

SQL Server 2012

数据库基础及应用

©主编 叶符明 王 松

SQL Server 2012

数据库基础及应用

主 编	叶符明	王 松	
副主编	朱小云	赖乔乔	杨 华
参 编	韩 琰	王 颖	李 静
	李雯婷	穆肇南	杨云勇
	周 海	陈家文	

 **北京理工大学出版社**
BEIJING INSTITUTE OF TECHNOLOGY PRESS

内 容 简 介

本书共分十一章, 主要包括: 数据库基础知识、SQL Server 数据库基础、数据库管理、表的操作、数据完整性、索引和视图、Transact-SQL 语言基础、存储过程与触发器、游标与事务、SQL Server 2012 的安全机制、使用 C# 开发应用程序。为方便开展实训练习, 本书中专门编写了上机指导部分。

本书主要适用于高等院校的计算机专业学生使用, 也可适用于非计算机专业学生学习数据库基础知识, 还可作为普通读者普及数据库基础知识的学习书籍。

版权专有 侵权必究

图书在版编目 (CIP) 数据

SQL Server 2012 数据库基础及应用/叶符明, 王松主编. —北京: 北京理工大学出版社, 2013. 10

ISBN 978 - 7 - 5640 - 8416 - 5

I. ①S… II. ①叶… ②王… III. ①关系数据库系统 - 高等职业教育 - 教材 IV. ①TP311. 138

中国版本图书馆 CIP 数据核字 (2013) 第 243541 号

出版发行 / 北京理工大学出版社有限责任公司

社 址 / 北京市海淀区中关村南大街 5 号

邮 编 / 100081

电 话 / (010) 68914775 (总编室)

82562903 (教材售后服务热线)

68948351 (其他图书服务热线)

网 址 / <http://www.bitpress.com.cn>

经 销 / 全国各地新华书店

印 刷 / 北京地质印刷厂

开 本 / 787 毫米 × 1092 毫米 1/16

印 张 / 15.5

字 数 / 354 千字

版 次 / 2013 年 10 月第 1 版 2013 年 10 月第 1 次印刷

定 价 / 40.00 元

责任编辑 / 赵 轩

李志敏

文案编辑 / 赵 轩

责任校对 / 周瑞红

责任印制 / 马振武

图书出现印装质量问题, 请拨打售后服务热线, 本社负责调换



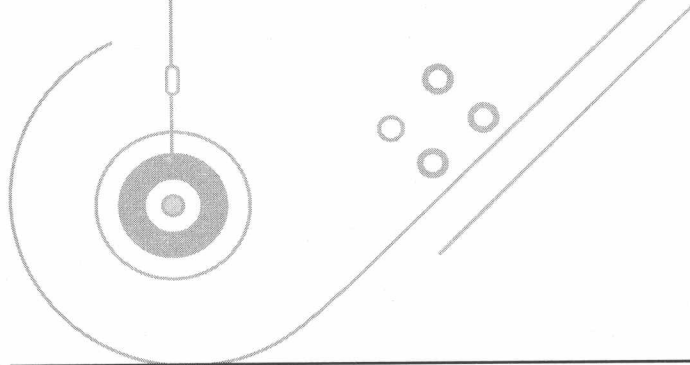
前言

数据库技术是现代信息科学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。《数据库基础及应用》课程是计算机信息管理、计算机应用技术、电子商务等专业的核心课程。

本教材不是完全讲授数据库原理，而是培养学生在了了解数据库基本知识的基础上，能够应用 SQL Server 工具，实现数据库的建立、表的建立，数据的录入、修改和查询等技术。教材编写对应项目式教学，全书以学生成绩系统项目为例，以项目式教学过程中完成具体任务的过程来讲授知识点，使学生了解数据库开发过程、开发步骤、掌握开发技术。

授课过程中，为了引导学生进行自主性、创新性、开放性学习，同时加强学生对专业技能的应用训练，针对每一章节的学习过程，我们采取“课堂讲授项目学习，个人独立项目巩固，小组特色项目提高”的组织方式。要求对学生进行分组，教师在课堂教学全过程采用“学生成绩管理”系统项目讲解知识，学生实现对知识点的认知和理解；在实训课程中要求每位同学独立完成“图书管理”系统项目的具体任务，学生实现知识点的验证及巩固；课后要求每组学生必须完成特色数据库项目的同步开发，完成数据库的设计和建立，数据的录入管理和处理等操作，学生实现数据库技术的综合应用及提高。

