

高职高专精品课程成果教材

SQL Server

数据库程序设计

杭国英 主编

赠教学资源包



机械工业出版社
CHINA MACHINE PRESS

高职高专精品课程成果教材

SQL Server 数据库程序设计

主 编 杭国英

副主编 刘 健

参 编 李宇红 张 伦

车冬生 徐 爽



机械工业出版社

本书将 SQL Server 2000 和 VB.NET 结合在一起, 利用一个完整的实例贯穿全书, 介绍了如何应用 VB.NET 实现一个 SQL Server 2000 的数据库管理系统。本书内容包括: 数据库概述、VB.NET 简介、SQL Server 2000 概述、数据库与表、数据查询、视图与索引、存储过程与触发器、SQL Server 系统维护、SQL Server 代理服务、销售管理系统的设计与实现等。

本书充分考虑到高职学生的特点和社会需求, 注重理论联系实际, 注重技能培养。具有理论难度适中、叙述简洁、图文并茂、突出重点、通俗易懂等特点。本书不仅适合作为高等职业院校的教材或教学辅导书, 也可作为社会培训数据库应用高级技术人才的教材和自学教材。

本书配有教学资源包 (含电子教案、源程序、实验指导书、教学大纲等), 可登录 www.cmpbook.com 免费下载, 或联系编辑: kongxijun@163.com。

图书在版编目 (CIP) 数据

SQL Server 数据库程序设计/杭国英主编. —北京: 机械工业出版社, 2006.9

高职高专精品课程成果教材

ISBN 7-111-19919-7

I . S... II . 杭... III . 关系数据库 - 数据库管理系统, SQL Server 2005 - 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 (2006) 第 111654 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: 孔熹峻 版式设计: 冉晓华 责任校对: 李秋荣

封面设计: 马精明 责任印制: 杨 曦

北京机工印刷厂印刷

2006 年 9 月第 1 版第 1 次印刷

184mm × 260mm · 14 印张 · 342 千字

0 001—4 000 册

定价: 21.00 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) 68326294

编辑热线电话 (010) 68354423

封面无防伪标均为盗版

前 言

随着计算机技术与网络通信技术的发展,数据库技术已成为信息社会中对大量数据进行组织与管理的重要手段,是网络信息化管理系统的基础。

数据库技术是现代信息科学与技术的重要组成部分,是计算机数据处理与信息管理的核心。数据库技术研究、解决了计算机信息处理过程中大量数据有效地组织和存储的问题,实现了在数据库系统中减少数据存储冗余、数据共享、保障数据安全以及高效地检索数据和处理数据。

Microsoft SQL Server 2000 是基于客户机/服务器模式的新一代数据库管理系统,它在电子商务、数据仓库和数据库解决方案等方面起着重要作用。另一方面,.NET 的出现,提供了一种新的开发环境,在这种环境下,开发人员几乎可以开发出所有在 Windows 上运行的应用程序,而 VB.NET 已成为使用这种架构的最受欢迎的开发工具之一。因此,本书将 SQL Server 2000 和 VB.NET 结合在一起,利用一个完整的实例贯穿全书,介绍如何应用 VB.NET 实现一个 SQL Server 2000 的数据库管理系统。

本书共 10 章,内容包括数据库概述、VB.NET 简介、SQL Server 2000 概述、数据库与表、数据查询、视图与索引、存储过程与触发器、SQL Server 系统维护、SQL Server 代理服务、销售管理系统的设计与实现等。

本书的设计思路是:结合高职学生的特点,全书自始至终贯穿一个完整的实例,每章解决实例中的一个问题。第 4~7 章单独设立一节实验与指导,用来布置练习或实验,并给出指导内容,适合教师在课堂讲授后指导学生练习。本书先介绍数据库的基本概念,然后引入 VB.NET,之后将 VB.NET 和数据库结合的应用融于各章中逐步介绍,最后是一个完整的销售数据库管理系统的实现。

本书充分考虑到高职学生的特点和社会需求,注重理论联系实际,注重技能培养。具有理论难度适中、叙述简洁、图文并茂、突出重点、通俗易懂等特点。本书不仅适合作为高等职业院校的教材或教学辅导书,也可作为社会培训数据库应用高级技术人才的教材和自学教材。

为了提高学习效率和教学效果,本书配有教学资源包,内容包括:《SQL Server 数据库程序设计》教学大纲、电子教案、习题集、实验指导手册、书中所用程序源代码和综合实例(商品销售管理系统)等。可登录 www.cmpbook.com 免费下载,或联系编辑: kongxijun@163.com。

本书由杭国英任主编并统稿,刘健任副主编。参加编写工作的有杭国英(第 1 章、第 3 章),刘健(第 2 章、第 4 章、第 8 章),李宇红(第 5 章),张伦(第 6 章、第 7 章、第 9 章),车冬生、徐爽、刘健共同编写了第 10 章。

在本书的编写过程中,特别是在设计综合实例的过程中,西单友谊集团电子商务部的林亚部长、沈煜副部长给予了我们很大的帮助,使综合实例更符合实际,在此对他们表示衷心感谢!

