

普通高等工科教育规划教材



数据库原理及应用

范剑波 主编
方飞 副主编

1.13

机械工业出版社
China Machine Press

普通高等工科教育规划教材

数据库原理及应用

主 编 范剑波
副主编 方 飞
参 编 范海绍
主 审 张 森



机械工业出版社

本书是在多年教学改革和应用实践的基础上，并通过作者两轮讲义使用后进行改写而成的。本书内容包括：数据库技术概论、关系数据库、结构化查询语言、关系数据库设计理论、数据库查询的实现与优化、数据库保护、数据库设计、数据库应用系统开发与维护和客户/服务器应用系统。

本书为普通高等工科教育规划教材，适合于计算机专业及相关专业本科作为教材使用，也可供高专、高职计算机专业学生选用，并可供计算机工程技术人员参考。

图书在版编目 (CIP) 数据

数据库原理及应用/范剑波主编. —北京：机械工业出版社，2001.4

普通高等工科教育规划教材

ISBN 7-111-08757-7

I. 数… II. 范… 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 (2001) 第 07873 号

机械工业出版社 (北京市百万庄大街 22 号 邮政编码 100037)

责任编辑：贡克勤 版式设计：霍永明 责任校对：张莉娟

封面设计：方 芬 责任印制：郭景龙

北京京丰印刷厂印刷·新华书店北京发行所发行

2001 年 4 月第 1 版·第 1 次印刷

787mm×1092mm^{1/16}·9.75 印张·236 千字

0 001—5000 册

定价：15.00 元

凡购本书，如有缺页、倒页、脱页，由本社发行部调换
本社购书热线电话 (010) 68993821、68326677-2527

前 言

我们通过十多年数据库课程教学与科研工作的实践，又经过近五年国家教育部计算机重点试点专业教学改革探索，形成了计算机应用专业《数据库原理及应用》课程的教学模式：由实践（VFP 程序设计）→理论（数据库原理及应用）→再实践（数据库课程设计）→再提高（毕业设计、科研），取得了明显的效果，学生的数据库设计能力及实际编程动手能力均有了较大的提高。本书就是在此基础上，并通过作者两轮讲义使用后进行改写而成的。本书内容包括：数据库技术概论、关系数据库、结构化查询语言、关系数据库设计理论、数据库查询的实现与优化、数据库保护、数据库设计、数据库应用系统开发与维护和客户/服务器应用系统。教学时数为 60~70 学时。

本书的特点是集理论、应用和系统于一体。理论的目标是最终能进行关系数据库的设计；应用的目标是能按软件工程的规范和数据库设计的步骤来进行数据库信息管理系统的开发；系统的目标是通过使用 SQL Server 7.0 和客户/服务器计算模式来进行网络数据库的开发和应用。本书将 SQL Server 7.0 数据库管理系统的内容结合到数据库原理中去讲，避免了部分内容的重复，也避免了在原理部分讲解时无实际系统作为参照时的空对空现象。此外，本书增加了数据库应用系统的开发和维护内容，使学生能运用关系数据库的设计理论和软件工程的规范来开发一个实际的数据库应用系统；本书还增加了客户/服务器应用系统的内容，使学生能进行网络数据库方面的开发和应用。

本书每章末附有小结、思考题和习题，供学生平时练习之用。最后附有 SQL Server 7.0 的实验和数据库课程设计的具体要求和题目，供学生实践环节使用。书中打 * 的为要求较高的教学内容，供教学选用。

本书为普通高等工科教育规划教材，适合于计算机专业及相关专业本科作为教材使用，也可供高专、高职计算机专业学生选用，并可供计算机工程技术人员参考。

本书第 1、2、3、4、5、7、9 章和附录 B 的内容由浙江大学宁波分校范剑波编写；第 8 章和附录 A 的内容由宁波职业技术学院方飞编写；第 6 章由宁波工商职业技术学院范海绍编写。全书由范剑波统稿并任主编，方飞任副主编。

本书由浙江大学计算机学院张森教授担任主审，张教授对教材的编写提纲和内容提出了很好的意见和建议，上海理工大学孔凡才教授对本教材的编写和出版给予了很大的帮助，在此一并表示衷心的感谢。此外，还要感谢我系郭建康老师在数据库课程建设中所作的贡献。最后感谢我系张定龙主任、程刚副主任的支持和帮助。限于编者水平，书中难免存在错误和不妥之处，殷切期望广大读者给予指正。[电子邮件地址：bfan@263.net，来信寄：宁波市后河巷 20 号（315016），宁波高等专科学校电子技术系]

