

# 第 3 章

## 数据库管理技术

### 3.1 概 述

数据库技术是 20 世纪 60 年代后期产生并发展起来的数据管理技术,是计算机学科的一个重要分支,它的出现极大地促进了计算机应用向各行各业的渗透。本章以数据库管理技术为重点,介绍数据库的相关概念、定义、关系数据库的基本理论、标准查询语言、数据库的设计方法,以及数据库的安全性和完整性等。

#### 3.1.1 基本概念

在数据处理领域中,存在两个最基本的概念,数据(data)和信息(information)。其中信息是现实世界中事物的存在方式或运动状态的反映。信息是可以感知的;信息是可以存储、压缩、加工、传递、共享、扩散、再生和增值的;信息的传递需要物质载体,信息的获取和传递要消耗能量。而数据是将现实世界中对客观事物的各种描述信息记录下来,形成可以识别的一组文字、数字或符号,它是对客观事物的反映和记录。

数据与信息是密切相关的,信息是向人们(或机器)提供关于现实世界有关事物的知识,数据则是描述事物信息的符号记录,二者不可分离,但又有一定的区别。

数据库(DataBase,DB)是按照一定格式,在计算机内存储的数据的“仓库”。数据库中的数据按照一定模型组织,查找和存取效率高,数据冗余小,可为多用户共享。

数据库管理系统(DataBase Management System,DBMS)是位于用户与操作系统之间用于管理数据库的软件系统。数据库管理系统要在操作系统的支持和控制下运行,其主要功能包括数据定义功能、数据操纵功能、数据库的运行管理及数据库的建立与维护等。

数据库系统(DataBase System,DBS)从广义上说,是指计算机系统引入数据库之后的系统,一般由硬件系统、数据库、数据库管理系统、数据库应用系统和用户构成。从狭义上说,数据库系统可以看做是基于数据库的应用系统。

#### 3.1.2 数据库管理技术发展史

数据库管理技术是应数据管理任务的需要而产生的。数据管理是数据处理的中心问

题,包括对数据的分类、组织、编码、存储、检索和维护等。随着计算机硬件和软件技术的发展,数据库管理技术不断更新完善,其发展过程可划分为四个阶段:人工管理阶段、文件系统阶段、数据库系统阶段和高级数据库系统阶段。

### 1. 人工管理阶段

20世纪50年代末期以前,计算机主要用于科学计算,数据量不大,没有磁盘等直接存取的存储设备,只有纸带、卡片和磁带等外存,也没有专门的软件系统负责数据管理。这个阶段,由于受到计算机的软硬件等方面的限制,只能由人工完成对数据的处理。

### 2. 文件系统阶段

20世纪50年代末期至60年代中期,计算机不仅用于科学计算,而且还大量用于信息管理。在硬件方面,外存储器有了磁盘等直接存取的存储设备;在软件方面,操作系统所提供的文件管理子系统,专门用于管理数据。数据可以以文件的形式长期保存在外部存储器上,用户可以根据需要反复使用这些数据。文件系统是应用程序与数据之间的一个接口,应用程序通过文件系统建立和存储数据文件,使得数据和程序具有了一定的独立性。

但文件系统存在很大的局限性,例如,数据仍然是面向应用的,数据的逻辑结构一旦改变,仍需要修改应用程序,不是一个弹性的无结构数据集合;另外,文件系统中数据冗余度大,由于相同数据重复存储、各自管理,因此容易造成数据的不一致。

### 3. 数据库系统阶段

20世纪60年代以后,随着计算机在数据管理领域的普遍应用,数据管理规模的日趋增大,数据量的急剧增加,以及多用户、多应用的实现,促成了数据库和数据库管理系统的诞生,并提出了数据库系统的概念。它的指导思想是对所有的数据实行统一、集中、独立的管理,使数据存储独立于使用数据的程序之外,并能实现数据的共享。保证数据的完整性和安全性,提高了数据管理效率。数据库系统为用户提供了更广泛的数据独立性,进一步减少了数据的冗余度,为用户提供了方便的操作接口。

