

SQL Server教程 从基础到应用

郑阿奇 主编



机械工业出版社
China Machine Press

计算机基础课程系列教材

SQL Server教程 从基础到应用

郑阿奇 主编



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

SQL Server 教程: 从基础到应用 / 郑阿奇主编. —北京: 机械工业出版社, 2015.3
(计算机基础课程系列教材)

ISBN 978-7-111-49601-4

I. S… II. 郑… III. 关系数据库系统—高等学校—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2015) 第 047571 号

本书以 Microsoft SQL Server 2012 中文版为平台, 系统地介绍数据库基础、SQL Server 的主要功能和综合应用等内容。SQL Server 教程部分主要包括数据库创建、表的创建和操作、数据库的查询和视图、游标、T-SQL、索引、数据完整性、存储过程和触发器、备份与恢复、系统安全管理、SQL Server 其他功能等。实验部分训练 SQL Server 基本操作和基本命令。实习部分通过创建学生成绩管理系统, 介绍使用目前流行的开发平台 (包括 PHP 5、Java EE、ASP.NET 4.5、VB 6.0) 操作 SQL Server 2012 数据库。

本书可作为大学本科、高职高专数据库课程教材和社会培训教材, 也可供广大数据库应用开发人员参考阅读。

出版发行: 机械工业出版社 (北京市西城区百万庄大街 22 号 邮政编码: 100037)

责任编辑: 余 洁

责任校对: 殷 虹

印 刷: 北京诚信伟业印刷有限公司

版 次: 2015 年 3 月第 1 版第 1 次印刷

开 本: 185mm×260mm 1/16

印 张: 23

书 号: ISBN 978-7-111-49601-4

定 价: 45.00 元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88378991 88361066

投稿热线: (010) 88379604

购书热线: (010) 68326294 88379649 68995259

读者信箱: hzjsj@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光 / 邹晓东

前 言

20 世纪 80 年代后期, Microsoft、Sybase 和 Ashton-Tate 三家公司共同开发了最初的 SQL Server。1988 年, 该产品被移植到 OS/2 上; 1992 年, SQL Server 被移植到 Windows NT 平台上。1993 年, SQL Server 4.2 面世, 它是桌面数据库系统, 虽然其功能相对有限, 但是采用 Windows GUI 为用户提供了易于使用的界面。Microsoft 公司专注于 Windows NT 平台上的 SQL Server 开发, 并于 1995 年发布了 SQL Server 6.05, 该版本提供了廉价的可以满足众多小型商业应用的数据库方案。后来 SQL Server 不断更新, 先后推出 6.5 版、7.0 版、2000 版、2005 版、2008 版、2012 版和 2014 版。目前 SQL Server 已经是市场上最流行的大中型关系数据库管理系统。

为了适应市场的需要, 我国高校的许多专业都开设了 SQL Server 数据库管理系统课程。本书以 Microsoft SQL Server 2012 为平台, 结合作者近年来教学与应用开发的实践, 在简单介绍数据库基础后, 系统介绍 SQL Server, 然后介绍 SQL Server 综合应用。

SQL Server 教程部分包括数据库创建、表的创建和操作、数据库查询和视图、游标、T-SQL、索引、数据完整性、存储过程和触发器、备份与恢复、系统安全管理等内容。这部分内容的介绍不再强调命令的格式, 而是突出主要功能和配套举例, 详细格式和功能说明可参考有关文档。

实验部分训练 SQL Server 基本操作和基本命令, 其数据库自成系统。

实习部分以当前流行的数据库应用开发工具(包括 PHP 5、Java EE、ASP.NET 4.5 和 VB 6.0 等)为平台, 开发具有相同功能的同一个数据库应用系统。选用的实例既典型又小而精, 教和学都非常方便。

本书配有教学课件、配套的客户端 /SQL Server 2012 应用系统数据库和所有源程序文件。有需要的教师可从华章网站(www.hzbook.com)免费下载。

本书由南京师范大学郑阿奇主编, 参加本套书编写的还有顾韵华、梁敬东、丁有和、徐文胜、彭作民、崔海源、徐卫军、王燕平、汤玫、周怡明、刘博宇、郑进、陶卫冬、严大牛、周怡君、吴明祥、时跃华、赵青松等。此外, 还有许多同志对本书提供了很多帮助, 在此表示感谢!

由于编者水平有限, 书中错误在所难免, 敬请广大读者批评指正。

作者邮箱: easybooks@163.com。

编 者

目 录

前言

第 0 章 数据库基础	1
0.1 数据库基本概念	1
0.1.1 数据库与数据库管理系统	1
0.1.2 数据模型	2
0.1.3 关系型数据库语言	4
0.2 数据库设计	4
0.2.1 概念数据模型	4
0.2.2 逻辑数据模型	6
0.2.3 物理数据模型	7
0.3 数据库应用系统	7
0.3.1 应用系统的数据接口	7
0.3.2 C/S 架构的应用系统	9
0.3.3 B/S 架构的应用系统	10
习题	10

