



新世纪高职高专实用规划教材

• 计算机系列

SQL Server 2000 基础教程

SQL Server 2000 JICHU JIAOCHENG

刘勇 周学军 编著



清华大学出版社

内 容 简 介

本书是作者针对高职高专教学的特点,根据多年从事计算机职业教育的经验编写而成的。编写中注意引入微软认证内容,以突出高职教育的特点。全书主要内容有:数据库基础知识,SQL Server 2000 的安装方法和安装要求,数据库操作,数据表操作,访问权限和管理权限,SQL 查询语言和 Transact-SQL 程序设计基础,索引,视图,数据备份与数据恢复,数据转换,自动化工作,数据库复制,开发 SQL Server 应用程序,使用 ASP 开发 Web 数据库。

本书可作为高职高专计算机专业学生的教材,也可以作为希望在操作技能上得到提高的本科生学习数据库课程的参考书。

版权所有,翻印必究。举报电话:010-62782989 13501256678 13801310933

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

本书防伪标签采用清华大学核研院专有核径迹膜防伪技术,用户可通过在图案表面涂抹清水,图案消失,水干后图案复现;或将表面膜揭下,放在白纸上用彩笔涂抹,图案在白纸上再现的方法识别真伪。

图书在版编目(CIP)数据

SQL Server 2000 基础教程/刘勇,周学军编著. —北京:清华大学出版社,2005.2

(新世纪高职高专实用规划教材 计算机系列)

ISBN 7-302-10415-8

I. S… II. ①刘…②周… III. 关系数据库—数据库管理系统, SQL Server 2000—高等学校:技术学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2005)第 007633 号

出 版 者:清华大学出版社 地 址:北京清华大学学研大厦

<http://www.tup.com.cn> 邮 编:100084

社 总 机:010-62770175 客 户 服 务:010-62776969

组稿编辑:刘建龙

文稿编辑:杨作梅

封面设计:陈刘源

排版人员:李欣

印 刷 者:北京市清华园胶印厂

装 订 者:北京市密云县京文制装订厂

发 行 者:新华书店总店北京发行所

开 本:185×260 印 张:16 字 数:376 千字

版 次:2005 年 2 月第 1 版 2005 年 2 月第 1 次印刷

书 号:ISBN 7-302-10415-8/TP·7073

印 数:1~5000

定 价:22.00 元

本书如存在文字不清、漏印以及缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。联系电话:(010)62770175-3103 或(010)62795704

《新世纪高职高专实用规划教材》序

编写目的

目前,随着教育的不断深入,高等职业教育发展迅速,进入到一个新的历史阶段。学校规模之大,数量之众,专业设置之广,办学条件之好和招生人数之多,都大大超过了历史上任何一个时期。然而,作为高职院校核心建设项目之一的教材建设,却远远滞后于高等职业教育发展的步伐,以至于许多高职院校的学生缺乏适用的教材,这势必影响高职院校的教育质量,也不利于高职教育的进一步发展。

目前,高职教材建设面临着新的契机和挑战:

(1) 高等职业教育发展迅猛,相应教材在编写、出版等环节需要在保证质量的前提下加快步伐,跟上节奏。

(2) 新型人才的需求,对教材提出了更高的要求,即教材要充分体现科学性、先进性和实用性。

(3) 高职高专教育自身的特点是强调学生的实践能力和动手能力,教材的取材和内容设置必须满足不断发展的教学需求,突出理论和实践的紧密结合。

有鉴于此,清华大学出版社在相关主管部门的大力支持下,组织部分高等职业技术学院的优秀教师以及相关行业的工程师,推出了一系列切合当前教育改革需要的高质量的面向就业的职业技术实用型教材。

系列教材

本系列教材主要涵盖以下领域:

- 计算机基础及其应用
- 计算机网络
- 计算机图形图像处理与多媒体
- 电子商务
- 计算机编程
- 电子电工
- 机械
- 数控技术及模具设计
- 土木建筑
- 经济与管理
- 金融与保险