### 4. 高级数据库系统阶段

20世纪70年代以来,随着计算机技术的不断发展,出现了分布式数据库、面向对象数据库 and 知识数据库,这些统称为高级数据库。

(1) 分布式数据库:随着计算机网络通信技术的迅速发展,使得分布在不同地理位置上的计算机对数据能够实现高度的共享,因此产生了分布式数据库系统。分布式数据库是分布在计算机网络中不同结点上的数据集合。它在物理上是分布的而在逻辑上是统一的。在分布式数据库系统中,允许适当的数据冗余,以防止因个别结点上数据的失效,而导致整个数据库系统的瘫痪,而且多台处理机可以并行工作,提高了数据处理的效率。

(2) 面向对象的数据库系统:面向对象的数据库系统是面向对象方法与数据库技术相结合的产物。在面向对象的数据库系统中将程序和方法也作为对象存储并由面向对象

数据库管理系统(OODBMS)统一管理。这样实现了应用程序和数据之间真正的共享。

(3) 知识数据库系统: 数据库技术(DB)和人工智能技术(AI)的结合推动了知识数据库(KD)系统的发展,知识库系统将人类具有的知识以一定的形式存入计算机,实现对知识方便有效的使用及管理。

(4) 数据仓库系统: 数据仓库是一个面向主题的、集成的、不可更新的、随时间不断变化的数据集,它用于支持企业或组织的决策分析处理。

数据仓库系统是一个信息提供平台,它首先从业务处理系统中获得数据,以一定模型组织和存储数据,并为用户提供各种从数据中获取信息和知识的手段。

### 3.1.3 关系数据库定义

关系数据库是将数据表示为表的集合(二维表),通过建立二维表之间的关系来定义结构的一种数据库。关系数据库是最常用的数据库类型,70 年代以来新发展的 DBMS 系统中,近百分之九十都是关系型数据库管理系统,其中涌现出了许多性能优良的商品化关系数据库管理系统。例如,小型数据库系统有 FoxPro、Access、Paradox 等;大型数据库系统有 Oracle、SQL Server、Sybase、Informix、Ingres 和 DB2 等。

关系数据库的一个关系就是一张二维表。表 3.1 描述了学生信息表的样例,表中每一行代表一条记录,每一列代表学生的一个属性,也称为字段,每个字段中所有数据的类型必须一致,例如“出生日期”字段,该列所有数据都是“日期”类型数据。

表 3.1 学生信息表样例

| 学 号     | 姓 名 | 性 别 | 班 级       | 出生日期       |
|---------|-----|-----|-----------|------------|
| 0910257 | 李玉  | 女   | 通信工程 0901 | 1991-08-29 |
| 0910258 | 张力  | 男   | 通信工程 0901 | 1992-06-13 |
| 0910259 | 徐亦洋 | 男   | 通信工程 0901 | 1992-07-03 |
| 0910260 | 马飞宇 | 男   | 通信工程 0902 | 1991-03-09 |
| 0910261 | 张小方 | 女   | 通信工程 0902 | 1992-01-09 |

关系数据库一般都由一组关系组成,针对不同的应用,所构造的关系表结构都有所不同。好的关系数据库一般都要求冗余数据应尽可能少,不存在插入异常不存在删除异常,不存在更新异常等。

### 3.1.4 面向对象数据库定义

面向对象(Object-Oriented,OO)的思想最初出现于挪威奥斯陆大学和挪威计算中心研制的仿真语言 Simula 67 中。随后,美国加州的 Xerox 研究中心推出 Smalltalk 68 和 Smalltalk 80 语言,使面向对象程序设计方法得到完善的实现。20 世纪 80 年代中期,计算机厂商在 Smalltalk 语言基础上,增加了数据库定义语言和操纵语言,并允许数据结构出现任意级嵌套和递归,形成面向对象数据语言(OPAL)。1993 年 Object Data

Management Group (ODMG) 国际组织推出基于对象的工业化的 OODB 标准——ODMG93。1997 年 ODMG 公布第二个标准 ODMG97, 内容涉及对象模型、对象定义语言、对象交换格式、对象查询语言以及这些内容与 C++、Smalltalk 和 Java 之间的绑定。OODBS 是一个面向对象的数据库系统, OODBS 应该满足两个标准: 首先它是一个数据库系统, 具备数据库系统的基本功能; 其次是一个面向对象系统, 充分支持完整的面向对象概念和机制。因此可以将一个 OODBS 表达为“面向对象系统+数据库能力”, 这类系统的好处是可与面向对象程序设计语言一体化, 使用者不需要学习新的数据库语言。