范剑波

2001 年 1 月 12 日于宁波

目 录

前言

第 1 章 数据库技术概论	1	2.2 关系代数	32
1.1 文件技术与数据库技术	1	2.2.1 传统的集合运算	32
1.1.1 文件技术的特点与 局限性	1	2.2.2 专门的关系运算	33
1.1.2 数据库技术的产生与 发展	3	2.2.3 关系代数表达式	34
1.1.3 数据库技术的特点	4	小结	36
1.1.4 数据库基本概念	6	思考题	36
1.2 数据模型	7	习题	37
1.2.1 概念模型	7	第 3 章 关系数据库结构化 查询语言	38
1.2.2 数据模型三要素	10	3.1 SQL 概述	38
1.2.3 层次模型*	11	3.1.1 标准 SQL 的特点	38
1.2.4 网状模型*	15	3.1.2 关系数据库的三级 模式结构	39
1.3 数据库系统结构	18	3.1.3 SQL Server 中的 T-SQL 对 标准 SQL 的扩充	39
1.3.1 数据库系统的三级 模式结构	18	3.2 T-SQL 的数据定义	40
1.3.2 数据库系统组成	20	3.2.1 SQL Server 中的 数据库	40
1.4 数据库管理系统	22	3.2.2 SQL Server 中的数据库 表和索引	41
1.4.1 DBMS 基本功能	22	3.3 T-SQL 数据查询功能	43
1.4.2 DBMS 的组成	23	3.3.1 SELECT 语句	44
1.4.3 DBMS 的评价与 选择	24	3.3.2 附加子句	46
小结	26	3.3.3 复杂查询	47
思考题	26	3.4 T-SQL 数据更新功能	49
习题	27	3.4.1 数据插入、修改和 删除语句	49
第 2 章 关系数据库	28	3.4.2 批处理	51
2.1 关系模型的基本概念	28	3.4.3 流程控制语言	52
2.1.1 关系的通俗解释	28	3.5 T-SQL 的视图、存储过程和 触发器	54
2.1.2 关系的数学定义	29	3.5.1 T-SQL 的视图	54
2.1.3 关系模型	30	3.5.2 T-SQL 的存储过程	57
2.1.4 关系数据库管理 系统	31		

3.5.3 T-SQL 的触发器 *	58	6.1.2 SQL Server 系统的 安全性	83
3.6 T-SQL 数据控制	61	6.2 数据完整性	85
3.6.1 语句权力的授予与 收回	61	6.2.1 完整性约束条件	85
3.6.2 对象权力的授予与 收回	62	6.2.2 SQL Server 系统的 完整性	85
小结	62	6.3 事务处理	88
思考题	63	6.3.1 事务的基本概念	88
习题	63	6.3.2 并发操作与数据的 不一致性	89
第 4 章 关系数据库设计理论	64	6.3.3 SQL Server 系统的并发 控制	90
4.1 关系模式的存储异常与 数据依赖	64	6.3.4 SQL Server 系统的恢复 策略	91
4.2 函数依赖的基本概念	65	小结	92
4.3 关系的规范化	67	思考题	92
4.3.1 第一范式	67	习题	92
4.3.2 第二范式	68	第 7 章 数据库设计	93
4.3.3 第三范式	69	7.1 数据库设计概述	93
4.3.4 BCNF 范式	69	7.1.1 数据库和信息系统	93
4.3.5 多值依赖 *	70	7.1.2 数据库设计的特点	93
4.3.6 第四范式 *	72	7.1.3 数据库设计方法 简述	94
4.4 模式分解的优劣	73	7.1.4 数据库设计步骤	94
小结	74	7.2 需求分析	95
思考题	74	7.2.1 需求分析的任务和 方法	95
习题	74	7.2.2 数据字典和数据 流图	96
第 5 章 数据库查询的实现 与优化	76	7.3 概念结构设计	97
5.1 概述	76	7.3.1 概念结构	97
5.2 查询优化的一般策略	78	7.3.2 概念结构设计的方法与 步骤	98
5.2.1 逻辑层优化的一般 策略	78	7.3.3 数据抽象与局部视图 设计	98
5.2.2 物理层优化的一般 方法	79	7.3.4 视图的集成	102
小结	80	7.4 逻辑结构设计	105
思考题	80	7.4.1 E-R 图向关系模型的 转换	106
习题	80	7.4.2 数据模型的优化	107
第 6 章 数据库保护	81		
6.1 安全性	81		
6.1.1 安全性控制的一般 方法	81		

