

高等职业教育计算机技术专业贯通制教材

# 数据库原理及应用

■ 主编 黄斌生 ■ 副主编 李 晔 葛文芹 ■

本书配有电子教学参考资料包



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

高等职业教育计算机技术专业贯通制教材

# 数据库原理及应用

主 编 黄斌生  
副主编 李 晔 葛文芹

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING



## 内容简介

本书内容包括数据库、关系数据库基础理论, 关系数据库标准语言, 数据库安全与保护知识, 数据库应用技术和数据库设计方法, 系统地介绍了数据库的基本概念、基本原理和应用技术, 最后以案例形式整体描绘了数据库应用系统的开发过程。本书内容精练、结构清晰, 并以 SQL Server 实例来阐述枯燥的理论, 直观易懂。

本书参考学时为 64 学时, 可作为中等职业学校、高等职业学校、成人高等学校和职业技术学院计算机及相关专业的教材。

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

数据库原理及应用 / 黄斌生主编. —北京: 电子工业出版社, 2008. 7  
高等职业教育计算机技术专业贯通制教材  
ISBN 978-7-121-06731-0

I. 数… II. 黄… III. 数据库系统 - 高等学校: 技术学校 - 教材 IV. TP311.13

中国版本图书馆 CIP 数据核字 (2008) 第 092217 号

策划编辑: 施玉新

责任编辑: 施玉新 牛旭东

印刷: 北京季蜂印刷有限公司

装订: 三河市万和装订厂

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开本: 787 × 1092 1/16 印张: 12.5 字数: 313.6 千字

印次: 2008 年 7 月第 1 次印刷

印数: 3 000 册 定价: 19.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 [zlt@phei.com.cn](mailto:zlt@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010) 88258888。

# 前 言



数据库技术是应用最广泛的计算机科学技术,因而数据库原理及应用课程成为计算机应用专业的主干课程之一。希望读者通过本书的学习能形成数据库系统设计理念,具有一定的理论基础和实践操作能力,能开发简单的数据库应用系统并适应数据库管理员等岗位要求。

本书根据学生特点,以职业教育的性质、任务和培养目标为出发点,坚持以就业为导向、以能力培养为本位的原则,突出教材的实用性和适用性。数据库原理是用于指导开发数据库应用系统的基本原理,理论应用于实践才有意义,也只有在实践中才易于消化吸收。因而,本书尽量简化理论,以必须、够用为度,帮助读者从实践中体会理论,从而架构完整的数据库知识体系,并能将其运用于数据库应用系统开发。

在内容选取、章节安排、难易程度等方面充分考虑教与学的需要,力求使教材概念准确清晰,重点明确,并根据内容需要采用不同的组织结构。

本书内容包括数据库和关系数据库的一些基本概念、原理,并以 SQL Server 2000 为平台,介绍了关系数据库标准语言、数据库的安全与保护知识及数据库应用技术,然后以实例详细描述了数据库设计的方法和步骤,最后以案例的形式给出开发数据库应用系统的整个流程,使读者学习完本书后就能运用于实践。

本书参考学时为 64 学时,可作为中等职业学校、高等职业学校、成人高等学校和职业技术学院计算机及相关专业的教材。

本书由黄斌生(张渚职业学校)主编,葛文芹(惠山职业教育中心)、李晔(无锡市广播电视大学)为副主编,钱杰(宜兴成职教研室)、吴军(无锡高等师范学校)、谢漪(惠山职业教育中心)参与编写,无锡交通高等职业学校许春勤教授主审。

由于编者水平有限,书中难免有不足之处,敬请读者批评指正,不胜感激。

联系 E-mail: H720126@ Hotmail. com

为了方便教师教学,本书还配有教学指南、电子教案及习题答案(电子版)。请有此需要的教师登录华信教育资源网([www.huaxin.edu.cn](http://www.huaxin.edu.cn)或[www.hxedu.com.cn](http://www.hxedu.com.cn))免费注册后再进行下载,有问题时请在网站留言板留言或与电子工业出版社联系(E-mail: [hxedu@phei.com.cn](mailto:hxedu@phei.com.cn))。