OODB 没有统一可行的标准, 存在一定的问题和局限性。目前, OODB 还缺乏坚实的形式化理论的支持, 没有一个切实可行的标准, 缺乏数据库的许多功能, 以致产品之间的兼容性和可移植性比较差, 这已成为它广泛应用的主要障碍。此外, OODB 涉及程序设计语言和 DBS 集成这一复杂问题, 更需要用标准来加以约束。所以制定规范已刻不容缓。另外, OODBS 不能与关系数据库系统(RDBS)兼容, 没有提供使 RDB 和 OODB 相互转换的“通道”或“桥”, 以使用户根据需要, 发挥两种数据库各自的优势。它也缺乏 RDBS 成功使用的一些功能特点, 如非过程化查询语言、视图和授权。另外, 由于其不支持现行标准的 SQL, 因此失去通用性方面的优势, 使其应用领域受到很大限制。典型的商品化的 OODBS 有 ONTOLOGIC 公司的 ONTOS DB。

### 3.1.5 典型商用数据库管理系统

#### 1. DB2

DB2 是 IBM 公司的产品, 是一个支持多媒体、Web 应用的关系数据库管理系统, 其功能足以满足大中型公司的需要, 并可灵活地服务于中小型电子商务解决方案。DB2 数据库核心又称作 DB2 公共服务器, 采用多进程多线程体系结构, 支持从 PC 到 UNIX, 从中小型机到大型机, 从 IBM 到非 IBM(HP 及 Sun UNIX 系统等)各种操作平台, 并分别根据相应平台环境作了调整和优化, 以便能够达到较好的性能。DB2 可以在主机上以主/从方式运行, 也可以在客户机/服务器环境中运行。其中服务器平台可以是 OS/400、AIX、OS/2、HP-UNIX、SUN-Solaris 等操作系统, 客户机平台可以是 OS/2、Windows、DOS、AIX、HP-UX 和 Sun Solaris 等操作系统。

DB2 核心的特色是支持面向对象的编程, 支持多媒体应用程序的开发, 具有备份和恢复能力, 支持存储过程和触发器, 支持 SQL 查询, 支持异构分布式数据库访问, 以及数据复制。

#### 2. Oracle

Oracle 在数据库领域一直处于领先地位, 1984 年, 甲骨文公司首先将关系数据库转到了桌面计算机上。其后在 Oracle 5 版本中, 率先推出了分布式数据库、客户/服务器结构等崭新的概念。Oracle 6 首创行锁定模式, 并支持对称多处理计算机。Oracle 10 版本则支持面向对象技术, 成为关系-对象数据库系统。目前, Oracle 产品覆盖了大、中、小型等几十种机型, Oracle 数据库已成为世界上使用最广泛的关系数据库系统之一。

Oracle 产品采用标准 SQL,能够与 IBM SQL/DS、DB2、INGRES、IDMS/R 等兼容; Oracle 可运行于很宽范围的硬件与操作系统平台上,也能与多种通信网络相连,支持各种协议(TCP/IP、DECnet、LU6.2 等);其内部提供了多种开发工具,能极大地方便用户进行进一步的开发。

### 3. Sybase

Sybase 把“客户机/服务器数据库体系结构”作为开发产品的重要目标。开发团队吸取了 INGRES 的研制经验,以满足联机事务处理应用的要求,于 1987 年推出了 Sybase SQL Server,称为大学版 INGRES 的第三代产品。

Sybase System 11.5 支持企业内部各种数据库应用需求,如数据仓库、联机事务处理、决策支持系统和小平台应用等。

Sybase 是一个面向联机事务处理,具有高性能的、高可靠性的、功能强大的关系型数据库管理系统。Sybase 数据库的多库、多设备、多用户、多线程等特点,极大地丰富和增强了数据库功能。因为 Sybase 数据库系统是一个复杂的、多功能的系统,所以对 Sybase 数据库系统的管理就变得十分重要,管理的好坏与数据库系统的性能息息相关。

