

新

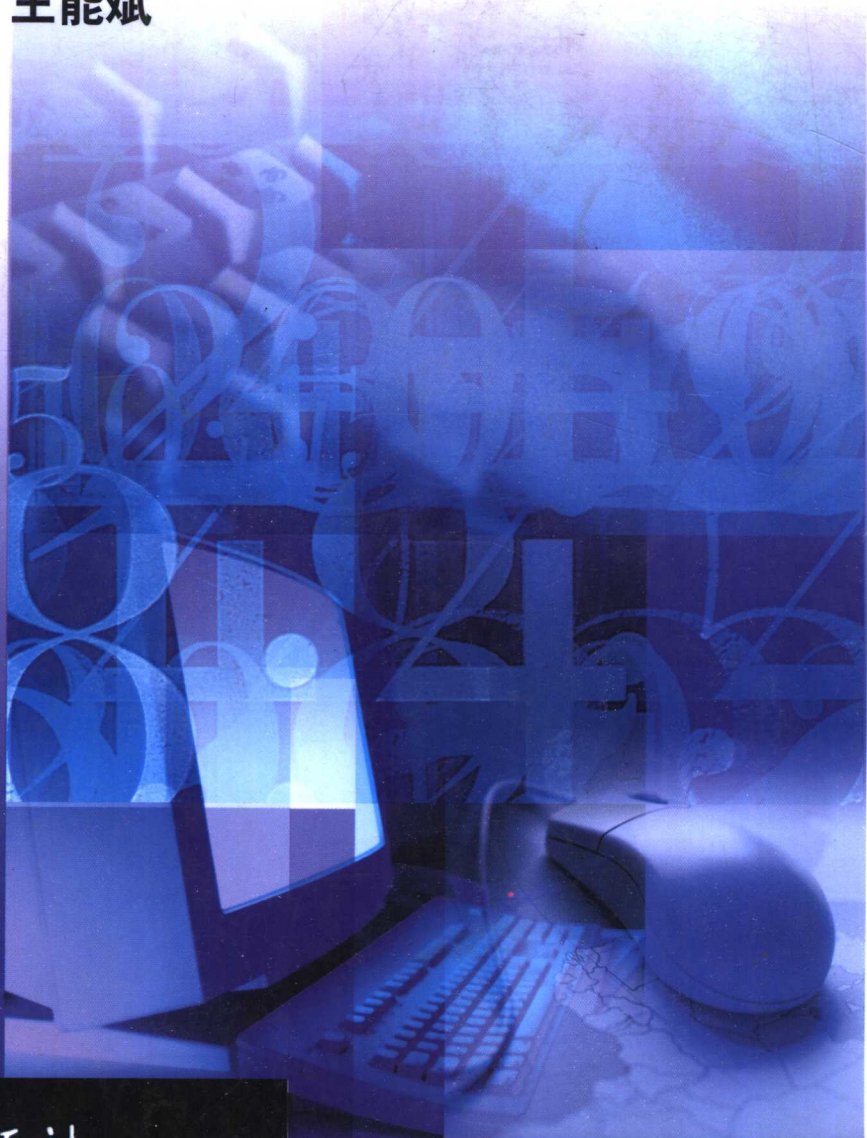


世纪高等学校计算机系列教材

数据库系统教程

下册

王能斌



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

新世纪高等学校计算机系列教材

数据库系统教程

下册

王能斌

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是一部全面、系统、深入阐述数据库系统原理的教程和科技参考书。全书共有六篇 23 章,分上、下两册。第一、二篇分在上册,共有 12 章;第三至六篇分在下册,共有 11 章。

第三篇介绍对象和对象-关系数据库系统及其实现技术。第四篇介绍非传统数据管理,包括空间数据库、时态数据库和多媒体数据库。第五篇介绍分布环境下的数据访问与管理,包括分布式数据库系统、并行数据库系统、异构多数据源的访问、XML 及其在数据管理中的应用。第六篇阐述数据仓库和 OLAP,介绍数据库在决策、数据挖掘、工作流、电子商务等领域中的应用。

本书取材新颖、内容丰富、文笔精炼、可读性好,着重对基本概念、基本原理和基本技术的阐述。其目的在于使读者学习以后,对数据库的主要新技术有系统、准确和深刻的理解,从而为参加研究、开发工作和阅读文献打好基础。本书可以作为计算机专业或其他专业研究生数据库课程的教材和高年级本科生的选修课教材或自学教材,也可作为广大科技人员在职进修的自学读物和工作中的参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