编 者

2008 年 5 月



# 目 录



<b>第 1 章 数据库系统概述</b> .....	1
1.1 数据管理及其技术的发展 .....	1
1.1.1 数据、信息与数据管理 .....	1
1.1.2 数据管理技术的发展历程 .....	2
1.2 数据库基本概念 .....	4
1.2.1 数据库 .....	4
1.2.2 数据库管理系统 .....	5
1.2.3 数据库系统 .....	5
1.2.4 数据库技术的发展历程 .....	6
1.3 数据模型与数据模式 .....	6
1.3.1 数据模型 .....	6
1.3.2 数据模式 .....	9
习题 .....	11
<b>第 2 章 关系数据库</b> .....	13
2.1 关系模型概述 .....	13
2.1.1 关系数据结构 .....	13
2.1.2 关系操作 .....	13
2.1.3 数据完整性约束 .....	14
2.2 关系数据库概述 .....	14
2.2.1 基本概念 .....	14
2.2.2 关系的性质 .....	15
2.2.3 关系模式 .....	15
2.3 关系代数 .....	15
2.3.1 传统的集合运算 .....	16
2.3.2 专门的关系运算 .....	16
2.3.3 关系代数应用 .....	17
2.4 关系演算 .....	19
2.4.1 元组关系演算 .....	19
2.4.2 域关系演算 .....	20
2.5 关系系统 .....	21
2.5.1 关系系统的定义和分类 .....	21
2.5.2 全关系系统的基本准则 .....	21
习题 .....	23
<b>第 3 章 关系数据库标准语言</b> .....	25
3.1 SQL Server 的安装与配置 .....	25
3.1.1 系统需求与安装 .....	25

3.1.2	SQL Server 服务	30
3.2	SQL 概述	31
3.2.1	SQL 的功能与特性	31
3.2.2	SQL 语句结构与书写格式	32
3.2.3	T-SQL 基础	32
3.3	数据库与基本表	33
3.3.1	数据库的创建与维护	33
3.3.2	基本表的建立与维护	35
3.3.3	数据操作	39
3.3.4	创建与维护索引	41
3.3.5	完整性约束	43
3.4	查询	46
3.4.1	单表查询	46
3.4.2	连接查询	50
3.4.3	嵌套查询	54
3.4.4	集合查询	57
3.5	视图	57
3.5.1	视图设计	57
3.5.2	通过视图修改数据	60
	习题	62
	上机	63
<b>第4章</b>	<b>数据库安全与保护</b>	<b>66</b>
4.1	数据库的安全管理	66
4.1.1	数据库安全性概述	66
4.1.2	SQL Server 安全管理	67
4.2	事务与并发控制	77
4.2.1	事务概述	77
4.2.2	控制事务	78
4.2.3	事务设计	79
4.2.4	并发控制	80
4.3	数据的备份与恢复	83
4.3.1	故障类型	84
4.3.2	恢复模型	84
4.3.3	备份类型	85
4.3.4	备份和还原操作	86
4.3.5	自动化管理备份	93
4.4	数据的导入导出与数据库复制	95
4.4.1	数据的导入与导出	95
4.4.2	数据库复制	98
	习题	111
	上机	112
<b>第5章</b>	<b>数据库应用技术</b>	<b>114</b>
5.1	数据库应用体系结构	114
5.1.1	客户/服务器体系结构	114

5.1.2	浏览器/服务器体系结构	115
5.1.3	分布式数据库系统	116
5.2	开放数据库互连	117
5.2.1	ODBC 简介	117
5.2.2	建立 ODBC 数据源	118
5.3	Visual Basic 应用程序访问数据库	120
5.4	Web 应用程序访问数据库	127
	习题	132
	上机	133
<b>第 6 章</b>	<b>数据库设计</b>	<b>135</b>
6.1	数据库设计概述	135
6.1.1	数据库设计的目标	135
6.1.2	数据库设计的方法	136
6.1.3	数据库设计基本过程	136
6.2	需求分析	137
6.2.1	需求分析的任务	137
6.2.2	需求分析的方法	138
6.3	概念结构设计	145
6.3.1	何谓概念结构设计	145
6.3.2	概念结构设计的要求及方法	145
6.3.3	局部概念结构设计	146
6.3.4	全局概念结构设计	148
6.4	逻辑结构设计	151
6.4.1	概念模型向关系模型转换规则	151
6.4.2	用关系规范化理论对关系数据模型进行优化	153
6.5	物理数据库设计	157
6.5.1	数据库物理设计目标及步骤	157
6.5.2	数据库物理设计的主要内容	158
	习题	160
<b>第 7 章</b>	<b>数据库应用开发</b>	<b>162</b>
7.1	系统设计	162
7.1.1	需求分析	162
7.1.2	概念结构设计	167
7.1.3	数据库设计	168
7.2	功能模块设计与实现	173
7.2.1	系统功能模块	173
7.2.2	界面设计及主要代码	174
7.3	系统运行与调试	188
7.3.1	调试方法	188
7.3.2	调试过程	188
	参考文献	190

