

微型计算机实用数据库技术大全

微型计算机 实用数据库技术大全



UNIFY
ORACLE
FOXPRO

主 编 袁晓洁 朱耀庭 编 者 辛运伟 于 红

天津科技翻译出版公司

1.13-01

微型计算机实用数据库技术大全

主	编	袁晓洁	朱耀庭
编	者	袁晓洁	朱耀庭
		辛运伟	于红

天津科技翻译出版公司

津新登字 90(010)号

责任编辑:王秀兰

微型计算机实用数据库技术大全

袁晓洁 朱耀庭 主编

天津科技翻译出版公司出版

(邮政编码:300192)

新华书店天津发行所发行

河北省雄县胶印厂印刷

* * *

开本:787×1092 1/16 印张:35 字数:870 千字

1994 年 10 月第 1 版 1994 年 10 月第 1 次印刷

印数:5000 册

ISBN 7-5433-0743-X

TP·19 定价 32.00 元

内 容 简 介

《微型计算机实用数据库技术大全》是一本集目前国际上流行的关系数据库管理系统 ORACLE、UNIFY、FoxPro 为一体的数据库实用教材。全书分四部分,详细讨论了关系数据库模型、关系数据库标准查询语言 SQL 和三种不同的关系数据库管理系统:ORACLE、UNIFY 及 FoxPro。书中对如何使用这三种数据库管理系统作了详实的说明。该书既可以作为高等院校计算机系和相关学科师生学习数据库理论及其应用的教学参考书,又可以作为一切微型计算机数据库应用软件开发人员的常备工具书。

前 言

作者多年来一直从事数据库的教学与科研工作,曾开发了多个实用的数据库管理应用软件,诸如工厂经营管理、财务管理、人事管理;学校教学管理、科研管理、图书管理;证券交易、期货交易等等。其中许多是用 ORACLE、UNIFY 和 FoxPro 开发的。在开发过程中往往苦于没有详实的资料而东奔西走,非常希望能有一本全面介绍目前各种关系数据库的书籍。正是从这一点出发,几位作者根据多年来的教学和科研工作的经验编辑出版了该书,奉献给大家,希望能够对广大读者有所帮助。

全书共分四部分。第一篇数据库概论详细介绍了什么是关系数据模型,什么是数据库管理系统,以及广泛应用于关系数据库系统的标准查询语言 SQL。第二篇 ORACLE 以 ORACLE5.1 版为重点,详细介绍了在 DOS 下从安装到使用的全部过程。这一篇共八章,包括概述、SQL * PLUS 指南、表格设计 SQL * FORMS、报表软件 SQL * REPORT、交互式图形工具 SQL * GRAPH、电子表格 SQL * CALC、预编译接口以及 ORACLE 的运行与管理。第三篇 UNIFY 数据库管理系统及其应用共六章。内容包括概述、数据库设计与维护、如何编辑屏幕表格、ENTER 数据录入和表格查询、SQL 的使用、UNIFY 的系统管理、报表处理、UNIFY 与 C 语言的接口等。第四篇 FoxPro 以最新的 FoxPro2.5 版为重点,共十三章,内容包括数据库的建立、数据库信息的维护、排序和检索、数据库查询、报表生成、VIEW 窗口及相关数据库、屏幕格式生成器、菜单生成器的使用、应用程序设计、应用程序接口等。每一篇都配有命令及速查手册。全书自始至终都有简单的编程实例供参考。

本书由袁晓洁、朱耀庭主编,参加编写的有朱耀庭、袁晓洁、辛运伟、于红、刘淑贞、田耕、徐雷、吴拉朵、万广林。本书编辑出版过程中得到了南开大学计算机与系统科学系有关领导的大力支持,特别是软件教研室陈有祺教授、朱瑞香教授、韩维恒教授的大力帮助。在此表示衷心的感谢。天津科技翻译出版公司的王秀兰编辑从本书的提纲拟定、组稿到编辑,自始至终作了大量的工作,在此一并致谢。

由于编者水平有限,编写时间仓促,书中难免有不当之处,敬请读者指正。

编 者

1994 年 9 月 18 日

目 录

第一篇 数据库概论

第一章 关系数据模型与数据库系统	(1)
§ 1.1 文件技术与数据库技术	(1)
§ 1.2 数据模型	(4)
§ 1.3 关系数据模型	(9)
§ 1.4 关系数据库管理系统	(13)
§ 1.5 DBMS 的评价与选择	(14)
§ 1.6 一个典型的教学管理数据库的设计	(16)
第二章 关系数据库标准语言——SQL	(21)
§ 2.1 概述	(21)
§ 2.2 SQL 数据定义功能	(21)
§ 2.3 SQL 数据查询	(23)
§ 2.4 数据更新功能	(29)
§ 2.5 建立和使用视图	(31)
§ 2.6 索引	(33)
§ 2.7 SQL 数据控制功能	(33)