另外,系列教材还包括大学英语、大学语文、高等数学、大学物理、大学生心理健康等基础教材。所有教材都有相关的配套用书,如实训教材、辅导教材、习题集等。

教材特点

为了完善高等职业技术教育的教材体系,全面提高学生的动手能力、实践能力和职业素质,特意聘请有实践经验的高级工程师参与系列教材的编写,采用了一线工程技术人员与在校教师联合编写的模式,使课堂教学与实际操作紧密结合。本系列丛书的特点如下:

(1) 打破以往教科书的编写套路,在兼顾基础知识的同时,强调实用性和可操作性。

(2) 突出概念和应用,相关课程配有上机指导及习题,帮助读者对所学内容进行总结和提高。

(3) 设计了“注意”、“提示”、“技巧”等带有醒目标记的特色段落,使读者更容易得到有益的提示与应用技巧。

(4) 增加了全新的、实用的内容和知识点,并采取由浅入深、循序渐进、层次清楚、步骤详尽的写作方式,突出实践技能和动手能力。

读者定位

本系列教材针对职业教育,主要面向高职高专院校,同时也适用于同等学历的职业教育和继续教育。本丛书以三年制高职为主,同时也适用于两年制高职。

本系列教材的编写和出版是高职教育办学体制和运作机制改革的产物,在后期的推广使用过程中将紧紧跟随职业技术教育发展的步伐,不断吸取新型办学模式、课程改革的思路和方法,为促进职业培训和继续教育的社会需求奉献我们的力量。

我们希望,通过本系列教材的编写和推广应用,不仅有利于提高职业技术教育的整体水平,而且有助于加快改进职业技术教育的办学模式、课程体系和教学培训方法,形成具有特色的职业技术教育的新体系。

新世纪高职高专实用规划教材

计算机系列编委会

主任 吴文虎

副主任 边奠英

委员 (以姓氏笔画为序)

万国平 王洪发 王庆延 邓安远

孙 辉 孙远光 朱华生 朱烈民

李 萍 杨 龙 杨扶国 邱 力

易镜荣 苑鸿骥 柏万里 胡剑锋

黄 俭 黄学光 黄晓敏 曾 斌

熊中侃 廖乔其 蔡泽光 魏 明

前 言

数据库技术是计算机科学中最活跃、应用最广泛的分支之一，在各类高职高专院校中，很多专业开设了数据库课程。然而，以前的数据库课程主要介绍小型数据库，随着数据库技术的不断发展，特别是网络的飞速发展，传统的小型数据库逐渐难以满足各单位对数据管理的要求，中、大型数据库得到越来越多的重视。SQL Server 2000 是一种既可用于中小型单位、又可用于大型企业的数据库。近年来，已有不少关于这方面的书问世，但多数是面向计算机专业本科生的。本书是根据高职高专院校的培养目标，注意引入微软认证内容，突出高职教育的特点，同时，注意使教学内容与实验实习环节相结合。在确定编写内容时，既注意了数据库管理人员的要求，也考虑了数据库开发人员的要求。为了使本书便于学生自学，编写过程中没有重点介绍过于繁杂的语句参数。该书可作为计算机类专业高职高专学生的教学用书，也可以作为数据库管理和开发人员的参考书。

本书力求帮助读者从初识 SQL Server 2000，成长为具有管理和维护 SQL Server 2000 服务器的技能，并能针对应用，进行简单编程。由于本书汲取了许多优秀书籍和国内外关于这门技术的精华，还融入了作者多年的实践经验，列举了大量具体实例，因此读者在学习时可以举一反三，在学习 SQL Server 2000 基础知识的同时，了解 SQL Server 2000 在实际工作中的应用形式和方法。

本书共分 14 章，通过大量的讲解和实例，使读者能够循序渐进地学习到 SQL Server 2000 的基础知识、技术要点和程序开发知识。如果读者以前学过数据库理论知识则能够很容易地掌握本书的内容。

本书主要由江西交通职业技术学院刘勇和江西现代职业技术学院周学军编著，参加本书编写的还有刘造新、黄侃、宋荣、肖辉辉、陆亚维、王琦、黄晓敏、熊慧芳、余艳、武国祥和周西柳等。