第一部分 数据库管理系统

第 1 章 SQL Server 2012 简介和安装	14
1.1 SQL Server 简介	14
1.1.1 SQL Server 2012 服务器组件、 管理工具和联机丛书	15
1.1.2 SQL Server 2012 的不同版本 及支持功能	16
1.2 SQL Server 2012 的安装和运行	18
1.2.1 SQL Server 2012 安装环境	18
1.2.2 SQL Server 2012 的安装	19
1.2.3 SQL Server 2012 的运行	23
习题	24
第 2 章 数据库创建	25
2.1 数据库及其数据库对象	25
2.2 以界面方式创建数据库	28
2.2.1 数据库的创建	28

2.2.2 数据库的修改和删除	30
2.3 以命令方式创建数据库	33
2.3.1 创建数据库	33
2.3.2 修改数据库	37
2.3.3 删除数据库	40
2.3.4 数据库快照	40
习题	42
第 3 章 表的创建和操作	44
3.1 表结构和数据类型	44
3.1.1 表和表结构	44
3.1.2 数据类型	45
3.1.3 表结构设计	50
3.2 以界面方式创建表	52
3.2.1 创建表	52
3.2.2 修改表结构	53
3.2.3 删除表	54
3.3 以命令方式创建表	54
3.3.1 创建表	54
3.3.2 修改表结构	56
3.3.3 删除表	58
3.4 以界面方式操作表数据	58
3.4.1 插入记录	58
3.4.2 删除记录	59
3.4.3 修改记录	60
3.5 以命令方式操作表数据	60
3.5.1 插入记录	60
3.5.2 修改记录	63
3.5.3 删除记录	63
习题	65
第 4 章 数据库的查询和视图	67
4.1 数据库的查询	67
4.1.1 选择查询结果输出列	67
4.1.2 选择查询条件	72

4.1.3 指定查询对象	78	6.6.1 标量函数	133
4.1.4 连接	80	6.6.2 内嵌表值函数	134
4.1.5 指定查询结果分组方法	84	6.6.3 用户定义函数的删除	137
4.1.6 指定查询结果分组后的 筛选条件	85	习题	137
4.1.7 将查询结果排序	86	第 7 章 索引	139
4.1.8 SELECT 语句的其他语法	87	7.1 索引的分类	139
4.2 视图	89	7.2 索引的创建	140
4.2.1 视图的概念	89	7.3 重建索引	143
4.2.2 创建视图	90	7.4 索引的删除	143
4.2.3 查询视图	93	习题	144
4.2.4 更新视图	94	第 8 章 数据完整性	145
4.2.5 修改视图的定义	96	8.1 数据完整性分类	145
4.2.6 删除视图	96	8.2 实体完整性	146
习题	97	8.3 域完整性	149
第 5 章 游标	98	8.4 参照完整性	154
5.1 声明游标	98	习题	157
5.2 打开游标	101	第 9 章 存储过程和触发器	158
5.3 读取数据	101	9.1 存储过程	158
5.4 关闭和删除游标	104	9.1.1 存储过程的类型	158
习题	104	9.1.2 存储过程的创建与执行	159
第 6 章 T-SQL	105	9.1.3 存储过程的修改	164
6.1 SQL 与 T-SQL	105	9.1.4 存储过程的删除	165
6.2 常量、变量与数据类型	106	9.1.5 以界面方式操作存储过程	165
6.2.1 常量	106	9.2 触发器	166
6.2.2 数据类型	108	9.2.1 触发器的类型	167
6.2.3 变量	111	9.2.2 触发器的创建	167
6.3 运算符与表达式	114	9.2.3 触发器的修改	173
6.4 流程控制语句	119	9.2.4 触发器的删除	174
6.4.1 BEGIN...END 语句块	119	9.2.5 以界面方式操作触发器	174
6.4.2 条件语句	119	习题	175
6.4.3 CASE 语句	121	第 10 章 备份与恢复	176
6.4.4 无条件转移语句	122	10.1 概述	176
6.4.5 循环语句	122	10.1.1 备份和恢复需求分析	176
6.4.6 返回语句	123	10.1.2 数据库备份的概念	177
6.4.7 等待语句	124	10.1.3 数据库恢复的概念	179
6.4.8 错误处理语句	124	10.2 数据库备份	179
6.5 系统内置函数	124	10.2.1 创建备份设备	179
6.5.1 系统内置函数简介	124	10.2.2 以命令方式备份数据库	181
6.5.2 常用系统标量函数	125	10.2.3 以界面方式备份数据库	185
6.6 用户定义函数	132	10.3 数据库恢复	186
		10.3.1 检查点	186