数据库系统教程. 下册/王能斌. —北京:电子工业出版社,2002.8

新世纪高等学校计算机系列教材

ISBN 7-5053-7828-7

I. 数… II. 王… III. 数据库系统—高等学校—教材 IV. TP311.13

中国版本图书馆 CIP 数据核字(2002)第 051731 号

责任编辑:张荣琴 特约编辑:吴明卒

印 刷:北京东光印刷厂

出版发行:电子工业出版社 <http://www.phei.com.cn>

北京市海淀区万寿路 173 信箱 邮编 100036

经 销:各地新华书店

开 本:787×1092 1/16 印张:16.25 字数:416 千字

版 次:2002 年 8 月第 1 版 2003 年 7 月第 2 次印刷

印 数:5 000 册 定价:21.00 元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。
联系电话:(010)68279077

总 序

为了认真贯彻《中国教育改革和发展纲要》和教育部“面向 21 世纪教育振兴行动计划”精神,适应现代社会、经济、科技、文化,特别是教育的发展方向,适应培养新世纪计算机人才的需要,根据计算机科学技术学科发展和新世纪高等学校教学内容和课程体系改革的要求,我们决定编写这套《新世纪高等学校计算机系列教材》。

教学改革和教学质量是高等学校的头等大事。教学改革的关键在于教材的改革,我们组织编写这套系列教材的指导思想是:立足于新世纪培养创造型开拓型人才的培养目标,立足于高速发展的计算机科学技术,科学地预测本学科前沿发展趋势;注重教材内容的思想性、科学性、先进性、系统性和广泛的适应性,继承与改革现有教材内容和体系;注重加强能力的培养,切实提高学生的综合素质。同时,为了适合我国国情,适应绝大多数高等学校计算机科学与技术专业类的教师知识结构和课程改革的实际情况,一方面,我们尽力拓宽教材内容,充实和加强有关理论,引用新的科技成果;另一方面,我们兼顾各门课程的历史、现状与发展,在具体章节的内容编排上力求循序渐进,顺理成章。在保证学科系统的前提下,充分考虑到学生的接受能力,使教材体系尽量符合学生的认知规律,便于学生接受、理解、掌握和巩固所学知识,促使学生学以致用,理论密切联系实际。

本套系列教材的组织编写得到了中国计算机学会、江苏省计算机学会的大力支持和帮助,凝聚了参加编审出版工作的许多教师和编辑的心血和汗水,在此一并对他们表示衷心的感谢。

限于我们的水平、能力和经验,本套系列教材的编审与出版工作一定还存在着很多错漏和其他不如意之处,我们真诚地希望使用这套教材的高校教师、学生和广大读者积极提出宝贵意见,以鞭策我们不断提高教材的编写、修订和出版质量,为我国的计算机科学技术专业教材建设服务。

《新世纪高等学校计算机系列教材》编辑委员会

前 言

本书是一部面向本科生和研究生的数据库系统教程。全书共有六篇 23 章,分为上、下两册。

第一篇为序篇,其中第 1 章概括地介绍数据管理的发展,数据库系统的构成以及基本的术语和概念;第 2 章阐述有代表性的数据模型,分别介绍其概念、结构、约束和操作。

第二篇共分为 10 章,全面阐述关系数据库系统,包括关系数据库语言、DBMS 原理及实现技术、数据库设计以及数据库的运行、维护和管理。关系数据库是当前应用最广、技术最成熟的一种数据库。对于数据库的初学者,先通过关系数据库全面掌握数据库的基本概念、理论、技术和方法,再在此基础上学习关系数据库的各种扩充或非关系数据库,不失为一种循序渐进、符合认识规律的学习方法。如果学时有限,作为数据库的入门课,也可以只学关系数据库。本书把关系数据库的内容全部集中在第二篇中,并列为上册的主要内容,其目的在此。

第三篇论述对象和对象-关系数据库系统。对象数据模型是继关系数据模型之后的最重要的数据模型,也是支持各种非传统数据库应用的基础。本篇的第 13 章介绍对象数据模型和对象数据库语言;第 14 章介绍对象 DBMS 的实现技术。

第四篇论述非传统数据的管理,这是近 20 多年来数据库的重要成就之一,其应用愈来愈多。本篇按空间数据库、时态数据库和多媒体数据库分别阐述其所需的特殊功能及实现方法。

第五篇论述分布环境下的数据访问与管理。除系统地介绍分布式数据库管理系统外,还介绍并行数据库系统以及用得比较多的各种异构多数据源访问技术。XML 已成为描述半结构化数据的标准语言;XML 文档管理也将成为数据库的标准功能之一。第 21 章专门阐述此问题。

第六篇从实现技术角度介绍数据库所支持的一些新应用。数据库对决策的支持是其中用得较广的一种,第 22 章介绍了数据仓库和 OLAP。第 23 章介绍数据挖掘、工作流和电子商务等三个以数据库为基础的应用,并结合工作流扼要地介绍非传统事务模型及其应用,结合电子商务简单介绍分布环境下的安全问题及解决途径。

