表作为举例的对象)。

表 1-1 EMP 表

EMPNO 编号	ENAME 姓名	JOB 工种	MGR 经理	HIREDATE 受雇日期	SAL 工资	COMM 佣金	DEPTNO 部门号
7936	SMITH	CLERK	7902	17_DEC_80	800		20
7499	ALLEN	SALESMAN	7698	20_FEB_81	1600	500	30
7521	WARD	SALESMAN	7698	22_FEB_81	1250	500	30
7566	JONES	MANAGER	7893	02_APR_81	2975		20
7654	MARTIN	SALESMAN	7698	28_SEP_81	1250	1400	30
7698	BLAKE	MANAGER	7893	01_MAY_81	2850		30
7782	CLARK	MANAGER	7893	09_JUN_81	2450		10

续表

EMPNO 编号	ENAME 姓名	JOB 工种	MGR 经理	HIREDATE 受雇日期	SAL 工资	COMM 佣金	DEPTNO 部门号
7788	SCOTT	ANALYST	7566	09_NOV_81	3000		20
7839	KING	PRESIDENT		17_NOV_81	5000		10
7844	TURNER	SALESMAN	7698	08_SEP_81	1500	0	30
7876	ADAMS	CLERK	7788	23_SEP_81	1100		20
7900	JAMES	CLERK	7698	03_DEC_81	950		30
7902	FORD	ANALYST	7566	03_DEC_81	3000		20
7934	MILLER	CLERK	7788	23_JAN_82	1300		10

表 1-2 DEPT 表

DEPTNO 部门号	DNAME 部门名称	LOC 地址
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATION	BOSTON

表 (TABLE): 又称关系, 由表名、列名及若干行组成。例如在员工关系中表名是 EMP, 列名包括 EMPNO、ENAME 等, 每一行数据描述了一个职工的情况, 表的结构有时也称关系模式。表的主要优点有:

第一, 形式直观, 一般用户易于接受和理解, 有极好的用户基础;

第二, 结构严密, 易于在计算机上表示, 是精确高效的信息描述工具。

列 (FIELD): 又称字段、域、属性。表中的每列都包含一类信息, 例如 JOB 表示员工的工种, 在关系数据库中列的顺序并不重要。

行 (ROW): 又称元组 (TUPLE)。表中的每行由若干个字段组成, 描述一个对象的信息, 每个字段描述了该对象的一种属性或性质, 例如在 EMP 表中第一行描述了这样一个职工: 编号为 7936, 名叫 SMITH 的办事员, 1980 年 12 月 17 日进入公司工作, 月工资 800 美元等等。每行可以有一个或若干列的集合, 用以标识此行, 这样的列的集合称为基键 (PRIMARY KEY)。行通常不应完全重复, 如果两行在全部字段上的值均相等, 则认为它们是相同的行, 行与行之间的顺序也是不重要的。

值域 (DOMAIN): 值域反映了由系统管理的数据信息的类型和范围。表中每列都以某个值域为基础取得数据, 在关系模型中允许多个列以同一值域为基础。

表和列的命名规定: 表名在整个数据库中必须唯一, 列名在一个表中必须唯一, 但在不同的表中可以出现相同的名字, 表名和列名应尽可能带有一定的意义并尽量简单。

1.3 关系数据库技术的新进展

目前数据库技术日新月异,新技术从不专属于某个厂商,通常是各有突破,又相互看齐,再各寻新路。随着硬件技术的发展,一些崭新的功能正日益走近广大用户。具有代表性的有:

并行操作:对查询、索引、数据装载、备份/恢复等操作进行并行处理;

并发控制机制:充分利用记录级(行级)封锁和无竞争的读操作最大限度地减少并发操作引起的资源竞争和等待;

基于成本的查询优化:采用具有智能处理能力的算法,计算出最好的查询路径,优化数据存取的效率;

资源限制器:数据库管理员据此限制每一个用户对系统资源的消耗量,进行有效的管理和配置;

连续操作能力:系统中的关键性操作如备份、恢复及其它数据库管理工作都可以联机地进行,与前台的事务处理互不干扰;

完整性约束:利用数据库的存储过程和触发器可以开发一些公共过程来保证系统的完整性,以此提高数据库的可靠性,减少开发成本;