7.5 数据库的物理设计	108	9.1.1 客户/服务器系统的 优点	124
7.5.1 数据库物理设计的要求 与内容	108	9.1.2 客户/服务器系统的基本 组成	125
7.5.2 评价物理结构	109	9.1.3 Client/Server 的几个 概念	126
7.6 数据库的实施和维护	109	9.1.4 开放式数据库 连接 ODBC	126
7.6.1 数据的载人和应用程序的 调试	109	9.2 VFP6.0 对客户/服务器系统结构 的支持	127
7.6.2 数据库的试运行	110	9.2.1 客户/服务器系统的 设计	127
7.6.3 数据库的运行和 维护	110	9.2.2 远程视图直接访问 SQL Server	130
小结	111	9.2.3 使用 SQL pass-through 技术 实现应用系统	132
思考题	111	小结	139
习题	112	思考题	140
第 8 章 数据库应用系统开发与 维护	113	习题	140
8.1 应用系统开发模型	113	附录	141
8.1.1 瀑布模型	113	附录 A SQL Server7.0 实验 指南	141
8.1.2 原型模型	115	实验一 使用 SQL Server7.0	141
8.2 数据库应用系统开发实例	116	实验二 T-SQL 的数据查询	141
8.2.1 用户需求分析	116	实验三 数据检索——高级主题	142
8.2.2 数据结构设计	118	实验四 定义基本表、建立索引和 修改数据	142
8.2.3 软件结构设计	119	实验五 设计数据完整性	142
8.2.4 程序流程设计	119	实验六 创建视图和存储过程	143
8.2.5 运行设计	120	实验七 综合练习实验	143
8.3 在 SQL Server7.0 中创建 数据库	120	附录 B 数据库课程设计大纲	144
小结	122	参考文献	147
思考题	123		
习题	123		
第 9 章 客户/服务器应用 系统	124		
9.1 客户/服务器系统	124		

第 1 章 数据库技术概论

在电子计算机刚刚问世的前 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。随着计算机的不断发展,计算机的应用已经从科学研究部门扩展到了各企业和行政部门。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的一门新技术而发展起来的。

数据处理是指对各种形式的数据进行收集、储存、加工和传播的一系列活动的总和。其目的之一是从大量的、原始的数据中抽取、推导出对人们有价值的信息,以作为行动和决策的依据;目的之二是为了借助计算机科学地保存和管理复杂的大量的数据,以便人们能方便而充分地利用这些宝贵的信息资源。

数据库技术所研究的问题就是如何科学地组织和储存数据,如何高效地获取和处理数据。数据库技术作为数据管理的最新技术,目前已广泛应用于各个领域,数据库系统已成为当今计算机系统的重要组成部分。

本章主要介绍数据库的基本概念与技术,它们包括:数据模型、数据库体系结构和数据库管理系统等。这些内容是学习后面几章的基础。

1.1 文件技术与数据库技术

在 50 年代到 60 年代期间,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;与此同时,数据结构设计和数据管理技术研究的软件技术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。

1.1.1 文件技术的特点与局限性

在文件技术阶段,数据管理的主要特点可概括为如下几点:

1) 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。

2) 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。

3) 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。

4) 数据不再仅仅属于某个特定的程序,而可以重复使用。

文件技术的上述特点使得这项技术在 20 世纪 60 年代得到了充分的发展,把计算机应用推向了一个新的高潮。但由于文件结构的设计仍然是基于特定用途的,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,因

而限制了它的进一步发展。

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,从而使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。现举一例加以说明。

某学校的学生处、教务处和卫生院均要使用计算机对学生的有关信息进行管理,但其各自处理的内容不同,如用文件系统实现,可按如下方式进行组织:

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,……。为此,学生处的应用程序员必须定义一个文件F1,该文件结构中的记录应包括上述几个数据项。

教务处要处理的信息包括:学号,姓名,系名,年级,专业,课名,成绩,学分,……。显然,教务处的应用程序员需定义一个文件F2,该文件结构中的每一记录包括以上几个数据项。

类似地,当卫生院要记录和处理学生的有关健康情况信息时,其创建的文件F3应包括下列数据项:学号,姓名,系名,年级,专业,年龄,性别,身高,体重,健康状况,……。

这样当上述三个部门共用某台计算机时,该计算机的外存中要同时保存F1,F2,F3三种文件,可这三种文件中均有学生的学号、姓名、系名、年级和专业等信息,因此重复的数据项达到了1/3以上,数据冗余将会产生以下问题:

1) 数据冗余不仅浪费存储空间,更严重的是带来潜在的 inconsistency。由于数据存在多个副本,所以当发生数据更新时,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。比如,某学生因故需从数学系转到计算机系,当学生处得到该信息后,将该生所属的系名改为计算机系,因而F1文件中保存了正确的信息。但若教务处和卫生院没有得到此信息,或者没有及时更改F2和F3文件,这就造成了数据的不一致性。由于数据的使用价值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。可以想像,当这种情况发生在军事、航天,金融等行业部门时,其后果是非常严重的。

2) 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对的(见图1-1),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。一旦逻辑结构改变,就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,也会影响文件的数据结构的改变。比如,学生处在管理学生信息时发现,只登记学生年龄