10.3.2 以命令方式恢复数据库	187	12.2.1 锁定粒度	220
10.3.3 以界面方式恢复数据库	190	12.2.2 锁定模式	220
10.4 附加数据库和复制数据库	190	12.3 自动化管理	221
10.4.1 附加数据库	190	12.3.1 SQL Server 代理	222
10.4.2 复制数据库	191	12.3.2 操作员	223
习题	191	12.3.3 作业	223
第 11 章 系统安全管理	192	12.3.4 警报	223
11.1 SQL Server 2012 的安全机制	192	12.3.5 数据库邮件	223
11.1.1 SQL Server 身份验证模式	192	12.3.6 维护计划向导	224
11.1.2 SQL Server 安全性机制	193	习题	224
11.1.3 SQL Server 数据库安全验证	193		
11.2 建立和管理用户帐户	194	第二部分 数据库管理系统实验	
11.2.1 以界面方式管理用户帐户	194	实验 1 SQL Server 2012 环境	226
11.2.2 以命令方式管理用户帐户	197	实验 2 创建数据库和表	228
11.3 角色管理	199	实验 3 表数据的插入、修改和删除	232
11.3.1 固定服务器角色	199	实验 4 查询和视图	236
11.3.2 固定数据库角色	201	实验 5 T-SQL 编程	244
11.3.3 自定义数据库角色	203	实验 6 索引和数据完整性	248
11.3.4 应用程序角色	205	实验 7 存储过程和触发器	252
11.4 数据库权限的管理	206	实验 8 备份和恢复	257
11.4.1 授予权限	206	实验 9 数据库的安全性	260
11.4.2 拒绝权限	209		
11.4.3 撤销权限	210	第三部分 数据库应用实习	
11.5 数据库架构的定义和使用	211	实习 0 创建实习数据库	266
11.5.1 以界面方式创建架构	211	实习 1 PHP 5/SQL Server 2012 学生成绩管理系统	272
11.5.2 以命令方式创建架构	212	实习 2 Java EE/SQL Server 2012 学生成绩管理系统	290
习题	213	实习 3 ASP.NET 4.5/SQL Server 2012 学生成绩管理系统	321
第 12 章 其他概念	214	实习 4 VB 6.0/SQL Server 2012 学生成绩管理系统	344
12.1 事务	214	附录 学生成绩数据库表样本数据	358
12.1.1 事务与 ACID 属性	214		
12.1.2 多用户使用问题	215		
12.1.3 事务处理	215		
12.1.4 事务隔离级	218		
12.2 锁定	220		

第 0 章

数据库基础

为了更好地学习 SQL Server，首先需要介绍数据库的基本概念，如果已掌握数据库原理，那么本章的数据库原理部分仅仅作参考。

0.1 数据库基本概念

0.1.1 数据库与数据库管理系统

1. 数据库

数据库 (DB) 是存放数据的仓库，而且这些数据存在一定的关联，并按一定的格式存放在计算机上。从广义上讲，数据不仅包含数字，还包括文本、图像、音频、视频等。

例如，把一个学校的学生、课程、学生成绩等数据有序地组织并存放在计算机内，就可以构成一个数据库。因此，数据库由一些持久的相互关联的数据的集合组成，并以一定的组织形式存放在计算机的存储介质中。

2. 数据库管理系统

数据库管理系统 (DBMS) 是管理数据库的系统，它按一定的数据模型组织数据。DBMS 应提供如下功能。

- 1) 数据定义功能：可定义数据库中的数据对象。
- 2) 数据操纵功能：可对数据库表进行基本操作，如插入、删除、修改、查询等。
- 3) 数据的完整性检查功能：保证用户输入的数据满足相应的约束条件。
- 4) 数据库的安全保护功能：保证只有具有权限的用户才能访问数据库中的数据。
- 5) 数据库的并发控制功能：使多个应用程序可在同一时刻并发地访问数据库的数据。
- 6) 数据库系统的故障恢复功能：当数据库出现运行故障时可进行数据库恢复，以保证数据库可靠运行。
- 7) 在网络环境下访问数据库的功能。
- 8) 它是方便、有效地存取数据库信息的接口和工具。编程人员通过程序开发工具与数据库的接口编写数据库应用程序。数据库系统管理员 (DataBase Administrator, DBA) 通过提供

的工具对数据库进行管理。

数据库管理系统是一个系统软件。目前,比较流行的DBMS有:SQL Server、Oracle、MySQL、Sybase、DB2、Access、Visual FoxPro等。其中,SQL Server是目前最流行的大中型关系数据库管理系统之一,被广泛应用于各种数据库应用场合。本书介绍的是SQL Server 2012版。

3. 数据库系统

数据、数据库、数据库管理系统与操作数据库的应用程序,加上支撑它们的硬件平台、软件平台和与数据库有关的人员一起构成了一个完整的数据库系统。图0-1描述了数据库系统的构成。

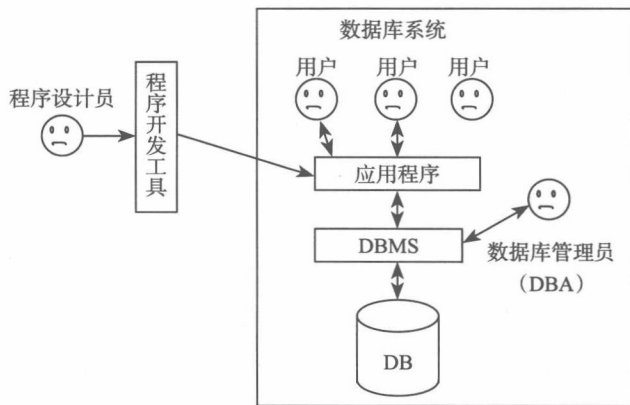


图0-1 数据库系统的构成

0.1.2 数据模型

数据库管理系统根据数据模型对数据进行存储和管理,数据库管理系统采用的数据模型主要有层次模型、网状模型和关系模型。

1. 层次模型

层次模型将数据组织成一对多关系的结构,采用关键字来访问其中每一个层次的每一部分。它存取方便且速度快;结构清晰,容易理解;数据修改和数据库扩展容易实现;检索关键属性十分方便。但结构不够灵活;同一属性数据要存储多次,数据冗余大;不适合于拓扑空间数据的组织。