由于编写时间仓促,难免有遗漏错误之处,恳请广大读者批评指正,并提出宝贵的意见和建议。

编 者

目 录

前言

第 1 章 数据库概述	1	第 3 章 SQL Server 2000 概述	29
1.1 数据库基本知识	1	3.1 SQL Server 2000 简介	29
1.1.1 数据库的基本概念	1	3.2 SQL Server 2000 的安装	30
1.1.2 数据管理技术的发展	2	3.2.1 SQL Server 2000 的环境需求	30
1.1.3 数据模型	4	3.2.2 SQL Server 2000 的安装步骤	31
1.1.4 数据库系统的模式结构	6	3.3 SQL Server 2000 的常用工具	34
1.2 关系数据库	7	3.3.1 SQL Server 企业管理器	35
1.2.1 关系模型与关系数据库	7	3.3.2 SQL Server 查询分析器	35
1.2.2 关系的完整性	8	3.3.3 SQL Server 服务管理器	36
1.3 关系数据库设计	9	3.4 习题	38
1.3.1 E-R 关系模型数据库设计	9	第 4 章 数据库与表	39
1.3.2 关系模型的规范化	12	4.1 数据库的创建与使用	39
1.3.3 关系数据库的关系	13	4.1.1 数据库的组成	39
1.3.4 数据库的一般设计方法	14	4.1.2 系统数据库	40
1.4 习题	15	4.1.3 创建数据库	41
第 2 章 VB.NET 简介	17	4.1.4 修改数据库	45
2.1 VS.NET 与 VB.NET 简介	17	4.1.5 删除数据库	46
2.1.1 Visual Studio.NET 简介	17	4.1.6 附加与分离数据库	47
2.1.2 VB.NET 简介	17	4.2 表的创建与使用	48
2.2 VS.NET 安装与启动	19	4.2.1 数据类型	48
2.2.1 VS.NET 的安装步骤	19	4.2.2 创建表	50
2.2.2 VS.NET 的启动	21	4.2.3 修改表	55
2.3 变量和类型	22	4.2.4 删除表	56
2.3.1 值类型	22	4.3 约束的创建与使用	57
2.3.2 引用类型	23	4.3.1 创建约束	57
2.4 面向对象语法	24	4.3.2 查看约束的定义	59
2.4.1 面向对象的概念	24	4.3.3 删除约束	59
2.4.2 面向对象的实现	25	4.4 ADO.NET 访问数据库	60
2.5 一个简单的 VB.NET 示例	26	4.4.1 ADO.NET 访问数据库的步骤	60
2.6 习题	28	4.4.2 ADO.NET 对象模型	60
		4.4.3 ADO.NET 数据连接方法	61
		4.5 实验与指导	61



实验 1: 创建销售管理数据库	61	6.3 实验与指导	117
实验 2: 创建销售库的表	62	6.4 习题	123
4.6 习题	64	第 7 章 存储过程与触发器	124
第 5 章 数据查询	66	7.1 存储过程的创建与使用	124
5.1 SQL 语言简介	66	7.1.1 存储过程概述	124
5.1.1 概述	66	7.1.2 创建存储过程	127
5.1.2 使用注释	67	7.1.3 修改存储过程	138
5.2 SELECT 语句	68	7.1.4 删除存储过程	140
5.2.1 SELECT 语句概述	68	7.2 触发器的创建与使用	140
5.2.2 简单查询	70	7.2.1 触发器概述	140
5.2.3 使用 WHERE 子句	74	7.2.2 创建触发器	142
5.2.4 使用 ORDER BY 子句	76	7.2.3 修改触发器	145
5.2.5 使用 INTO 子句	77	7.2.4 删除触发器	146
5.2.6 使用 GROUP BY 子句	78	7.3 实验与指导	148
5.2.7 使用 COMPUTE BY 子句	79	实验 1: 利用 VB.NET 调用存储过程	148
5.2.8 使用集合函数	80	实验 2: 触发器的创建	154
5.2.9 高级查询	83	7.4 习题	155
5.3 INSERT 语句	90	第 8 章 SQL Server 系统维护	156
5.3.1 INSERT 语句的语法格式	90	8.1 SQL Server 的安全性管理	156
5.3.2 INSERT 语句应用	90	8.1.1 安全性管理概述	156
5.4 UPDATE 语句	90	8.1.2 服务器的安全性管理	157
5.4.1 UPDATE 语句的语法格式	91	8.1.3 数据库的安全性管理	159
5.4.2 UPDATE 语句应用	91	8.1.4 表的安全性管理	164
5.5 DELETE 语句	92	8.2 备份与恢复	165
5.5.1 DELETE 语句的语法格式	92	8.2.1 数据库的备份	165
5.5.2 DELETE 语句应用	92	8.2.2 数据库的恢复	168
5.6 实验与指导	93	8.3 习题	169
5.7 习题	97	第 9 章 SQL Server 代理服务	170
第 6 章 视图与索引	99	9.1 概述	170
6.1 视图的创建与使用	99	9.1.1 更改 SQL Server 代理服务的状 态	170
6.1.1 视图概述	99	9.1.2 设置 SQL Server 代理	171
6.1.2 创建视图	100	9.2 操作员管理	174
6.1.3 通过视图管理数据	105	9.3 警报管理	175
6.1.4 修改视图	105	9.3.1 创建用户定义错误	175
6.1.5 删除视图	107	9.3.2 创建事件警报	176
6.2 索引的创建与使用	108	9.3.3 创建性能警报	178
6.2.1 索引的概念	108	9.3.4 定义对警报的响应	179
6.2.2 索引分类	109	9.4 作业管理	179
6.2.3 创建索引	109		
6.2.4 修改和删除索引	114		