第二篇 ORACLE 数据库系统及其应用

第三章 ORACLE 概述	(36)
§ 3.1 ORACLE 的主要特点	(36)
§ 3.2 ORACLE 的体系结构	(37)
§ 3.3 MS-DOS 下 ORACLE V5.1 的安装	(39)
§ 3.4 ORACLE V5.1 的运行管理	(42)
第四章 SQL * PLUS 用户指南	(43)
§ 4.1 使用 SQL * PLUS 的预备知识	(43)
§ 4.2 标准 SQL 的扩充	(46)
§ 4.3 SQL * PLUS 命令	(48)
§ 4.4 授权(Grant)机制	(51)
§ 4.5 SQL * PLUS 速查手册(ORACLE V5.1B)	(52)
第五章 表格设计 SQL * FORMS	(67)
§ 5.1 SQL * FORMS 介绍	(67)
§ 5.2 FORM 的快速生成	(72)
§ 5.3 运行 FORM 时的功能键速查手册	(75)
§ 5.4 FORM 的运行	(80)
§ 5.5 设计 FORM 时的功能键速查手册	(85)
§ 5.6 设计 FORM	(87)

§ 5.7	块级设计——生成和定义	(94)
§ 5.8	域级设计——生成和定义	(103)
§ 5.9	屏幕画面设计	(111)
§ 5.10	触发器设计	(115)
第六章	报表软件 SQL * Report	(149)
§ 6.1	SQL * Report 概述	(149)
§ 6.2	报表正文格式化程序 RPF	(150)
§ 6.3	报表生成程序 RPT	(157)
§ 6.4	高级报表打印	(166)
第七章	交互式图形工具 SQL * Graph	(168)
§ 7.1	概述	(168)
§ 7.2	绘制折线图	(169)
§ 7.3	绘制直方图	(171)
§ 7.4	绘制饼形图	(172)
§ 7.5	图形的完善	(173)
第八章	电子表格软件 SQL * Calc	(177)
§ 8.1	SQL * Calc 概述	(177)
§ 8.2	SQL * Calc 功能键速查手册	(179)
§ 8.3	SQL * Calc 启动和屏幕画面	(181)
§ 8.4	表格数据的编辑操作	(183)
§ 8.5	电子表格命令	(188)
§ 8.6	使用 ORACLE 表中数据	(200)
第九章	预编译接口	(205)
§ 9.1	PRO * C 程序的结构	(205)
§ 9.2	数据更新操作	(210)
§ 9.3	数据库查询	(213)
§ 9.4	事务管理	(215)
§ 9.5	错误检测和恢复	(217)
§ 9.6	动态 SQL 语句	(221)
§ 9.7	运行 ORACLE 预编译	(226)
§ 9.8	其它应用	(229)
第十章	ORACLE 的运行与管理	(230)
§ 10.1	启动和停止数据库(IOR)	(230)
§ 10.2	显示系统实用程序(ODS)	(233)
§ 10.3	修改显示终端(CRT)实用程序	(234)
§ 10.4	数据装入程序(ODL)	(237)
§ 10.5	卸出和装入程序(EXPORT/IMPORT)	(241)
§ 10.6	ORACLE 数据库结构	(245)

第三篇 UNIFY 数据库系统及其应用

第十一章	UNIFY 使用初步	(248)
------	------------	-------

§ 11.1 概述.....	(248)
§ 11.2 数据库设计和维护.....	(253)
第十二章 屏幕格式的建立及其使用.....	(267)
§ 12.1 建立和修改屏幕格式.....	(267)
§ 12.2 ENTER—数据录入和格式查询	(272)
第十三章 UNIFY SQL 的应用	(278)
§ 13.1 UNIFY SQL 的扩充	(278)
§ 13.2 注册 SQL 使用的屏幕格式及 SQL 屏幕的使用.....	(279)
§ 13.3 UNIFY SQL 命令参考	(280)
第十四章 UNIFY 的系统管理	(292)
§ 14.1 系统安全维护.....	(292)
§ 14.2 事务登录.....	(297)
§ 14.3 增加、修改或删除菜单	(299)
§ 14.4 将程序注册到系统中.....	(301)
§ 14.5 数据库装载.....	(303)
第十五章 RPT ——报表处理程序	(305)
§ 15.1 报表书写的概念和简单实例.....	(305)
§ 15.2 RPT 的执行	(308)
§ 15.3 格式化 SQL 和 RPT	(309)
§ 15.4 RPT 命令参考	(310)
第十六章 UNIFY 的 C 语言接口	(318)
§ 16.1 如何编译并装入 UNIFY C 程序	(318)
§ 16.2 函数使用实例.....	(322)
§ 16.3 ENTER 裁剪工具及实例	(329)
§ 16.4 C 语言接口参考.....	(336)
第四篇 FoxPro 数据库系统及其应用	
第十七章 FoxPro 简介	(349)
§ 17.1 FoxPro 的发展概况	(349)
§ 17.2 FoxPro 2.5 的特点	(350)
§ 17.3 FoxPro 的安装和启动	(350)
§ 17.4 FoxPro 使用初步	(352)
第十八章 数据库的建立.....	(360)
§ 18.1 数据库基本概念.....	(360)
§ 18.2 创建一个数据库文件.....	(360)
§ 18.3 数据库的字段类型.....	(364)
§ 18.4 修改数据库文件结构.....	(365)
第十九章 数据库信息的维护.....	(369)
§ 19.1 打开和关闭一个数据库文件.....	(369)
§ 19.2 追加新记录.....	(371)
§ 19.3 修改记录.....	(373)