历经 40 年的发展,数据库在理论和技术上都有丰富的积累,并正在蓬勃地发展。国际上新出版的数据库教材大部分都在千页左右。为了使读者能系统地掌握数据库内容,不致在读后有支离破碎之感,作者初次尝试按上述系统组织本教程。尚祈海内外同行和广大读者,不吝赐教,以资改进。

本书着重基本原理、基本概念和基本技术的阐述,力图使读者对数据库系统有全面、深入、系统的领会。各种 DBMS 产品,除用于举例外,不做专门介绍。具体技术很容易过时,而基本原理、概念和技术可起相对持久的作用。在掌握了基本原理、概念和技术后,读者不难通过阅读手册或培训材料较快地熟悉产品,并以宽阔的视野分析和评价产品。

教材不同于文献综述,教材应主要阐述相对成熟的内容。对于那些尚在探讨中的新内容,本书除在展望中偶尔提及外,不做详细介绍。如果读者因研究需要,须深入了解某一研究方向的发展和动态,可查阅有关文献。

标准是技术发展的反映或导向。虽然其中难免有折衷平衡的成分,但总比个别产品全面、

系统、周到些。本书尽可能向现有标准靠拢,但不介绍标准本身。在数据库的诸多标准中,SQL:1999也许是最重要的。在本书中,介绍了递归查询、存储过程、按角色授权、触发子、对象-关系数据库、SQL/CLI等SQL:1999新增的内容。

本书可满足各种课程设置的需要。作为入门的数据库概论或DBMS概论课程,可选用上册的内容。下册的四篇可作为本科生高年级选修课或研究生课程的教材。这四篇的内容相对独立,可以根据学时及研究方向选学其中的一部分或全部。上册也可用做非计算机专业研究生课程教材。目前学生看参考书比较少,无适合的参考书是原因之一。本书的内容超出一般课堂教学的需要,可以选择其中的一些内容作为学生的参考材料或自学材料。本书便于自学,可供广大科技人员学习、参考。

本书是由徐宝文教授倡议编写的;Michigan大学朱强教授和Nova Southeastern大学计算机及信息研究生院孙俊平教授赠送美国新出版的数据库教材多本,这都是对作者宝贵的支持,作者在此表示诚挚的谢意。本书是在作者过去的科研、教学工作和编写教材实践的基础上写成的,对于曾在这方面提供过帮助的谢希仁教授、徐洁磐教授以及本校教师董逸生、孙志挥、徐宏炳、金远平、陈钢、徐立臻和作者所指导过的历届研究生表示感谢。对于曾支持过作者研究经费的国家自然科学基金委员会、863高技术计划、原国防科工委、原电子工业部等部门表示感谢。电子工业出版社吴浩源先生多年来给予作者很大的帮助。夫人包桂林女士长期以来给予作者充分的理解、支持和慰勉,使作者能在愉快的氛围中笔耕不辍,终成此书。对此作者深表感谢。

由于编者水平有限,书中难免存在一些缺点和错误,殷切希望广大读者批评指正。对本书的意见请按电子邮件地址 nbwang@seu.edu.cn 反馈给作者,谢谢。

王能斌
于玄武湖畔
2002年3月26日

目 录

下 册

第三篇 对象和对象-关系数据库系统

第 13 章 数据模型和数据库语言

13.1 引言	(2)
13.1.1 关系数据库的弱点	(2)
13.1.2 对象和对象-关系数据库的兴起	(6)
13.2 对象数据库	(7)
13.2.1 概述	(7)
13.2.2 对象模型	(8)
13.2.3 对象定义语言	(13)
13.2.4 对象查询语言	(14)
13.3 对象-关系数据库	(17)
13.3.1 概述	(17)
13.3.2 行类型	(18)
13.3.3 抽象数据类型	(22)
13.3.4 聚集类型和区分类型	(24)
习题	(24)

第 14 章 系统实现中的一些问题

14.1 查询处理与优化	(25)
14.2 存储结构	(26)
14.2.1 物理簇集	(27)
14.2.2 内外存存储结构的转换	(29)
14.3 并发控制	(31)
14.4 方法的实现、管理和调用	(33)
14.5 模式演化	(34)
14.5.1 对象数据库中的模式演化	(34)
14.5.2 模式修改	(35)
14.5.3 数据修改	(38)
14.6 版本管理	(39)
14.7 结束语	(40)
习题	(41)

第四篇 非传统数据的管理

第 15 章 空间数据库

15.1	引言	(44)
15.2	空间数据模型	(45)
15.2.1	空间数据类型	(45)
15.2.2	空间数据的操作	(45)
15.3	空间索引	(49)
15.3.1	空间索引的基本概念及发展	(49)
15.3.2	kd-树	(52)
15.3.3	G-树	(53)
15.3.4	R-树及其变种	(56)
15.4	空间数据库系统结构	(60)
习题		(62)