9.4.1 创建作业分类	179	10.3 人员管理	190
9.4.2 创建作业	180	10.4 商品管理	195
9.4.3 设置作业步骤	182	10.4.1 商品添加	195
9.4.4 设置作业的调度计划	184	10.4.2 商品查询	196
9.4.5 设置通知消息	184	10.4.3 修改商品信息	198
9.5 习题	185	10.5 库存管理	204
第 10 章 销售管理系统的设计与实现	186	10.6 销售统计	209
10.1 概述	186	10.7 习题	214
10.2 用户管理	188	参考文献	215

第1章 数据库概述

随着计算机技术与网络通信技术的发展,数据库技术已成为信息社会中对大量数据进行组织与管理的重要手段,是网络信息化管理系统的基础。

数据库技术是现代信息科学与技术的重要组成部分,是计算机数据处理与信息管理的核心。数据库技术研究、解决了计算机信息处理过程中大量数据有效地组织和存储的问题,实现了在数据库系统中减少数据存储冗余、数据共享、保障数据安全以及高效地检索数据和处理数据。

本章主要介绍数据库的基本知识、关系数据库的特点、建立和设计方法等。

1.1 数据库基本知识

数据库技术是计算机科学技术的一个重要分支。目前,数据库技术已经成为数据处理的公用支撑技术,数据库系统也早已遍布政府机关、社会团体和企业单位,存储着他们赖以正常运转的数据资源,显著地提高了工作效率和质量,产生出巨大的社会、经济效益。

对于学习和使用数据库的人们来说,首先要了解和掌握数据库的基本概念、数据管理技术的发展、数据模型和数据库系统的模式结构等一些最基本的知识。

1.1.1 数据库的基本概念

1. 数据

数据(Data)是客观事物存在方式和运动状态的反映。在大多数人的头脑中,可能被狭义地理解成由数字0~9组成的数字符号序列。从计算机科学的角度看,广义的“数据”是指那些可以被计算机接受并能被计算机处理的符号序列。

在现实世界中,常见的数据形式可归纳为以下3种类型:

- 1) 数值型数据:对客观事物进行定量记录的符号,如数量、价格、年龄和度数等。
- 2) 字符型数据:对客观事物进行定性记录的符号,如姓名、性别、单位、地址等。
- 3) 特殊型数据:对客观事物进行形象特征和过程记录的符号,如声音、视频、图像等。

2. 数据库

数据库(DataBase, DB)从字面上理解,就是存放大量“数据”的“仓库”。在计算机中,数据库是指为了满足某一部门中多个用户的多种应用的需要,按照一定的数据模型在计算机中组织、存储和使用的相互联系的数据集合。

举例来说,每个人都有很多朋友和熟人,为了保持联系,常常用一个记事本将他们的姓名、地址、电话等信息都记录下来,以方便联系。这个“通讯录”就是一个最原始的由人工管理的“数据库”,每个人的姓名、地址、电话等信息就是这个数据库中的数据项。可以随意在记事本这个“数据库”中添加新朋友的个人信息,也可以由于某个朋友的家庭地址或者其他变动而修改其相关信息,如图1-1所示。



编号	姓名	办公电话	住宅电话	移动电话	传真号码	电子邮件地址	生日
2	安琪	(010) 8237-9482	(010) 8542-3497	(010) 139101111	(010) 8542-9483	andrewf@anywhere.com	1955-10-15
1	黎明	(206) 555-3412	(206) 555-3497	(206) 555-7777	(206) 555-3413	janetl@anywhere.com	1964-11-14
3	王辉	(010) 6555-8122	(010) 8278-3497	(010) 139555555	(010) 8278-8123	margiep@anywhere.com	
4	君如	(010) 6734-8080	(010) 8765-3422		(010) 8765-7877	nancy@anywhere.com	
5	李军	(206) 555-1234				StevenB@hfs.com	

图 1-1 通讯录

数据库的定义：数据库是与应用程序彼此独立的、以一定的组织方式存储在一起的、彼此相互关联的、具有较少冗余的、能被多个用户共享的数据集合。数据库可以用人工建立、维护和使用，也可以通过计算机建立、维护和使用。当然，我们关心的是后者。计算机化的数据库一般都是通过应用程序或数据库管理系统来建立、维护 and 使用的。

3. 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是一个通用的软件系统，由一组计算机程序构成。数据库管理系统能够对数据库进行有效的管理，包括存储管理、安全性管理、完整性管理等。数据库管理系统提供了一个软件环境，使用户能方便快速地建立、维护、检索、存取和处理数据库中的信息。

