



21世纪计算机系列规划教材

- 遵循“循序渐进”和“理论联系实际”的教学原则
- 注重内容的实用性，培养学生的专业能力
- 适合高等院校应用型本科及高职高专“数据库应用技术”或“数据库原理及应用”课程

SQL Server 2008 数据库实用教程

姚一永 吕峻闽 主 编
靳紫辉 陈 婷 汤来锋 副主编

配备
电子教案



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

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

21 世纪计算机系列规划教材

SQL Server 2008 数据库 实用教程

姚一永 吕峻闽 主 编

靳紫辉 陈 婷 汤来锋 副主编

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书以微软公司最新数据库产品 SQL Server 2008 为平台,在简明介绍传统数据库理论的基础上,详细阐述了 SQL Server 2008 系统的安装、使用和维护,强化实践教学和综合应用,并给出 VB 和 C#平台上完整的学生学籍管理系统开发过程,有利于读者对照学习,使读者能轻松自如地在 SQL Server 2008 平台上开发出一个完整的应用系统。

本书清晰描述了 SQL Server 2008 的主要功能及使用方法,突出重点和难点,同时注重对实际技能的培养。每章后都有习题并提供相关实验内容。

本书既可作为计算机及相关专业师生的教材,也可供数据库开发技术人员使用。

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

SQL Server 2008 数据库实用教程 / 姚一永, 吕峻闽主编. —北京: 电子工业出版社, 2010.2

(21 世纪计算机系列规划教材)

ISBN 978-7-121-10279-0

I. S… II. ①姚…②吕… III. 关系数据库—数据库管理系统, SQL Server 2008—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2010) 第 014580 号

策划编辑: 徐建军

责任编辑: 徐 萍 文字编辑: 徐 磊

印 刷: 北京市顺义兴华印刷厂

装 订: 三河市双峰印刷装订有限公司

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: 16 字数: 409.6 千字

印 次: 2010 年 2 月第 1 次印刷

印 数: 5 000 册 定价: 29.00 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 zltz@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

前 言

信息技术的飞速发展大大推动了社会的进步,也逐渐改变了人们的生活、工作和学习方式。数据库技术和网络技术是信息技术中的重要支柱。自 20 世纪 70 年代以来,数据库技术的发展已使得信息技术的应用从传统的计算方式转变为现代化的数据管理方式。在现代社会中,数据库技术的应用无处不在。当今各种热门的信息系统开发领域,如管理信息系统、企业资源计划系统、供应链管理系统、客户关系管理系统、电子商务系统、决策支持系统、智能信息系统等,都离不开数据库技术强有力的支持。据国际上有关机构统计,目前世界上存在一千多万个正在使用的数据库。Microsoft SQL Server 系统是一个典型的关系型数据库管理系统,它起步于 20 世纪 80 年代后期,是微软品牌中的一个重要产品。微软公司在 Microsoft SQL Server 产品方面投入了巨大的开发力量,持续不断地研发新技术以满足用户不断增长和变化的需求,从而使得该产品功能越来越强大,用户使用越来越方便,系统的可靠性越来越高,应用也越来越广泛。

本书主要基于对本科生实施大 IT 教育的理念,积极探索对非 IT 专业大学生进行 IT 教育的有效途径,结合近年来我们的教学和开发实践经验,以当前流行的 SQL Server 2008 数据库平台为实例,详略结合,突出基本,既汲取现有教学资料中合理的内容,又在对传统教学内容的介绍上有所创新。

全书共分为 12 章,内容涵盖了 SQL Server 2008 系统应用的方方面面,不仅包括数据库经典理论的介绍,还包括 SQL Server 2008 数据库产品的详细安装方法,数据库和数据表的创建、修改和查询,T-SQL 语言的使用方法,存储过程,事务处理,数据完整性,数据备份和安全性管理等高级应用。对于各个知识点的讲解,都配有大量可实际运行的实例,可供读者边学习边实践,以方便读者快速、全面地掌握 SQL Server 的使用方法和技巧。

本书由姚一永、吕峻闽主编,靳紫辉、陈婷、汤来锋任副主编并负责编写相应各章节,郭进负责应用部分的程序设计。参加本书编写的还有黄纯国、杨大友、陈斌、陈小宁、王静、王玉晶、张英、宁涛等。同时西南财经大学天府学院信息技术教研中心和现代技术中心的各位老师为本书提供了许多帮助,在此,编者对以上人员致以最诚挚的谢意!在编写本书的过程中参考了相关的图书和资料,在此也对这些资料的相关作者深表感谢。