教学环节	教学目的及要求	教学内容及任务	项目名称
课堂讲授	认知、理解	章节知识点讲授及学习	学生成绩系统
实训练习	巩固、掌握	章节知识点验证及巩固	学生成绩系统（验证） 图书管理系统（巩固）
课后作业	应用、提高	综合应用及提高	小组自选项目（进销存管理系统、人力资源管理系统、财务管理系统酒店管理系统……）



本书在选材上从实用和教学的角度出发，深入浅出地介绍了数据库基础及应用相关知识。主要内容包括：第1章介绍数据库基础知识；第2章介绍 SQL Server 数据库基础；第3章介绍数据库管理，主要包括数据库的创建、查看修改、删除、备份与还原等；第4章介绍表的操作，主要包括表结构的创建、修改、删除和的表数据操作等；第5章介绍数据完整性，主要包括数据完整性的重要作用及约束、默认、规则等具体实现方法；第6章索引和视图，主要介绍索引和视图的作用及应用；第7章 Transact-SQL 语言基础；第8章存储过程与触发器，介绍了存储过程与触发器的功能及应用；第9章游标与事务，介绍了游标与事务的功能及应用；第10章 SQL Server 2012 的安全机制主要介绍了账户、用户、角色、权限等安全机制；第11章数据库开发技术，简单介绍了用 C# 开发数据库应用程序的实例；第12章为配合教学实训提供了详细的上机指导。

本书由叶符明、王松担任主编；朱小云、赖乔乔、杨华担任副主编；参加编写的还有韩琰、李静、王颖、李雯婷、穆肇南、杨云勇、周海、陈家文等。本书凝聚了作者多年从事《数据库基础及应用》教学的实际经验，书中所有代码均经正确运行后截图显示，图示清晰。

由于时间仓促以及水平有限，书中错误和不当之处在所难免，恳请专家、老师和读者批评指正。

编 者

目录

第1章 数据库基础..... 1

- 1.1 数据库管理系统..... 1
 - 1.1.1 数据与数据处理..... 1
 - 1.1.2 计算机数据管理..... 1
 - 1.1.3 数据库系统..... 2
 - 1.1.4 数据库管理系统..... 4
- 1.2 数据模型..... 4
- 1.3 关系数据库..... 7
- 1.4 关系型数据模型的范式结构..... 10

第2章 SQL Server 数据库概述..... 13

- 2.1 SQL 及 Transact-SQL 简介..... 13
- 2.2 SQL Server 发展简史..... 13
- 2.3 SQL Server 2012 特点及配置要求..... 14
- 2.4 SQL Server 2012 的安装过程..... 16
- 2.5 SQL Server 2012 的启动..... 29

第3章 数据库管理..... 31

- 3.1 数据库的存储结构..... 31
- 3.2 创建数据库..... 32
- 3.3 修改数据库..... 38
- 3.4 查看数据库..... 41
- 3.5 删除数据库..... 43
- 3.6 数据库的备份及还原..... 45
 - 3.6.1 备份/还原数据库..... 45
 - 3.6.1 分离/附加数据库..... 52

第4章 表的操作..... 56

- 4.1 数据类型..... 56
 - 4.1.1 系统数据类型..... 56
 - 4.1.2 创建及删除用户定义数据类型..... 59

4.2 表的创建	62
4.3 表的修改	68
4.4 表的删除	73
4.5 表的数据操作	74
4.5.1 向表中插入数据	74
4.5.2 修改表中数据	78
4.5.3 删除表中数据	80

第5章 数据完整性 83

5.1 数据完整性	83
5.1.1 实体完整性	83
5.1.2 域完整性	83
5.1.3 参照完整性	84
5.1.4 用户定义完整性	84
5.2 约束	84
5.2.1 CHECK 约束	84
5.2.2 PRIMARY KEY 约束	87
5.2.3 FOREIGN KEY 约束	91
5.2.4 UNIQUE 约束	94
5.2.5 DEFAULT 约束	97
5.3 默认	100
5.3.1 创建默认对象	100
5.3.2 绑定默认	101
5.3.3 解除默认对象的绑定	102
5.3.4 删除默认对象	103
5.4 规则	103
5.4.1 创建规则	104
5.4.2 绑定规则	104
5.4.3 解除规则的绑定	105
5.4.4 删除规则	106