# 第1章 数据库系统概述



## 本章学习要点

通过本章的学习，了解信息与数据的联系及区别；了解数据管理技术的发展历程和数据库技术的发展状况；能区分 DB、DBS、DBMS 这些基本概念；理解数据模型的概念、意义及三级数据模型间的关系；理解数据模型与数据模式间的联系与区别；通过三级模式和两级映射充分理解数据的独立性。

## 1.1 数据管理及其技术的发展

### 1.1.1 数据、信息与数据管理

#### 1. 数据与信息

当今社会被称为信息社会，信息是什么？随着信息的地位和作用的不断增强，人们对信息的认识也不断加深。较普遍地描述为：信息是反映客观世界中各种事物的特征和变化并可借助某种载体加以传递的有用知识。

数据是人们为了反映客观世界而记录下来的可以鉴别的符号。日常生活中，我们把数据理解为能进行算术运算的数字，但计算机中的数据包含范围很广，文字、图形、图像、声音和视频等都能以数字形式表示（即数字化），因而都是数据。

从数据与信息的概念可以看出，它们都是人们为了反映客观世界而存在的，因而二者是有联系的：信息是通过对数据进行提炼、加工的结果，是对数据赋予一定意义的解释。其关系如图 1.1 所示。

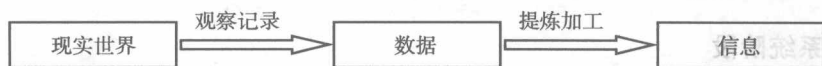


图 1.1 数据与信息的关系

二者又有区别：信息一定是数据，但数据不一定是信息。因为信息有着严格的有用性要求与限定，而数据则无此要求。要从杂乱无序、纷繁复杂的数据中提炼出有用的信息必须通过数据处理，从而管理信息。例如，学生的成绩经过处理得到的平均分，就能称为信息，当然，单个成绩对于其个人来说也是信息，将其与之前比较也能获得信息，但一组数据放在那就不能称为信息。也就是说，根据需要对数据处理后能获得信息，不同的人出于不同的目的，可从同一组数据中获得不同的信息，而且获得的信息有正确的也有错误的。



总之,信息与数据是两个不可分割的概念。信息需用数据来表征,对这些数据再加工又可得新的数据,这些新数据又传递着新的信息。因而,习惯上在一些非严格的场合,常视为同义。例如数据处理又可称为信息处理,数据管理也可称为信息管理。

## 2. 数据管理

数据管理概念的提出始于人们对提高数据处理效率的研究。数据处理是指对数据进行收集、组织、加工、存储、提取及传输等一系列过程。数据管理是指对数据进行分类、组织、检索和维护等工作,所以数据管理是数据处理的核心问题。

### 1.1.2 数据管理技术的发展历程

面对海量的数据,如何有效地进行管理?这就对数据管理提出了更高的技术要求。随着计算机技术的发展,数据管理技术也得到了极大的提高。从数据管理的技术与效率来看,可将其划分为3个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

#### 1. 人工管理阶段

在计算机发明之前(20世纪50年代中期以前),自然是人工管理数据,制作表格、统计数据全靠手工完成,一个数据错误就得重新来过。如果有多个主管部门,就要制作多份格式不同的表格,效率极其低下。这种方式在我国还大量存在。

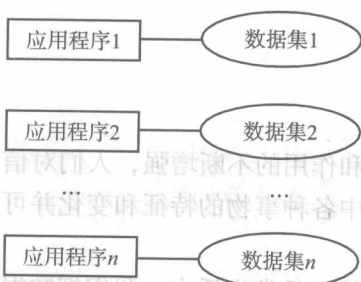


图 1.2 人工管理阶段

计算机发明之初,主要用于科学计算,没有专门管理数据的软件,处理数据时需要开发专用程序,且原始数据与最终结果都需要人工记录,所以称为人工管理阶段,如图 1.2 所示。可以用程序设计语言中的交互式输入语句来解释,一个一个输入数据,数据不能保存,下次要用,还得重新输入。

人工管理阶段的主要特点如下。

- (1) 数据不保存在机器中(那时没有外存)。
- (2) 没有专用软件对数据进行管理,应用程序中不仅要规定数据的逻辑结构,而且要设计物理结构。
- (3) 只有程序的概念,没有文件的概念。
- (4) 数据不共享,即数据面向应用,一组数据对应一个程序,也就是说当相同的一组数据被多个应用程序调用时,需要在各自的程序中重复定义。

#### 2. 文件系统阶段



计算机高级语言、操作系统和磁盘的出现使得用文件来管理数据成为可能,计算机的文件系统将数据存储在磁盘文件中,应用程序通过文件系统对文件中的数据进行存取和加工。这一阶段(20世纪50年代后期到20世纪60年代中期)的特点如下。