第 16 章 时态数据库

16.1	引言	(63)
16.2	概念和术语	(64)
16.3	TSQL2 语言	(67)
16.3.1	TSQL2 的时态数据模型	(67)
16.3.2	TSQL2 语言的特点	(69)
16.4	时态 DBMS 的实现	(76)
习题		(79)

第 17 章 多媒体数据库

17.1	引言	(81)
17.1.1	多媒体数据简介	(81)
17.1.2	多媒体数据的特点及多媒体数据库	(83)
17.2	存储系统	(84)
17.3	元数据及其生成	(86)
17.4	查询及索引	(89)
17.5	等时、同步和演示管理	(93)
17.5.1	演示管理	(93)
17.5.2	数据流管理	(94)
17.5.3	同步	(96)
习题		(97)

第五篇 分布环境下的数据访问与管理

第 18 章 分布式数据库管理系统

18.1	分布式数据库系统	(100)
18.2	数据分布策略	(101)
18.2.1	数据分布的目的	(101)
18.2.2	数据分布的方式	(101)
18.2.3	关系的分割	(102)
18.2.4	数据分布带来的问题	(104)
18.3	分布式数据库系统结构	(104)
18.4	数据目录的分布及管理	(105)
18.4.1	数据目录的分布策略	(106)
18.4.2	分布式数据库系统中的命名	(107)
18.4.3	分布式 DBMS 数据目录管理举例	(108)
18.5	查询分解和优化	(109)
18.5.1	查询树的变换	(110)
18.5.2	复本的选择	(111)
18.5.3	查询树的分解	(111)
18.5.4	全局查询优化的一般问题	(112)
18.5.5	半连接在分布连接中的应用	(113)
18.5.6	用直接连接进行分布连接	(115)
18.5.7	多复本的更新	(116)
18.6	分布式数据库系统中的恢复技术	(117)
18.6.1	两步提交协议	(117)
18.6.2	三步提交协议	(118)
18.7	分布式数据库系统中的并发控制	(119)
18.7.1	分布式数据库系统中的两段封锁	(119)
18.7.2	全局死锁的检测	(121)
18.8	逻辑上分布的 DDBS 的概念	(122)
18.9	移动数据库	(124)
18.10	结束语	(125)
习题		(125)

第 19 章 并行数据库系统

19.1	引言	(127)
19.2	并行化粒度及操作的并行性	(129)
19.2.1	并行化粒度	(129)
19.2.2	操作的并行性	(129)
19.3	并行数据库系统结构	(130)
19.4	数据划分	(133)
19.5	并行算法	(134)

19.5.1 并行排序	(134)
19.5.2 并行连接	(135)
19.5.3 其他关系运算的并行执行	(137)
19.6 查询优化	(138)
19.7 结束语	(141)
习题	(141)

第 20 章 异构多数据源的访问

20.1 引言	(142)
20.2 ODBC	(142)
20.2.1 ODBC 的由来和发展	(142)
20.2.2 基本原理和方法	(143)
20.2.3 SQL/CLI 内容简介	(145)
20.2.4 与嵌入式 SQL 的比较	(149)
20.3 COM/DCOM	(150)
20.3.1 COM/DCOM 的由来和发展	(150)
20.3.2 COM 简介	(150)
20.3.3 DCOM 简介	(154)
20.4 CORBA 及其在异构数据集成中的应用	(156)
20.4.1 CORBA 的设计思想和发展	(156)
20.4.2 CORBA 的基本概念	(157)
20.4.3 CORBA 的系统结构	(158)
20.4.4 CORBA 在异构数据集成中的应用	(161)
20.5 Java Beans	(162)
20.6 万维网与数据库	(165)
20.6.1 引言	(165)
20.6.2 万维网与数据库系统互操作	(166)
习题	(168)

第 21 章 XML 及其在数据管理中的应用

21.1 引言	(169)
21.2 XML 文档	(171)
21.3 XML 文档类型定义	(174)
21.4 XML 模式	(176)
21.5 XML 查询语言	(182)
21.5.1 概述	(182)
21.5.2 XPath	(183)
21.5.3 XSLT	(185)
21.5.4 XQuery	(187)
21.6 XML 在数据管理中的应用	(189)
21.6.1 XML 的应用及 XML 文档管理	(189)
21.6.2 XML 文档关系模式的生成	(190)

21.6.3 查询处理	(192)
习题	(195)

第六篇 数据库的一些新应用

第 22 章 数据库在决策中的应用

22.1 决策支持系统及其对数据库系统的要求	(198)
22.2 数据仓库框架	(199)
22.3 数据仓库的基本数据模式	(200)
22.4 数据仓库的基本操作	(202)
22.5 OLAP 的实现技术	(207)
22.5.1 关系 OLAP 和多维 OLAP	(208)
22.5.2 OLAP 的查询优化	(210)
22.6 位图索引	(211)
22.7 OLAP 数据的维护	(215)
习题	(218)