由于编者水平所限，不当之处在所难免，恳请各位专家和读者朋友指正。E-MAIL: ly@jxjtxy.com。

目 录

第 1 章 数据库基础知识1	2.5 习题..... 24
1.1 数据库系统.....1	第 3 章 数据库操作 25
1.1.1 数据、信息和数据库.....1	3.1 准备使用 SQL Server 2000..... 25
1.1.2 数据库系统.....2	3.1.1 查看安装结果..... 25
1.2 数据模型.....3	3.1.2 启动、停止、暂停和修改 SQL Server 2000 服务..... 28
1.2.1 研究现实世界的方法.....3	3.1.3 使用 OSQL、SQL Server 企业管理器和 SQL 查询分析器..... 32
1.2.2 数据模型.....3	3.1.4 数据库结构..... 36
1.2.3 关系型数据库.....4	3.2 创建和配置用户数据库..... 38
1.3 关系型数据库的设计.....7	3.2.1 创建用户数据库..... 38
1.3.1 数据库设计过程.....7	3.2.2 设置数据库选项..... 44
1.3.2 E-R 方法.....8	3.2.3 管理用户数据库大小..... 47
1.4 习题.....8	3.3 将数据库文件存放在多个磁盘上..... 50
第 2 章 MS SQL Server 2000 的 安装方法和安装要求9	3.3.1 RAID 简介..... 50
2.1 SQL Server 2000 的各个版本.....9	3.3.2 文件组..... 51
2.2 安装准备.....10	3.3.3 分离数据库与附加数据库.... 51
2.2.1 安装要求.....10	3.4 习题..... 54
2.2.2 SQL Server 2000 的 安装配置选项.....13	第 4 章 数据表操作 56
2.3 SQL Server 2000 安装程序.....14	4.1 数据类型..... 56
2.3.1 运行安装程序.....14	4.1.1 系统数据类型..... 56
2.3.2 选择安装方式.....16	4.1.2 用户定义的数据类型..... 61
2.3.3 选择安装类型.....16	4.2 数据表的建立..... 64
2.3.4 选择服务的登录账户、 身份验证模式和许可模式.....18	4.2.1 使用企业管理器创建..... 64
2.3.5 确定排序规则和 选择网络协议.....20	4.2.2 使用 CREATE TABLE 语句创建..... 67
2.4 安装命名实例和无人值守安装.....21	4.3 修改表的结构和删除表..... 68
2.4.1 安装 SQL Server 2000 命名实例.....21	4.3.1 使用企业管理器修改表..... 68
2.4.2 无人值守安装.....22	4.3.2 使用企业管理器删除表..... 69
	4.4 表的关联..... 70

4.4.1 建立表的关联.....	70	6.3 SELECT 高级查询.....	106
4.4.2 删除表的关联.....	72	6.3.1 数据汇总.....	106
4.5 记录的添加和修改.....	72	6.3.2 联接查询.....	109
4.5.1 记录的添加和修改.....	73	6.3.3 子查询.....	112
4.5.2 记录的删除.....	73	6.4 习题.....	113
4.6 习题.....	74	第 7 章 索引	114
第 5 章 访问权限和管理权限	76	7.1 索引及其分类.....	114
5.1 身份验证过程与授权过程.....	76	7.2 创建索引.....	115
5.1.1 身份验证模式.....	76	7.2.1 聚集索引和非聚集索引.....	115
5.1.2 Windows 和 SQL Server		7.2.2 主键约束和惟一性约束.....	115
身份验证的比较.....	77	7.2.3 使用索引创建向导.....	115
5.1.3 设置验证模式.....	78	7.2.4 使用索引管理器.....	118
5.2 创建和管理登录.....	80	7.2.5 使用 CREATE	
5.2.1 创建和管理		INDEX 语句.....	119
服务器登录账号.....	80	7.3 索引的维护和使用.....	120
5.2.2 创建和管理		7.3.1 查看索引.....	120
用户登录账号.....	83	7.3.2 删除索引.....	121
5.3 数据库特定的权限.....	84	7.3.3 在 SELECT 语句中	
5.3.1 权限的分类.....	84	使用索引.....	122
5.3.2 管理对象权限.....	85	7.4 习题.....	122
5.3.3 管理语句权限.....	88	第 8 章 视图的创建和使用	123
5.4 角色的使用.....	88	8.1 概述.....	123
5.4.1 服务器角色.....	89	8.1.1 视图的概念.....	123
5.4.2 数据库角色.....	89	8.1.2 视图的优点.....	123
5.5 习题.....	94	8.2 创建视图.....	123
第 6 章 SQL 查询语言和		8.2.1 使用向导创建视图.....	123
Transact-SQL 基础	95	8.2.2 使用企业管理器	
6.1 SQL 语言.....	95	创建视图.....	126
6.1.1 SQL 语言的概述.....	95	8.2.3 使用 CREATE VIEW	
6.1.2 SQL 语言分类.....	96	命令创建视图.....	129
6.2 Transact-SQL 基础.....	96	8.3 使用视图.....	129
6.2.1 标识符.....	96	8.3.1 通过视图添加数据.....	129
6.2.2 数据查询.....	98	8.3.2 通过视图更改数据.....	130
6.2.3 数据库的修改与删除.....	100	8.3.3 通过视图删除数据.....	130
6.2.4 数据的插入、		8.3.4 使用视图进行数据查询.....	130
修改和删除.....	103	8.4 视图的维护.....	131
6.2.5 表的修改与删除.....	105	8.4.1 修改视图.....	131