- (1) 数据以文件形式长期保存在磁盘等介质上。
- (2) 由专门的软件对数据进行管理,程序和数据从物理上分开,即数据的逻辑结构与物理结构有了区别,这样就不需要再关心数据的物理位置,只需通过文件名就可调用数据。

- (3) 文件组织呈现多样化,有索引文件和链接文件等。

(4) 数据不再属于某个特定程序，可以重复使用。

图 1.3 为文件系统阶段示意图。可以程序设计语言中的数据文件来理解。

但随着数据管理规模越来越大，文件系统暴露出以下缺陷。

(1) 数据共享性差，导致冗余度大。在文件系统中，一个数据文件基本上对应于一个应用程序，当不同的应用程序需要使用具有部分相同的数据时，必须建立各自的数据文件，造成数据在多个文件中重复存储。

(2) 数据不一致性。如上所述，当数据在多个文件中重复存储时，修改一个数据必须同时修改涉及的所有文件，一旦失误就会造成各文件中的数据不一致。

(3) 数据联系弱。数据存在于各个文件中，相互独立，没有联系。

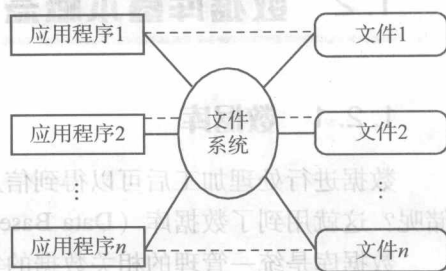


图 1.3 文件系统阶段

### 3. 数据库系统阶段

随着计算机技术的发展，数据库管理系统（20世纪60年代中期以后）出现了，它标志着数据管理技术的飞跃。数据库系统阶段结构如图 1.4 所示。其主要特点如下。

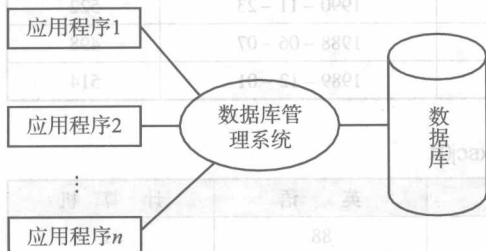


图 1.4 数据库系统阶段

(1) 数据结构化。数据结构化是数据库的主要特征之一，也是数据库系统与文件系统的本质区别。数据库系统阶段采用复杂的数据模型表示数据结构，数据模型不仅能描述数据本身的特点，还能描述数据间的联系，有关内容见数据模型部分。

(2) 有较高的数据独立性。数据的独立性是指应用程序与存储在磁盘上的数据库中的数据相对独立。例如，学生档案表中有很多信息，在某一用户界面上只是显示其中的一部分，如果修改表的结构但没有影响到这一部分，则应用程序无须改动。

数据库系统有较高的数据独立性是因为其数据结构分成用户逻辑结构、整体逻辑结构和物理结构三级，从而使得数据具有物理独立性和逻辑独立性。后面的数据模式部分有更详细的说明。

(3) 数据的共享性高，冗余度低。数据库是从整体出发进行数据描述的，数据不仅面向某个应用，而且面向整个应用程序，即整个应用程序共享这些数据，从而减少了冗余，也避免了数据的不一致性。

(4) 数据库系统为用户提供方便的用户接口，可以使用终端命令操作数据，也可以用程序方式操作数据库。

(5) 数据库系统提供了 4 个方面的数据控制功能——数据库的恢复、并发控制、数据完整性和数据安全性，以保证数据库中数据是安全的、正确的和可靠的。

(6) 对数据的操作不一定以记录为单位，还可以数据项为单位，增加了系统的灵活性。

## 1.2 数据库基本概念

### 1.2.1 数据库

数据进行处理加工后可以得到信息,那么,收集到的原始数据那么多,该如何组织和存储呢?这就用到了数据库(Data Base, DB)。

数据库是统一管理的相关数据的集合,可直观地理解为存储数据的仓库。本书只介绍当前主流数据库——关系数据库。一个关系数据库中可存放多个基本表,每个基本表存储一个关系的所有数据。例如:我们有个名为xs的数据库用于存放学生的一些数据,其中包含学生档案信息表xsda,学生成绩表xscj01、xscj02(分别记录二次考试的成绩),分别如表1.1、表1.2和表1.3所示。

表 1.1 学生档案信息表 xsda

学号	姓名	性别	年龄	出生日期	成绩
060201001	吴晨	男	17	1990-01-27	533
060201002	张燕	女	18	1989-08-07	511
060201003	李子明	男	18	1989-04-11	488
070101007	张卫国	男	17	1990-11-23	522
070102006	张子兴	男	19	1988-06-07	498
070102007	吴敏	女	18	1989-12-01	514