第 23 章 其他应用举例

23.1 数据挖掘	(219)
23.1.1 概述	(219)
23.1.2 数据挖掘过程	(220)
23.1.3 数据挖掘的基本方法	(221)
23.1.4 关联规则	(222)
23.1.5 序列挖掘	(224)
23.1.6 分类	(227)
23.1.7 聚类分析	(230)
23.2 工作流	(232)
23.2.1 概述	(232)
23.2.2 工作流的概念、功能和结构	(233)
23.2.3 支持事务的工作流	(234)
23.3 电子商务	(237)
23.3.1 概述	(237)
23.3.2 电子商务系统的特点	(237)
23.3.3 电子商务的安全	(238)
23.4 结束语	(239)
习题	(240)
参考文献	(241)

第三篇 对象和对象-关系 数据库系统

第 13 章 数据模型和数据库语言

第 14 章 系统实现中的一些问题

面向对象数据模型是继关系数据模型以后最重要的数据模型。两者各有所长,也各有所短,因而各有其适用范围。它们不是互相排斥或取代,而是相辅相成。

本篇在关系数据库的基础上介绍对象和对象-关系数据库系统。第 13 章分别介绍对象数据库和对象-关系数据库的数据模型和数据库语言,尽可能向 ODMG 工业标准和 SQL:1999 标准靠拢。第 14 章介绍 ODBMS 和 ORDBMS 实现技术的一些特点,与 RDBMS 相同或类似的技术不再重复。

第 13 章 数据模型和数据库语言

13.1 引言

13.1.1 关系数据库的弱点

自 20 世纪 80 年代以来,关系 DBMS(以下简称 RDBMS)一直是 DBMS 产品的主流。随着计算机应用的发展,关系数据库也逐步暴露其弱点,这些弱点主要表现在下列 4 个方面。

1 有限的数据类型

关系数据库所能支持的数据类型限于 RDBMS 所定义的有限的几种基本数据类型,表 3-1 是其一例。尽管 SQL 标准和 RDBMS 厂商都在不断地扩充基本数据类型,仍难满足多种应用的需要,特别是多媒体数据管理的需要。为了存取声音、图形、图像、动画、影视、文本等多媒体数据,不少 RDBMS 增加了二进制大对象(Binary Large Objects,简称 BLOB)数据类型。然而,对 BLOB 的结构及操作,DBMS 没有定义。用 BLOB 虽可存储形形色色的多媒体数据,但对其操作或按内容查询,DBMS 却无能为力,只有交给应用程序去处理。严格地说,BLOB 不能算是 DBMS 中的一种数据类型,它只能算是 DBMS 管理下的一种文件,DBMS 仅仅为应用程序做一些相当于打开、关闭之类的文件操作。利用 BLOB 存取多媒体数据,不可避免地增加应用程序开发者的负担,而且由用户的应用程序直接处理数据库中的共享数据,也有碍数据的安全和一致性的维护,有违数据库的基本宗旨。总之,目前 RDBMS 中用 BLOB 存取多媒体数据,是权宜之计,不是根本的解决办法。实际应用中数据类型的多样化和 RDBMS 中数据类型的有限性,依然是一个突出的矛盾。

2 缺少全系统惟一、不依赖于属性值、类似于 OID 的标识符

在 2.6 节中,曾介绍过 OID 的概念。OID 是由系统生成的、不依赖于属性值的、全系统惟一的对象标识符。OID 可以用做对象的替身来访问它,甚至可以区别两个属性值完全相等的对象。OID 比起对象本身一般要短小,OID 可以通过系统提供的机制快速地映射成对象在内存或磁盘上的地址。实际上,OID 可以看成对象的逻辑指针。逻辑指针由系统自动维护,不会出现“指针悬空”等不一致问题,它为对象提供不依赖于存储空间分配的快速访问机制。在实际应用中,常会遇到由多个对象构成的所谓复杂对象,例如一台机器、一个软件、一本书等。在访问复杂对象时,常常要从一个对象追寻到另一个对象,而且往往会层层追踪,这有点像网状、层次数据库中曾经出现过的导航式访问,不过网状、层次数据库中用的是物理指针。对于导航式访问,OID 是方便、有效的访问手段。

关系数据库是按属性值选取数据的,例如用主键值选取某个元组,按属性值的限制条件选取一个元组集等。总之,关系数据库是按数据的内容访问数据的,也就是平常所称的联想访问

(associative access)。联想访问的优点是：只须指明访问的要求，不须关心访问的过程。在学习关系数据库时，读者很熟悉这种访问方法，而且也体会其优点。但是，联想访问不是在所有情况下都是方便的。例如，在访问复杂对象的内部结构时，导航式访问就显得直观、有效，应用联想访问反而显得低效和不易理解，这可从下面将要介绍的例子看出。