§ 19.4	删除记录	(373)
§ 19.5	浏览数据	(374)
第二十章	排序和索引	(379)
§ 20.1	准备工作	(379)
§ 20.2	排序	(379)
§ 20.3	索引	(381)
第二十一章	数据库查询	(389)
§ 21.1	查询条件	(389)
§ 21.2	使用 LIST、LOCATE 和 SEEK 命令进行查询	(390)
§ 21.3	使用 RQBE 窗口查询	(393)
第二十二章	简单报表的生成	(397)
§ 22.1	建立快速报表	(397)
§ 22.2	编辑快速报表	(402)
§ 22.3	建立格式报表	(403)
§ 22.4	编辑格式报表	(404)
第二十三章	VIEW 窗口及相关数据库	(406)
§ 23.1	VIEW 窗口的组成	(406)
§ 23.2	视图文件	(407)
§ 23.3	相关数据库概念	(408)
§ 23.4	建立和使用关系数据库	(409)
§ 23.5	在 RQBE 中使用相关数据库	(413)
第二十四	复杂报表的生成	(415)
§ 24.1	使用页格式设计对话框	(415)
§ 24.2	设计页标题	(415)
§ 24.3	记录的分组	(416)
§ 24.4	建立计算字段	(417)
§ 24.5	报表字段的格式定义	(419)
§ 24.6	在报表中筛选记录	(421)
第二十五章	屏幕生成器的使用	(424)
§ 25.1	建立快速屏幕	(424)
§ 25.2	使用屏幕设计窗口	(424)
§ 25.3	修改字段属性	(428)
§ 25.4	建立按钮	(428)
§ 25.5	生成屏幕代码	(430)
§ 25.6	运行屏幕	(430)
§ 25.7	屏幕生成器的高级使用技巧	(431)
§ 25.8	用应用程序生成器生成应用程序	(433)
第二十六章	菜单生成器的使用	(435)
§ 26.1	菜单组成元素	(435)
§ 26.2	菜单设计窗口	(435)

§ 26.3	使用菜单设计窗口.....	(437)
§ 26.4	MENU 菜单	(439)
§ 26.5	生成菜单代码.....	(442)
§ 26.6	运行菜单代码程序.....	(442)
第二十七章	应用程序设计.....	(443)
§ 27.1	FoxPro 程序设计语言	(443)
§ 27.2	程序实例.....	(448)
第二十八章	项目管理器.....	(450)
§ 28.1	打开项目管理器.....	(450)
§ 28.2	管理项目.....	(450)
§ 28.3	建立应用程序.....	(451)
第二十九章	应用程序接口.....	(453)
§ 29.1	API 编程规则	(453)
§ 29.2	API 例程.....	(453)
§ 29.3	获取参数.....	(454)
§ 29.4	返回结果.....	(455)
附录 A	FoxPro 命令	(456)
附录 B	FoxPro 函数	(520)

第一篇 数据库概论

第一章 关系数据模型与数据库系统

在电子计算机问世的前 10 年中,主要是用于解决科学研究和工程设计中的数值计算问题。这个时期它的技术与应用只被少数科学家所拥有。随着计算机自身不断的发展,计算机的应用由科学研究部门扩展到各种企业和行政部门。因此,在计算机的三大主要应用领域,即科学计算、数据处理和过程控制中,数据处理迅速上升为计算机应用的主要方面,数据库技术就是作为数据处理中的一门新技术而发展起来的。本章扼要介绍有关数据库技术的发展简史以及数据库的主要概念与技术,特别详细地介绍了 80 年代流行的关系数据库方面的基础知识,从而为初学数据库的读者学习后面几章奠定基础。

§ 1.1 文件技术与数据库技术

在 50 年代到 60 年代之间,计算机不仅用于科学计算,而且已经大量用于数据处理方面,此时计算机外存储器有了磁鼓、磁盘等直接存取的存储设备,为计算机进行事务管理奠定了硬件基础。另外软件技术主要是数据结构设计和数据管理技术的研究也得到了迅速的发展,从而出现了专门的管理数据的软件,这就是所谓的“文件技术”阶段。

在文件技术阶段,数据管理的主要特点是:外存储器成为计算机系统不可缺少的组成部分,用户可以随时通过程序对文件进行查询、修改和增删等处理;文件组织形式日益多样化,索引文件、直接存取文件、链接文件纷纷出现,既可以满足批处理应用的需要,又能有效地实现记录的随机存取;出现了专用数据管理系统的软件,即文件系统,它能对驻留在外存储器上的数据文件实施统一管理,这种专用数据管理软件构成了操作系统的一个重要组成部分,由于应用程序不再需要了解数据在存储介质上的实际地址,因而大大减少了程序设计的工作量;数据不再仅仅属于某个特定的程序,而可以重复使用。由于文件结构的设计仍然是基于特定的用途,程序仍然是基于文件的特定物理结构和存取方法编制的,因此,数据结构与程序之间的依赖关系并未根本改变。