表 1.2 学生成绩表 xscj01

学号	语文	数学	英语	计算机
060201001	80	92.5	88	91
060201002	77	68	79	82
060201003	69	72.5	80.5	78
070101007	55	66	77	68
070102006	67	54	66	63
070102007	73	68	55	88

表 1.3 学生成绩表 xscj02

学号	语文	数学	英语	计算机
060201001	83	90	91.5	88
060201002	83	77	85	80
060201003	79	78	77	89
070101007	66	71	72	77
070102006	58	66	56	71
070102007	67	48	65	82

数据库中不仅仅包含表,还有视图等其他对象,而且可以在表间进行关系运算以形成复杂的多表操作,将在第3章中详细介绍。

以数据库这种方式组织存储数据使得数据独立于程序,能为多个用户所共享,因而被广

泛使用。

### 1.2.2 数据库管理系统

数据库管理系统 (Data Base Management System, DBMS) 是系统软件, 是位于用户与操作系统之间的一层数据管理软件, 为用户或应用程序提供访问 DB 的方法, 包括 DB 的建立、查询、更新及各种数据控制。按其功能可分为三部分。

#### 1. 数据定义

DBMS 提供数据定义语言 (DDL), 用户通过它对数据库中的数据对象进行定义, 可创建数据库、基本表、视图和索引等。

#### 2. 数据操纵

DBMS 的数据操纵语言 (DML) 用于对数据库中的数据进行查询、插入、删除和修改等操作。有时为了突出查询功能 (查询语言 SQL) 可将其作为一个独立部分。

#### 3. 数据库控制

这是 DBMS 的核心功能, 所有数据库的操作都要在其统一管理和控制下进行, 以保障数据库的安全性及完整性, 由数据控制语言 (DCL) 负责。

### 1.2.3 数据库系统

数据库系统 (Data Base System, DBS) 是实现有组织地、动态地存储大量关联数据, 方便多用户访问的计算机软件、硬件和数据资源组成的系统, 即采用了数据库技术的计算机系统。图 1.5 即为 DBS 层次图, 说明如下。

#### 1. 硬件

DBS 对硬件要求较高, 这是因为数据量大, 因而存储器容量要大、传输与处理速度要快。

#### 2. 软件

操作系统是必不可少的, DBMS 是核心, 为了开发应用系统还需要具有数据库接口的开发工具, 如 Visual Basic 和 Delphi 等。

#### 3. 数据

信息社会, 数据是一种重要的资源。数据库系统就是通过对数据进行处理获得信息的系统。

#### 4. 人员

数据库分析、设计、管理、维护和使用人员都是 DBS 的组成部分。用户提出需求; 系统分析员分析需求, 确定系统的基本功能、数据库结构和应用程序的设计等; 应用程序员根

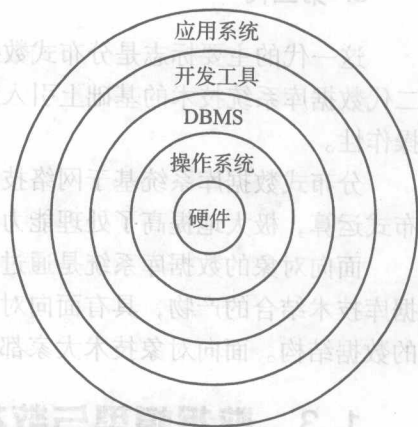


图 1.5 DBS 层次图



据系统的功能需求设计、编写和调试应用程序；数据库管理员（DBA）则参与整个应用程序的开发过程，监视和控制系统的运行并将运行状况反馈给设计开发人员以改进系统性能，在数据库系统中的地位很重要。

### 1.2.4 数据库技术的发展历程

数据库技术起源于20世纪60年代，是目前应用最广泛的计算机技术，日常生活中无处不在，包括商场、银行和人事管理等部门。

数据库技术的发展是以数据模型的发展为主线的，并据此划分为三代。

#### 1. 第一代

20世纪70年代的层次数据库系统与网状数据库系统为第一代。其确立了支持三级抽象模式的体系结构，数据定义语言与数据操纵语言相对独立，用存取路径来表示数据之间的联系。

#### 2. 第二代

美国的 E. F. Codd 在1970年提出的关系模型奠定了关系数据库的理论基础。关系模型概念简单、清晰，数据间的联系用关系来表示，以关系代数为理论基础，数据独立性强，数据库语言采用说明性语言，编程容易。因为关系数据库具有巨大的优势，因而从1974年出现第一个关系数据库系统（RDBMS）开始，到20世纪80年代后几乎所有新开发的数据库系统都是关系型的，典型的有 FoxBase（Visual FoxPro 早期版本）、Access、Sysbase、Oracle 和 SQL Server 等。关系数据库的出现极大地促进了计算机的推广应用。现今，人们的生活、工作都要与各种各样的数据库打交道。