第6章 索引和视图 107

6.1 索引	107
6.1.1 索引的分类	107
6.1.2 创建索引	108
6.1.3 索引的查看	111
6.1.4 索引的删除	113

6.2 视图	114
6.2.1 视图的概念	115
6.2.2 创建视图	115
6.2.3 查询视图	120
6.2.4 重命名视图	122
6.2.5 修改视图的定义	123
6.2.6 删除视图	125

第7章 Transact-SQL 语言基础

7.1 常量与变量	128
7.1.1 常量	128
7.1.2 变量	128
7.2 运算符和表达式	130
7.3 数据查询操作	131
7.3.1 查询的基本语法	131
7.3.2 简单查询	131
7.3.3 查询结果排序	139
7.3.4 查询统计	140
7.3.5 分组查询	142
7.3.6 连接查询	143
7.3.7 子查询	146
7.4 程序流程控制语句	148
7.4.1 IF...ELSE 语句	148
7.4.2 无条件转移语句——GOTO 语句	149
7.4.3 WHILE、BREAK、CONTINUE 语句	149
7.5 系统内置函数	151
7.5.1 行集函数	151
7.5.2 聚合函数	151
7.5.3 标量函数	152
7.6 用户定义函数	152
7.6.1 用户函数的定义	152
7.6.2 用户函数的调用	153
7.6.3 用户函数的删除	153

第8章 存储过程与触发器

8.1 存储过程	154
----------------	-----

8.1.1 存储过程基础	154
8.1.2 创建存储过程	154
8.1.3 执行存储过程	157
8.1.4 修改、查看、重命名、删除存储过程	159
8.1.5 存储过程应用	161
8.2 触发器	166
8.2.1 触发器基础	166
8.2.2 创建触发器	167
8.2.3 查看、修改和删除触发器	170
8.2.4 触发器应用	173

第9章 游标与事务 174

9.1 游标	174
9.1.1 游标基础	174
9.1.2 游标的使用步骤	174
9.1.3 游标应用	176
9.2 事务	177
9.2.1 事务基础	177
9.2.2 事务操作	178

第10章 SQL Server 2012 的安全机制 181

10.1 创建用户账户及登录	181
10.2 数据库用户	187
10.3 数据库角色	188
10.4 数据库权限	189
10.5 数据库架构	191

第11章 使用 C#开发应用程序 192

第12章 210

上机指导	210
实训一 数据库文件管理	210
实训二 表的操作	211

实训三	表数据的操作	213
实训四	约束	215
实训五	默认和规则	216
实训六	关系、索引和视图	217
实训七	SELECT 查询	218
实训八	程序设计	219
实训九	存储过程	219
实训十	触发器	220
实训十一	游标和事务	221
实训十二	安全认证与访问权限	221
实训十三	数据库开发技术	221

第1章

数据库基础

本章学习导读

现如今人类社会已进入信息时代,计算机也已广泛应用于信息处理领域,为计算机应用的普及打开了一个崭新的局面。目前各种类型的计算机已经广泛应用于情报检索系统、图书管理系统、办公信息系统、银行信息系统、交通信息系统、人事管理系统及财务管理系统等各种企事业单位的管理信息系统中。在管理信息系统中,数据处理的特点是数据量大、类型多、机构复杂,并且对数据的存储、检索、分类和统计的要求也比较高。如何对系统内的大量数据进行组织和管理,为用户准确、快速地提供所需要的信息,并使之能为更多的程序共享,这是数据库系统的基本应用之一。

数据库技术产生于20世纪60年代末期。数据库技术的出现,使得数据处理能力得到了极大的提高,可靠性不断增加,成本也不断降低,从而推动了计算机应用的普及。

数据库系统有3个重要的组成部分:经组织后可供用户使用的数据库、介于数据库与应用程序之间的数据库管理系统以及提供给用户使用的各类应用程序。计算机应用人员只有掌握了数据库系统的基础知识,熟悉数据库管理系统的特点,才能开发出适用的数据库应用系统。



1.1 数据库管理系统

1.1.1 数据与数据处理

数据是数据库中存储的基本对象。数据是对客观事物属性的描述与记载,是物理符号。在数据处理领域中,数据不仅包括数字、字母、文字和其他特殊字符组成的文本形式的数据,而且还包括图形、图像、动画、影像和声音等多媒体数据,但是最多、最基本的仍然是文字数据。