透明集成:透明的分布式数据库以及透明的网点技术支持新老系统的透明集成,这种集成可跨越不同的硬件、网络 and 系统软件平台;

两阶段提交:可以更有效地支持分布式事务处理,与其它数据库或文件系统连接,甚至与根本不支持两阶段提交的老系统一起,实现两阶段提交处理,保护数据的一致性;

数据复制技术:采用高效安全的方法将常用的数据复制到多个节点上,这是主要的分布支撑手段之一;

异种系统的集成:可以将各种数据源、常用的开发工具、应用软件以及第三方软件产品集成在一起,构成一个既符合工业标准,又易于用户接受的灵活的应用环境;

多线索服务器结构:提供多个共享服务进程,每个进程可服务于多个用户,大幅度地降低了资源需求,提高了系统,特别是存储器的效率;

支持先进的体系结构:配置有可伸缩性能和动态装载平衡功能,支持 SMP、MPP 等结构,并行服务器系统还支持集群机器和海量并行系统;

高可靠性:数据库及数据文件、数据库对象均可在线备份和恢复,充分发挥诸如硬盘镜像、多 CPU 等容错硬件平台的功能。

当然,用户是否必须拥有这些功能,或者是否有必要的硬件条件和客观需求以实现这些功能是另一回事。总之,数据库技术的发展已为我们的数据库应用提供了越来越丰富的选择空间。

1.4 主要厂商及数据库产品特点

1.4.1 几家主导厂商简介

数据库管理系统是典型的高技术产品,软件开发与研究是厂商的主要工作,“制造”相对简单一些。所以这一领域的新技术主要来自厂商之间的竞争,它们你追我赶,形成了相对的垄断局面。客观地说,几家顶尖厂商的产品是同一数量级的,技术实力相当,但由于用户对

大型软件的依赖集中在后续服务上，如培训、应用集成和系统开发。所以对厂商的特点和经营情况亦要有充分了解。

Microsoft：最大的软件厂商，主导数据库产品为 SQL Server，该产品初期与 Sybase 公司共同开发，1994 年后独立发展，主要的操作系统平台是 MS 的 Windows NT，是偏重微机平台的产品，特点是与 MS 的开发工具如 Visual Basic、Visual C++ 等配合极好，易于让 Windows 的开发者接受。MS 另有一个很著名的产品 FoxPro，许多人也称其为数据库，但 MS 仅称其为“个人数据库”，巧妙地否认此产品有完整的数据安全和共享等数据库必须具有的功能，所以 FoxPro 不是一个真正的数据库管理系统，而仅仅是数据库和文件系统之间的过渡。

IBM：最大的硬件厂商，主导数据库产品为 DB2，主要的操作系统平台是 IBM 自己的 OS/2 和其它主机操作系统，由于 IBM 有能力将软件和硬件融为一体，充分发挥各自的优势，使 DB2 在高端平台上稳占一席之地，而且由于 IBM 公司是主流机的霸主，是关系数据库的倡导者，其它数据库厂商也纷纷把与 DB2 的兼容作为自己的目标。

CA：仅次于 MS 的软件厂商，主导数据库产品是 CA Ingres，Ingres 是最早实现的关系数据库管理系统，后来被 CA 公司收购。CA 是一家侧重微机以上机型的软件业务的公司，而且经营应用软件，所以 CA Ingres 以其对大型系统的适应能力和应用配套能力上的优势占有自己的市场份额。

ORACLE：第三大软件公司，最大的数据库管理系统公司，主导产品是 ORACLE 7，ORACLE 长期以来雄居数据库市场的龙头宝座，它的产品线长，操作系统平台丰富，第三方支持多，技术优势大，更为重要的是数据库市场中最大的用户群已成为公司的显赫资本。ORACLE 曾被誉为最成功的数据库专业公司，但目前它在向系统软件、通用工具和应用软件方向发展。

Sybase：专业的数据库软件公司，主导数据库产品是 SQL Server 10，它曾与 MS 合作，这导致它最早在低端平台上引入图形界面，在办公系统领域赢得了大量客户，从而在客户/服务器结构中取得了巨大的成功，具有最高的市场占有率。

Informix：专业的数据库软件公司，主导数据库产品是 Informix Online，操作系统平台侧重于 UNIX 系统，曾发表第一个基于 UNIX 系统的关系数据库管理系统，其低端产品是 UNIX 平台上装机量最大的，因而影响十分广泛，当然 Informix 的技术水准亦十分先进，高端产品常为军队、银行等部门的大型应用项目选用。