为了方便教学,可登录 www.hxedu.com.cn 免费下载与本书配套的教学资源。

由于编者水平有限,加之时间仓促,书中难免有不妥之处,敬请读者批评指正,以便在今后的修订中不断改进。

编 者

目 录

第 1 章 数据库基本概念和 SQL Server 2008 环境	(1)
1.1 数据库系统的基本概念	(1)
1.1.1 数据与数据处理	(1)
1.1.2 数据管理的发展历程	(2)
1.1.3 数据库系统	(4)
1.1.4 数据模型	(5)
1.1.5 逻辑结构设计	(9)
1.2 SQL Server 2008 简介	(11)
1.3 SQL Server 2008 的安装	(15)
1.3.1 SQL Server 2008 的环境需求	(15)
1.3.2 SQL Server 2008 的安装步骤	(16)
1.4 配置 SQL Server 2008	(25)
1.5 SQL Server 2008 管理工具	(29)
1.6 SQL Server 配置管理器	(30)
1.7 SQL Server 2008 系统数据库	(33)
1.8 Transact-SQL 语言简介	(34)
习题	(34)
第 2 章 数据库和表创建	(36)
2.1 SQL Server 基本概念	(36)
2.1.1 数据库	(36)
2.1.2 表	(36)
2.2 用界面方式创建数据库和表	(39)
2.2.1 数据库的创建、修改和删除	(39)
2.2.2 表的创建、修改和删除	(44)
2.3 使用命令方式创建数据库和表	(50)
2.3.1 使用 CREATE DATABASE 创建数据库	(50)
2.3.2 使用 ALTER DATABASE 修改数据库	(53)
2.3.3 使用 DROP DATABASE 删除数据库	(53)
2.3.4 使用 CREATE TABLE 创建表	(53)
2.3.5 使用 ALTER TABLE 修改表	(55)
2.3.6 使用 DROP TABLE 删除表	(56)
习题	(56)
第 3 章 表数据操作	(57)
3.1 界面操作表数据	(57)
3.1.1 插入记录	(57)
3.1.2 删除记录	(58)
3.1.3 修改记录	(58)

3.2 命令操作表数据	(58)
3.2.1 使用 INSERT 语句插入表数据	(58)
3.2.2 使用 DELETE 或 TRUNCATE 语句删除数据	(60)
3.2.3 使用 UPDATE 语句修改数据	(61)
习题	(61)
第 4 章 数据库的查询和视图	(62)
4.1 连接、选择和投影	(62)
4.1.1 选择 (Selection)	(62)
4.1.2 投影 (Projection)	(63)
4.1.3 连接 (Join)	(63)
4.2 数据库的查询	(65)
4.2.1 选择列	(65)
4.2.2 选择行	(68)
4.2.3 FROM 子句	(74)
4.2.4 连接	(74)
4.2.5 数据汇总	(74)
4.2.6 排序	(76)
4.3 视图	(77)
4.3.1 视图概念	(77)
4.3.2 创建视图	(77)
4.3.3 查询视图	(80)
4.3.4 更新视图	(80)
4.3.5 修改视图的定义	(83)
4.3.6 删除视图	(84)
4.4 游标	(84)
4.4.1 游标概念	(84)
4.4.2 声明游标	(85)
4.4.3 打开游标	(86)
4.4.4 读取数据	(86)
4.4.5 关闭游标	(87)
4.4.6 删除游标	(87)
习题	(87)
第 5 章 T-SQL 语言	(89)
5.1 常量、变量与数据类型	(89)
5.1.1 常量	(89)
5.1.2 数据类型	(91)
5.1.3 变量	(93)
5.2 运算符与表达式	(95)
5.3 流程控制语句	(100)
5.3.1 IF...ELSE 语句	(100)
5.3.2 无条件转移 (GOTO) 语句	(101)

5.3.3	WHILE 语句	(101)
5.3.4	RETURN 语句	(102)
5.3.5	WAITFOR 语句	(102)
5.4	系统内置函数	(103)
5.4.1	系统内置函数介绍	(103)
5.4.2	常用系统内置函数	(103)
5.5	用户定义函数	(112)
5.5.1	用户定义函数的定义与调用	(112)
5.5.2	用户定义函数的删除	(114)
	习题	(115)
第 6 章	索引与数据完整性	(116)
6.1	索引	(116)
6.1.1	索引的分类	(116)
6.1.2	索引的创建	(116)
6.1.3	索引的删除	(118)
6.2	默认值约束及默认值对象	(118)
6.2.1	在表中定义及删除默认值约束	(119)
6.2.2	默认值对象的定义、使用与删除	(120)
6.3	数据完整性	(121)
6.3.1	数据完整性的分类	(121)
6.3.2	域完整性的实现	(122)
6.3.3	实体完整性的实现	(124)
6.3.4	参照完整性的实现	(129)
	习题	(132)
第 7 章	存储过程和触发器	(133)
7.1	存储过程	(133)
7.1.1	存储过程的类型	(133)
7.1.2	用户存储过程的创建与执行	(133)
7.1.3	用户存储过程的编辑修改	(138)
7.1.4	用户存储过程的删除	(139)
7.2	触发器	(140)
7.2.1	利用 SQL 语句创建触发器	(140)
7.2.2	利用 SQL Server Management Studio 创建触发器	(142)
7.2.3	触发器的修改和删除	(142)
	习题	(144)
第 8 章	备份恢复与导入/导出	(145)
8.1	备份和恢复概述	(145)
8.1.1	备份和恢复需求分析	(145)
8.1.2	数据库备份和恢复的基本概念	(145)
8.2	备份操作和备份命令	(146)
8.2.1	创建备份设备	(146)