图0-2为某学校按层次模型组织的数据示例。

2. 网状模型

网状模型具有多对多类型的数据组织方式。它能明确而方便地表示数据间的复杂关系;数据冗余小。但网状结构复杂,增加了用户查询和定位的困难;需要存储数据间联系的指针,这使得数据量增大;数据的修改不方便。

图0-3为按网状模型组织的数据示例。

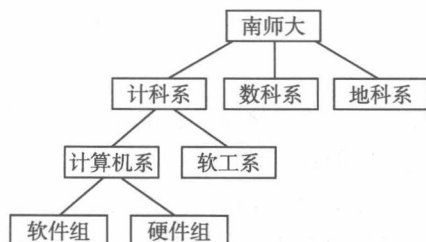


图0-2 按层次模型组织的数据示例

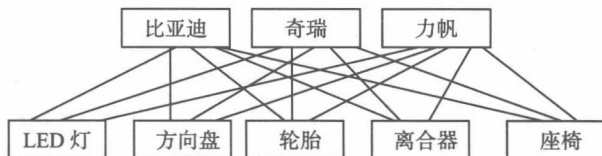


图0-3 按网状模型组织的数据示例

3. 关系模型

关系模型以记录组或二维数据表的形式组织数据,以便于利用各种实体与属性之间的关系进行存储和变换,既不分层,也无指针,是建立空间数据和属性数据之间关系的一种非常有效的数据组织方法。它结构特别灵活,概念单一,满足所有布尔逻辑运算和数学运算规则

形成的查询要求；能搜索、组合和比较不同类型的数据；增加和删除数据非常方便；具有更高的数据独立性和更好的安全保密性。但数据库较大时，查找满足特定关系的数据费时，而且无法表达空间关系。

例如，在学生成绩管理系统涉及的“学生”、“课程”和“成绩”3个表中，“学生”表涉及的主要信息有：学号、姓名、性别、出生时间、专业、总学分、备注；“课程”表涉及的主要信息有：课程号、课程名、开课学期、学时和学分；“成绩”表涉及的主要信息有：学号、课程号和成绩。表0-1～表0-3分别描述了学生成绩管理系统中“学生”、“课程”和“成绩”3个表的部分数据。

表0-1 “学生”表

学号	姓名	性别	出生时间	专业	总学分	备注
191301	王林	男	1995-02-10	计算机	50	
191303	王燕	女	1994-10-06	计算机	50	
191308	林一帆	男	1994-08-05	计算机	52	已提前修完一门课
221302	王林	男	1994-01-29	通信工程	40	有一门课不及格，待补考
221304	马琳琳	女	1995-02-10	通信工程	42	

表0-2 “课程”表

课程号	课程名	开课学期	学时	学分
0101	计算机基础	1	80	5
0102	程序设计与语言	2	68	4
0206	离散数学	4	68	4

表0-3 “成绩”表

学号	课程号	成绩	学号	课程号	成绩
191301	101	80	191308	101	85
191301	102	78	191308	102	64
191301	206	76	191308	206	87
191303	101	62	221302	101	65
191303	102	70	221304	101	91

表格中的一行称为一个记录，一列称为一个字段，每列的标题称为字段名。如果给每个关系表取一个名称，则有 n 个字段的关系表的结构可表示为：关系表名（字段名1，…，字段名 n ），通常把关系表的结构称为关系模式。

在关系表中，如果一个字段或几个字段组合的值可唯一标识其对应记录，则称该字段或字段组合为码。

例如，表0-1中的“学号”可唯一标识每一个学生，表0-2中的“课程号”可唯一标识每一门课程。表0-3中的“学号”和“课程号”可唯一标识每一个学生的一门课程的成绩。

有时一个表可能有多个码，比如，在表0-1中，若姓名不允许重名，则“学号”、“姓名”均是“学生”表的码。对于每一个关系表，通常可指定一个码为“主码”，在关系模式中，一般用下横线标出主码。

设表0-1的名称为xsb，关系模式可分别表示为：

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

设表0-2的名称为kcb，关系模式可分别表示为：

kcb (课程号, 课程名, 开课学期, 学时, 学分)

设表 0-3 的名称为 cjb, 关系模式可分别表示为:

cjb (学号, 课程号, 成绩, 学分)

通过上面的分析可以看出, 关系模型更适合组织数据, 所以使用最广泛。目前, 主流的关系型数据库管理系统 (RDBMS) 包括 Oracle、SQL Server、MySQL、Access 和 Visual FoxPro 等。

0.1.3 关系型数据库语言

结构化查询语言 (Structured Query Language, SQL) 是用于查询关系数据库的结构化语言。SQL 的功能包括数据查询、数据操纵、数据定义和数据控制 4 部分。