OID 不但有利于数据的访问，也有利于数据的管理和维护。主键只是在一个关系中是惟一的，而不是在全系统是惟一的。随着数据库的修改，主键的值也会变动。尽管在设计数据库时，设计者总是力图选取一些相对稳定、不重复的属性或属性组作为主键，例如学号、居民身份证号等。但是它们毕竟是学生、居民的属性，在一定范围内用来识别实体是可行的，但用做全系统范围内的识别标志是不够的。因为这些属性只适用于学生和居民，且在不同的学校和同一学校的不同时期，学号的格式未必相同；居民身份证也不是每人都有，年龄未满 16 岁以及 1986 年以前去世的人就没有居民身份证。总之，用某些属性值来识别实体总是不可避免地带来这样或那样的局限性。根本出路在于数据的管理者，即 DBMS，赋予每个被管理的实体或对象一个由系统生成的、惟一不变的标识符，即 OID。在不少 RDBMS 中，为了识别元组，常常为每个元组提供一个由系统生成的元组标识符，即 tid。但 tid 仅在一个关系中是惟一的，它仍取代不了 OID 的作用。

如 2.6.4 小节所述，OID 有逻辑 OID 和物理 OID 之分。物理 OID 依赖于对象的地址，用得不多。在本章中，所有 OID 均指逻辑 OID。

3 不支持用户可定义的或系统可扩充的函数或运算

一般的 RDBMS 只支持常用的算术、逻辑、字符串、位串、时间等的运算和一些比较运算以及少许聚集函数。超出此范围，只有交给应用程序去处理。这不但对用户带来不便，而且也会严重影响数据库的性能。这可以从下例看出。

例 13-1 设有关系 image(图像)，其模式定义为

```
CREATE TABLE image
  (image-no CHAR(8) NOT NULL,
   image-body BIT VARYING (1048576),
   PRIMARY KEY(image-no));
```

此关系有两个属性：image-no(图像号)和 image-body(图像体)。图像体以 BLOB 形式存储。在 SQL-92 中，BLOB 可表示为变长位串类型，即 BIT VARYING，括号中的无符号整数表示允许的最大位数，在本例中为 1024×1024 位。

本例要求将所有图像沿顺时针方向旋转 45° 。由于 DBMS 不提供这种旋转图像的操作，应用程序只能要求 DBMS 依次取出图像体，传给应用进程，由应用程序对图像进行旋转处理后，再传至 DBMS 进程，以更新原有的图像。每处理一幅图像，须访问数据库两次（一取，一存），须在应用进程与 DBMS 进程之间传送图像两次（一来，一往）。这样处理的效率是很低的。尤其在客户/服务器系统中，应用进程与 DBMS 进程一般分处于不同机器上，这样处理的效率更低。

如果 DBMS 有一个函数，例如 ROTATE(image, degree)，可将图像 image 沿顺时针方向旋转 degree 角度，则上述处理便可写成下面的 SQL 语句。

```
UPDATE image
  SET image-body=ROTATE(image-body,45);
```

这样，应用程序只要向 DBMS 发出上述的 SQL 语句，DBMS 就可以完成所有处理，不需要应用程序的干预。不但省去了图像在 DBMS 进程与应用进程之间的传送，且在整个处理过

程中,应用程序只须访问数据库一次。而且在处理每幅图像时,读-修改-写的全过程都可在一个缓冲区中完成。

从例 13-1 可以看出,如果 DBMS 能够根据应用的需要,允许扩充一些函数或运算,可以方便用户,提高性能。至于由谁来扩充,可以是用户,也可以是系统管理人员。前者的好处是给用户很大的自由,用户可以根据应用的需要自行扩充,但是由于用户定义的函数和运算在质量上难以保证,可能影响 DBMS 的可靠运行。由系统管理人员定义函数和运算,可望在质量上有所保证,但用户的自由不得不有所限制。在现有的 RDBMS 中,都不允许用户或系统管理人员扩充函数或运算,限制了它的功能,这是一个很大的缺陷。