8.2.2 使用对象资源管理器进行数据库备份	(148)
8.3 恢复操作和恢复命令	(149)
8.3.1 检查点 (check point)	(149)
8.3.2 数据库的恢复命令	(150)
8.3.3 使用对象资源管理器恢复数据库	(150)
8.4 导入/导出	(152)
8.4.1 导入/导出概念	(152)
8.4.2 使用 BCP 实用程序导入/导出数据	(152)
8.4.3 使用导入/导出向导	(153)
习题	(157)
第9章 SQL Server 2008 安全管理	(158)
9.1 安全管理概述	(158)
9.2 SQL Server 验证模式	(159)
9.2.1 Windows 身份验证模式	(160)
9.2.2 混合身份验证模式	(161)
9.2.3 设置身份验证模式	(161)
9.3 数据库账号	(163)
9.3.1 服务器的登录账号	(163)
9.3.2 数据库用户账户	(165)
9.4 固定服务器角色	(166)
9.4.1 服务器角色概述	(167)
9.4.2 服务器角色管理	(168)
9.5 数据库角色	(169)
9.5.1 固定数据库角色	(169)
9.5.2 自定义数据库角色	(172)
9.5.3 应用程序角色	(173)
9.6 数据库权限	(174)
9.6.1 权限概述	(174)
9.6.2 管理权限	(175)
9.6.3 继承权限	(177)
习题	(177)
第10章 其他	(178)
10.1 复制	(178)
10.2 事务	(180)
10.2.1 什么是事务	(180)
10.2.2 ACID 属性	(181)
10.2.3 使用事务	(182)
10.2.4 事务的举例	(188)
10.2.5 分布式事务	(189)
10.3 自动化管理基础	(190)
10.3.1 自动化管理概述	(190)

10.3.2	自动化管理元素	(191)
10.4	配置数据库邮件	(193)
10.4.1	数据库邮件概述	(193)
10.4.2	配置数据库邮件过程	(193)
10.4.3	使用邮件配置文件	(194)
10.5	操作员	(195)
10.6	警报	(196)
10.6.1	标准事件警报	(196)
10.6.2	自定义事件警报	(198)
10.6.3	性能警报	(199)
10.6.4	WMI 警报	(200)
10.7	作业	(201)
10.7.1	概述	(201)
10.7.2	创建本地作业	(202)
10.7.3	创建多服务器作业	(204)
10.8	维护计划向导	(205)
	习题	(207)
第 11 章	VB/SQL Server 开发与编程	(208)
11.1	Visual Basic 数据库访问方法	(208)
11.1.1	Data 控件访问 SQL Server 数据库	(208)
11.1.2	ADO 访问 SQL Server 数据库	(208)
11.2	ODBC 数据源配置和可视化数据管理器	(210)
11.2.1	ODBC 数据源配置	(210)
11.2.2	可视化数据管理器	(214)
11.3	VB/SQL Server 编程——学生信息管理系统	(215)
11.3.1	用户界面设计	(216)
11.3.2	Data 数据控件设置和数据绑定	(218)
11.3.3	VB/SQL 数据库代码实现	(220)
第 12 章	C#.NET/SQL Server 开发与编程	(225)
12.1	ADO.NET 概述	(225)
12.1.1	ADO.NET DataSet 组件	(225)
12.1.2	.NET 数据提供程序集	(225)
12.2	C#.NET 数据库操作关键类	(226)
12.2.1	SqlConnection	(226)
12.2.2	SqlDataAdapter	(226)
12.2.3	SqlCommand	(226)
12.2.4	SqlDataReader	(227)
12.3	C#.NET/SQL Server 编程——学生信息管理系统	(227)
12.3.1	用户界面设计	(227)
12.3.2	C#.NET/SQL 数据库代码实现	(232)

第1章 数据库基本概念和 SQL Server 2008 环境

本章先主要介绍数据库的基本概念、数据管理的发展过程、数据模型的基本知识、数据库系统的组成部分,接着介绍 SQL Server 2008 的入门知识,包括 SQL Server 2008 的特点、安装方法、文件系统,以及它的管理工具和配置方法等。

通过对本章的学习,可以了解为什么要使用数据库,以及数据技术的重要性,也可为以后章节的学习打下基础。

- 理解关系数据库的组成和基本概念
- 理解实体与关系模型的概念
- 掌握安装 SQL Server 2008 的方法
- 掌握配置 SQL Server 2008 的方法
- 了解 SQL Server 2008 的重要管理工具