文件技术在 20 世纪 60 年代中得到了充分的发展,把计算机应用推向一个新的高潮。60 年代中期以后,计算机在数据处理领域的应用迅速发展,由个别部门的应用逐步发展成多个部门的普遍应用,由简单孤立的单项应用发展为彼此相关的复杂应用,从而管理的规模更加庞大,数据量急剧增长,共享性也更强了,这就带来了数据管理上的一些新问题,我们举一例说明(例 1)。

例如,某学校学生处、卫生院、教务处均要处理学生的有关信息,但各自处理的内容不同,如用文件系统实现,可按如下方式组织。

学生处要处理的信息包括:

学号、姓名、系名、年级、专业、年龄、性别、籍贯、政治面目、家庭住址、个人履历、社会关系……等。为此,学生处的应用程序员必须定义一个文件 F1,其记录包括上述几个数据项。

卫生院要处理的信息是:

学号、姓名、系名、年级、专业、年龄、性别、身高、体重、健康状况……等。显然,卫生院必须定义文件 F2,其每一记录包括上述数据项。

类似地,当教务处也要进行数据处理时,其相应的文件 F3 则可能包括下列数据项:

学号、姓名、系名、年级、专业、课名、成绩、学分……等。

这样当上述三个部门共用该计算机时,外存中要同时保存 F1、F2、F3 三种文件,其中重复的数据项竟达到 1/3 以上,严重冗余。

数据冗余不仅浪费存储空间,更严重的是带来潜在的不一致性。由于数据存在多个副本,当发生数据更新时,就很可能发生某些副本被修改而另一些副本被遗漏的情况,从而使数据发生不一致,影响数据的正确性和可靠性。由于数据的使用价值在很大程度上依赖其可靠性,所以,这种不一致的后果是不可忽视的。

另一方面,在传统文件技术阶段,文件是为某一特定应用服务的,应用程序和文件是一一对一的,这就造成了应用程序与数据结构过分地互相依赖,而且系统很难扩充。一旦数据的逻辑结构改变,必须修改应用程序和文件结构的定义,反之,应用程序的改变,也将影响文件的数据结构的改变。因此,用户迫切希望应用程序能够不受数据结构变化的影响。

此外,文件系统缺乏对数据操作进行控制的方法,并且对于数据的安全性、保密性、正确性等方面的控制,完全要用户程序自己负责,这使得应用程序的编制相当繁琐。

综上所述,传统的文件技术有许多缺点,不能满足人们的要求,因此迫切需要新的数据管理技术来实现对数据的共享,实现数据与程序的独立性,并提供安全性和完整性,也就是说在操作系统上必须有一个软件系统——数据库管理系统 DBMS(Data Base Management System),在建立、运用和维护时对数据库进行统一控制,这就是数据库技术。

60 年代后期的三大事件标志着数据管理技术进入了数据库阶段,它们是:

1)1969 年美国 IBM 公司的数据库管理系统 IMS(Information Management System)问世。IMS 的数据模型是层次结构的。

2)同年美国 CODASYL(Conference On Data System Language)委员会公布其研究成果 DBTG 报告,该报告确定并建立了数据库系统的许多概念、方法和技术。DBTG 所提议的方法是基于网状结构的。

3)1970 年 IBM 公司 San Jose 研究实验室的研究员 E. F. Codd 发表了题为“大型共享数据库数据的关系模型”论文,提出了数据库的关系模型,奠定了关系数据库的理论基础。

此外,1978 年美国计算机与信息处理国家标准委员会规划与需求委员会 DBMS 研究组的研究成果 SPARC 报告的发表,是数据发展史上的又一重要事件,它标志着数据库技术进入了成熟的阶段。

与文件系统相比,数据库技术提供了对数据的更高级更有效的管理,它有如下主要特点:

1)数据共享性。数据共享的意义是多种应用、多种语言互相覆盖地共享数据集合。在传统的文件组织中,每个数据文件是特定的应用所私有的,如图 1-1。而数据库是从整体观点来看待和描述数据的,数据不再是面向某一应用,而是面向整个系统,如图 1-2。这可以大大减少数据的冗余度,既节约存储空间,减少存取时间,又可避免数据之间的不相容性和不一致性。前面所举文件系统例 1,如从数据库角度进行组织,如图 1-3 所示。