是不够的,还应记录下学生的出生年月,因此需在F1的文件中加一个数据项,这就必须修改F1的文件结构,同时还要修改涉及F1文件的应用程序;另一方面,若用户原先是用Pascal语言编制的访问F1文件的应用程序,现想改用C语言,那么,F1文件的数据结构也必须随之改变。由此可看出在文件技术阶段,扩充已有的系统是很困难的,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。

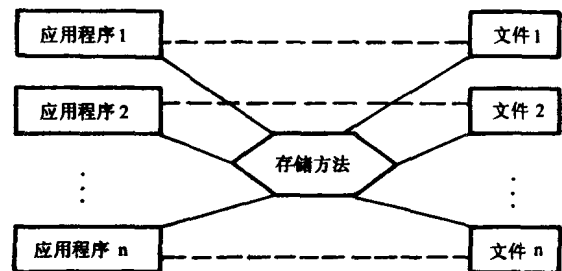


图 1-1 应用程序与文件之间一对一关系

3) 文件系统缺乏对数据操作进行控制的方法, 对于数据的安全性, 保密性和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, 在操作系统之上必须有一个软件系统——数据库管理系统 DBMS(Data Base Management System), 在数据库的建立、运用和维护时, 对数据库进行统一的控制, 这就是数据库技术。

1.1.2 数据库技术的产生与发展

数据库技术从 20 世纪 60 年代中期产生到今天仅仅 30 多年的历史, 但其发展速度很快、使用范围很广。60 年代末出现了第一代数据库——网状数据库和层次数据库, 70 年代出现了第二代数据库——关系数据库。目前关系数据库已逐渐淘汰了网状数据库和层次数据库, 成为当今最为流行的商用数据库系统。而 80 年代出现的以面向对象为主要特征的数据库系统又在向关系数据库系统提出挑战。以下几个事件标志着数据库技术日益成熟的过程。它们是:

1) 1969 年 IBM 公司研制、开发了 DBMS 的商品化软件 IMS (Information Management System), 它是层次模型数据库系统的典型代表。

2) 美国数据系统语言协商会 (The Conference On Data Systems Languages) 下属的数据库任务组 DBTG (Data Base Task Group) 对数据库方法进行了系统的研究、讨论, 于 60 年代末、70 年代初提出了若干报告, 称为 DBTG 报告。DBTG 报告确定并建立了数据库系统的许多概念、方法和技术, 它是网状模型数据库系统的典型代表。

3) 1970 年 IBM 公司 San Jose 研究室的 E. F. Codd 在美国计算机学会会刊 “Communication of the ACM” 上发表了题为 “大型共享数据库数据的关系模型” (“A Relational Model of Data for Shared Data Banks”) 的著名论文, 提出了数据库的关系模型, 开创了数据库关系方法和关系数据理论的研究, 为关系数据库技术奠定了理论基础。E. F. Codd 本人作为关系数据库的创始人和奠基人, 获得了 1981 年 ACM 图灵奖。

自 E. F. Codd 提出关系模型数据库方法以后, 1974 年 IBM 公司就提出了一种基于关系方法实现对数据库存取的 SQL (Structured Query language) 语言, 并经大约 5 年的努力, 成功地研制了一个实现 SQL 语言的关系数据库系统原型 System R, 该系统在 IBM 370 上运行。之后, IBM 又将 SQL 语言引入到 DB2 (IBM DataBase2) 中, 配置在 MVS 上运行, 并于 1983 年推出 DB2 产品, 表明了关系模型数据库系统终于被人们所接受, 并得到最终的肯定。目前, 有影响的关系模型数据库系统有许多, 值得一提的 ORACLE 系统采用 SQL 语言, 用 C 语言开发, 使 ORACLE 不仅使用方便、速度快且移植性好。1979 年开始推出 ORACLE 第一版, 它是由核心和一组支持软件组成的软件包。现在已推出第八版, 增加了与各种语言、决策支持工具以及网络通信软件的接口, 形成了开放式体系结构, 是当前唯一可以通用于大、中、小和微型计算机的产品。关系数据库系统的出现, 促进了数据库的小型化和普及化, 使得在微型机上配置数据库系统成为可能。较流行的 ORACLE、UNIFY、SQL、Server 等较完善的关系数据库系统也都有其微机版本。

4) 从 80 年代以来, 数据库技术在商业领域的巨大成功刺激了其他领域对数据库技术需求的迅速增长。这些新的领域为数据库应用开辟了新的天地。另一方面在应用中提出的一些新的数据管理的需求也直接推动了数据库技术的研究与发展, 尤其是面向对象数据库系统

(Object Oriented DataBase System) 的研究与发展。1990 年美国高级 DBMS 功能委员会发表了“第三代数据库系统宣言”的文章，提出了第三代 DBMS 应具有的三个基本特征如下：

- ① 第三代数据库系统应支持数据管理、对象管理和知识管理。
- ② 第三代数据库系统必须保持或继承第二代数据库系统的技术。
- ③ 第三代数据库系统必须对其他系统开放。