1.1 数据库系统的基本概念

当人们收集大量数据,并抽取出某个应用所需要的数据时,应将其保存起来以供进一步加工处理,保存数据的地方则称为数据库(Database, DB),即存放数据的仓库,只是这个仓库是建立在计算机的存储设备上的。在计算机中大量的数据通常不会被随机存放,而是按一定的格式有组织地存放,所以数据库的定义是长期储存在计算机内、有组织的、可共享的大量数据的集合。数据库中的数据按一定的数据模型组织、描述和储存,具有较小的冗余度、较高的数据独立性和易扩展性,并可以为各种用户共享。数据库中的数据具有永久存储、有组织、可共享三个基本特点。本节将重点介绍数据库系统中常用的几个基本概念,包括数据、数据处理、数据库和数据库管理系统,以方便读者更好地理解数据库系统的功能和特点。

1.1.1 数据与数据处理

数据(Data)是人们用于记录客观事物情况的符号表示,是表示现实世界中的物体、事件、位置、概念等的未经加工的原始素材,是通过物理观察得来的事实和概念。例如,“四川省内8所高校2009年本科扩招近3000人”,其中的数据除了“8所”、“2009年”和“3000人”外,还包括“四川”、“高校”和“本科”。数据的概念在数据处理领域已经被大大拓宽了,其表现形式不仅仅是数字,也可以是文字、图形、图像、声音等。数据也是计算机领域中的术语,属于软件范畴。软件是由程序与数据两部分构成的,数据是计算机软件中程序加工的原料与结果。

信息(Information)是关于现实世界事物的存在方式或运动状态的反映的综合,是数据中所包含的意义,是加工处理后的数据,是数据所表达的内容。具体来说,是一种被加工为特定形式的数据,但这种数据形式对接收者来说是有意义的,而且对当前和将来的决策具有明显的或实际的价值。数据是用来记录信息的可识别的符号,是信息的具体表现形式。描述事物的符号记录称为数据。这些符号可以经过数字化处理后存入计算机。目前数据的概念在数据处理领域中已被大大地拓宽了,因为计算机存储和处理的对象十分广泛。表示这些对象的数据也越来越复杂了。可用多种不同的数据形式表示同一信息,而信息不随数据形式的不同而改变。数据

是信息的符号表示或载体,信息是数据的内涵,是数据的语义解释。数据的表现形式还不能完全表达其内容,不能反映其信息,需要经过解释,数据和数据的解释是不可分的。例如,任意一个数字 19,可以表示一个人的年龄,也可以是 19 号、19 排或其他解释。同样一条数据“王伟,男,1991.06,工商”,可以表达这样一个信息,王伟是一个工商专业的学生,男,出生年月为 1991 年 6 月,也可以对该数据作另外的解释,王伟,男,1991 年 6 月参加工作,是工商专业的老师。数据的解释是对数据含义的说明,数据的含义称为数据的语义,数据与其语义是不可分的。因此,数据是符号化的信息,信息是语义化的数据,如上例中的数据 19 被赋予了特定的语义,就具有了传递信息的功能。一般来说,从信息转换为数据需要进行特征抽取,而从数据还原为信息需要经过数据解释。在一些不是很严格的场合,对信息和数据没有严格的区分,有时甚至相互替换使用,如信息处理与数据处理、信息采集与数据采集等。

数据处理是将数据转换成信息的过程,包括对数据的收集、存储、检索、加工、变换、传输等一系列活动。其基本目的是从大量的、杂乱无章的、难以理解的原始数据中抽取和推导出有价值的信息。可用下式简单地表示数据、数据处理与信息的关系:

数据→数据处理→信息

数据是输入,而信息是输出结果。“数据处理”的真正含义是为了产生信息而处理数据。数据处理的中心问题是数据管理。

1.1.2 数据管理的发展历程

数据库技术是伴随着数据管理任务的需要而产生的,数据处理包括数据的收集、存储、检索、加工、变换、传输等一系列活动,而数据管理则是对数据进行分类、组织、编码、存储、检索和维护。数据管理是数据处理的中心问题。人们研制计算机的初衷就是为了进行复杂的数据运算,但是随着计算机技术的发展,其应用远远超出了这个范围。在计算机软件、硬件的基础上,在需求的推动下,数据管理技术经历了人工管理、文件系统管理和数据库管理(分布式)三个阶段。

1. 人工管理阶段

20 世纪 50 年代初,计算机主要用于科学计算,没有操作系统和管理数据的软件,也没有磁盘等直接存取数据的设备,只有纸片、卡片、磁带,数据以符号的形式存放在穿孔卡片上,并以批处理方式运行。数据的组织和管理只能由程序员以人工手段进行,在计算机上无法保存、检索数据。不同程序之间的数据是相互孤立的,逻辑上没有任何联系。人工管理阶段的特点如下。

(1) 数据不独立。一组数据对应一组程序,这就使得程序依赖于数据,一旦数据的逻辑结构或者物理结构发生变化,程序就必须做相应的修改,这样就加重了程序员的负担。

(2) 数据不保存。当时的计算机主要用于科学计算,数据不需要长期保存。计算某一问题时导入,用完即删除,应用软件与系统软件均如此。

(3) 数据非共享。数据是面向应用程序的,一组程序对应一组数据,当多个程序涉及相同的数据时,也无法互相利用、互相参照,必须各自定义,因此程序与程序间存在着大量的冗余数据。