0.2 数据库设计

数据模型按不同的应用层次分成 3 种类型: 分别是概念数据模型、逻辑数据模型、物理数据模型。

0.2.1 概念数据模型

概念数据模型 (conceptual data model, 简称概念模型) 是面向数据库用户的实现世界的模型, 主要用来描述世界的概念化结构, 它使数据库设计人员在设计的初始阶段, 摆脱计算机系统及 DBMS 的具体技术问题, 集中精力分析数据以及数据之间的联系等, 与具体的数据管理系统无关。概念数据模型只有转换成逻辑数据模型, 才能在 DBMS 中实现。

概念模型用于信息世界的建模, 一方面它应该具有较强的语义表达能力, 能够方便直接地表达应用中的各种语义知识, 另一方面它应该简单、清晰、易于用户理解。在概念数据模型中, 最常用的是 E-R 模型、扩充的 E-R 模型、面向对象模型及谓词模型。

通常, E-R 模型把每一类数据对象的个体称为“实体”, 而每一类对象个体的集合称为“实体集”。例如, 学生成绩管理系统主要涉及“学生”和“课程”两个实体集。其他非主要的实体可以很多, 如班级、班长、任课教师、辅导员等实体。

把每个实体集涉及的信息项称为属性。例如, “学生”实体集的属性有: 学号、姓名、性别、出生时间、专业、总学分和备注。“课程”实体集的属性有: 课程号、课程名、开课学期、学时和学分。

实体集中的实体彼此是可区别的。如果实体集中的属性或最小属性组合的值能唯一标识其对应实体, 则将该属性或属性组合称为码。码可能有多个, 对于每一个实体集, 可指定一个码为主码。

如果用矩形框表示实体集, 用带圆角的矩形框表示属性, 用线段连接实体集与属性; 当一个属性或属性组合指定为主码时, 在实体集与属性的连接线上标记一条斜线, 则可以用如图 0-4 所示形式描述学生成绩管理系统中的实体集及每个实体集涉及的属性。

实体集 A 和实体集 B 之间存在各种关系, 通常把这些关系称为“联系”。通常将实体集及实体集联系的图示称为实体 (entity) - 联系 (relationship) 模型, 即 E-R 模型。

E-R 图就是 E-R 模型的描述方法, 即实体 - 联系图。通常, 关系数据库的设计者使用 E-R 图来对信息世界建模。在 E-R 图中, 使用矩形表示实体型, 使用椭圆表示属性, 使用菱形表示联系。从分析用户项目涉及的数据对象及数据对象之间的联系出发, 到获取 E-R 图的

这一过程称为概念结构设计。

两个实体集 A 和 B 之间的联系可能是以下 3 种情况之一。

1. 一对一的联系 (1:1)

一对一的联系是指 A 中的一个实体至多与 B 中的一个实体相联系，B 中的一个实体也至多与 A 中的一个实体相联系。例如，“班级”与“班长”这两个实体集之间的联系是一对一的联系，因为一个班级只有一个班长，反过来，一个班长只属于一个班级。“班级”与“班长”两个实体集的 E-R 模型如图 0-5 所示。

2. 一对多的联系 (1:n)

一对多的联系是指 A 中的一个实体可以与 B 中的多个实体相联系，而 B 中的一个实体至多与 A 中的一个实体相联系。例如，“班级”与“学生”这两个实体集之间的联系是一对多的联系，因为一个班级可有若干学生，反过来，一个学生只能属于一个班级。“班级”与“学生”两个实体集的 E-R 模型如图 0-6 所示。

3. 多对多的联系 (m:n)

多对多的联系是指 A 中的一个实体可以与 B 中的多个实体相联系，B 中的一个实体也可与 A 中的多个实体相联系。例如，“学生”与“课程”这两个实体集之间的联系是多对多的联系，因为一个学生可选多门课程，反过来，一门课程可被多个学生选修。“学生”与“课程”两个实体集的 E-R 模型如图 0-7 所示。