### 4. SQL Server

SQL Server 是 Microsoft 公司开发的大型关系数据库系统。SQL Server 的功能比较全面、效率高,可以作为大中型企业或单位的数据库平台。SQL Server 在可伸缩性与可靠性方面做了许多工作,近年来在许多企业的高端服务器上得到了广泛的应用。同时,该产品继承了 Microsoft 产品界面友好、易学易用的特点,与其他大型数据库产品相比,在操作性和交互性方面独树一帜。SQL Server 可以与 Windows 紧密集成,这种设计使 SQL Server 能充分利用操作系统所提供的特性,不论是应用程序的运行速度,还是系统事务的处理速度,都能得到较大的提升。

SQL Server 具有单进程多线程的体系结构。由于 SQL Server 只有一个服务器进程,所有的客户都连接在这个进程上。该进程又细分为多个并发的线程,它们共享数据缓冲区和 CPU 时间,能及时捕捉各用户进程所发出的存取数据请求,然后按一定的调度算法处理这些请求,比操作系统直接对这些请求进行调度高效得多。

### 5. Microsoft Access

Microsoft Access(Microsoft Office Access)是由 Microsoft 发布的关系型数据库管理系统,它结合了 Microsoft Jet 数据库引擎和图形用户界面两项特点,Access 是 Microsoft Office 办公套件中一个重要成员,现在它已经成为世界上非常流行的桌面数据库管理系统,适用于中小型商务企业,用以存储和管理商务活动所需要的数据。Access 不仅是一个数据库,而且它具有强大的数据管理功能,它可以方便地利用各种数据源,生成窗体(表单)、查询、报表和应用程序等。Access 相对于其他中小型数据库管理系统,更加简单易学,无论用户是要创建个人使用的独立桌面数据库,还是部门或中小公司使用的数据库,都可以使用 Access 作为数据库平台。例如,使用 Access 处理公司的客户订单数

据,管理自己的个人通讯录,记录和处理科研数据等。

Access 最大的特点是界面友好,简单易用,和其他 Office 成员一样,极易被一般用户所接受。因此,在许多低端数据库应用程序中,经常使用 Access 作为数据库平台;在初次学习数据库系统时,很多用户也是从 Access 开始的。

## 3.2 关系数据库规范化理论

### 3.2.1 数据模型

#### 1. 数据模型的基本概念

模型是对现实世界事物特征的模拟和抽象,数据模型是对现实世界数据特征的抽象。从现实世界事物的客观特性到计算机中对事物特性的具体数据表述要经历现实世界、信息世界和计算机世界 3 个数据领域。在这 3 个领域中,对数据的描述采用不同的术语。

(1) 现实世界:是独立于人们意识之外的客观世界,其中存在着客观事物及相互间的关系。一个客观存在并且可以被识别的事物称为个体。个体可以是实际的,也可以是抽象的,例如,一个学生、一门课程都属于实际的个体;而天气、爱好都属于抽象的个体。事物之间可以存在一定的关系,而这些关系由事物本身的性质决定,如学生选修课程、学生借阅书籍等。现实世界的客观事物之间往往存在着错综复杂的关系,如果将这些关系进行选择、命名、分类等抽象处理,则进入了信息世界。

(2) 信息世界:是指现实世界在人脑中形成的概念,因此又被称为概念世界。现实世界中的个体在信息世界中被称为实体,信息世界对实体及实体之间的关系使用概念模型来描述。概念模型都是较为抽象的,它们与具体的数据库或计算机平台无关。

(3) 计算机世界:将信息世界中描述事物及事物之间关系的概念模型转化为计算机中物理结构上的描述,称为物理模型。

在描述现实世界中事物的客观特性时,首先将现实世界的事物及其关系抽象为信息世界的信息模型,然后再将信息模型抽象转换为计算机世界的数据库模型。所以说,数据库模型是现实世界两级抽象的结果。因此,现实世界是信息源泉,是设计数据库的出发点,也是使用数据库的最终结果。信息模型和数据库模型是现实世界事物及其关系的两级抽象,其中数据库模型是实现数据库系统的根据。