DBMS 是数据库系统的核心软件，学习使用数据库，通常学习某个 DBMS 的使用方法，在关系数据库中有许多 DBMS 系统，例如：Access、FoxBASE、FoxPro、Oracle 等。

4. 数据库系统

狭义地讲，数据库和数据库管理系统加在一起就构成了数据库系统；广义地讲，数据库系统是由数据库、硬件、软件和数据库管理员 4 个部分组成的。

1.1.2 数据管理技术的发展

数据库技术是计算机科学技术中发展最快的重要分支之一，它已经成为计算机信息系统和应用系统的重要技术支撑。数据库技术所研究的问题就是如何科学地组织和存储数据，如何高效地获取和处理数据，而数据处理的中心问题是数据管理。因此，数据管理的发展是数据库技术发展的一个重要标志。

数据管理指的是对数据的分类、组织、编码、存储、检索和维护。自 20 世纪 60 年代末、70 年代初以来，随着数据库技术的不断发展和完善，数据管理技术在 30 多年中主要经历了 4 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

早期的数据处理，限于计算机技术的发展而主要用于科学计算，数据本身不能独立存储提供应用，数据只能是附属于计算机程序的一部分，随程序一起运行或撤销。

人工数据管理的特点如下：

- 数据不保存。一组数据对应于一个应用程序，应用程序与其处理的数据结合成一个整体。在进行计算时，系统将应用程序与数据一起装入，用完后就将它们撤销，释放被占用的数据空间和程序空间。

- 没有软件对数据进行管理。程序员不仅要规定数据的逻辑结构，还要在程序中设计物理结构，如存储结构、存取方法、输入输出方式等。因此，数据与程序不具有独立性，如果数据在存储上改变了，程序员就必须修改程序。



- 没有文件概念。数据的组织方法由应用程序开发人员自行设计和安排。
- 数据面向应用。即如果两个应用程序使用相同的数据，也必须各自定义自己的数据存储和存取方式，不能共享相同的数据定义，因此造成了程序和程序之间大量的相同数据的冗余。

2. 文件系统阶段

随着计算机硬件技术的发展，特别是有了可以直接存储的外存储设备，软件上也有了数据管理软件，使得计算机管理的数据可以以文件的形式保留在外存上，这样可以通过对数据文件的存取进行数据的插入、查询、修改、删除等操作，但是对文件数据的访问操作只能以数据记录为单位，不能以数据项为单位进行。

文件数据管理的特点如下：

- 数据能够长期保存在磁盘上。用户可经常对在外存上保存的数据进行查询、修改、插入和删除等操作。
- 有了提供文件与程序之间进行存取方法转换的软件。文件的逻辑结构与存储结构由系统进行转换，使程序与数据有了一定的独立性。程序员不必关心数据的物理位置，可以集中精力考虑算法，并且节省了维护程序的工作量。
- 文件形式的多样化。由于有了直接存取存储设备，也就有了索引文件、链接文件、直接存取文件等。对文件的记录可以顺序访问、随机访问。文件之间相互独立，必须用程序来实现文件与文件之间的联系。
- 数据存取以记录为单位。

3. 数据库系统阶段

随着计算机技术的快速发展，大容量硬盘出现，硬件价格也趋于下降，对大量数据进行管理的需求越来越多。同时，数据库理论不断完善，且出现了数据库管理系统软件商品，数据库技术应用日益普及，发展趋于成熟。数据库系统相比文件系统来说，克服了文件系统的缺陷，在数据管理方面有了重大的飞跃。

数据库技术的产生，使数据与程序可以相互独立，最大程度地减少了存储数据的冗余、最大限度地为多个程序或用户所共享。

数据库系统的特点如下：

- 数据库中的数据是结构化的。文件系统中的数据是无结构的，数据库系统不仅考虑数据项之间的联系，而且考虑记录之间的联系。这种联系是通过存储路径来实现的。

- 数据冗余度小，易扩充。文件系统中的数据均是与某个应用相联系，而数据库中的数据面向系统，减少了数据冗余，实现了不同应用间的数据共享，如图 1-2 所示。

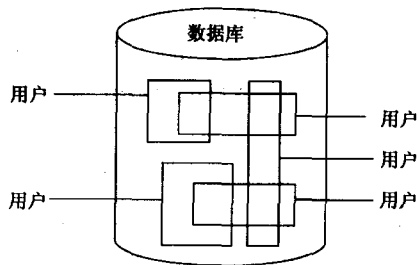


图 1-2 数据共享

- 较高的数据和程序独立性。
- 数据库为用户提供了方便的接口。用户可以用数据库系统提供的查询语言和交互式命令操纵数据库，也可以用高级语言编写程序来操纵数据库，拓宽了数据库的应用范围。
- 数据的最小存取单位是数据项。在数据库中用户既可以存取数据库中某一个数据项



或一组数据项，也可以存取一个记录或一组记录。

1.1.3 数据模型

数据模型是数据库系统中的一个关键概念，各种机器上的数据库系统都是基于某种数据模型的。因此，了解数据模型的基本概念是学习数据库的基础。

数据模型是指描述记录内的数据项间的联系和记录之间的联系的数据结构形式。它应满足三方面要求：能较真实地模拟现实世界；能容易被人所理解；便于在计算机上实现。

在数据库的发展历史上，常用的数据库模型有3种：层次模型（Hierarchical Model）、网状模型（Network Model）和关系模型（Relational Model）。