(4) 数据无管理。数据由相应的程序管理,没有专门的管理数据的软件,应用程序除了要规定数据的逻辑结构外,还要规定数据的物理结构,包括存储结构、存储方法、输入输出方法,这就给应用程序设计人员增加了很大的负担。

2. 文件系统管理阶段

20 世纪 50 年代后期到 60 年代中期,出现了磁盘、磁鼓等直接存储设备。处理方式上不仅有了批处理,而且能够联机实时处理。计算机在一种称为操作系统的软件的指挥下工作,数据以文件的形式存储在计算机上,操作系统中有了专门用于数据管理的软件,一般称为文件系统。

在文件系统中,文件被某一应用所有,数据与程序的独立性差,数据的逻辑结构一旦改变就必须修改应用程序,文件系统只是减轻了程序员对物理设备存取负担,它并不理解数据的语义,只负责存储,数据的语义信息只能由程序来解释。也就是说,数据收集以后怎么组织,以及数据取出来之后按什么含义应用,只有全权管理它的程序知道。一个应用若想共享另一个应用生成的数据,必须同另一个应用沟通,了解数据的语义与组织方式。在这样的系统中还存在很多弊端。像在学生管理系统中,学校中有教务、学工等诸多部门,分别管理学生的学籍、选课、人事等数据。各个部门使用不同的应用程序,每个应用程序管理一组数据,即数据是面向某个应用的,数据之间是相互隔离的,这致使相同的信息可能在几个地方重复存储,导致数据的冗余度较大,这种冗余除了导致存储和访问开销增大以外,还可能导致数据不一致,即同一数据的不同副本的值不一样。用文件系统管理数据具有如下特点。

(1) 数据可以长期保存。这一阶段,计算机不仅用于科学计算,还大量用于信息管理。由于计算机大量用于数据处理,数据需要长期保留在外存上反复进行保存、查询、修改、插入和删除等基础性工作,因此在计算机中长期保存数据成为现实。

(2) 由文件系统管理数据。由专门的软件即文件系统进行数据管理,文件系统把数据组织成相互独立的数据文件,利用“按文件名访问,按记录进行存取”的管理技术,对文件进行修改、插入和删除的操作。文件系统实现了记录内的结构性,但整体无结构。程序和数据之间由文件系统提供存取方法进行转换,使应用程序与数据之间有了一定的独立性,程序员可以不必过多地考虑物理细节,而将精力集中于算法。而且数据在存储上的改变不一定反映在程序上,这大大节省了维护程序的工作量。但是,文件系统仍存在一些缺点。

(3) 数据共享性差,冗余度大。在文件系统中,一个文件基本上对应于一个应用程序,即文件仍然是面向应用的。当不同的应用程序具有部分相同的数据时,也必须建立各自的文件,而不能共享相同的数据,因此数据的冗余度大,浪费存储空间。同时由于相关数据的重复存储、各自管理,容易造成数据的不一致性,给数据的修改和维护带来了困难。

(4) 数据独立性差。文件系统中的文件是为某一特定应用服务的,文件的逻辑结构对该应用程序来说是优化的,因此要想对现有的数据再增加一些新的应用会很困难,系统不容易扩充。一旦数据的逻辑结构改变,必须修改应用程序,修改文件结构的定义。应用程序的改变,如应用程序改用不同的高级语言等,也将引起文件的数据结构的改变。因此数据与程序之间仍缺乏独立性。可见,文件系统仍然是一个不具有弹性的无结构的数据集合,即文件之间是孤立的,不能反映现实世界事物之间的内在联系。

3. 数据库管理阶段

20 世纪 60 年代后期,计算机应用于管理的规模更加庞大,数据量急剧增加,硬件技术的发展使计算机联机存取大量数据成为可能。硬件价格下降,而软件价格上升,使开发和维护系统软件的成本增加。计算机上存储的数据量急剧增加,要求计算机能够联机实时处理各种数据,解决多用户、多应用共享数据的需求,文件系统的管理方法已无法适应开发应用系统的需要。因此为了解决多用户、多应用的共享数据需求,使数据可以得到更广泛的应用,数据库技

术应运而生。数据库从诞生之日起就迅速发展,从简单到复杂,从单机到网络,与网络通信技术、面向对象程序设计技术、并行计算技术等相互渗透,相互融合,逐渐发展成为完整的系统,称为数据库系统。数据库系统管理数据比文件系统具有明显的优点,从文件系统到数据库系统,是数据管理技术上质的飞跃。用数据库系统管理数据具有如下特点。

(1) 数据结构化。在数据库中,数据不再像文件系统中的数据那样从属于某个特定的应用,而是按照某种数据模型组织成为一个结构化的数据整体。它不仅描述了数据本身的特性,即数据内部的结构化,而且描述了数据与数据之间的种种联系,这使数据库系统中实现了整体数据的结构化。