8.4.2	视图更名.....	131	第 11 章 自动化工作.....	163
8.4.3	视图信息的查询修改.....	131	11.1 SQL Server 代理.....	163
8.4.4	视图的删除.....	132	11.1.1 自动化管理执行元素.....	164
8.5	习题.....	133	11.1.2 设置 SQL Server 代理.....	164
第 9 章 数据库备份与数据库恢复.....	134	11.2 定义操作员.....	167	
9.1 数据库备份及其备份类型.....	134	11.2.1 利用企业管理器 创建操作员.....	167	
9.1.1 数据库备份概述.....	134	11.2.2 使用 Transact-SQL 语言 创建和管理操作员.....	168	
9.1.2 数据库备份的类型.....	134	11.3 配置警报.....	168	
9.2 备份数据库、文件、文件组 和事务处理日志.....	135	11.3.1 定义错误信息.....	169	
9.2.1 备份设备.....	135	11.3.2 设置警报.....	170	
9.2.2 创建数据库备份.....	136	11.3.3 使用 Transact-SQL 语言 创建和管理警报.....	172	
9.3 恢复数据库.....	139	11.4 创建作业.....	172	
9.3.1 数据库的修复.....	139	11.4.1 新建作业.....	172	
9.3.2 恢复数据库.....	141	11.4.2 作业分类.....	175	
9.4 习题.....	144	11.4.3 使用 Transact-SQL 语言 创建和管理作业.....	175	
第 10 章 数据转换.....	145	11.5 创建数据库维护计划.....	175	
10.1 使用“DTS 导入/导出 向导”工具.....	145	11.5.1 数据库维护.....	175	
10.1.1 使用“DTS 导入/ 导出向导”工具导入 EXCEL 工作簿数据.....	145	11.5.2 维护工作自动化.....	176	
10.1.2 使用“DTS 导入/导出 向导”工具导出 数据到 VFP.....	148	11.6 习题.....	179	
10.1.3 利用“DTS 导入/导出 向导”将 VFP 数据 导出到 Excel 工作簿.....	151	第 12 章 数据库复制.....	180	
10.2 使用 DTS 设计器.....	151	12.1 复制简介.....	180	
10.2.1 DTS 设计器简介.....	151	12.1.1 SQL Server 复制服务.....	180	
10.2.2 创建一个数据传输和 转换数据包.....	153	12.1.2 SQL Server 复制模型.....	182	
10.2.3 执行数据包.....	156	12.1.3 SQL Server 复制类型.....	184	
10.3 保存、打开 DTS 数据包.....	160	12.1.4 事务的一致性.....	185	
10.3.1 保存 DTS 数据包.....	160	12.2 配置复制.....	186	
10.3.2 打开 DTS 数据包.....	160	12.2.1 配置出版服务器.....	186	
10.4 习题.....	161	12.2.2 创建出版物.....	188	
		12.3 执行复制.....	191	
		12.3.1 订阅出版物.....	191	
		12.3.2 删除复制.....	194	
		12.4 合并复制的冲突管理.....	194	
		12.4.1 数据冲突.....	194	
		12.4.2 创建和解决冲突.....	195	