下面简要介绍这3种模型。

1. 层次模型

层次模型是数据库系统中最早使用的一种模型。数据的层次模型用树形结构来表示各类实体的类型和实体之间的联系。在现实世界中许多实体之间的联系很自然地呈现出一种层次关系，例如：家族关系、行政机构等。

在数据库中满足下述两个条件的“基本层次联系”的集合称为层次模型。

- 1) 有且只有一个节点无双亲节点，这个节点就是根节点。
- 2) 其他节点有且仅有一个双亲节点。

在层次模型中，每个节点表示一个记录类型（实体），记录之间的连线表示节点之间的联系。每个节点的上方的节点称为该节点的双亲节点，而其下方的节点称为该节点的子节点。没有子节点的节点称为叶节点。

层次模型的结构示意图如图1-3所示。

在图1-3中，R1为根节点，同时是R2、R3的双亲节点。R2、R3是R1的子节点，同时又分别是R4、R5、R6的双亲节点。R6是R7的双亲节点，R7是R6的子节点。R4、R5、R7是叶节点。

从图1-3上可以看出层次模型像一棵倒立的树，节点的双亲是惟一的。

层次模型适用于描述客观存在的事物中主次之分的结构关系，具有层次分明、结构清晰的特点。它的缺点是层次模型只能反映记录类型（简称“记录型”）间的一对多的关系，不能反映多对多的关系。

图1-4是一个“教师学生”层次数据库。在这个层次数据库中有4个记录型。记录型“院系”是根节点，由系号、系名、系主任名3个字段组成。它有两个子节点：教研室和学生。记录型“教研室”是院系的子节点，同时也是授课的双亲节点，它由室号、室名、室主任名3个字段组成。记录型“学生”由学号、姓名、成绩3个字段组成。记录型“授课”由课号、班级、教师姓名3个字段组成。“学生”与“授课”是叶节点，它们没有子节点。由“院系”到“教研室”、由“教研室”到“授课”、由“院系”到“学生”均是一对多的联系。

2. 网状模型

由于层次结构不能描述多对多的关系，因而产生了网状模型。网状模型就是在层次模型

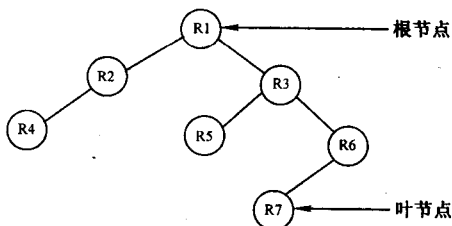


图 1-3 层次模型的结构示意图

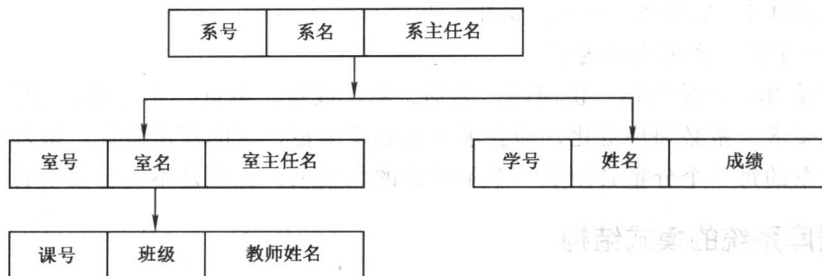


图 1-4 “教师学生”层次数据库

的基础上取消层次模型的限制，将树形结构变成网状结构。网状模型是以记录型为节点的网路，反映的是现实世界中较为复杂的事物间的联系。

广义上，任意一个连通的基本层次联系的集合就是一个网状模型。在数据库中，把满足下列条件的基本层次联系集合称为网状模型：

- 1) 允许有任意个（包括零个）无双亲节点。
- 2) 允许一个节点可以有多于一个的双亲节点。
- 3) 两个节点之间可以有二种或两种以上的关系。

图 1-5 为网状模型示意图。

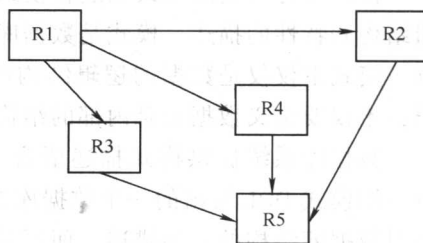


图 1-5 网状模型示意图

网状模型能够更为直接地描述现实世界，具有良好的性能，存取效率较高。但是，由于网状模型的结构比较复杂，而且随着应用环境的扩大，数据库的结构就变得越来越复杂，不利于用户使用。

3. 关系模型

关系模型是当前最重要的一种数据模型。20 世纪 80 年代以来，几乎所有的数据库系统都是关系型的。关系模型是建立在严格的数学概念的基础上的。

从用户的角度看，关系模型的数据结构是一个二维表，它使用表格描述实体间的关系，由行和列组成。每个表格就是一个关系。下面以产品表（如图 1-6 所示）为例，介绍关系模型中的一些术语。