(2) 实现数据共享。在数据库系统中,由于数据是面向整个系统的,数据库中的数据实现了按某种数据模型组织为一个结构化的数据,实现了多个应用程序、多个用户甚至多种语言能够共享一个数据库中的数据(目前数据共享已可实现更大范围的共享),所以大大提高了数据的利用率,也提高了工作效率。数据共享是数据库先进性的重要体现。

(3) 数据冗余低,易扩充。由于数据可共享,减少了重复的数据存储,节约了存储空间,而且数据的共享还能避免数据之间的不相容性与数据的不一致性。由于数据面向整个系统,是有结构的数据,因此它不仅可以被多个应用共享使用,而且容易增加新的应用,使得数据库系统弹性大,易于扩充,可以适应各种用户的要求。选取整体数据子集或加上一部分数据,便可用于不同的新应用需求。

(4) 数据独立性高。数据库的另一个主要特性之一是实际的数据与使用的数据的程序是分离的,即数据与程序相互独立,互不依赖,不会因为一方的改变而改变另一方,这样大大简化了应用程序设计与维护的工作量,同时数据也不会随程序的结束而消失,可以长期保留在系统中。

数据的独立性包括数据的物理独立性和数据的逻辑独立性,它们的实现是由数据库管理系统的二级映像功能来保证的。

(5) 数据由 DBMS 统一控制。数据库管理系统(DBMS)是统一控制和管理数据的软件。由于数据库中数据的共享,可能会造成多个用户同时存取数据,甚至会在同一时间存取同一数据,如果不进行数据控制,数据库中的数据就不正确了。为了保证数据的完整性、安全性,数据库管理系统必须具备以下功能。

① 数据的安全性保护。数据的安全性是指每个用户只能按指定方式使用和处理指定数据;DBMS 可保护数据以防止不合法使用造成的数据的泄密和破坏。

② 数据的完整性检查。系统通过设置一些完整性规则以保证数据的正确性、有效性和相容性,从而将数据控制在有效的范围内,或保证数据之间满足一定的关系。

③ 并发控制。当多个用户同时对数据进行修改、存取时,对多用户的并发进程加以控制和协调,防止相互干扰而得到错误的结果或破坏数据库的完整性。

④ 数据库恢复。在计算机系统出现各种故障时,DBMS 可将数据库从错误状态恢复到某一已知的正确状态。

1.1.3 数据库系统

数据库系统在今天的信息社会中有着广泛的应用,它是信息技术的核心。数据库系统一般由数据库、数据库管理系统、数据库管理员、硬件平台及软件平台组成。DBMS 是数据库系统的基础和核心。

1. 数据库

顾名思义,数据库是在计算机存储设备上按一定格式存储数据的仓库。科技飞速发展的今天,人们借助计算机和数据库来保存和管理大量的数据,以便更有效地利用这些信息资源。

长期储存在计算机内的、有组织的、可共享的数据集合称为数据库。数据库中的数据按一定的数据模型组织、描述和储存,具有较小的冗余度、较高的数据独立性和易扩展性,并可为各种用户共享。

2. 数据库管理系统

数据库管理系统是管理数据库的系统软件,它负责数据库中的数据组织、数据操纵、数据维护,并保护控制数据不受破坏。

3. 数据库管理员

数据库管理员(DataBase Administrator, DBA)是一类特殊的用户,他担负着全面管理和控制数据库系统的任务。其主要职责包括数据库设计、数据库维护和改善系统性能,提高系统效率。随着时间的推移,用户的需求会发生变化,DBA 还需对数据库进行较大的改造,包括修改当初的部分设计,即进行数据库的重构造。

4. 硬件平台

在数据库系统中,数据量都很大,并且由于数据库管理系统功能的完善使得其自身规模也很大,因此数据库系统对硬件资源的要求较高。这包括要求计算机具有足够大的内存,用于存放操作系统、DBMS 的核心模块、数据缓冲区及应用程序,有足够大和安全性好的磁盘或磁盘阵列等存储设备存放数据,有足够的存储设备做数据备份,同时还要求有较高的通信能力,以提高数据传送能力。

5. 软件平台

数据库系统的软件主要包括以下几项。操作系统,它作为基础软件支持 DBMS 运行;数据库管理系统,建立、使用、维护和配置数据库的系统软件;用于开发应用程序的高级语言及其编译系统;以数据库管理系统为核心的应用开发工具;为某个特定应用环境开发的数据库应用系统。

1.1.4 数据模型

在现实世界中许多模型,这些模型都是对现实世界中某个对象特征的模拟和抽象,如飞机模型、汽车模型就是对现实世界的飞机和汽车的模拟和抽象。数据模型也是一种模型,只不过它是对现实世界里的数据特征的抽象。由于计算机不能直接处理现实世界的具体事物,因此人们必须先把具体事物转换成计算机能够处理的数据,即把现实世界中具体的人、物、活动、概念等用数据模型来抽象、表示和处理,即先进行数字化。这些就需要我们建立一个数据模型。例如,学生管理信息系统,人们通常应了解在该系统中应有哪些数据,这些数据之间有什么联系,以及如何组织这些数据并将其合理地存放在数据库中,以便有效地对其进行管理。