12.5 习题	196	13.3 使用 Visual Basic 开发 SQL Server 应用程序	218
第 13 章 开发 SQL Server 应用程序	197	13.3.1 Visual Basic 概述	218
13.1 概述	197	13.3.2 Visual Basic 开发界面介绍	219
13.1.1 数据库应用程序 开发方法	197	13.3.3 连接数据库	220
13.1.2 ODBC 简介	197	13.3.4 显示数据的两种方式	222
13.1.3 ADO 简介	198	13.4 习题	224
13.2 使用 Delphi 开发 SQL Server 应用程序	198	第 14 章 使用 ASP 开发 Web 数据库	225
13.2.1 Delphi 概述	198	14.1 ASP 概述	225
13.2.2 Delphi 的集成开发环境	199	14.2 配置 ASP 的运行环境(IIS)	226
13.3.2 利用 Delphi 的 BDE 组件开发 SQL Server 2000 的应用程序	200	14.3 ASP 基础	229
13.2.4 利用 ADO 方式开发 SQL Server 2000 应用程序	213	14.3.1 ASP 程序结构	229
		14.3.2 ASP 内置对象	229
		14.4 使用 ASP 集成 SQL 数据库	234
		14.5 习题	239

第 1 章 数据库基础知识

开始学习 SQL Server 2000 之前,先了解一些数据库的基础知识,有利于对 SQL Server 2000 的理解和应用。本章主要介绍数据库系统的概念、数据模型和关系型数据库的基本设计方法。

1.1 数据库系统

1.1.1 数据、信息和数据库

1. 数据(Data)

数据是对客观事物特征所进行的一种抽象化、符号化表示。例如,某人身高 2 米、体重 75kg、年龄 18 岁,其中 2、75、18 就是数据。除了这些具有数值特征的数据外,还有一些不具有数值特征数据,例如,表示姓名的字符串“王小明”、“Ritney Spears”以及表示性别的字符“男”与“女”等,它们也是一种数据。

2. 信息(Information)

信息是指有一定含义的、经过加工(处理)的、对决策有价值的的数据。例如,“1998 年全国工商税收完成 8552 亿元”是一条信息,其中 1998、年、全国、工商税收、8552 和亿元等都只是数据,所以说数据可以表示信息。

3. 数据库(Database, DB)

数据库就是存储数据的仓库,是按一定组织方式存储的相互有关的数据集合,这些数据不仅彼此关联且可动态变化。例如,一个学生成绩管理系统包含“学生数据表”(见表 1.1)、“课程数据表”(表 1.2)和“选课数据表”(表 1.3)等数据。

表 1.1 学生数据表

学号	姓名	性别	出生日期	入学时间	系别
99201	王海	男	07/16/82	09/01/99	计算机
99203	李娜	女	08/23/80	09/01/99	计算机
00205	李晓	女	08/20/82	09/01/00	管理

表 1.2 课程数据表

课程编码	课程名	课时	学分
01	数学	200	5
02	计算机应用基础	140	5
03	英语	240	5
04	语文	180	4

表 1.3 选课数据表

学号	课程编码	成绩	选课时间
99201	01	89	09/01/99
99201	02	94	09/01/99
99203	01	88	09/01/99
99203	03	80	09/01/99
99205	02	98	09/01/00
99201	03	86	09/01/00

1.1.2 数据库系统

1. 数据库系统构成

数据库系统是指计算机系统中引进数据库后的系统构成，主要包括以下几方面：计算机硬件、数据库、数据库管理系统、相关软件和人员。其中数据库管理系统是一个数据库管理软件，它的主要职能是维护数据库，接受和完成用户程序或命令提出的数据增加、删除和修改等处理任务。

2. 数据库系统的特点

(1) 数据的结构化

在同一数据库中可以有许多数据文件，这些组织文件是有联系的，且在整体上服从一定的结构形式。

(2) 数据共享

共享是数据库系统的目的，也是它的重要特点。一个数据库中的数据不仅可为同一机构中的各个部门所共享，还可为不同机构、不同地域甚至不同国家的用户所共享。

(3) 数据的独立性

在早期的数据管理方式下，数据结构和应用程序相互依赖，一方的改变总要导致另一方的改变。数据库系统则较大程度地减小了这种相互依赖，在某种程度上实现了数据独立性。

(4) 可控冗余度

数据共享范围较小时，每个用户使用的数据相对较为固定，并各自管理自己的数据。随着数据的共享范围扩大，便出现大量的数据重复现象，这就是冗余。数据的过度冗余带来的问题是数据维护工作量增加大、数据的一致性差，而另一方面，保留少量重复数据又