1) 关系：一个关系也就是通常所说的一张表，如图 1-6 中的产品表。

2) 元组：表中的一行就是一个元组，或称为一条记录。

3) 属性：表中的一列即为一个属性（也称字段），每个属性有一个名称即属性名（也称字段名）。例如，图 1-6 中的产品表有 8 列，对应 8 个属性：产品 ID、产品名称、供应商、类别、单位数量、单价、库存量、订购量。

4) 关键字：表中的某个属性组可以惟一地确定一个元组，如图 1-6 中的产品 ID，可以惟一确定一个产品，即该关系中的关键字。

5) 关系模式：对关系的描述。一般的表示方法为：

产品表							
产品ID	产品名称	供应商	类别	单位数量	单价	库存量	订购量
1	苹果汁	佳佳乐	饮料	每箱24瓶	¥ 18.00	39	0
2	牛奶	佳佳乐	饮料	每箱24瓶	¥ 19.00	17	40
3	番茄酱	佳佳乐	调味品	每箱12瓶	¥ 10.00	13	70
4	盐	康富食品	调味品	每箱12瓶	¥ 22.00	53	0
5	蒜油	康富食品	调味品	每箱12瓶	¥ 21.35	0	0
6	酱油	妙生	调味品	每箱12瓶	¥ 25.00	120	0

图 1-6 产品表



关系名 (属性 1, 属性 2, ……, 属性 n)

例如图 1-6 中的关系可表示为:

产品 (产品 ID, 产品名称, 供应商, 类别, 单位数量, 单价, 库存量, 订购量)

关系模型要求关系必须规范化, 即要求关系必须满足一定的规范条件。最基本的一条规范条件是: 关系的每一个分量必须是一个不可分的数据项, 也就是不允许表中有表。

1.1.4 数据库系统的模式结构

数据库系统的体系结构是数据库系统的总体框架。尽管实际的数据库系统软件产品多种多样, 支持不同的数据模型, 使用不同的数据库语言, 建立在不同的操作系统之上, 数据的存储结构也不尽相同, 但是绝大多数数据库系统在总体框架上都具有三级模式的结构特征。

数据库系统的三级模式结构是: 外模式、模式和内模式。

1. 模式

模式也称为概念模式或逻辑模式, 它是数据库的总体框架, 是对数据库中全体数据的逻辑结构和特性的描述。模式是数据库数据在逻辑级上的视图, 它通常以某一种数据模型为基础。模式不仅仅是数据的逻辑结构的定义, 而且要定义与数据有关的安全性、完整性的要求; 不仅要定义数据记录内部的结构, 而且要定义这些数据项之间的联系。

数据库系统提供模式描述语言 (Data Description Language, DDL) 来严格地表示以上内容。用模式 DDL 写出的一个数据库逻辑定义的全部语句, 称为某一个数据库的模式。模式是对数据库结构的一种描述, 而不是数据库本身, 它是装配数据的一个框架, 所有数据都是按这一框架进行装配的, 如图 1-7 所示。

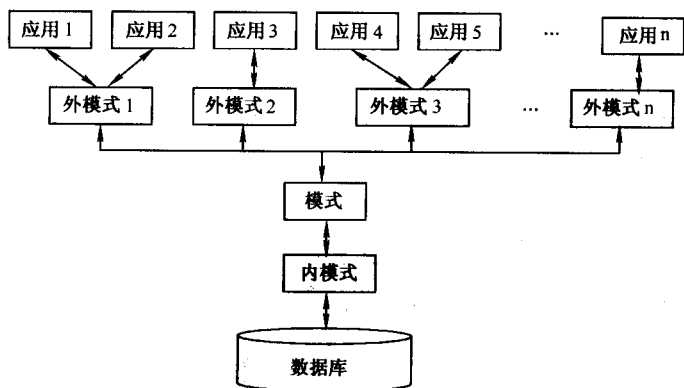


图 1-7 数据库系统的三级模式图

2. 外模式

外模式也称子模式或用户模式, 是数据库用户看到的数据视图, 即与某一应用有关的数据的逻辑表示。不同用户的外模式可以相互覆盖, 同一外模式可以为某一用户的任意多个应用所启用。一个应用只能启用一个外模式。

外模式通常是模式的子集, 也可以是整个模式。由于用户的不同需求, 不同用户的外模式的描述是不同的, 即使是对同一数据的描述也有可能不同。



外模式用外模式描述语言 (Subschema Data Description Language, SDDL) 描述用户数据视图。用 SDDL 写出的一个用户数据视图的逻辑定义的全部语句称为此用户的外模式。

3. 内模式

内模式是全体数据库数据的内部表示或者低层描述, 用来定义数据的存储方式和物理结构, 因此内模式也称为存储模式。

内模式规定了数据在介质上的物理组织形式和记录寻址方式, 定义物理存储块的大小和溢出处理方法等。它与模式是对应的。

内模式用内模式数据描述语言 (Data Storage Description Language, DSDL) 进行描述。

数据库系统的三级模式结构将数据库的全局逻辑结构同用户的局部逻辑结构和物理存储结构分开, 方便了用户对数据库的组织和使用。

1.2 关系数据库

关系数据库 (Relation DataBase) 是建立在严密的数学基础之上的, 它应用数学方法来处理数据库中的数据。

关系数据库是目前各类数据库中最重要、最流行的数据库, 也是目前使用最广泛的数据库系统。20 世纪 70 年代以后开发的数据库管理系统产品几乎都是基于关系的。

1.2.1 关系模型与关系数据库

关系模型的逻辑结构是一个二维表, 它由行 (记录) 和列 (字段) 组成。每个表格就是一个关系。关系模型具有以下性质:

- 1) 在一个关系中, 每一数据项不可再分, 是最基本的数据单位。
- 2) 在一个关系中, 同一列数据项要具有相同的数据类型。
- 3) 在一个关系中, 各列的顺序可以任意地排列。
- 4) 在一个关系中, 不允许有相同的字段名。
- 5) 在一个关系中, 每一行是一个个体事物多个字段值的集合。
- 6) 在一个关系中, 不允许有完全相同的记录行。
- 7) 在一个关系中, 记录的顺序可以任意地排列。

关系数据库是依照关系模型设计的若干个关系的集合。

在关系数据库中, 称一个“关系” (一张二维表) 为一个数据表 (简称表), 表是由表结构和数据组成的。

一个关系数据库由若干个表组成, 每一个表都具有相对的独立性, 这一独立性的惟一标志是表的名字, 称为表文件名。数据库中不允许有重名的表, 因为对表中数据的访问首先是通过表文件名来实现的。

在关系数据库中, 各表之间可以相互关联, 表之间的这种联系是依靠每一个独立表内部的相同属性字段建立的。

有关表的建立和表间关联关系的建立请参见本书第 4 章和第 5 章的相关介绍。



1.2.2 关系的完整性

关系模型的完整性规则是用来约束关系的，以保证数据库中数据的正确性和一致性。关系模型的完整性共有三类：实体完整性、参照完整性和用户定义的完整性。数据完整性由实体完整性和参照完整性规则来维护，实体完整性和参照完整性是关系模型必须满足的完整性约束条件，将由关系系统自动支持。

1. 实体完整性

规则一即实体完整性规则：若属性 A 是基本关系 R 的主属性，则属性 A 不能取空值（空值包括“不知道”或“无定义”的值）。主属性不能为空，是指主码整体不能为空，例如：姓名、生日和家庭住址组合作为主码，则“姓名”、“生日”、“家庭住址”都不能取空值，而不是整体不为空。

在关系数据库系统中有各种各样的关系，即各种各样的表，如基本表、查询表、视图表等。基本表是客观存在的表，是实际存储数据的逻辑表示；查询表是查询结果所对应的表；视图表是由基本表或视图表导出的表，是虚拟表，不对应实际存储的数据。实体完整性规则是针对基本表的。

对于实体完整性的说明如下：

- 一个基本关系对应着一个现实世界的实体集。
- 现实世界中的实体是可区分的，即它们具有某种惟一的标识。
- 关系模型中由主码作为惟一性标识。
- 主码不能取空值，因为主码取空值说明存在某个不可标识的实体，与第二点矛盾。

2. 参照完整性

现实世界中的实体之间往往存在一定联系，在关系模型中实体与实体的联系是用关系来描述的。参照完整性即是有关关系之间能否正确进行联系的规则。两个表能否正确进行联系，外码是关键。

外码的定义：设 F 是基本关系 R 的一个或一组属性，但不是关系 R 的主码，另有关系 S ，关系 S 的主码为 K_s ，若 F 与 K_s 相对应，则称 F 是基本关系 R 的外码。

【例】 学生和院系两个实体由以下两个关系表示，主码用下划线标明。

学生（学号、姓名、院系号）

院系（院系号、院系名）

“院系号”是学生表的一组属性，但不是学生表的主码，“院系号”与院系表的主码相对应，则“院系号”是学生表的外码。学生表中某个属性的取值要参照院系表属性的取值。我们可以清楚的看到外码“院系号”是联系学生表和院系表的桥梁，两个关系进行联系就是通过外码实现的。

规则二即参照完整性规则：若属性（或属性组） F 是基本关系 R 的外码，它与基本关系 S 的主码 K_s 相对应（基本关系 R 和 S 不一定是不同的关系），则对于 R 中每一个元组在 F 上的值必须为：

- 或者取空值（ F 的每个属性值均为空值）。
- 或者等于 S 某个元组的主码值。

在上例中学生关系中的“院系号”可以为空值，表示尚未给该学生分配院系；或者非空



值,但必须是院系关系中属性“院系号”的某个元组值,表示不能把学生分到一个根本不存在的院系。即被参照关系“院系”中一定存在一个元组,它的主码值等于参照关系“学生”中的外码值。

3. 用户定义的完整性

实体完整性和参照完整性用于任何关系数据库系统。用户定义的完整性则是针对某一具体数据库的约束条件,由应用环境决定,它反映了某一具体应用所涉及的数据必须满足的语义要求。例如,成绩的取值用户会定义在0~100之间。关系模型应提供定义和检验这类完整性机制,以便用统一的方法处理它们而不要由应用程序承担这一功能。