#### 2. 数据模型的三要素

数据库结构的基础是数据模型,数据模型的三个要素是数据结构、数据操作和数据的约束条件。

(1) 数据结构:用于描述实体及其关系按照何种方式存储,是对系统静态特性的描述。

(2) 数据操作:用于描述对数据可以进行何种操作(如查询、插入、删除和修改),是对系统动态特性的描述。

(3) 数据的约束条件: 完整性约束规则的集合, 即对于具体的应用数据, 必须遵循特定的语义约束条件, 以保证数据的正确、有效和相容。

数据模型的类型分为概念数据模型和基本数据模型。概念数据模型也称为信息模型, 是按用户的观点对数据和信息建模, 是现实世界到信息世界的第一层抽象, 强调其语义表达功能, 易于用户理解, 主要用于数据库设计。这类模型中最著名的是实体关系模型, 简称 E-R 模型; 基本数据模型是对现实世界中数据的第二层抽象, 是按计算机世界的观点对数据建模, 描述数据库中数据的存储和组织方式, 即如何表示实体以及实体之间的关系, 它是数据库系统的核心和基础。基本的数据模型有层次模型、网状模型和关系模型等。

### 3. E-R 模型

E-R 模型也称为实体关系模型或实体联系模型, 是描述信息世界、建立概念模型的实用工具。数据库设计过程一般要先给出 E-R 模型图描述, 再进一步转换为相应 DBMS 所支持的数据模型。E-R 模型涉及 3 个主要概念: 实体、属性和实体关系。

(1) 实体: 可以被识别的, 客观存在的事物即为实体。例如一个学生就是一个实体, 是真实世界的学生在大脑中的概念。实体也可以是抽象的, 如天气、满意度等。

(2) 属性: 实体具有的某一特性即被称为属性, 其中能够唯一标识一个实体的属性被称为主键属性。例如, 一个学生的属性可以有学号、姓名、性别和班级等组成, 其中学号属性能够唯一的标识一个学生, 为主键属性。

属性又有“型”和“值”之分, 例如, 学生实体的学号、姓名、班级等都是属性的型(类型), 而属性的值就是具体的内容, 如学生 0910257、李玉、通信工程 0901 等分别是学号、姓名、班级属性的值。由此可见若干属性型可以表现一个实体的类型, 若干个属性值的集合(即元组)则构成了实体的值。

具有相同属性的实体的集合称之为“实体集”, 如所有的学生。

(3) 实体间的关系: 实体集之间存在的联系被称为关系。关系分为两种: 一种是实体内部各属性之间的关系; 另一种是实体之间的关系。这里要讨论的是实体之间的关系, 两个实体集之间存在一对一、一对多和多对多的关系类型。

① 一对一关系(1:1): 设 A、B 为两个实体集, 若 A 中的每个实体至多和 B 中的一个实体有关系, 反过来, B 中的每个实体也至多和 A 中的一个实体有关系, 则称 A 对 B 或 B 对 A 是 1:1 关系。例如, 学校和正校长之间就是 1:1 关系。

② 一对多关系(1:n): 如果 A 中的每个实体和 B 中的多个实体有关系, 而 B 中的每个实体至多和 A 中的一个实体有关系, 称 A 对 B 是 1:n 关系。例如, 学校和教师之间就是 1:n 关系; 班级和学生之间也是 1:n 关系。

③ 多对多关系(m:n): 如果 A 中的每个实体和 B 中的多个实体有关系; 反过来, B 中的每个实体也和 A 中的多个实体有关系, 称 A 对 B 是 m:n 关系或 B 对 A 是 m:n 关系。例如, 一门课程可以同时有多个学生选修; 一个学生也可以同时选修多门课程, 课程和学生之间是多对多关系。

为了直观地表示信息世界中实体及实体之间的关系, 在 E-R 模型图中, 实体使用矩

形框来表示;实体的属性使用椭圆形框来表示,其中主键属性加下划线表示;实体间的关系则用菱形框来表示,并用无向线段连接各图形框。

例如,描述学校教务管理系统,系统中涉及学生、课程和教师等实体。假设学生实体有学号、姓名、性别、出生日期和班级等属性;课程实体有课程号、课程名、学分和学时数等属性;教师实体有教工号、姓名、性别、职称和教研室等属性;其中学生实体通过选课与课程实体建立了关系,即某个学生选修某门课程可以得到相应的成绩,其关系类型为多对多关系;教师实体通过授课与课程实体建立了关系,假设该关系为一对多关系,即某个教师可以教授多门课程,而某门课只能由一位教师教授。该教务管理系统的 E-R 模型图如图 3.1 所示。