可以提高查询效率,因此,在数据库设计时,适当控制冗余度十分重要。数据库系统具有可控冗余度的特点。

1.2 数据模型

1.2.1 研究现实世界的方法

现实世界是存在于人们头脑之外的客观世界,现实世界中的事物可以分成“对象”和“性质”两类。

我们将现实世界在人们头脑之中的反映称作观念世界,在观念世界中,客观事物称之为实体,例如人、部门、表格、物体、课程等都是实体,实体是可以明确识别的对象。实体的某一特征称为属性。例如,人有姓名、性别和年龄等属性,属性是客观事物的性质的抽象描述。

观念世界是数据库设计人员在认识现实世界中的实体及实体间联系后进行的一种抽象,这种抽象不能被计算机所接受,它必须转换成计算机能够接受的数据世界。数据世界是对观念世界的数字化。

综上所述,现实世界、观念世界和数据世界的相互关系是:现实世界 $\xrightarrow{\text{抽象}}$ 观念世界 $\xrightarrow{\text{转换}}$ 数据世界。

1.2.2 数据模型

随着现实世界抽象成观念世界再转换成数据世界,现实世界中的实体模型最后转换成数据世界中的数据模型。数据模型就是各个数据对象以及他们之间相互关系的集合。数据模型的重要任务之一是指出数据间的联系(主要是实体间的联系)。数据模型的设计方法决定了数据库的设计方法。目前主要的设计方法有三种:关系方法、层次方法和网络方法。

1. 关系模型

关系模型是出现较晚的数据模型,它以数学理论为基础构造数据模型,将每一个实体集合视作一张二维表,即关系表。例如,表 1.1 就表示一个实体集,横向一行表示“学生”实体中的一个具体的成员,称作一条记录,每条记录由若干数据项组成,其中每个数据项表示该实体的一个属性,称作字段。

在关系模型中,基本的数据结构是表格,记录之间联系是通过称作“键”的字段来体现的。例如,在表 1.1、表 1.2 和表 1.3 所示的数据模型中,带下划线的字段便是“键”,表 1.3 中含有两个分别来自表 1.1 和表 1.2 的“键”(学号和课程编码),第一条记录中“学号”键值为“99201”,“课程编码”键值为“01”,这就表示学号为“99201”的学生学习课程编码为“01”的课程的成绩为 89 分,而该学生的基本情况可从表 1.2 中查得,同样,该门课程的情况可从表 1.3 中查得。可见,学生实体与课程实体之间的联系是通过选课进行关联的。

2. 层次模型

层次模型的结构是树形结构，树的节点是实体，树的枝是联系。树中只有一个节点没有父节点，这个节点称为树根，其余的每个节点有且只有一个父节点，另外，上一层实体和下一层实体之间的联系是一对多的关系。

3. 网状模型

数据的网状模型是以实体为节点的网络结构，其特点是一个子节点可以有多个父节点，并可以有多个节点没有父节点，此外，两个节点之间可以有两种或多种联系。

1.2.3 关系型数据库

1. 关系型数据库的含义

关系型数据库是指一些相关的表和其他数据库对象的集合，其含义可以从以下三方面把握：

- 在关系数据库中，信息存放在二维表格中，一个关系数据库包含多个数据表格(简称为数据表或表)，每个表包含行(记录)和列(字段)。通常，一个数据库有多个表。
- 数据库所包含的表之间是有关联的，关联由主键和外键所体现的参照关系实现。
- 数据库不仅包含表，还包含其他的数据库对象，例如视图、存储过程和索引等。