4 不能清晰表示和有效处理复杂对象

如所周知,关系数据模型必须遵守第一范式的限制,即所有属性必须是原子的。在实际应用中,常常要求属性为非原子的,即允许用某些类型构造子(type constructor)构造属性。例如一位学生可以选多门课,如果能用集合表示选课这个属性,则表示起来就方便得多。又如一台机器由若干种部件组成,每种部件又由若干种零部件组成。部件实质上是非原子属性,它是一个元组,由组成它的零部件构成。在关系数据模型中,为了遵守第一范式的限制,只能用部件的主键(例如部件号)作为外键来参与组成。若要查询机器的某个部件的组成,必须进行一次相当于连接的关系操作,而不能自然地由机器查询其部件,由部件查询其组成的零、部件。上述的集合、元组说明这些非原子属性的构造方式,统称为类型构造子。除了集合、元组外,常用的类型构造子还有表、数组、可重复集合、队列和堆栈等。凡含有由类型构造子构造的非原子属性的对象称为复杂对象。关系数据模型虽可用第一范式表示这些复杂对象,但表示不直观,处理的效率不高。这可以通过下面的例子看出。

例 13-2 图 13-1 是台式计算机的硬件构成图,试写出其关系模式。

若用关系数据模型表示图 13-1 的内容,可设计两个表(关系),见表 13-1 和表 13-2。

表 13-1 零部件表

编号	是否组合件	名 称
P1	是	台式计算机
P2	是	主机
P3	是	显示器
P4	是	辅助设备
P5	否	主机箱
P6	否	主机板
P7	否	I/O 板
P8	否	电源
P9	否	软盘驱动器
P10	否	硬盘驱动器
P11	否	光盘驱动器
P12	否	显示器及外壳
P13	否	显示器电源及线路
P14	否	鼠标器
P15	否	键盘
P16	否	打印机

表 13-2 装配表

部件编号	含有零部件编号
P1	P2
P1	P3
P1	P4
P2	P5
P2	P6
P2	P7
P2	P8
P2	P9
P2	P10
P2	P11
P3	P12
P3	P13
P4	P14
P4	P15
P4	P16

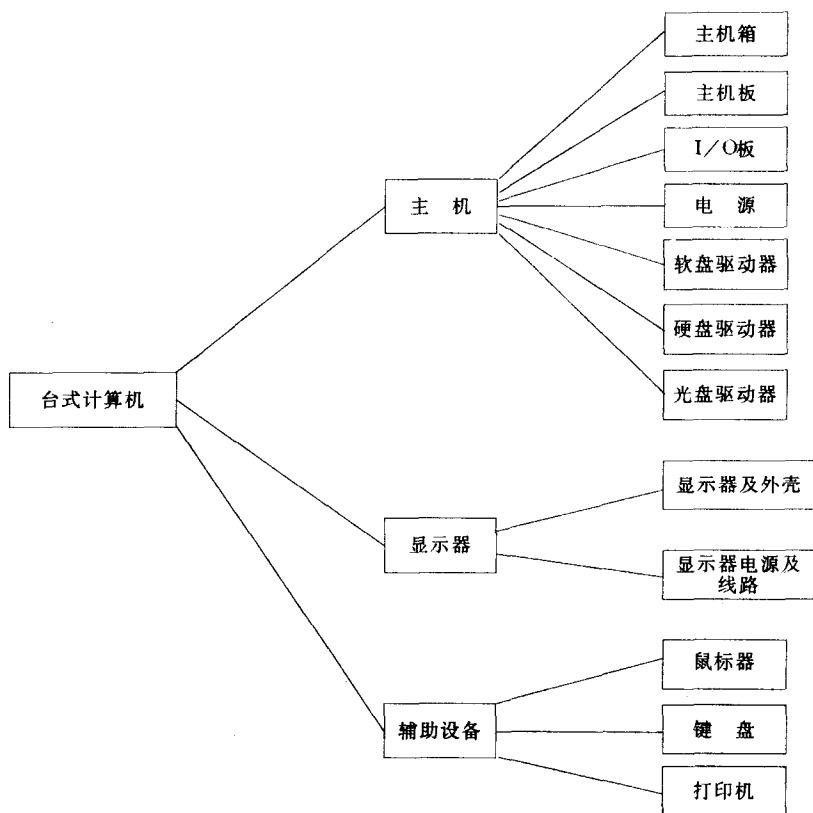


图13-1 台式计算机的硬件构成图

由于在关系数据模型中,属性不能是集合,P1 含有{P2,P3,P4}这一简单事实须用三个元组表示;P2 含有七个零件,须用七个元组表示;余类推。为了减少数据冗余和关系模式的规范化(参看第 10 章),装配关系和零部件的描述不得不分在表 13-1 和表 13-2 中。这对有些处理会带来不便和开销。事实上,表 13-1 中的“是否组合件”属性是冗余的,因为凡是“编号”出现在表 13-2“部件编号”属性中的都是组合件,否则不是组合件。为了提高处理效率,设置了此冗余属性。读者可以作为一个作业,设计一个生成 P1 材料单(即列出构成 P1 的所有零件)的程序。在设计时,可以使用嵌入 C 语言的 SQL 和创建必要的临时关系。由于部件中可以包含部件,这是一个递归查询问题。现有 RDBMS 有些还不支持递归查询,使用嵌入式 SQL 是必要的。读者不难发现,此项处理过程是繁琐和低效的。实际的机器构成远比图 13-1 复杂。图 13-1 中有些零件,例如主机箱,还可分成很多零件或部件,例如插件座、螺钉、风扇等,有些零件还可能在同一部件中使用多个。因此,在关系数据库上处理材料单这样一种典型的复杂对象问题是可能的,但远不是直观和有效的。