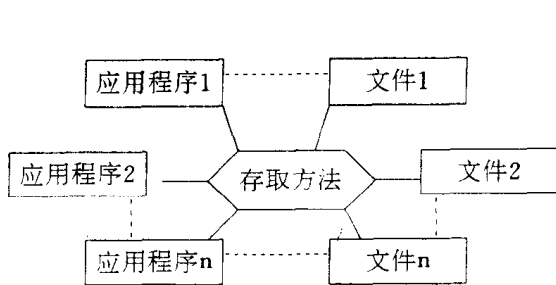


图 1—1

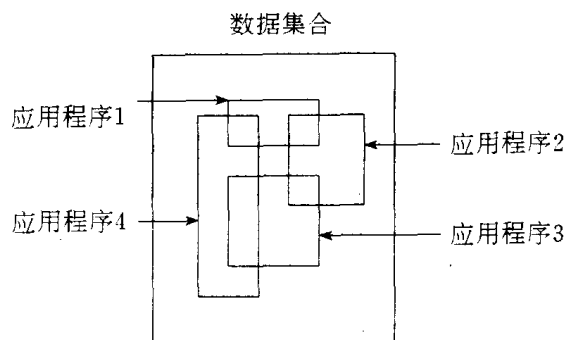


图 1—2

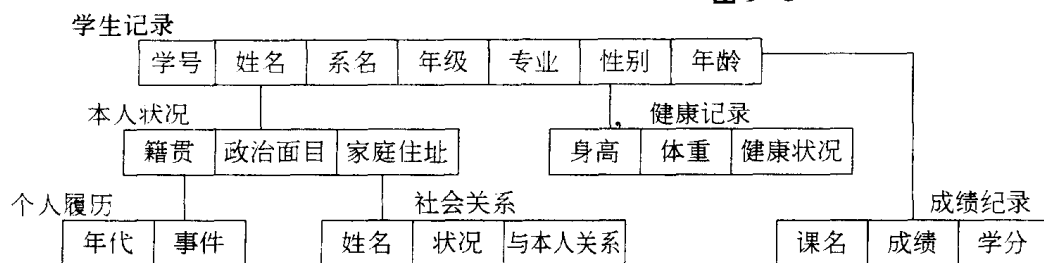


图 1—3

由该结构看出数据不再分属各个应用程序,而是集中存储在一起。对一个特定的组织而言,除了由于需要保密强加的某些限制以外,这些数据是为该组织的各个下属部门和各个方面的应用所共享的,这样就大大提高了数据的使用价值。另外,当应用需求改变或增加时,只要重新选取不同子集或者加上一小部分数据,便可以有更多的用途,满足新的要求,使得系统很容易扩充。它的这个特点是源于数据库中数据项之间及记录之间是有相互联系的,它是高度结构化的,而文件系统很难达到这一点。

2) 具有较高的数据和程序的独立性。在使用数据库时,应用程序对存储结构有较高的独立性。这种独立性是由系统在存储结构和逻辑结构之间提供的映象来获得的,这样当存储结构或者说物理结构改变时,只要相应地改变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,从而使建立在逻辑结构之上的应用程序也保持不变,这称为物理独立性。另一方面,一个数据库系统所拥有的数据比某个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改变时,局部逻辑结构可以保持不变,而程序员根据局部逻辑结构编写的程序也可以不变,这就是所谓的逻辑独立性。提高数据独立性是数据库所追求的一个主要目标。数据库中数据与应用程序的关系如图 1—4 所示。

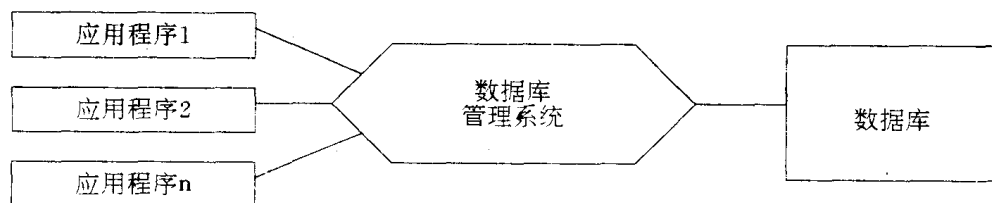


图 1—4

3)对数据实行集中统一的控制。系统提供统一的数据定义、增删、检索以及更新操作的手段并且统一控制数据的安全性、完整性和保密性。由于数据库中的信息非常重要,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。

另外,在数据库中由于是共享机制,如遇多用户同时使用数据库情况,这种并发的存取动作如果不加以控制势必造成严重的后果,数据库的这种功能一般称作数据的操作一致性。另一方面,数据库中的数据是对客观世界中某些实体性质的反映,它有一定语义含义,如性别的值只可能是男或女两个值,会计记帐时,收支应当平衡,也就是在数据库中能体现收入之和减支出之和等于剩余数,这就是说,无论何时,数据库中的数据应当满足这种正确性要求,这称作数据语义一致性。

由于数据库系统具有上述特征,它的出现使信息系统的研制从围绕加工数据的程序为中心转变到围绕共享的数据库来进行,我们可以给数据库下如下一个定义:数据库是以一定的组织方式存储在计算机中相互有关数据的集合,它能以最佳方式、最少的重复、最大的独立性为多种应用服务。鉴于上述特点,大型复杂的信息系统大多以数据库为核心,数据库系统在计算机应用中起着越来越重要的作用。