信息是客观世界中各种事物(包括数据)变化、相互作用和特征的反应,是一个抽象的概念。

数据是信息的具体表现形式,信息是数据有意义的表现。信息来源于数据,当数据传递给有关的人,通过相互作用后,可能会给人带来某种信息。可见,数据与信息既有联系又有区别。

数据处理是指将数据转换成信息的过程。从数据处理的角度来看,信息是一种被加工成特定形式的数据,并且这种数据形式对于数据接收者来说是有意义的。因此,数据处理也称信息加工。人们有时说“信息处理”,其真正的含义是,为了产生信息而处理数据,通过处理数据获得信息。例如,每个人都有“出生日期”这一基本的原始数据,而每个人的“年龄”这一信息则可以通过当前的年份与“出生日期”中的年份进行一个减法计算而得到。

1.1.2 计算机数据管理

数据管理是数据处理的核心和基础。数据管理是指如何对数据进行分类、组织、编码、存储和维护。其主要任务是收集信息,然后将信息用数据表示并按类别组织保存。实际上,管人、管财、管

物或管事的工作就是数据管理工作。

从世界上第一台计算机诞生以来,在应用需求的推动下,在计算机硬件、软件发展的基础上,数据管理技术经历了人工管理、文件系统和数据库系统三个阶段。

① 人工管理阶段

20 世纪 50 年代中期以前,计算机主要用于科学计算。当时的外部存储器只有纸带、卡片、磁带,没有磁盘等直接存取的存储设备;没有操作系统,也没有专门管理数据的软件,数据由计算或处理它的程序自行携带。数据处理是批处理。

这一阶段的特点是:数据不具有独立性,一组数据对应一组程序;数据发生变化则必须对程序作出修改;数据不保存、不共享。

② 文件系统阶段

20 世纪 50 年代后期到 60 年代中期,这时计算机开始大量用于管理中的数据处理工作。硬件方面,有了磁盘、磁带等直接存取的存储设备;软件方面,出现了操作系统和专门的数据管理软件,这时的数据管理软件叫文件系统。数据处理不仅有批处理,而且能够联机实时处理。

这一阶段的特点是:数据与程序有了一定的独立性,程序和数据分开存储,有了数据文件,并且数据保存在数据文件中可反复使用;由于数据文件是对应于某个具体的应用程序的,所以在不同的应用程序中使用相同的数据时,仍需建立自己的数据文件,因此共享性差、冗余大,数据独立性差。

③ 数据库系统阶段

20 世纪后期以来,需要计算机管理的数据量急剧增长,对数据共享的需求日益增加,文件系统管理的方式已不能满足信息系统发展的需要。为了实现计算机对数据的统一管理,达到数据共享的目的,人们发展了数据库技术。

这一阶段的特点是:数据采用特定的数据模型,数据库中的数据是有结构的,这种结构由数据库管理系统所支持的数据模型表现出来,数据模型不仅可以表示事物内部各数据项之间的联系,而且可以表示事物与事物之间的联系,从而反映出现实世界中事物之间的联系;有统一的数据控制功能,数据库可以被多个用户或应用程序共享,数据的存取往往是并发的,即多个用户同时使用同一个数据库;必须给数据库系统提供必要的保护措施,包括并发访问控制功能、数据的安全性控制功能和数据的完整性控制功能;可以实现数据共享并减少数据冗余;具有较高的数据独立性。

1.1.3 数据库系统

数据库系统是指计算机系统中引入数据库后的系统,一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员(DBA)和用户构成。数据库的建立、使用和维护等工作仅靠数据库管理系统是远远不够的,还要有专门的人员来完成,这些人被称为数据库管理员。通常,把数据库系统简称为数据库。

① 数据库系统的组成

数据库系统(Data Base System, DBS)一般由数据库、软件、硬件和用户构成。

(1) 数据库

数据库(DataBase, DB)是指长期存储在计算机内,有组织的、统一管理的相关数据的集合。数据库中的数据按照一定的数据模型组织和存储,可共享并且冗余度较小,数据之间相互联系而又有较高的独立性。

(2) 软件

数据库系统的软件包括数据库管理系统(DataBase Management System,DBMS)、操作系统(Operating System,OS)、数据库应用系统(DataBase Application System,DBAS)等。DBMS是数据库系统的核心软件,负责数据的存取、维护和管理,是用户使用数据库的接口,其他应用程序需要通过DBMS访问数据库。