关系数据库的弱点还可举出不少,例如关系间缺少继承机制,不利于模式的演化等。作者认为,上述 4 点是最本质的。还有些弱点,例如缺少规则处理能力,不是关系数据库所固有的,本书也没有把它说成关系数据库的弱点。事实上,现在已有不少 RDBMS 增加了规则处理功能。

13.1.2 对象和对象-关系数据库的兴起

为了克服关系数据库的弱点,早在 20 世纪 80 年代初期,也正是关系数据库逐步成为数据库主流的时候,人们就开始探讨关系数据库后的下一代数据库问题。关系数据库的固有缺点是由于关系数据模型引起的。人们研究的重点一开始就很自然地集中在数据模型上。早期提出的有代表性的数据模型有非一次范式(Non-First normal Form,简称 NF²)关系数据模型和语义数据模型。它们都从不同的方面,部分地克服了关系数据模型的一些缺点。然而最有影响的要数稍后崭露头角的面向对象数据模型。面向对象数据模型在本书的 2.6 节中介绍过。读者在学习本章前最好再仔细阅读该节。从 2.6 节可知,面向对象数据模型完全可以克服上述关系数据库的四个弱点。加之,面向对象程序设计语言和面向对象设计方法近年来得到日益广泛的应用。应用的需要和用户的要求也促使 DBMS 逐步在不同程度上反映面向对象数据模型的特点。20 世纪 80 年代中后期是人们探索面向对象数据模型与 DBMS 结合的时代,其意义及规模不亚于 20 世纪 70 年代中后期在关系数据库方面的实践。

20 世纪 80 年代中期以后,不少公司,特别是一些新成立的小公司,把开发以面向对象数据模型为基础的对象数据库管理系统(Object Data Base Management System,简称 ODBMS)看成新的机会,纷纷投入 ODBMS 的研制和开发,推出了一批早期的 ODBMS 产品,例如 Gemstone, Ontos, O₂, Itasca, Versant, ObjectStore 等。这些产品有下列 3 个特点。

(1) 由于当时还没有公认的面向对象数据模型,更无这方面的标准。这些产品在数据模型和数据语言方面不少借鉴面向对象程序设计语言,例如 Smalltalk-80, C++, Common-Lisp 等。虽然也有些 ODBMS,例如法国的 O₂,自行设计了 SQL 风格的查询语言,但它与 SQL 并不兼容。

(2) 这些 ODBMS 产品在 DBMS 结构、功能和实现技术方面借鉴了 RDBMS,例如普遍采用客户/服务器结构,以事务作为数据库的处理单元,具有并发控制、恢复和安全控制功能,在物理层采用簇集、索引等技术以提高性能等。但是 ODBMS 毕竟不同于 RDBMS,在实现时有其特点。这些产品为 ODBMS 的实现提供了最初的经验。

(3) 当时认为计算机辅助设计(Computer-Aided Design,简称 CAD)、计算机集成制造(Computer-Integrated Manufacturing,简称 CIM)等是 ODBMS 最主要的潜在应用领域。因此 ODBMS 普遍地增加了 CAD, CIM 所需要的功能,例如版本管理、长事务管理和 check out/check in 机制(参阅 4.4 节)等。这些功能并非 ODBMS 所特有, RDBMS 也可具有此功能。

与此同时,不少大学和研究单位也开始重视对象数据库的研究,或研制 ODBMS 原型,或进行专题研究。对象数据库成为当时数据库研究的热点之一。不少的人认为,对象数据库将要取代关系数据库,成为下一代数据库的主流。形势的发展并没有像这些人所预料的那样。ODBMS 产品虽然以较快的速度发展,但在 DBMS 市场上所占的份额很小,对传统的 RDBMS 并不构成威胁,更谈不上取代了。分析其原因,主要有下列 3 点。

(1) 目前数据库最大的应用领域仍是传统的企事业管理信息系统。在这类应用中,虽有处理复杂对象以及扩充数据类型和相应的函数、运算的要求,但大量的任务攸关(mission critical)的应用仍是联想访问。ODBMS 胜任不了这样的任务。

(2) 如前所述, ODBMS 的设计主要面向 CAD, CIM 之类的应用。它不适宜于很多用户共享数据和大量事务联机处理的应用环境。ODBMS 的安全措施及恢复机制也满足不了企事业