数据库技术与网络通信技术、人工智能技术、面向对象程序设计技术、并行计算技术等互相渗透、互相结合，成为当前数据库技术发展的主要特征。

我国数据库技术的发展大致始于 70 年代中、后期，其标志是 1977 年的第一次数据库技术研讨会。自 1982 年起，几乎每年由该学会下设的数据库学组召开一次全国性的学术交流会，会议起到了交流我国数据库应用与有关科研成果、指导我国数据库学术发展的作用。经过十多年的努力，我国数据库技术水平已从初期的学习、理解、试探阶段发展到消化、改造和创新阶段。

1.1.3 数据库技术的特点

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、多种语言互相覆盖的共享数据集合。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 所示。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 所示。这可以大大减少数据冗余度，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。前面所举文件系统中的例子，若从数据库角度进行组织，则如图 1-3 所示。

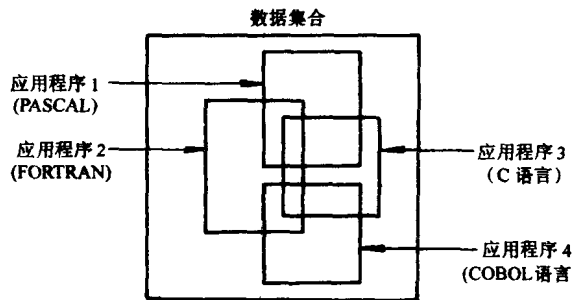


图 1-2 应用程序共享数据集合

由该结构可看出，数据不再分属于各个应用程序，而是集中存储在一起。对一个特定的组织而言，除了由于需要保密而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。在图 1-3 中，学生记录是为教务处、学生处和卫生院所共享的，若某个学生需要转系，则只要修改学生记录中的系名属性即可，这样就不会出现不一致性的情况。另外，除了共享的数据以外，各部门还可以有自己的私有数据，这些数据也放在数据库中，但为了防止其他部门对它进行访问，可加上各种保密限制，禁止非法获得这些数据。

由于数据库是高度结构化的，数据库中数据项之间及记录之间是有相互联系的，因此，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，而文件系统则很难达到这一点。

2. 具有较高的数据和程序的独立性

在文件系统阶段，对文件的访问实际上是对存储数据的物理介质或设备进行访问，一旦更换物理介质或改变文件存储结构，则势必会影响应用程序。而在使用数据库时，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像来获得的。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

像，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。在当前硬件迅猛发展的情况下，存储数据的物理介质及设备不断更新换代，存储方法和存取策略也不断改善，物理独立性可以使数据库的物理存储结构随技术发展不断加以调整，而不必修改应用程序。

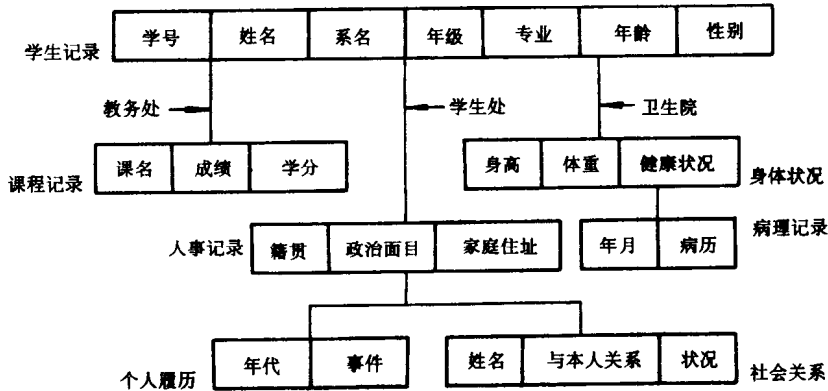


图 1-3 共享数据范例

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。例如，在图 1-3 中，假如学生处要登记每个学生的出生年月，那么可以在学生记录中加一个数据项（出生年月）。如果教务处和卫生院不需要访问该数据项，则其原来的应用程序可以不改变。提高数据独立性是数据库所追求的一个主要目标。在数据库中，数据与应用程序的关系如图 1-4 所示。

3. 对数据实行集中统一的控制

由于数据库中的数据为各种用户所共享，而计算机的共享一般是并发的，即多个用户可以同时使用数据库，因此数据库系统除了提供统一的数据定义、检索及更新操作手段外，还需提供控制数据安全性和完整性的方法，并能保障系统在并发存取数据时的正确执行。