§ 1.2 数据模型

数据库中存储的是数据,这些数据反映了现实世界中有意义、有价值的信息,它不仅反映数据本身的内容,而且反映数据之间的联系。那么如何抽象表示、处理现实世界中的数据和信息呢?这就需要使用数据模型这个工具,数据模型是数据库中用于提供信息表示和操作手段的形式框架,它是我们将现实世界转换为数据世界的桥梁。

目前广为使用的模型大致可分为两种:一种称为概念模型(信息模型),它的建立独立于任何计算机系统,如实体——联系模型,语义网络模型等。它的特点是完全不涉及信息在计算机中的表示问题,只描述组织中特定的有意义的信息结构,因而这类模型概念简单、清晰,易于用户理解,它的语义表达能力很强,能够方便、直接地表达应用中的各种语义知识。另一种称为基本数据模型,它是按计算机系统的观点对数据建模,如网状模型、层次模型、关系模型等。它的特点是与具体计算机系统密切相关,它直接面向数据库中数据的逻辑结构,它通常有一组严格定义了语法和语义的语言,人们可以使用它来定义、操纵数据库中的数据,它是现实世界中的信息最终在机器世界得到的反映。

这两种模型可看成两个过程。一般在数据库设计时,先调研某个企业、组织或部门的情况,为其建立概念模型,其次再将概念模型转换为基本数据模型,最终在计算机上得以实现。本节简略介绍概念模型中的实体——联系模型以及基本数据模型中的网状、层次模型,下一节再重点介绍目前最流行的关系模型。

1. **实体——联系模型** 实体——联系模型,即 E—R 模型(Entity—Relationship model),它是将现实世界中的客观对象抽象为某一种信息结构,这种结构并不依赖于计算机系统,它的基本语义单位是实体与联系,下面我们介绍其主要概念。

实体:客观存在并可相互区分的事物。从具体的人、物、事件到抽象的状态与概念都可以用实体抽象地表示,例如,在学校里,一个学生,一个老师,一门课程都是实体。实体不仅可指事物本身,也可指事物之间的具体联系,如,一个学生选了一门课程,某个老师订了一份报纸等,都可称为实体。一些实体的集合称为实体集,每个实体集都有一个名以便与其它实体集相区别,

例如,学生、课程、教师订报纸等都是实体集。

属性:实体所具有的某一特性。一个实体可由若干属性来刻画。例如,学生实体可由学号、姓名、年龄、性别、年级、系、专业等属性组成(7809, 王平, 19, 女, 92级, 计算机系, 软件)。这些属性组合起来表示了王平这个学生。另外,同类实体就是具有相同属性的实体,它们具有共同的特征和性质,一般称为实体型,如学生(学号,姓名,年龄,性别,年级,系,专业)就是一个实体型。

联系:是事物内部或事物之间语义关系的抽象表示。由于现实世界的事物之间是有联系的,那么在信息世界中也要反映出这种联系来,在实体——联系模型中将联系分为三类:

1) 一对一联系(1:1) 若有两个实体集 A 和 B,如果实体集 A 中至多有一个实体与 B 中的一个实体有联系,反之 B 中也至多有一个实体与 A 中的一个实体有联系,则称它们之间是一对一联系,记作 1:1。如,看电影时,观众与座位之间就是一对一的联系,因为一个人只能坐一个座位,一个座位只能由一个人来坐。

2) 一对多联系(1:n) 若有两个实体集 A 和 B,如果 A 中的一个实体与 B 中若干个实体相联系,但 B 中每个实体只与 A 中一个实体相联系,则称 A 与 B 之间为一对多联系,记作 1:n。例如,班级与学生之间是一对多联系,因为一个班可有若干个学生,但一个学生只能属于一个班。

3) 多对多联系 对于两个实体集 A 和 B,若 A 中一个实体与 B 中若干个实体相关联,B 中每一个实体也与 A 中多个实体相关联,则称 A 与 B 之间为多对多联系,记作 m:n。例如,学生与课程之间为多对多联系,因为一个学生可选多门课程,一门课程可由多个学生选。

E-R 模型可用图形形象加以表示,最常用的是 E-R 图,它是 P. P. S Chen 于 1976 年提出的。它的主要思想是以矩形框表示实体型,菱形框表示联系,椭圆形框表示属性,相应的实体名、联系名、属性名均写入框中,然后用无向线段将上述框联起来,在联系实体型的线段上标上联系的类型(1:n, 1:1, 或 m:n),上面几个例子可用图 1-5 表示。