#### 3. 第三代

这一代的主要标志是分布式数据库系统和面向对象数据库系统的出现。它们在继承了第二代数据库系统技术的基础上引入新技术，具有良好的可移植性、可连接性、可扩充性及互操作性。

分布式数据库系统基于网络技术，数据资源可以分布在各个结点上，数据处理可进行分布式运算，极大地提高了处理能力。第5章中有详细介绍。

面向对象的数据库系统是通过对象和类的概念建立的数据库模型，是面向对象技术与数据库技术结合的产物，具有面向对象技术的封装性和继承性等特点，能完整地描述现实世界的数据结构。面向对象技术大家都应该接触过，此处不再重复。

## 1.3 数据模型与数据模式

### 1.3.1 数据模型

数据是用来描述现实世界的，数据模型则是对数据特征的抽象，它描述数据的基本结构及其相互间的关系，以及定义在数据上的操作。

数据模型既要比较自然地模拟现实世界以易于为人们所理解（面向现实世界、面向用

户), 又要便于在计算机上实现 (面向实现、面向计算机)。但这两方面的要求往往是矛盾的, 这一矛盾目前无法完全解决, 但可以采用不同的模型适用于不同的对象这一方法。正如程序设计语言, 人们用高级语言编写程序, 写好的程序通过解释、编译成计算机能识别的机器语言, 从而实现了既面向机器又面向用户的目的。数据模型也采用类似的方法, 即根据不同的使用对象和应用目的, 采用多级数据模型。

人们按照数据模型与用户、机器距离的远近将其分为三级。

### 1. 概念数据模型

概念数据模型是一种独立于计算机系统的模型, 它不涉及信息在系统中的表示, 只是用来描述某个特定组织的概念化结构, 强调语义表达功能。即概念数据模型面向现实、面向用户, 而与 DBMS 无关。这样, 数据库设计人员在开发的起始阶段只需正确了解和描述现实世界即可。

最典型、最常用的概念数据模型为实体—联系模型 (E-R 模型或 E-R 图), 如图 1.6 所示。实体—联系模型直接从现实世界中抽象出实体类型及实体间的联系, 其包含实体、属性和实体间的联系三要素。

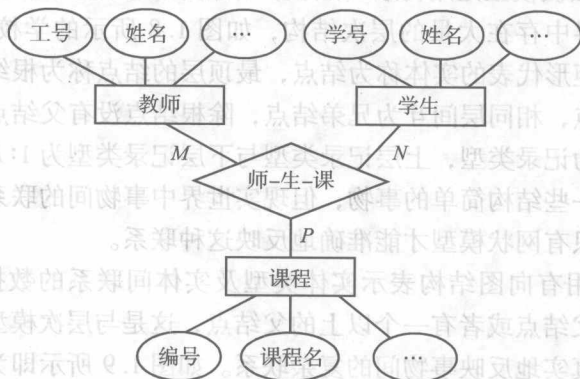


图 1.6 教师、学生、课程的 E-R 图

实体对应于现实世界中可区别的客观对象或抽象概念, 学生、教师等都是实体。属性是实体特征的抽象, 如学生的属性有姓名、年龄、性别等。实体间的联系对应于现实世界中各种客观对象或抽象概念之间的联系。如“学生、教师、课程”这 3 个实体间存在“教师教学生学习某课程”的联系。

E-R 图中用矩形框表示实体类型; 椭圆框表示实体类型和联系类型的属性; 菱形框表示实体间的联系。如图 1.6 所示就是一个简化的反映学生、教师、课程实体及其联系的 E-R 图, 图中  $M$ 、 $N$ 、 $P$  即表示实体间的联系是多对多。

由于现实世界中一个组织体内部包含多个实体, 因此从哪个角度选择实体、选择哪些实体是首先要考虑的问题; 其次要确定描述对象的属性, 并不是所有属性都要罗列; 再则现实世界中实体间充满着各种复杂的联系, 如图 1.7 所示, 有一对一 (1:1), 如某个班主任与所管理的班级; 一对多 (1:N), 如一个教师担任几门课程; 多对多 ( $M:N$ ), 如  $M$  个教师教  $N$  个学生。为了便于在关系数据库中实现, 我们应将一对多、多对多的关系转化为一对一的关系, 有关内容将在后面章节中逐步学习。