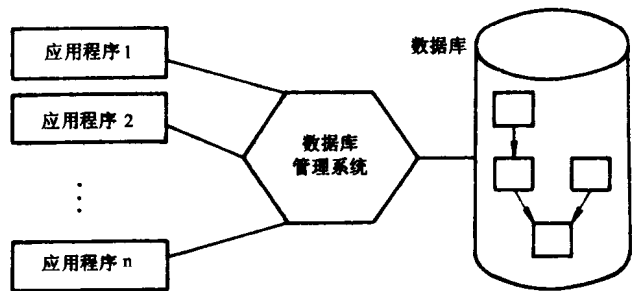


图 1-4 数据库中数据与应用程序的关系

一个较完善的数据库系统通常提供以下三个方面的数据控制功能：

(1) 数据的安全性控制 数据的安全性是指保护数据以防止不合法的使用所造成的数据泄密和破坏，这就要采取一定的安全保密措施。例如，系统用检查口令或其他手段来检查用户身份，合格用户才能进入数据库系统；提供用户保密级别和数据存取权限的定义机制，当用户对数据库执行操作时，系统自动检查用户能否执行这些操作，检查通过后才执行允许的操作。

(2) 数据的完整性控制 数据的完整性指数据的正确性、有效性与相容性。数据库中的数据是对客观世界中某些实体性质的反映，它有一定的语义含义，如性别的值只可能是

“男”或“女”；会计记账时，收支应当平衡，即收入之和减支出之和等于剩余数；月份是1~12之间的正整数；学生的学号是唯一的等等。数据库系统可以提供必要的功能，保证数据库中的数据在输入、修改过程中始终符合原来的定义和规定。

(3) 并发控制 当多个用户的并发进程同时存取、修改数据库时，可能会发生互相干扰而得到错误的结果，并使数据库完整性遭到破坏，因此必须对多用户的并发存取动作加以控制和协调。

鉴于数据库系统的上述特点，大型复杂的信息系统大多以数据库为核心，因而数据库系统在计算机应用中起着越来越重要的作用。

1.1.4 数据库基本概念

本节主要给出与数据库相关联的几个基本概念。

1. 数据库 (Data Base)

数据库是指在辅助存储器中的存储数据。这些数据是现实世界中的一些相关信息，它们在特定的组织（企、事业）中能为多种应用（或用户）服务。它们以数据模型所确定的数据结构方式存储，并能以有效的存取方法为操纵数据的语言提供快速响应。

2. 数据库管理系统 (Data Base Management System)

数据库管理系统 (DBMS) 是数据库的一个软件，它允许一个或多个使用者对数据库中的抽象数据提出请求（包括询问和修改），并以合乎使用者要求的格式提供给使用者。抽象数据的含义是指数据由数据模型表示。若数据以关系数据模型存储，则相应的管理系统称为关系数据库管理系统。

3. 数据库系统 (Data Base System)

大量的经过加工整理的、存储在数据库中的数据，由数据库管理系统管理，为多个不同的应用（或用户）共同使用的数据处理系统，可称为数据库系统。

由此可见，数据库系统有两个主要组成部分：数据库管理系统和数据库。其中，当然也包括了存放数据的存储介质及其设备和使用、维护数据库系统的人员。

数据库系统的研究包括三个方面：

(1) 数据库管理系统的研究 包括 DBMS 应具有什么样功能的原理性问题和如何实现的技术性问题的研究。当前，DBMS 的研究已从集中式数据库管理系统向分布式数据库管理系统 (DDBMS)、知识库管理系统 (KBMS) 等方面延伸，以及延伸到适应各种应用领域。

(2) 数据库理论的研究 主要围绕关系数据库理论、事务理论、逻辑与数据库（演绎数据库）、面向对象的数据库、知识库等方面的研究，探索新思想的表达、提炼、简化，最后使其为人们所理解；研究新算法以提高数据库效率。

(3) 数据库设计方法及工具的研究

数据库设计的主要含义是在数据库管理系统的支持下，按照应用要求为某一部门或组织设计一个结构良好、使用方便、效率较高的数据库及其应用系统。目前在这一领域是进行数据库设计方法、设计工具和理论的研究；数据模型和数据建模的研究；计算机辅助数据库设计方法及其软件系统的研究；数据库设计规范和标准的研究等等。

4. 数据模型

数据库的数据模型可以看作是一种形式化描述数据、数据之间的联系以及有关的语义约束规则的抽象方法。它规定数据如何结构化和一体化，以及规定对这种结构化数据进行何种操作。

当前流行的数据库系统中有四种最常用的数据模型，它们是：层次模型、网状模型、关系模型和面向对象模型。

5. 数据库语言

数据库语言由数据定义语言（DDL）和数据操纵语言（DML）组成，它为用户提供了交互式使用数据库的方法。数据定义语言负责描述和定义数据的各种特性，用户通过使用数据定义语言可将数据库的结构以及数据的特性通知给相应的 DBMS，从而生成存储数据的框架，DDL 用来定义数据库的模式。数据操纵语言实现对数据库数据的基本操作：检索、插入、修改和删除。DML 有两类：一类是嵌入主语言中的，如嵌入 COBOL、C 等高级语言中，这类 DML 语言本身不能独立使用，因此称为宿主型的；另一类是交互式命令语言，它们语法简单，可以独立使用，所以称为自主型或自含型的。