关系型数据库最大的特点是用二维表格表示实体，将具有相同属性的数据独立地存储在一个表格中。对任何一个表可以新增、删除和修改表中数据，而不会影响表中其他数据。

在层次模型和网状模型中，联系是用指针实现的，且一次查询只能访问数据库中的一条记录，存取效率不高。关系模型不用指针导航数据，而是用键来建立表之间的关联，其表简单，用户易懂，只需用简单的查询语句就可以对数据库进行操作，并不涉及存储结构、访问技术等细节。关系模型是数学化的模型。由于把表看成一个集合，因此集合论、数理逻辑等知识可引入到关系模型中。SQL 语言是关系数据库的代表性语言，并已得到广泛应用。当前大部分数据库都使用关系型数据库模型。

2. 关系型数据库与表

关系数据库由多个表以及其他数据库对象组成，表之间因为某些字段的相关性而产生关联。如图 1.1 所示，选课表通过“学号”与学生表相关联、同时通过“课程编码”与课程表相关联。

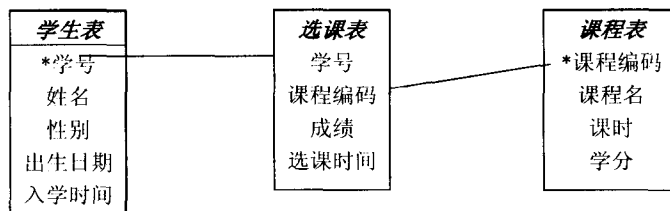


图 1.1 表与表之间的关联

在表 1.1、表 1.2 和表 1.3 所示的例子中，如果要查询学生的成绩，可以通过以上不同表的字段间的关联，形成如表 1.4 所示的查询结果。

表 1.4 学生成绩查询结果

学号	姓名	性别	系别	课程编码	课程名	课时	成绩
99201	王海	男	计算机	01	数学	200	89
99201	王海	男	计算机	02	计算机应用基础	140	94
99203	李娜	女	计算机	01	数学	200	88
99203	李娜	女	计算机	03	英语	240	80

将表 1.4 所示结果与表 1.1、表 1.2 和表 1.3 比较，可以看出，表 1.4 中数据重复较多，故采用图 1.1 所示的结构建立数据库，既可以节省大量的存储空间，又可以有效防止因重复数据带来的重复输入产生的错误，还方便了数据的修改。

在关系数据库的设计时，一般都要对所有需要的字段大致分类，将同一个表中重复的字段挑选出来组成新的表，以减少数据的重复现象。

3. 表的关联分类

在关系型数据库中，关联可以分成如下三类。

(1) 一对一关联

如果表 A 的一条记录至多与表 B 的一条记录相关联，反之亦然，则称表 A 与表 B 的联系为一对一关联，记为“1:1”。

一对一关联的情形较少，因为具有这种关联的两个表合并为一个表后不会增加数据的冗余度。

(2) 一对多关联

如果表 A 中每条记录可以与表 B 中任意多条记录相关联，而表 B 中每条记录至多与表 B 中的一条记录相关联，则称表 A 与表 B 的联系为一对多关联，记为“1:N”。

一对多关联的情形较常见，例如，一个专业有多名学生、一个系有多名老师等。

(3) 多对多关联

如果表 A 的一条记录可以与表 B 中任意多条记录相关联，反之亦然，则称表 A 与表 B 的联系为多对多关联，记为“M:N”。

多对多关联的例子也很多，例如，一名学生可以选修多门课程，而一门课程可以有多个同学选修。

4. 表的主键和外键

为了实现如图 1.1 所示的结构，需要在数据表中定义用于在表间建立关联的键。这便是表 1.4 中带星号的字段。

(1) 键

键是能惟一标识每个记录的字段或字段集。通常，采用学号、身份证号和课程编码等字段作为相应表的键，因为这类编号本身具有不重复的特点，而使用姓名、课程名等作为键的方法常常行不通，因为姓名、课程名等经常会出现重名。有时，也可以将若干个字段(即

字段集)定义为一个键,如表 1.3 中,可以将{学号,课程编码}定义为该表的键(假设同一学生重复选修同一门课程的情况不存在)。

(2) 主键

主键是恰好能惟一标识每个记录的字段或字段集。当一个字段集被定义成主键时,该字段集既要能惟一标识每个记录,又不能含有多余的字段。例如在表 1.3 中,字段集{学号,课程编码,选课时间}虽然可以惟一标识每个记录,但它包含了一个多余的字段“选课时间”,故不能用作主键。