HP：著名的计算机厂商，业务范围涉及计算机的各个领域，也包括数据库管理系统，主导数据库产品是 Turbo IMAGE，操作系统平台以 HP 自己的产品为主，它的用户群来自 HP 的传统用户，可以认为是 HP 的硬件优势带动了其在操作系统和数据库系统市场的发展。

Progress：数据库专业厂商，主导产品是 Progress ADE8，Progress 的知名度相对较小，但其 ADE V8 则是所有已发表的数据库管理系统中最先支持面向对象技术的，Progress 的市场份额较小，但仍然有相当的绝对数，比如世界各地的百事可乐工厂均同美国百事可乐公司一样，同步选用其作为数据库管理系统平台。

谁是最大的数据库系统厂商，标准很多，如销售额、装机数量、用户数、存取频率、存取数据量等，难于有所结论。应该说上述 8 家公司的产品都是业界颇有影响的产品，均不可小视。国际数据集团 (IDC) 曾报告了 1994 年上述 8 家公司在关系数据库管理系统市场上的收入，从销售额这一项上为 8 家公司排了位次，如表 1-3 所示。

表 1-3 1994 年在主要平台上数据库管理系统的授权收入

单位: 百万美元

	UNIX 收入	UNIX 份额	NetWare 收入	NetWare 份额	OS/2 收入	OS/2 份额	Windows NT 收入	Windows NT 份额
ORACLE	1010.2	42.5	94.9	56.2	22.7	18.4	6.7	8.6
Sybase	468.6	19.7	23.0	13.6	15.2	12.4	18.2	23.2
Informix	392.3	16.5	14.1	8.3	1.3	1.1	11.7	15.0
CA Ingres	113.1	4.8	2.3	1.4	0.9	0.7	0.9	1.2
HP	68.0	2.9						
Progress	46.9	2.0	8.0	4.8	5.7	4.6	5.7	7.2
IBM	4.9	0.2			46.5	37.9		
Microsoft					14.1	11.5	30.9	39.3

1.4.2 ORACLE 的产品线

所有关系数据库产品的功能基本相当, 其产品亦是类同的, 各个厂家的产品均得符合数据库及软件工程理论, 又受制于当时的硬件网络平台水平, 所以大同小异。为了初步了解关系数据库产品的组成结构, 我们在下面罗列了用户面最广、产品最齐全的 ORACLE 产品线, 另外, 为了增强可操作性, 同时照顾大多数用户的需要, 在后面的章节均以 ORACLE 7 数据库系统为背景展开讨论。

(1) 关系数据库管理系统 (RDBMS)

Oracle Parallel Server: 采用并行服务器技术, 支持几乎所有集群系统和海量并行处理系统, 它通过先进的高速缓存管理实现了最理想的性能/价格比、线性动态的性能缩放和优化、高可靠性, 并实现了数据库的统一和共享, 是关系型数据库功能的重大突破。

Oracle 7 Server: 真正释放出开放的关系数据库管理系统的潜能, 支持企业级的数据库环境, 具有高可靠性的性能设计以支持关键任务的应用, 具有可伸缩性, 以保证系统的扩展并适应将来的变化, 具有优化的性能/价格比, 以支持系统的开放性, 充分保护软件投资。

Oracle 7 Workgroup Server: 工作组服务器具备符合工业标准的数据库产品 ORACLE 7 的处理能力和性能, 并且使用非常方便, 通过使用工作组服务器中能加快和简化数据库管理的图形用户接口工具 (GUI), 可以方便地建立易于使用的可靠的客户/服务器应用。

Personal Oracle 7: 具有 ORACLE 7 数据库的所有功能和高性能, 但价格和使用的方便性与标准的 Windows 应用相当, 对于需要功能很强而应用又相对独立的数据库产品的用户和开发人员来说, Personal Oracle 7 是一个可以在 Windows 下运行的个人数据库产品, 又可以与 Oracle 7 Workgroup Server 连接构成分布数据环境。

(2) SQL 工具和编程接口

Oracle SQL * Plus: 提供强有力的、易于使用的查询定义和数据操作环境, 是标准 SQL