(3) 硬件

硬件是指存储数据库和运行数据库管理系统的硬件资源,包括物理存储数据库的存储设备(如磁盘、硬盘等)、CPU、内存、I/O通道和外部设备等。数据库一般存储在外存储器中,运行时需调入内存。随着数据库存储数据量的增大,要求有足够大的内存来存储操作系统、DBMS、应用程序及缓冲区,同时也要求有足够大的外存和较高的通道能力。

(4) 用户

用户是指一组使用数据库的不同人员,他们向数据库提出存储、维护和查询数据等各类请求。数据库系统中主要有4类用户:数据库管理员、系统分析与设计人员、应用程序员、终端用户(也称最终用户)。

数据库管理员(Database Administrator,DBA)是指全面负责数据库系统的管理、维护和正常使用的人员。DBA的主要职责是:设计数据库、定义数据库的各个方面、对数据库进行维护、改进和重新构造数据库系统等。

系统分析与设计人员(System Analyst and Designer,SAP)是指负责需求分析、数据库设计和系统功能设计的人员。

应用程序员(Application Programmer,AP)是指负责设计和编写应用程序的人员,同时也要负责程序的安装和调试。

终端用户(End User,EU)是指从计算机联机终端存储数据库的人员。他们通过数据库提供的终端命令或者菜单界面操作等交互方式来存取数据库中的数据,这种方法简单、易懂、易学、易用,适合非计算机专业的专业人员使用。

② 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式、内模式三级构成的,如图1-1所示。用户级对应外模式,概念级对应模式,物理级对应内模式,使不同级别的用户对数据库形成不同的视图。

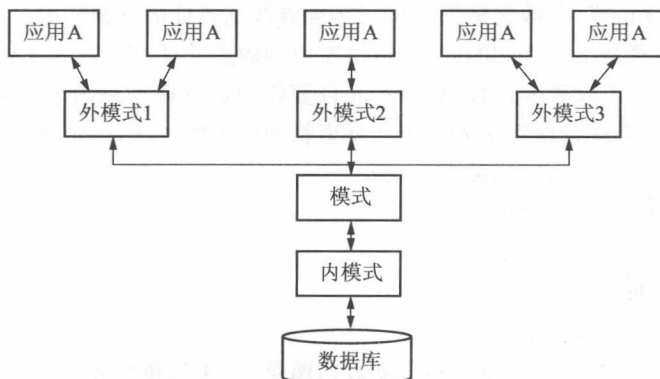


图 1-1 数据库系统的三级模式结构

(1) 模式

模式又称概念模式或逻辑模式,对应于概念级。它是由数据库设计者综合所有用户的数据,按照统一的观点构造的全局逻辑结构,是对数据库中全部数据的逻辑结构和特征的总体描述,是所有用户的公共数据视图(全局视图)。

(2) 外模式

外模式又称子模式或用户模式,对应于用户级。它是某个或某几个用户所看到的数据库的局部数据视图,是与某个应用有关的数据的逻辑表示。外模式是从模式中导出的一个子集,包含模式中允许特定用户使用的那部分数据。

(3) 内模式

内模式又称存储模式,对应于物理级。它是数据库中全体数据的内部表示或底层描述,是数据库最低一级的逻辑描述,它描述了数据在存储介质上的存储方式和物理结构,对应着实际存储在外存储介质上的数据库。

在 SQL Server 数据库体系结构中,外模式对应于视图,模式对应于基本表,内模式对应于存储文件。

1.1.4 数据库管理系统

在数据库系统阶段,为了科学地组织和存储数据,以便高效地获取和维护数据,出现了统一管理数据的专门软件系统——数据库管理系统。

数据库管理系统是帮助用户建立、使用和管理数据库的计算机软件系统,是位于用户和操作系统之间的数据库管理软件,也是数据库系统的核心。

数据库管理系统发展到今天,已有很多不同的数据库管理系统软件。根据应用领域的不同,可分为两大类:一类是大型网络数据库管理系统,常用的有 SQL Server、ORACLE、DB2、Sybase、Informix 等;另一类是小型桌面数据库管理系统,常用的有 Visual Foxpro、MS-Access、Dbase 等。