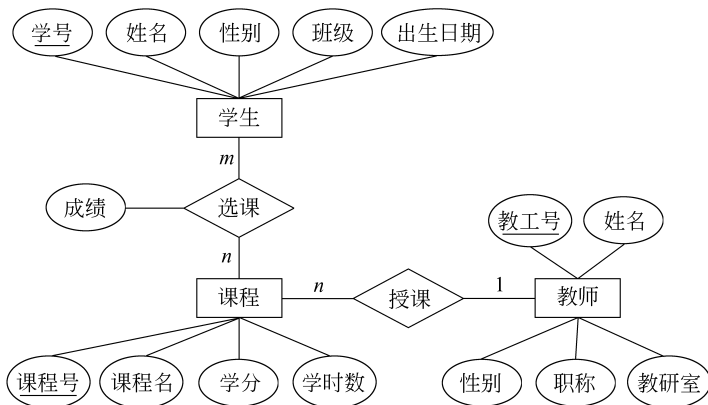


图 3.1 学校教务管理系统 E-R 模型图

#### 4. 基本数据模型

实体关系(E-R)模型是对现实世界客观事物之间关系的第一级抽象的结果。在计算机世界中,E-R 模型又被抽象为基本数据模型。基本数据模型是 E-R 模型的数据化,它将实体模型内部之间的关系抽象为字段(数据项)之间的关系,将实体模型之间的关系抽象为记录之间的关系。基本数据模型反映了这两类复杂关系的形式,它是对数据库中的数据组织结构化的描述。

基本数据类型的种类很多,目前常用的有层次模型、网状模型和关系模型。

##### 1) 层次模型

层次模型是数据库系统中最早使用的模型,它的基本结构为树形结构,每个结点表示一个实体集,实体之间的关系为一对多的关系。学院教务管理系统的层次模型如图 3.2 所示,层次模型中有且只有一个结点无双亲(这个结点称为根结点),其他结点则有且只有一个双亲结点。

层次模型具有结构简单、层次清晰、容易实现等优点,但对于一些非层次性结构(如多对多

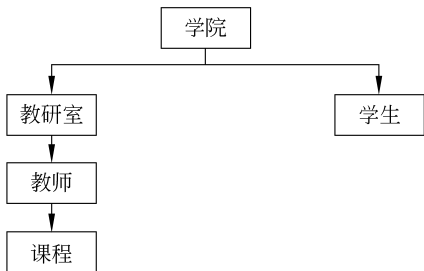


图 3.2 层次模型举例



关系),层次模型需要对关系进行分解,表现起来比较烦琐,也不直观。另外,动态访问层次模型中的数据(如插入或修改记录)时,对应用程序的编制比较复杂,且操作效率不高。

### 2) 网状模型

网状模型可以看作是层次模型的扩展。它采用网状结构表示实体及其之间的关系,其中一个结点可以没有父结点;也可以有多于一个的父结点,一个结点可以有多个双亲结点,多个结点可以无双亲结点。学院教务管理系统的网状模型如图 3.3 所示。

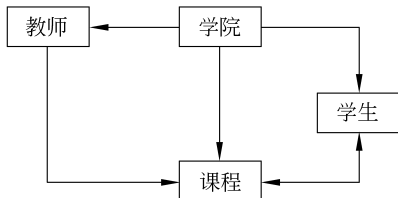


图 3.3 网状模型举例

与层次模型相比,网状提供了更大的灵活性,能更直接地描述现实世界,性能和效率也比较好。但网状模型结构复杂,用户不易掌握,实体关系变动后会涉及链接指针的调整,扩充和维护都比较复杂。