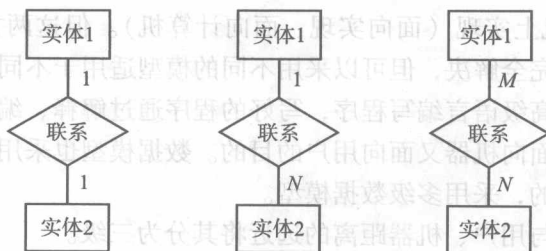


图 1.7 实体间关系的种类

E-R 模型接近于人的思维，与计算机无关，用户容易理解接受。但 E-R 模型只能说明实体间语义的联系，不能说明详细的数据结构，无法在计算机上实现。

## 2. 逻辑数据模型

数据模型第二级为逻辑数据模型，其既要面向用户又面向实现。用概念数据模型表示的数据必须转化为逻辑数据模型表示的数据才能在 DBMS 中实现。

典型的逻辑数据模型包括层次模型、网状模型和关系模型。

(1) 层次模型：层次模型是用树形（层次）结构表示实体间联系的数据模型，与目录树结构类似。现实世界中存在大量的层次结构，如图 1.8 所示的学校行政系统即为层次结构。层次结构中每个矩形代表的实体称为结点，最顶层的结点称为根结点，最底层的为叶结点，上下层为父子结点，相同层间互为兄弟结点，除根结点没有父结点外，其他结点有且只有一个父结点。结点为记录类型，上层记录类型与下层记录类型为  $1:N$  的联系。

层次模型能描述一些结构简单的事物，但现实世界中事物间的联系往往很复杂，不是层次模型所能描述的，只有网状模型才能准确地反映这种联系。

(2) 网状模型：用有向图结构表示实体类型及实体间联系的数据模型称为网状模型。网状模型允许结点无父结点或者有一个以上的父结点，这是与层次模型的根本区别，正是由于这一特点，才能更真实地反映事物间的复杂联系。如图 1.9 所示即为网状模型，图中描述了学生、教师、课程等多个实体间复杂的联系。从图中可以看出网状模型可描述多对多的联系，即由两个一对多的联系转换成  $M:N$  联系，从而能描述比层次模型 ( $1:N$ ) 更复杂的联系。

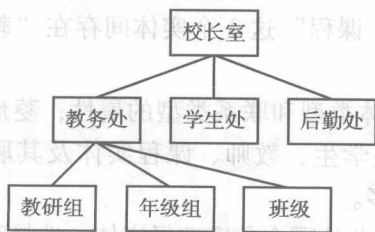


图 1.8 层次模型

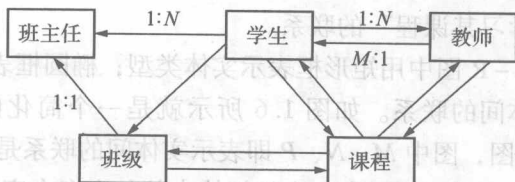


图 1.9 网状模型

网状模型的优点为：

- 能够直接地描述现实世界；
- 具有良好的性能，存取效率高。

网状模型的缺点为：

- 结构复杂；



• 语言复杂, 用户不易掌握和使用。

(3) 关系模型: 关系模型是由若干个关系模式组成的集合, 其主要特征是用二维表格结构表达实体集, 用键表示实体间的联系。每个关系实际上就是一张表格, 表 1.1、表 1.2 和表 1.3 就是 3 个关系。关系中的一行称为一个元组 (表中的记录), 一列称为属性 (表中的字段), 列的名称为属性名 (表中的字段名)。

关系模型既可以反映一对一的关系, 也可以通过转换反映一对多和多对多的关系, 所以能描述复杂的现实世界。同时, 由于用简单的二维表格来描述, 符合人的习惯, 简单易懂, 因而被广泛使用, 成为商品化数据库系统所采用的主流数据模型。

### 3. 物理数据模型

逻辑数据模型反映数据的逻辑结构而不反映数据的存储结构, 反映数据存储结构的数据模型称为物理数据模型。其不但与 DBMS 有关, 而且与操作系统和硬件有关, 即面向机器。对于一般用户来说, 物理数据模型是透明的, 即不需要了解。

概念数据模型只用于数据库的设计, 逻辑数据模型和物理数据模型用于 DBMS 的实现, 这种分级使得在设计开发时, 不同层次的开发人员可就某一方面展开讨论, 有时甚至不需要考虑物理数据模型。