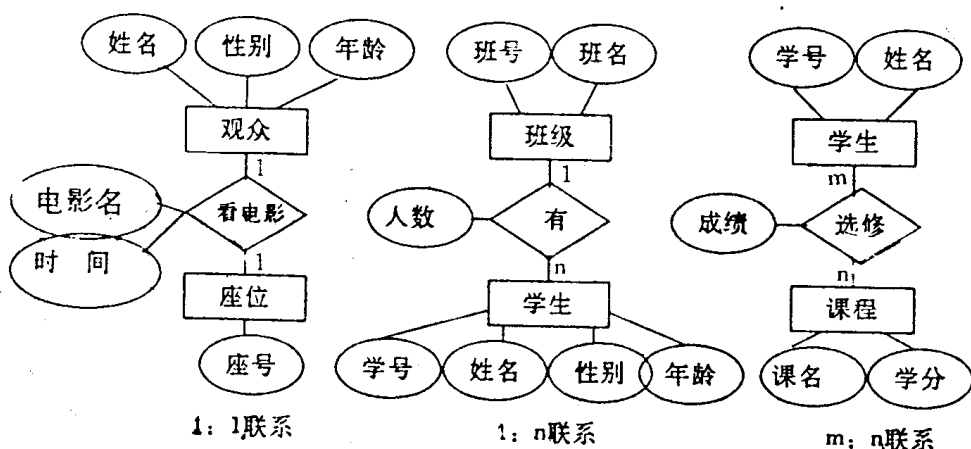


图 1-5

E-R 图是抽象描述现实世界的有力工具,它与计算机所支持的数据模型相独立,它更接近于现实世界。虽然现实世界丰富多彩,各种信息十分繁杂,但用 E-R 图可以很清晰地表示出其中的错综复杂关系,下面我们给出一个简单教务管理的 E-R 图(图 1-6)。

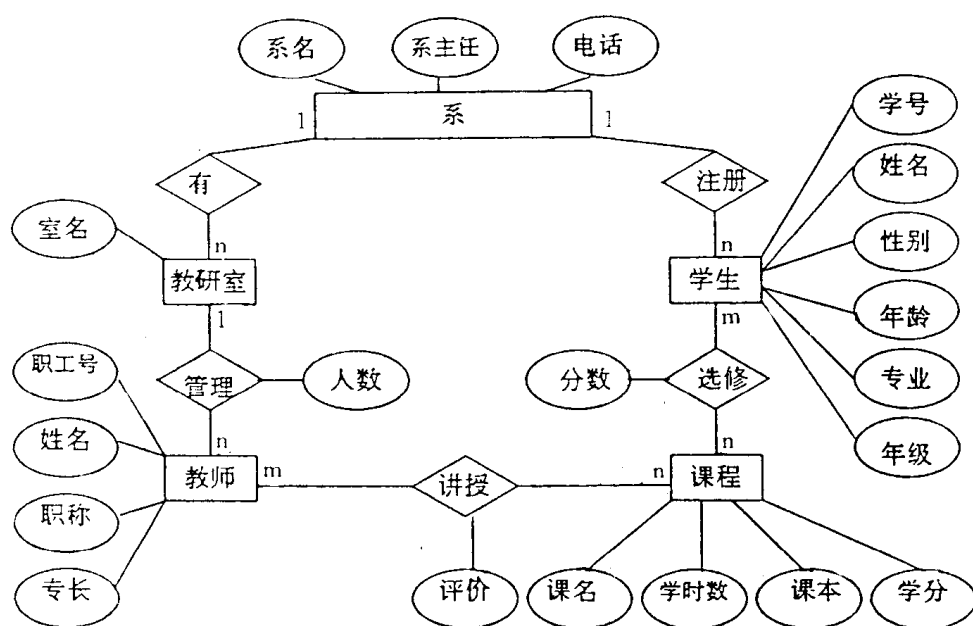


图1—6 教务管理E—R图

2. 数据模型 数据模型是对客观事物及其联系的数据描述,它反映了实体内部和实体之间的联系。数据模型设计的好坏直接影响数据库的性能,一个基本数据模型实际上给出了一个通用的在计算机系统上可实现的描述和动态地模拟现实世界信息结构及其变化的抽象方法。为了在计算机系统上实现这种方法,人们开发了相应的支持软件,这就是数据库管理系统(Data Base Management System,简称DBMS)。显然,DBMS 是因其支持的数据模型而异。

目前在数据库技术中使用最广泛的基本数据模型有三种:

层次模型(Hierachical Model);

网状模型(Network Model);

关系模型(Relational Model)。

这里简略介绍前两种,关系模型放入下一节详细说明。

1) 层次模型 层次模型是较早用于数据库技术的一种数据模型,使用该模型的典型系统是 IBM 公司的 IMS。层次模型的典型数据结构是以记录类型为结点的有向树,它把整个数据库的结构表示成为一个有序树的集合,而这些有序树的每一个结点是一个由若干数据项组成的逻辑记录型。

图 1—7 给出了一个采用层次模型的数据库模式的例子。它表示的是一个大工业公司的内部教育系统的信息。这个公司设有一个教育部门,其作用是开设一些对本公司的雇员进行培训的课程,每门课都在其组织内不同的地点开设。

显然该数据库有五个记录类型:课程,先行课,设置,教员和学生。课程称为根记录类型,它是先行课和设置的父记录类型,而先行课和设置称为课程记录型的子女记录类型。在层次模型中只有一个结点无双亲,这个结点即为根结点(如课程),其它结点是依据根才存在的,它们有且仅有一个双亲。同一双亲的结点称为兄弟结点,无子女的结点为树叶。在整个层次模型中,