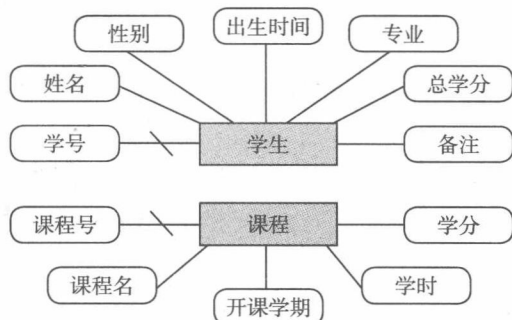


图 0-4 学生和课程实体集属性的描述

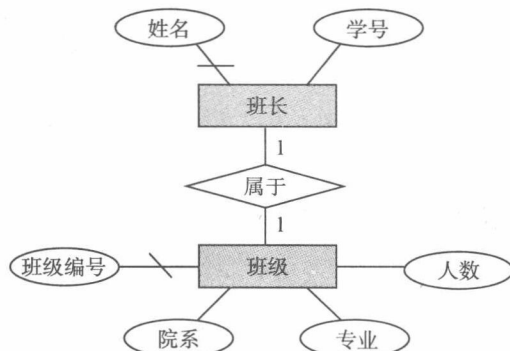


图 0-5 “班级”与“班长”实体集的 E-R 模型

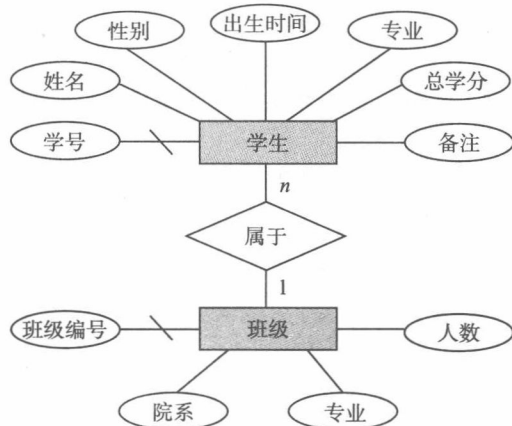


图 0-6 “学生”与“班级”两个实体集的 E-R 模型

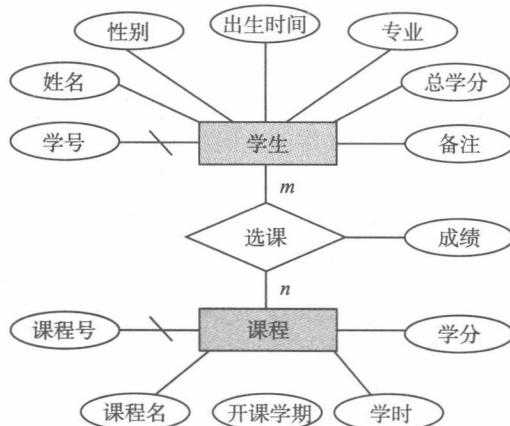


图 0-7 “学生”与“课程”实体集的 E-R 模型

0.2.2 逻辑数据模型

逻辑数据模型(logical data model, 简称逻辑模型)是用户从数据库看到的模型,是具体的DBMS所支持的数据模型。此模型既要面向用户,又要面向系统,主要用于数据库管理系统(DBMS)的实现。

前面用E-R图描述学生成绩管理系统中实体集与实体集之间的联系,为了设计关系型的学生成绩管理数据库,需要确定包含哪些表,每个表的结构是怎样的。

前面已经介绍了实体集之间的联系,下面介绍根据3种联系从E-R图获得关系模式的方法。

1. 1:1 联系的 E-R 图到关系模式的转换

1:1 的联系既可单独对应一个关系模式,也可以不单独对应一个关系模式。

1) 如果联系单独对应一个关系模式,则由联系属性、参与联系的各实体集的主码属性构成关系模式,其主码可选参与联系的实体集的任一方的主码。

例如,考虑图0-5描述的“班级(bjb)”与“班长(bzb)”实体集通过属于(syb)联系E-R模型,可设计如下关系模式(下划线表示该字段为主码):

```
bjb (班级编号, 院系, 专业, 人数)
bzb (学号, 姓名)
syb (学号, 班级编号)
```

2) 如果联系不单独对应一个关系模式,联系的属性及一方的主码加入另一方实体集对应的关系模式中。

例如,考虑图0-5描述的“班级(bjb)”与“班长(bzb)”实体集通过属于(syb)联系E-R模型,可设计如下关系模式:

```
bjb (班级编号, 院系, 专业, 人数)
bzb (学号, 姓名, 班级编号)
```

或者:

```
bjb (班级编号, 院系, 专业, 人数, 学号)
bzb (学号, 姓名)
```

2. 1:n 联系的 E-R 图到关系模式的转换

1:n 的联系既可单独对应一个关系模式,也可以不单独对应一个关系模式。

1) 如果联系单独对应一个关系模式,则由联系的属性、参与联系的各实体集的主码属性构成关系模式, n 端的主码作为该关系模式的主码。

例如,考虑图0-6描述的“班级(bjb)”与“学生(xsb)”实体集E-R模型,可设计如下关系模式:

```
bjb (班级编号, 院系, 专业, 人数)
xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)
syb (学号, 班级编号)
```

2) 如果联系不单独对应一个关系模式,则将联系的属性及1端的主码加入 n 端实体集对应的关系模式中,主码仍为 n 端的主码。

例如,图0-6描述的“班级(bjb)”与“学生(xsb)”实体集E-R模型可设计如下关系

模式:

bjb (班级编号, 院系, 专业, 人数)

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注, 班级编号)

3. $m:n$ 联系的 E-R 图到关系模式的转换

$m:n$ 的联系单独对应一个关系模式, 该关系模式包括联系的属性、参与联系的各实体集的主码属性, 该关系模式的主码由各实体集的主码属性共同组成。

例如, 图 0-7 描述的“学生 (xsb)”与“课程 (kcb)”实体集之间的联系可设计如下关系模式:

xsb (学号, 姓名, 性别, 出生时间, 专业, 总学分, 备注)

kcb (课程号, 课程名称, 开课学期, 学时, 学分)

cjb (学号, 课程号, 成绩)