目前的数据库系统均是基于某一数据模型的,数据模型是数据库的核心和基础。因此,我们必须掌握数据模型的基本概念及基本知识。

1. 数据模型的三要素及分类

数据模型具有以下 3 个要素。

1) 数据结构

数据结构用于描述系统的静态特性,即组成数据库的对象类型,研究与数据类型、内容、

性质有关的对象。数据结构包括两方面的内容：一是数据本身，如关系模型中的关系、域、属性等；二是数据之间的联系。

2) 数据操作

描述系统的动态特征，即对数据库中的对象允许执行的操作的集合，主要有检索和更新（包括插入、删除、修改）两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则（如优先级），以及实现操作的语言。

3) 数据的约束条件

数据的约束条件是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和存储规则，用以限定符合数据模型的数据库状态及状态的变化，以保证数据的正确、有效、一致。此外，数据模型还应该提供定义完整性约束条件的机制。

为了建立数据模型，首先需要对特定的某一现实世界进行抽象，即找出现实世界的特征，然后继续整理并以数据的形式存储在计算机世界的数据库中。这样抽象出的数据模型应满足 3 个方面的要求：一是能比较真实地模拟现实世界；二是容易为人所理解；三是便于在计算机上实现。

但现实世界和计算机世界存在很大的差异，直接将现实世界中的语义映射到计算机世界是非常困难的。要同时满足这 3 方面要求的数据模型目前还很难实现，即能真实模拟现实世界的模型常常无法在计算机中直接实现。所以在数据库的设计中引入一个信息世界作为通向计算机世界的桥梁。

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型，人们常常先将现实世界抽象为信息世界，然后将信息世界再转换为计算机世界。而将现实世界的客观对象转换为信息世界的信息抽象并不依赖于具体的计算机系统，它不是某个 DBMS 支持的数据模型，仅是一个概念级的模型；然后再把概念模型转换为计算机上某一特定的 DBMS 支持的数据模型。

由此，根据不同的应用目的，数据模型被分为两类。一类是概念数据模型（简称概念模型），它是按用户的观点来对数据和信息建模，用于把现实世界抽象到信息世界；还有一类是结构数据模型（简称数据模型），它是按计算机系统的观点对数据建模，用于把信息世界转换到计算机世界。两类数据模型的关系如图 1-1 所示。

从现实世界到信息世界，即建立概念模型，由数据库设计人员完成。从概念模型到数据模型的转换可以由数据库设计人员完成，也可以用专门的数据库设计工具辅助设计人员完成，各数据库管理系统均提供该工具。

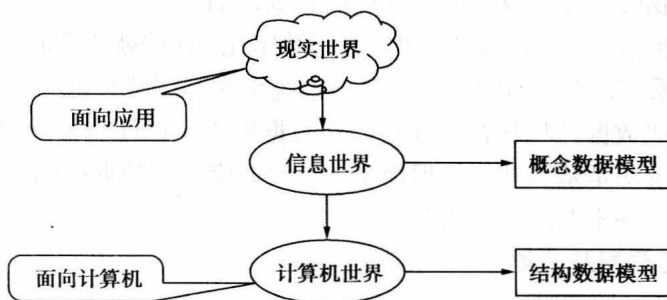


图 1-1 两类数据模型

2. 概念模型

概念模型是现实世界到信息世界的第一层抽象,是到计算机世界的一个中间层次,是数据库设计人员进行数据库设计的一个工具,同时也是数据库设计人员和用户之间进行交流的语言,因此就要求概念模型不仅要具有较强的语义表达能力,能够方便、直接地表达应用系统中的各种语义,还应该简单、清晰、易于用户理解。

概念模型只是将现实世界的客观对象抽象为某种信息结构,这种信息结构并不依赖于具体的计算机系统;而对应于计算机世界的模型则由数据模型描述,数据模型是数据库中实体之间及其联系的抽象描述。

数据模型是从计算机实现的观点来对数据建模,一般都有严格的形式化定义,以便于在计算机上实现。常见的数据模型包括层次模型、网状模型、关系模型、面向对象模型及对象关系模型等,其中关系模型是目前最流行的数据库数据模型。

数据模型是严格定义的一组概念的集合。这些概念精确地描述了系统的静态特性、动态特性和完整性约束条件。

实体-联系模型(E-R 模型)是基于对现实世界的这样一种认识而建立的,即世界是由一组称为实体的基本对象及这些对象间的联系组成的。此模型通过允许对应用系统模式进行定义来帮助数据库进行设计,应用系统模式代表了数据库的全局逻辑结构。E-R 模型是一种语义模型,模型的语义方面主要体现在模型力图去表达数据的意义。E-R 模型在将现实世界的含义和相互关联映射到概念模式方面非常有用,因此,许多数据库设计工具都利用了 E-R 模型的概念。