### 3) 关系模型

关系模型是目前应用最多、也最为重要的一种数据模型。关系模型建立在严格的数学概念基础上,采用二维表格结构来表示实体和实体之间的关系。例如,将图 3.1 所示学校教务管理系统 E-R 模型图转化成关系模型,则该关系模型的结构如图 3.4 所示。

|     |    |    |    |      |     |     |     |    |     |
|-----|----|----|----|------|-----|-----|-----|----|-----|
| 学生表 |    |    |    |      | 课程表 |     |     |    |     |
| 学号  | 姓名 | 性别 | 班级 | 出生日期 | 课程号 | 课程名 | 学时数 | 学分 | 教工号 |
|     |    |    |    |      |     |     |     |    |     |
| 教师表 |    |    |    |      | 选课表 |     |     |    |     |
| 教工号 | 姓名 | 性别 | 职称 | 教研室  | 课程号 | 学号  | 成绩  |    |     |
|     |    |    |    |      |     |     |     |    |     |

图 3.4 教务管理系统关系模型举例

关系模型中没有层次模型和网状模型中的链接指针,实体之间的关系是通过不同关系表中的同名属性来实现的。

例如,教师表和课程表中都有教工号属性,如果要查询某位教师所讲授的课程信息,可以先从教师表找到该教师的教工号,然后再在课程表中找到该教工号所对应的课程信息。从这一点可以看出,教工号字段建立了两个表之间的关系。另外,图 3.4 中的选课表也建立起了课程表和学生表之间的关系。

关系模型建立在关系代数理论之上,具有可靠的数学基础;关系模型的优点是概念清晰,结构简单,一对一、一对多和多对多的关系都可以很方便的描述;表示数据的形式统一,实体、实体之间的关系及查询结果都采用关系表示,用户比较容易理解;另外,关系模型的存取路径对用户是透明的,程序员不用关心具体的存取过程,减轻了程序员的工作负担,具有较好的数据独立性和安全保密性。

### 3.2.2 规范化理论

在关系模型中,数据库模式就是关系模式的集合。因此,关系数据库的设计就是关系模式的设计。关系数据库规范化理论是由 E. F. Codd 于 1971 年提出的,目的是要设计“好的”关系模式。规范化是在关系型数据库中减少数据冗余的过程。

#### 1. 数据冗余问题

假设有如下关系模式,关系名为学生管理,其中包含:学号、姓名、性别、班级、课程号、课程名、学分、成绩等属性,该关系模式中(学号,课程号)为主键。关系表中的数据如表 3.2 所示。

表 3.2 学生管理表

| 学号      | 姓名 | 性别 | 班 级       | 课程号   | 课程名   | 学分 | 成绩 |
|---------|----|----|-----------|-------|-------|----|----|
| 0910257 | 李玉 | 女  | 通信工程 0901 | 08102 | 计算机基础 | 3  | 85 |
| 0910258 | 张力 | 男  | 通信工程 0901 | 08102 | 计算机基础 | 3  | 75 |
| 0910257 | 李玉 | 女  | 通信工程 0901 | 03201 | 高数    | 4  | 76 |
| 0910258 | 张力 | 男  | 通信工程 0901 | 03201 | 高数    | 4  | 80 |
| ⋮       | ⋮  | ⋮  | ⋮         | ⋮     | ⋮     | ⋮  | ⋮  |

表 3.2 所示关系表存在以下问题:

(1) 数据冗余:如果一个学生选修多门课程,或一门课程同时由多个学生选修,将导致表中“姓名”、“性别”、“班级”、“课程名”等相同的属性值被多次重复存储,这就造成了数据的冗余。

(2) 不一致性:由于数据存储冗余,当更新某些数据项时,有可能一部分记录修改了,而另一部分未修改,造成存储数据的不一致性。

(3) 插入异常:如果要开设一门新课程,插入新的课程信息,但因为还没有学生选修该课程,导致新信息无法插入。因为该关系模式中(学号,课程号)为主键字段,因为还没有学生选修,因此“学号”字段为空,而关系数据模型规定主关键字不能为空,这就导致了插入异常。

(4) 删除异常:当删除某门课程时,则与该课程相连的所有学生信息也会被删除。如果某个学生只选修了这门课程,则该学生就将被彻底删除,这种现象叫做删除异常。

导致上述问题的原因就是所构造的关系模式中各属性之间互相关联,互相依赖,因此也互相制约。在构造关系模式时,必须从语义上区分出这些关联,将互相依赖的属性构成单独的模式。例如,将表 3.2 所示关系模式分解为表 3.3 所示的三个表,即可有效减少数据冗余,并消除上述异常问题。