关系模式 cjb 的主码是由“学号”和“课程号”两个属性组合起来构成的一个主码, 一个关系模式只能有一个主码。

至此介绍了根据 E-R 图设计关系模式的方法, 通常这一设计过程称为逻辑结构设计。

设计好一个项目的关系模式后, 就可以在数据库管理系统环境下创建数据库、关系表及其他数据库对象, 输入相应数据, 并根据需要对数据库中的数据进行各种操作。

0.2.3 物理数据模型

物理数据模型 (physical data model, 简称物理模型) 是面向计算机物理表示的模型, 描述了数据在存储介质上的组织结构, 它不但与具体的 DBMS 有关, 而且与操作系统和硬件有关。每一种逻辑数据模型在实现时都有其对应的物理数据模型。DBMS 为了保证其独立性与可移植性, 大部分物理数据模型的实现工作都由系统自动完成, 而设计者只设计索引、聚集等特殊结构。

0.3 数据库应用系统

0.3.1 应用系统的数据接口

客户端应用程序或应用服务器向数据库服务器请求服务时, 首先必须和数据库建立连接。虽然现有 DBMS 几乎都遵循 SQL 标准, 但不同厂家开发的 DBMS 有差异, 存在适应性和可移植性等方面的问题, 为此, 人们研究和开发了连接不同 DBMS 的通用方法、技术和软件接口。

1. ODBC 数据库接口

开放式数据库互连 (Open DataBase Connectivity, ODBC) 是微软公司推出的一种实现应用程序和关系数据库之间通信的接口标准。符合该标准的数据库可以通过用 SQL 语句编写的程序对数据库进行操作, 但只针对关系数据库。目前所有的关系数据库都符合该标准。ODBC 的本质是一组数据库访问 API (应用程序编程接口), 由一组函数调用组成, 其核心是 SQL 语句。

在具体操作时, 必须先用 ODBC 管理器注册一个数据源, 管理器根据数据源提供的数据库位置、数据库类型及 ODBC 驱动程序等信息, 建立 ODBC 与具体数据库的联系。这样, 只

要应用程序将数据源名提供给 ODBC，ODBC 就能建立与相应数据库的连接。

2. ADO 数据库接口

ADO (ActiveX Data Object) 是微软公司开发的基于 COM 的数据库应用程序接口，通过 ADO 连接数据库，可以灵活地操作数据库中的数据。使用 ADO 访问关系数据库有两种途径：一种是通过 ODBC 驱动程序，另一种是通过数据库专用的 OLE DB Provider，后者有更高的访问效率。

随着网络技术的发展，网络数据库及相关的操作技术也越来越多地应用到实际中，而数据库操作技术也在不断地发展完善。ADO 对象模型进一步发展成了 ADO.NET。ADO.NET 是 .NET Framework SDK 中用于操作数据库的类库总称，ADO.NET 相对于 ADO 的最大优势在于对数据的更新修改可在与数据源完全断开连接的情况下进行，然后把数据更新的结果和状态传回到数据源，这样大大减少了由于连接过多对数据库服务器资源的占用。

3. ADO.NET 数据库接口

ADO.NET 数据模型从 ADO 发展而来，但它不只是对 ADO 的改进，而是采用了一种全新的技术，主要体现在以下几个方面：

- 1) ADO.NET 不是采用 ActiveX 技术，而是与 .NET 框架紧密结合的产物。
- 2) ADO.NET 包含对 XML 标准的完全支持，这对于跨平台交换数据具有重要意义。
- 3) ADO.NET 既能在与数据源连接的环境下工作，又能在断开与数据源连接的条件下工作。特别是后者，非常适合于网络应用的需要，因为在网络环境下，始终做到与数据源保持连接，不符合网站的要求，不仅效率低，代价高，而且常会引发由于多个用户同时访问带来的冲突。因此，ADO.NET 系统集中主要精力解决在断开与数据源连接的条件下的数据处理问题。

ADO.NET 提供了面向对象的数据库视图，并且在其对象中封装了许多数据库属性和关系。最重要的是，它通过多种方式封装和隐藏了很多数据库访问的细节。可以完全不知道对象在与 ADO.NET 对象交互，也不用担心数据移动到另外一个数据库或者从另一个数据库获得数据等细节问题。通过 ADO.NET 访问数据库的接口模型如图 0-8 所示。

数据层是实现 ADO.NET 断开式连接的核心，从数据源读取的数据先缓存到数据集中，然后被程序或控件调用。数据源可以是数据库或 XML 数据。

数据提供者用于建立数据源与数据集之间的联系，它能连接各种类型的数据源，并能按要求将数据源中的数据提供给数据集，或者从数据集向数据源返回编辑后的数据。

4. JDBC 数据库接口

JDBC (Java DataBase Connectivity) 是 JavaSoft (原来 Sun 公司的业务部门) 开发的，用 Java 语言编写的用于连接和操作数据库的类和接口，可为多种关系数据库提供统一的访问方式。通过 JDBC 访问数据库包括 4 个主要组件：Java 应用程序、JDBC 驱动器管理器、驱动器和数据源。