1.2 数据模型

模型是对现实世界特征的模拟和抽象,在日常生活中时常可见,如地形图沙盘、按一定比例缩小的建筑模型、航模飞机等,而数据模型是对现实世界数据特征的抽象。

为了反映事物本身及事物之间的联系,数据库中的数据必须有一定的结构,并且这种结构由数据模型来表示。数据库不仅要管理数据本身,而且要使用数据模型来表示出数据之间的联系。可见,数据模型是数据库管理系统用来表示事物及事物间联系的方法,一个具体的数据模型应当正确地反映出数据之间存在的整体逻辑关系。

数据模型应满足三方面的要求:

- 1) 能比较真实地模拟现实世界。
- 2) 容易为人所理解。
- 3) 便于在计算机上实现。

根据模型应用的目的不同,可以将模型划分为两类:一类是概念模型,它是按用户的观点来对数据和信息建模的,主要用于数据库设计,是独立于计算机系统的数据模型;另一类是数据模型,主要包括网状模型、层次模型和关系模型等,它是按计算机系统的观点来对数据建模的,主要用于对数据库管理系统的实现。

为了把现实世界中的具体事物抽象化,并且组织为某一 DBMS 支持的数据模型,人们常常首先将现实世界抽象为信息世界,然后将信息世界转换为机器世界。也就是说,首先把现实世界中的客观对象抽象为某一种信息结构,这种信息结构并不依赖于具体的计算机系统,也不是某一个 DBMS 支持的数据模型,而是概念模型,然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型,这一过程如图 1-2 所示。

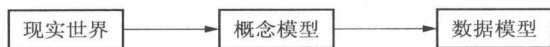


图 1-2 模型转换

① 概念模型

概念模型是对信息世界建模,要求方便、直接、简单、清晰地表示信息世界中的常用概念。概念模型的表示方法很多,其中最常用的是实体—联系 (Entity—Relationship) 方法,使用的工具是 E-R 图。

E-R 图中的三个基本要素:

(1) 实体

具体事物或抽象概念。如一个学生等。在 E-R 图中用矩形表示。

(2) 属性

实体所具有的某个特性。如学生的学号、姓名等。在 E-R 图中用椭圆形表示,并用无向边将其与相应的实体连接起来。

(3) 联系

实体间的相互联系。在 E-R 图中用菱形表示,并用无向边分别将其与有关的实体联系起来,同时,在无向边旁标上联系的类型。

实体间联系的 3 种类型:

1) 一对一联系 (1:1)。如果一个实体集 A 中的每一个实体,在实体集 B 中至多有一个 (也可以没有) 实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一联系。

例如,在一个班里,一个班级只有一个班长,而一个班长只在一个班中任班长职务,所以班长和班级是一对一联系。

2) 一对多联系 (1:n)。如果一个实体集 A 中的每一个实体,在实体集 B 中有 n 个实体与之联系,反之,实体集 B 中的每一个实体,在实体集 A 中至多有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多联系。

例如,在一个寝室里,一个寝室有多名学生,而一个学生只属于某一个寝室,所以寝室和学生是一对多联系。

3) 多对多联系 ($m:n$)。如果一个实体集 A 中的每一个实体,在实体集 B 中有 n 个实体与之联系,反之,实体集 B 中的每一个实体,在实体集 A 中也有 m 个实体与之联系,则称实体集 A 与实体集 B 具有多对多联系。

例如,在一个班级里,每一位任课教师有多名学生,而每一个学生也有多名任课教师,所以教师和学生是多对多联系。

② 数据模型

数据库领域中最常用的数据模型有三种,即层次模型、网状模型和关系模型。

下面对这 3 种模型进行介绍。

(1) 层次模型

用树形结构表示事物及其之间联系的模型称为层次模型。在这种模型中,数据被组织成由“根”开始的“树”,由根开始沿着不同的分支放在不同的层次上。如果不再向下分支,那么此分支序列中最后的结点称为“叶”。这种模型结构需满足以下两个条件:

- 1) 有且仅有一个结点无父结点,这个结点称为“根结点”。
- 2) 其他结点有且仅有一个父结点。

图 1-3 所示为一个学校行政机构的层次模型。

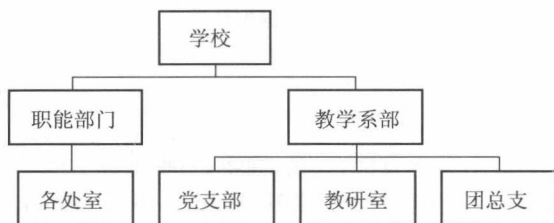


图 1-3 学校行政机构的层次模型

层次数据库采用层次模型作为数据的组织形式。典型的层次数据库管理系统是 1968 年 IBM 公司推出的 IMS 系统。

(2) 网状模型

用网状结构表示事物及其之间联系的模型称为网状模型。在这种模型中,网中的每一个结点代表一个事物类型。这种模型结构突破了层次模型的两点限制:允许结点有多于一个的父结点;可以有一个以上的结点没有父结点。因此,网状模型可以方便地表示各种类型的数据之间的联系。

图 1-4 所示为一个学校选课情况的层次模型。

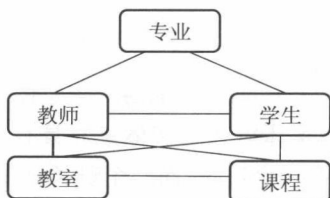


图 1-4 学校选课情况的层次模型

网状数据库采用网状模型作为数据的组织形式。典型的网状数据库管理系统是 20 世纪 70 年代美国数据系统研究会 CODASYL 下属的数据库任务组 DBTG 提出的 DBTG 系统。

(3) 关系模型

用二维表结构来表示事物及其之间联系的模型称为关系模型。在这个模型中,数据的逻辑结构是一张二维表,这种模型结构满足以下条件:

- 1) 每一列中的分量是类型相同的数据。
- 2) 列的顺序可以是任意的。
- 3) 行的顺序可以是任意的。
- 4) 表中的分量是不可再分割的最小数据项,即表中不允许有子表。
- 5) 表中的任意两行不能完全相同。

Course 的关系模型如表 1-1 所示。

表 1-1 Course 的关系模型

c_id	c_name	c_term	c_hour	c_score	c_category
030011	财务会计	1	64	4	必修
060003	C 语言程序设计	1	64	4	必修
060073	数据库基础及 SQL 应用	3	75	4	必修
061114	计算机导论	1	75	5	必修
062022	网页设计与制作	4	64	4	专业选修
083003	演讲口才	3	28	1.5	公共选修
980002	职业生涯与发展规划	1	32	1.5	必修

关系数据库采用关系模型作为数据的组织形式。关系模型与层次模型、网状模型的本质区别在于数据描述的一致性,模型概念单一。在关系模型数据库中,每一个关系都是一个二维表,使得描述事物的数据本身能够自然地反映它们之间的联系。而传统的层次模型和网状模型数据库是使用链接指针来存储数据和体现联系的。



1.3 关系数据库

目前数据库系统中支持的主要模型有:层次模型、网状模型、关系模型。关系模型由 E. F. Codd 于 1970 年首次提出。1970 年 E. F. Codd 在美国计算机学会会刊《Communication of the ACM》上发表的题为“A Relational Model of Data for Shared Data Banks”的论文,开创了数据库的新纪元。此后,他连续发表了多篇论文,这些论文为关系数据库的发展奠定了理论基础。关系数据库是采用关系模型作为数据的组织方式的数据库。关系数据库应用数学方法来处理数据库中的数据。自 20 世纪 80 年代以来,新推出的数据库管理系统几乎都支持关系模型,现在主流的数据库系统大多数都是基于关系模型的关系数据库系统(RDBS),SQL Server 就是其中之一。

关系模型三大组成要素:数据结构、关系的完整性规则和关系操作集合。

① 基本概念

关系:一个关系就是一张二维表。

属性:表中的一列称为一个属性,有时也称为一个字段(Field)。例如,读者信息表中的系别、班级、学号、姓名等字段。

域:一个属性的取值范围称为一个域。例如,性别的取值范围是“男”和“女”。

元组:表中的一行称为一个元组,有时也称为一条记录。

码:表中的某个属性,若它的值唯一地标识一个元组,则称该属性为“候选码”,若一个关系有多个候选码,则选定其中一个为主码,这个属性称为主属性。例如,Student 表中的“学号”的值唯一地标识了一个元组,所以称为“候选码”,也是主码。

关系模式:是对关系的描述,它包括关系名、组成该关系的属性名、属性到域的映像。通常简单记为:

关系名(属性名₁,属性名₂,……,属性名_n)。

例如,下表中 Student 表的关系模式可记为:

Student(s_dept,s_class,s_no,s_name,s_sex,s_birth,s_id)。