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具有意义、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。概念模型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、易于为用户所理解。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、层次模型和关系模型等。基本数据模型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。

鉴于关系模型的广泛流行，关系数据库越来越受到人们的重视，它已成为数据库领域的主流。本书在后几章将重点介绍关系数据库的基本思想与技术。在这一节只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。

1.2.1 概念模型

人类对错综复杂的现实世界进行某些特定问题研究时，常常需要将其抽象、提取主要因素，略去次要成份，经过归纳形成一个较清晰的轮廓进行具体研究。这样可使复杂的问题简化，从而易于处理。这种方法实际上就是建立模型的方法。

现实世界中事物千姿百态，但它们总能相互区别，主要方法是通过事物的特征来区分。如大学生的特征是：学号、姓名、年龄、性别、系、专业、年级等；椅子的特征是：椅子腿、椅子背、椅子座等。因此可以选择某些特征来识别各事物间的区别。其次，事物之间是息息相关的，如学生坐椅子，工人制造椅子等，表示事物之间是有联系的。概念模型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，下面介绍其主要概念。

1. 实体 (Entity)

实体是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，一位学生、一位老师、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，如一位学生选了一门课程、某个老师订了一份报纸等，都可称为实体。同型实体的集合称为实体集。例如，全体学生就是一个实体集。每个实体集都有一个名称以便与其他实体集相互区别，例如，学生、课程、教师订报纸等都是实体集。

2. 属性 (Attribute)

属性是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。例如：950818，徐展飞，18，男，95级，计算机系，软件，这些属性组合起来便表示了徐展飞这个学生。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。

3. 联系 (Relationship)

现实世界的事物之间是有联系的，这种联系必然要在信息世界中加以反映。学校中教与学的联系，可以以教师实体集与学生实体集两者间的关联表示：教师教学生；学生从教师的讲课中获取知识。在信息世界中，实体之间的联系可分为三类：

(1) 一对一联系 (1:1) 设有两个实体集 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。例如，看电影时，观念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) 设有两个实体集 A 和 B，如果 A 中的一个实体与 B 中若干个实体有联系，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 (m:n) 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。

4. 码 (Key)

一个属性或属性的集合，如果它的值（值集）能唯一地识别实体集中每个实体，就称该属性（或属性集）为实体集的码或键。例如，学号是学生实体的码，因为一个学生有一个属于自己的学号，在任一学校都不可能有两个学生具有相同的学号。

5. 值域 (Domain)

值域是属性的值的变化范围。例如，学号的值域为 6 位整数，姓名的值域为字符串集合，年龄的值域为 0~150 之间的整数，性别的值域为（男、女）等。

表示概念模型最常用的方法是实体——联系方法 (Entity-Relationship Approach)，它是 P. P. S. Chen 于 1976 年提出的。这个方法是用 E-R 图来描述某一组织的概念模型，其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

姓名均写入框中，然后用无向线段将上述框连起来，在连接实体的线段上标上联系的类型(1:1, 1:n, m:n)。上面几个例子可用图 1-5 表示。图 1-6 则给出了某学校教务管理的 E-R 图。