## 1.3.2 数据模式

数据模型是描述数据的手段, 数据模式则是用给定的数据模型对具体数据的描述。例如用关系模型来描述“学生”得到关系模式, 如用层次模型描述, 则得到的是层次模式。

数据模式是对数据库中全体数据的逻辑结构和特征的描述, 或者说是数据库所有数据元素类型的一个结构图。在程序设计语言中, 数据有型和值之分, 型是该数据所属数据类型的说明, 而值则是型的一个实例, 例如, 整数是型, 87 是其一个值。用数据模型描述数据时也有型与值之分, 对某一类数据的结构、联系和约束的描述是型的描述, 称为数据模式, 模式的一个具体值称为实例。同一数据模式下可以有多个实例, 也就是说, 模式是相对稳定的, 实例是相对变动的。如表 1.1 可用关系模式“学生 (学号、姓名、性别、年龄、出生日期、入学成绩)”来描述, 其中每一行具体数据为一实例。

当前流行的 DBMS 种类很多, 它们基于不同的数据模型, 使用不同的数据库语言, 但都采用相同的体系结构——三级模式和两级映像。

### 1. 三级模式结构

三级模式结构是指数据库系统的体系结构分为三级: 内部级 (内模式)、概念级 (模式) 和外部级 (外模式), 如图 1.10 所示。

(1) 内模式: 也称存储模式, 是数据物理结构和存储方式的描述, 是数据在数据库内部的表示方式。内模式依据全局逻辑结构对数据进行适当的组织和空间分配, 以优化运行效率。内模式对一般用户是透明的, 但它的设计直接影响数据库的性能, 因而, 对内模式有一定的了解是必要的。

一个数据库只有一个内模式。

(2) 模式: 即概念模式又称为逻辑模式, 因为它是对数据库中全体数据的逻辑结构和特征的描述。按内模式存储的数据通过 DBMS 的映射形成逻辑级上的视图, 此为所有用户





的公共数据视图，对此视图的操作也通过 DBMS 的映射反映到内模式，如在 DBMS 中我们看到的的基本表即为概念模式这一级。

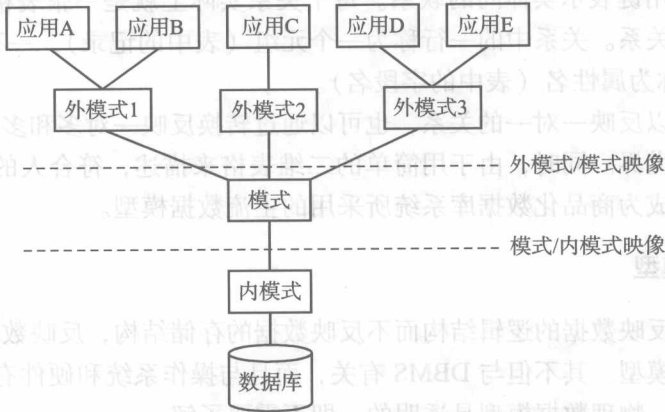


图 1.10 数据库系统的三级模式结构

一个数据库只有一个概念模式。

(3) 外模式：又称为子模式或用户模式，是用户与数据库的接口，是对用户用到的那部分数据的逻辑结构和特征的描述。因为用户对数据的需求不同，同时也不是所有数据都对用户公开，所以将概念模式中的一部分形成子集作为特定用户的数据视图。从中可以看出，一个概念模式可以衍生出多个子模式而适应用户的不同需求。

下面简单介绍一下这三级模式与数据模型间的关系。内模式是用物理数据模型对数据的描述，概念模式与外模式都是用逻辑数据模型对数据的描述，区别在于一个是全体，一个是局部。而概念模型是面向用户的，与机内的表示无关，后面章节会对其进行深入具体的介绍。

### 2. 两级映像与数据的独立性

数据按外模式的描述提供给用户，按内模式的描述存储在磁盘中，而概念模式提供了连接这两级的相对稳定的中间点，并使得任何一级的改变都不受另一级的牵制。也就是说在三级模式中间 DBMS 提供了两级映像，使数据具有独立性。

(1) 外模式/模式映像：外模式/模式映像用于定义外模式和概念模式间的对应性，在外模式中描述。模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构，每个模式都可有多个外模式，每个外模式对应于一个应用，也就是说应用程序是在外模式的基础上编制的。当模式需要变化时，只要不影响到应用程序所用到的外模式，就只需改变模式而不改变外模式，这样便保证了数据与程序的逻辑独立性。

(2) 模式/内模式映像：模式/内模式映像用于定义概念模式和内模式之间的对应性，一般在内模式中描述。模式描述数据库的全局逻辑结构，内模式描述了数据的存储结构，模式/内模式映像定义了它们之间的对应关系。当数据库的存储结构要进行修改时，模式/内模式映像能作出相应的变化，而不是去改变模式，从而不影响应用程序，这样就保证了数据与程序的物理独立性。

可以认为二级映像相当于两个缓冲，在其能力范围内使得对内模式的修改不影响模式，对模式的修改不影响外模式，从而保证了程序与数据的相对独立性。且映射是由 DBMS 提