主键实施实体完整性,故要求每个表必须有且仅有一个主键,即表中的主键是惟一的。

(3) 外键

若表 A 的主键出现在另一个表 B 的字段中,则该主键在表 B 中被称作外键。如图 1.1 的示例中,选课表中的学号和课程编码分别为学生表和课程表中的主键,故都是选课表的外键。

使用外键的主要目的在于

- 提供表之间的连接。例如,通过学号可以将学生表与选课表相连接。
- 可以根据外键的值检查输入数据的合法性。例如,在输入选课数据时,应保证输入的学号和课程编码分别是存在于学生表和课程表中的,否则,数据库可以拒绝接受。

5. 数据完整性

数据完整性用来确保数据库中的数据的正确性和可靠性。例如,数据库中某个表的数据更新后,所有与此相关的数据都要相应更新。

(1) 实体完整性

实体完整性是为了保证表中的数据惟一。实体完整性是由主键来实现的,主键的取值必须惟一,并且不能为空。

(2) 域完整性

域完整性可以保证数据的取值在有效的范围内。例如,可以限制性别字段的取值范围为{男,女,NULL},其中 NULL 表示空值,表示其值不确定。

(3) 参照完整性

参照完整性用于确保相关联的表间的数据保持一致,避免因一个表的数据修改,导致另一个表的相关数据失效。参照完整性一般是通过主键和外键来实现的,具体有如下两种方式:

- 级联 更改主键值时,所有外键值也随之改变;在主键所在表中删除记录时,外键所在的所有表中的相应记录将自动删除。
- 限制 若外键表中有相关记录,则不允许更改相关的主键值,也不允许删除该主键值所在的记录;若主键没有相应的值,则外键所在表中不能添加相应的记录。

1.3 关系型数据库的设计

1.3.1 数据库设计过程

在以前的数据管理中,数据是从属于程序的,数据文件的设计通常是应用程序设计的一部分。在数据库系统中,数据由 DBMS(数据库管理系统 Data Base Management System, DBMS)进行独立的管理,对程序的依赖性大为减少,而数据库的设计也逐渐成为一项独立的开发活动。

通常,一个数据库的设计都要经历需求分析、概念设计、逻辑设计和物理设计几个设计段。

1. 需求分析

需求分析就是对现实世界要处理的对象进行详细调查,在了解原系统的概况、确定新系统功能的过程中,获得用户对数据库的数据要求、功能要求、安全性和完整性要求。该阶段要求分析人员将系统需求写成用户和设计人员都能接受的说明书。

2. 概念设计

概念设计是将需求说明书中关于数据的需求,综合为一个统一的概念模型。概念模型是表达概念设计结果的工具,是设计人员对系统的抽象和概括,它能表达用户的要求,且独立于支持数据库的数据库管理系统和硬件系统。

概念设计的任务一般可分为三步来完成:首先进行数据抽象,设计局部概念模型;然后将局部概念模型综合成全局概念模型;最后是评审。在概念设计中,一种称作实体联系模型(Entity Relationship Model, ER)的方法被广泛采用。ER 直接从现实世界中抽象出实体及实体间联系,然后用实体联系图(ER 图)表示数据模型。设计 ER 图的方法称为 ER 方法。我们将在 1.3.2 小节具体介绍 ER 方法。

通常,一个数据库系统有多个用户,其中各个用户对数据的观点可能不一样,数据处理需求也可以不同。在进行数据库概念设计时,为了更好地模拟现实世界,一个有效的方法是先分别考虑各个用户的数据需求,形成若干个局部概念模型,并画出这些局部概念模型的 ER 图,然后,把这些 ER 图合并,消除冗余和可能存在的矛盾,得出系统总体的 ER 图。

3. 逻辑设计

概念设计的结果是得到一个与数据库管理系统无关的概念模型。而逻辑设计的目的是把概念设计所得到的全局 ER 模型,转换成与选用的具体机器上的数据库管理系统所支持的数据模型相符合的逻辑结构。对于关系型数据库,主要是完成表的结构和表与表之间的关联设计。

4. 物理设计

物理设计的任务是确定数据库的存储结构,主要包括确定数据库文件和索引文件的记