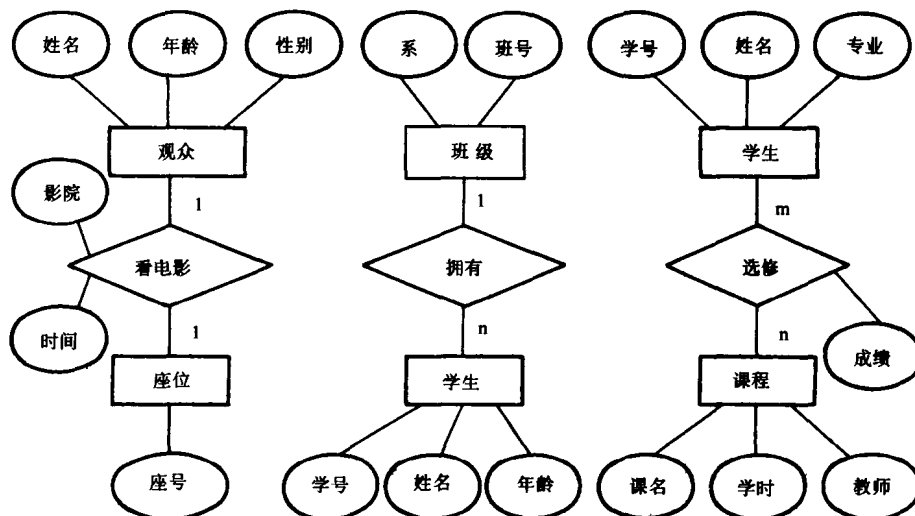


图 1-5 各种联系类型的 E-R 图

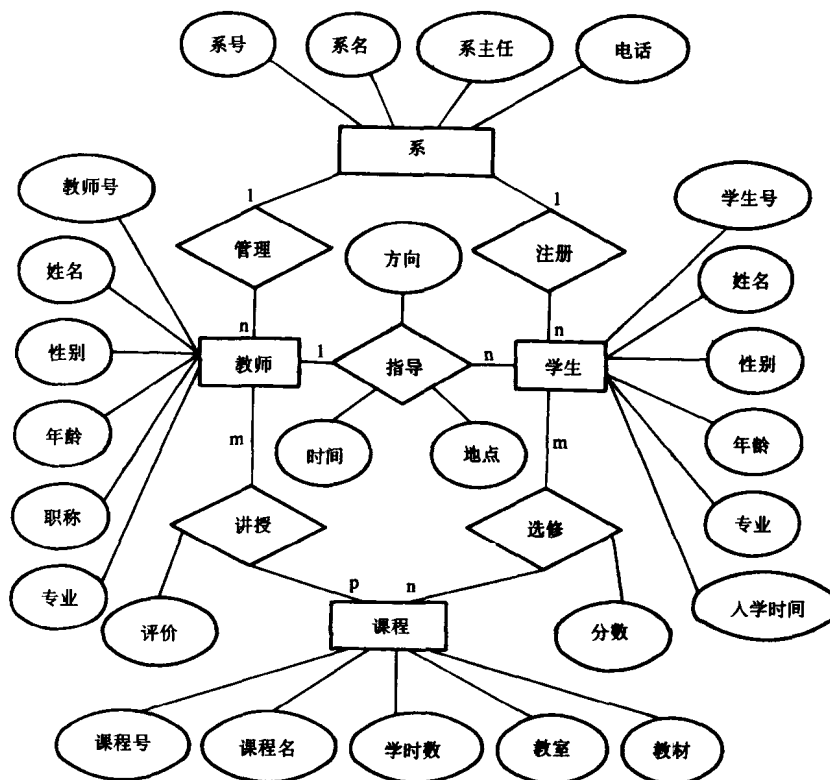


图 1-6 教务管理 E-R 图

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，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。下面我们只对图 1-6 教务管理的概念模型进行一些说明。

教务管理涉及的实体有：

- 系 属性有系号，系名，系主任，电话；
- 教师 属性有教师号，姓名，性别，年龄，职称，专业；
- 学生 属性有学生号，姓名，性别，年龄，专业，入学时间；
- 课程 属性有课程号，课程名，学时数，教室，教材。

这些实体之间的联系如下：

- 一个系有多个教师，一个教师只能属于一个系，因此系和教师具有一对多联系；
- 一个系有许多学生，但一个学生只能在一个系注册，因此系与学生也是一对多联系；
- 在某个时间某个地点一位教师可指导多个学生，但某个学生在某一时间和地点只能被一位教师所指导，因此教师与学生是一对多联系；
- 一位教师可讲授多门课程，一门课程可由多位教师来讲授，每位教师讲授某门课程都有一个评价，因此教师与课程之间是多对多联系；
- 一个学生可选修多门课程，一门课程允许多个学生选修，每个学生选修某门课程都有一个分数（成绩），因此学生与课程之间是多对多联系。

1.2.2 数据模型三要素

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，它实质上是一组向用户提供的规则，这种规则规定数据结构如何组织以及相应地允许进行何种操作。一般说来，一个数据库的数据模型至少必须包含以下三个组成部分：

1. 基本数据结构类型

它是数据模型的最基本成份，它规定如何把基本数据项组织成大的数据单位，并通过这种结构来表达数据项之间的联系。由于数据模型是向用户提供的。因此，其所规定的基本数据结构类型应该是简单的、基本的、易于为用户理解的；另一方面，这种基本数据结构类型还必须有足够强的表示能力，即由这些基本数据结构类型可以有效地表达数据之间各种复杂的语义相关关系。

在数据库系统中，通常按照数据结构的类型来命名数据模型，如层次结构、网状结构和关系结构的模型分别命名为层次模型、网络模型和关系模型。

2. 数据操作

数据操作是指对数据库中各种对象（型）的实例（值）允许执行的操作的集合，包括操作及有关的操作规则。数据库主要有检索和更新（包括插入、删除、修改）两大类操作。数据模型要定义这些操作的确切含义、操作符号、操作规则（如优先级别）以及实现操作的语言。

数据结构是对系统静态特性的描述，数据操作是对系统动态特性的描述。

3. 数据的约束条件

数据的约束条件是完整性规则的集合，完整性规则是给定的数据模型中数据及其联系所