3. 实体集

实体(Entity)是现实世界中客观存在并可相互区别的“物体”或“事件”。例如,学校中的每个同学、老师都是一个实体。每个实体有一组特征,其中一部分特征的取值可以唯一标示实体,如学生的身份证号、学号。实体可以是具体的事或物体,也可以是抽象的概念,如银行客户的贷款等。

实体集是具有相同类型及相同性质(或属性)的实体集合。例如,全体学生可被定义为学生实体集(Student),同样,教师实体集(Teacher)表示所有教师信息的集合。

属性(Attribute),实体所具有的某一特征称为属性。属性是实体集中每个成员具有的描述性性质,因此实体可以通过一组属性来表示。学生实体可能有学号、姓名、性别、出生日期、专业、家庭地址等属性,而对每个属性来说,各实体都有自己的属性值。

码(Key),能够唯一标示实体的属性集称为码,如身份证号、学号都可以作为学生实体的码。

域(Domain),每个属性具有一组相同数据类型的值的集合,该集合称为该属性的域或值集,如性别取值为男、女,学生姓名的域是某个长度内的所有字符串的集合。

实体型(Entity Type),一组具有相同属性的实体必然具有共同的特征和性质。用实体名及其属性名集合来抽象和刻画同类实体,称为实体型。例如,课程(课程号,课程名,学分)就是一个实体型。

因此数据库包括了一组实体集,每个实体集中包括一些相同类型的实体。在现实世界中,事物与事物之间,以及事物内部都是有联系的,而这些联系反映在信息世界中即表现为实体之间的联系和实体内部的联系。实体之间的联系指不同实体集之间的联系,而实体内部的联系通常指组成实体的各属性之间的联系。

4. 实体之间联系的类型

联系是多个实体间的相互关联。例如,定义学生与课程两个实体的关联,这一联系指学生

实体中的某个学生可以选择课程实体中的课程，联系集是同类联系的集合。

两个实体集 A 和 B 之间的联系可以分为 3 类。

1) 一对一联系 (1:1)

对于实体集 A 中的一个实体最多同实体集 B 中的一个实体相联系，B 中的一个实体也最多同 A 中的一个实体相联系，则称实体集 A 与实体集 B 是一对一联系，记为 1:1。

例如，在学生信息管理系统中，一个班级只有一个班长，而一个班长只在一个班中任班长，则班级与班长之间就是一对一联系，如图 1-2 左图所示。

2) 一对多联系 (1:n)

对于实体集 A 中的一个实体可以同实体集 B 中的任意数目的实体相联系，而 B 中的一个实体最多只同 A 中的一个实体相联系，则称实体集 A 与实体集 B 是一对多联系，记为 1:n。

例如，一个辅导员可以管理多个班级，而一个班级只有一个辅导员；一个教研室可以有多个老师，而一个老师只属于一个教研室，则辅导员与班级，教研室与老师之间具有一对多联系，如图 1-2 中图所示。

3) 多对多联系 (m:n)

对于实体集 A 中的一个实体可以同实体集 B 中的任意数目的实体相联系，反之，实体集 B 中的一个实体也可以同 A 中的任意数目的实体相联系，则称实体集 A 与实体集 B 具有多对多联系，记为 m:n。

例如，一个学生可以同时选修多门课程，而一门课程同时有若干个学生选修，则学生与课程之间具有多对多联系，如图 1-2 右图所示。

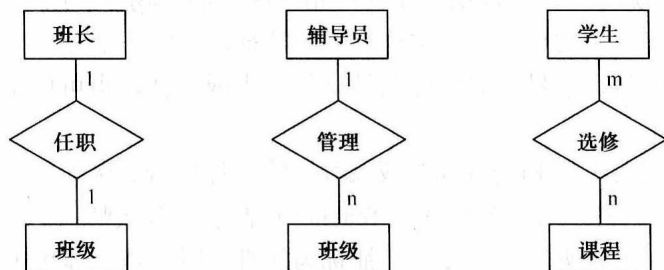


图 1-2 两个实体集之间的联系

在现实中，还存在两个以上的实体集之间的一对一、一对多、多对多联系。这种两个实体集以上的联系可以清晰地表示出几个实体集参与到一个联系集中，而在对应的二元联系中，难以体现这样的参与性约束。

例如，一个学生可以同时选修多门课程，一个老师在教一门课时对应多个学生，同时一门课程可以有若干个学生选修，一个老师可以教多门课，如图 1-3 左图所示。

另一个例子是工厂中常见的供应商、项目、零件 3 个实体集之间的多对多联系。一个供应商可以给多个项目供应多种零件，每个项目可以使用多个供应商供应的多种零件，每种零件可由多个供应商供给，则供应商、项目、零件三者之间是多对多的联系，如图 1-3 右图所示。

同样，在同一个实体集内部的各个实体之间也可以存在一对一、一对多、多对多联系。比如，教师实体集内部即存在领导与被领导的联系。