在实际系统中,这类完整性规则一般在建立库表的同时进行定义,应用编程人员不需再做考虑。如果某些约束条件没有建立在库表一级,则应用编程人员应在各模块的具体编程中通过程序进行检验和控制。

1.3 关系数据库设计

随着计算机技术的广泛应用,目前从小型的单项事务处理到大型的信息系统都采用数据库技术来保持数据的完整性和一致性,因此在应用系统的设计中,数据库搭建得是否合理变得日趋重要。

具体地说,数据库设计是指针对一个给定的应用环境,构造最优的数据库模式,建立数据库及其应用系统,使之能够有效地存储数据,满足各种用户的应用需求(信息要求和处理要求)。数据库设计是数据库在应用领域的主要研究课题。

目前我们所说的数据库设计大多是在一个现成的DBMS的支持下进行,即以一个通用的DBMS为基础开发数据库应用系统。

为了使数据库的设计趋于合理,人们进行了大量的研究和实践,提出了不少数据库设计的方法,如新奥尔良法(New Orleans)、规范化法和基于E-R模型的数据库设计方法等。这些方法的共同点都是将数据库设计纳入软件工程的范畴,把软件工程的原理、技术和方法应用到数据库设计中。

1.3.1 E-R 关系模型数据库设计

构建数据模型需要回答实体、关系和属性等问题。利用数据模型可以更容易理解数据的含义,理解每个用户对数据的要求、数据本身的特性、在应用程序中对数据使用的情况等。

数据库的设计最困难的一个方面就是设计人员、程序员和最终用户看待数据的方式不同。为了保证能够准确理解数据的本质,理解用户使用这些数据的方法,需要有一个通用模型,这个模型与技术实现无关,且没有二义性。E-R(Entity-Relationship, 实体-关系)模型就能很好地解决这些问题。

E-R建模是一种自上而下的数据库设计方法。通过标识模型中必须要表示的重要数据(实体)以及数据间的关系开始E-R建模,再增加细节信息,如实体和关系所要具有的信息(属性),以及实体、关系和属性上的限制等。

1. 实体

实体(Entity):一组有相同属性的对象,被用户标识为独立存在的对象集合。E-R建模



的一个基本概念是实体，代表一组现实世界中的对象集合，它们有相同的属性。每个对象必须在集合中被唯一地标识表示，叫做实体。

一个实体是独立存在的，代表一组物理存在（现实）或是概念存在（抽象）的对象的集合，如表 1-1 所示。

表 1-1 实体的构成

物理存在	概念存在
房屋	角色
电脑	租赁
工厂	注册

每个实体可以用矩形来表示，在矩形中写上实体的名字，名字通常是一个名词。在矩形中每个单词的第一个字母大写。图 1-8 显示了 Computer、Role、House 的图形化表示。

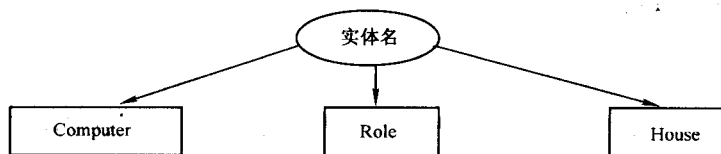


图 1-8 实体的图形化表示

2. 关系

关系 (Relationship)：实体之间的具有某种含义的关联。

一个关系是特定实体之间的一组关联。和实体一样，每个关系在集合中也要被唯一地标识。一个可以惟一标识的关联叫做关系事件 (Relationship Occurrence)。

每个关系有一个名字，此名字描述关系的功能。例如，实体产品与实体工厂通过关系出售关联，实体原料与实体工厂通过关系加工关联。

每个关系显示为连接关联实体的一个菱形，并用关系的名字标记。通常，关系由动词或动词短语命名。关系名短语的第一个英文字母要大写。在给定的 E-R 模型中，关系名应该是惟一的。一个关系仅在一个方向标记，代表关系仅在一个方向起作用（例如：工厂加工原料有意义，原料加工工厂则没有意义）。因此，一旦选择了关系名，必须在名字旁加箭头指示方向。图 1-9 表示了工厂加工原料和工厂出售产品的关系。

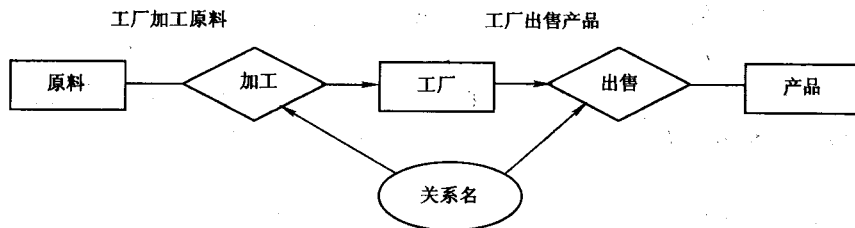


图 1-9 工厂加工原料和工厂出售产品的关系（二元关系）

在实体的关系中还有一个叫做关系的度 (Degree of a Relationship) 的概念，它是指在关系中参与的实体的数目。一个度数为 2 的关系叫做二元关系，在图 1-9 中显示的两个关系度