父结点与子结点的联系都是一对多的联系。图 1-8 显示了图 1-7 结构的一个具体值。

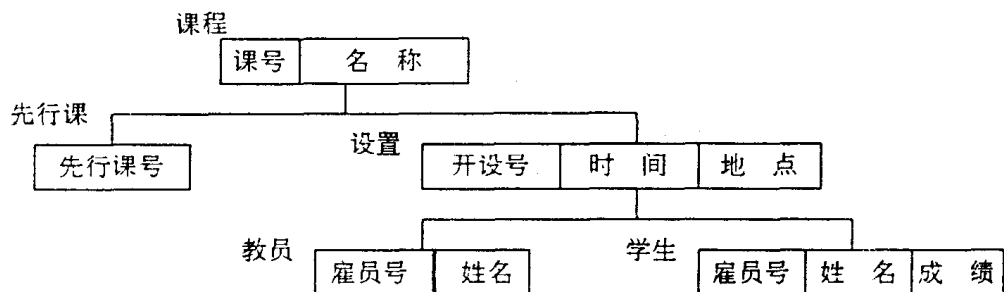


图 1-7 教育数据库的层次结构

在现实世界中,具有层次型联系的事物是很多的,如组织机构,各种生物或非生物的分类归属,需要区分主目和细目的文件等等,用层次模型来描述它们非常自然、直观。结构简单、层次清晰,易于理解,这些是层次模型突出的优点。但另一方面,层次模型很难表示出那些非层次型的系统,即使表示出来也非常不自然,另外层次模型的数据库不仅在逻辑上而且在物理上都是按有序树结构来组织的,这使得数据库模式适应变化的能力很差,如要对数据库模式稍加修改,就要整个地对数据库进行重新组织,这无疑大幅度地增加了系统的开销,层次模型的这一缺陷使它很难适应许多信息需求动态变化的应用环境。

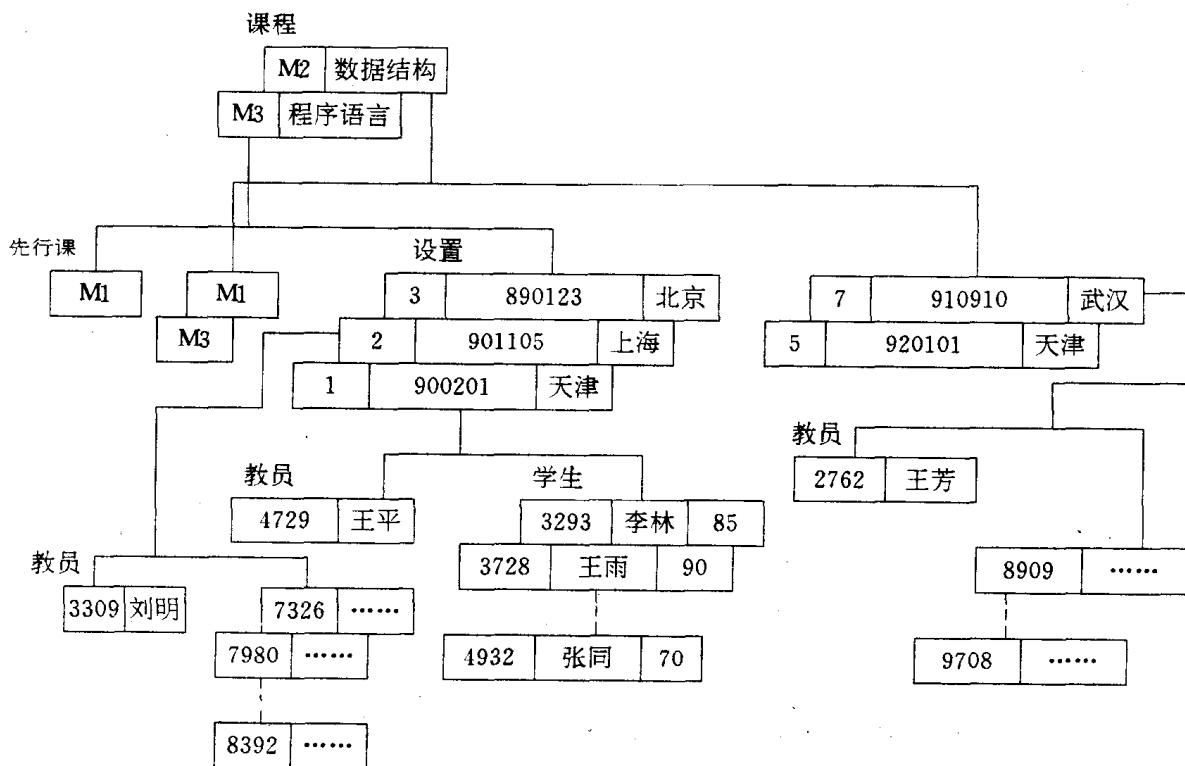


图1-8 教育数据库的树形实例