在 JDBC API 中有两层接口：应用程序层和驱动程序层，前者使开发人员可以通过 SQL

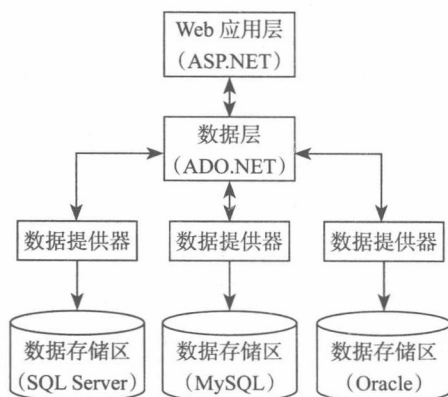


图 0-8 通过 ADO.NET 访问数据库的接口模型

调用数据库和取得结果,后者处理与具体数据库驱动程序的所有通信。

使用 JDBC 接口操作数据库有如下优点:

- 1) JDBC API 与 ODBC 十分相似,有利于用户理解。
- 2) 使编程人员从复杂的驱动器调用命令和函数中解脱出来,从而致力于应用程序功能的实现。
- 3) JDBC 支持不同的关系数据库,增强了程序的可移植性。

使用 JDBC 的主要缺点有:访问数据记录的速度会受到一定影响,此外,由于 JDBC 结构中包含了不同厂家的产品,这给数据源的更改带来了较大麻烦。

5. 数据库连接池技术

对于网络环境下的数据库应用,由于用户众多,使用传统的 JDBC 方式连接数据库,系统资源开销过大成为制约大型企业级应用效率的瓶颈,采用数据库连接池技术对数据库连接进行管理,可以大大提高系统的效率和稳定性。

0.3.2 C/S 架构的应用系统

DBMS 通过命令和适合专业人员的界面操作数据库。对于一般的数据库应用系统,除了 DBMS 外,还需要设计适合普通人员操作数据库的界面。目前,流行的开发数据库界面的工具主要有 Visual Basic、Visual C++、Visual C# 等。应用程序与数据库、数据库管理系统之间的关系如图 0-9 所示。

从图 0-9 中可看出,当应用程序需要处理数据库中的数据时,首先向数据库管理系统发送一个数据请求,数据库管理系统接收到这一请求后,对其进行分析,然后执行数据库操作,并把处理结果返回给应用程序。由于应用程序直接与用户交互,而数据库管理系统不直接与用户打交道,所以应用程序被称为“前台”,而数据库管理系统被称为“后台”。由于应用程序是向数据库管理系统提出服务请求,通常称其为客户端程序(client),而数据库管理系统是为应用程序提供服务,通常称其为服务器程序(server),所以又将这一操作数据库的模式称为 C/S(客户/服务器)架构。

应用程序和数据库管理系统可以运行在同一台计算机上(单机方式),也可以运行在网络环境中。在网络环境下,数据库管理系统在网络中的一台主机上运行,应用程序可以在网络上的多台主机上运行,即一对多的方式。例如,用 Visual Basic 开发的 C/S 架构的学生成绩管理系统的学生信息输入界面如图 0-10 所示。

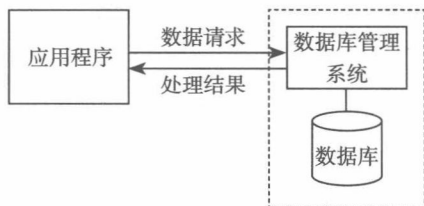


图 0-9 应用程序、数据库、数据库管理系统之间的关系



图 0-10 C/S 架构的学生成绩管理系统界面

0.3.3 B/S 架构的应用系统

基于 Web 的数据库应用采用三层（浏览器 / Web 服务器 / 数据库服务器）模式，也称为 B/S 架构，如图 0-11 所示。其中，浏览器（browser）是用户输入数据和显示结果的交互界面，用户在浏览器表单中输入数据，然后将表单中的数据提交并发送到 Web 服务器，Web 服务器接收并处理用户的数据，通过数据库服务器，从数据库中查询需要的数据（或把数据录入数据库），并将结果回送 Web 服务器，Web 服务器把返回的结果插入 HTML 页面，传送给客户端，在浏览器中显示出来。



图 0-11 三层 B/S 架构

目前，开发数据库 Web 界面的流行工具主要有 ASP.NET (C#)、PHP、Java EE 等。例如，用 ASP.NET 开发的 B/S 架构的学生成绩管理系统的学生信息录入页面如图 0-12 所示。



图 0-12 B/S 架构的学生成绩管理系统页面

习题

一、选择题

- SQL Server 是 ()。
 - 数据库
 - DBA
 - DBMS
 - 数据库系统
- SQL Server 组织数据采用 ()。
 - 层次模型
 - 网状模型
 - 关系模型
 - 数据模型
- () 是实体属性。
 - 形状
 - 汽车
 - 盘子
 - 高铁
- 在数据库管理系统中设计表属于 ()。
 - 概念结构设计
 - 逻辑结构设计
 - 物理结构设计
 - 数据库设计
- 图书与读者之间是 ()。
 - 一对一关系
 - 多对一关系
 - 多对多关系
 - 一对多关系