



普通高等教育“十一五”国家级规划教材
高等学校计算机教材

SQL Server

实用教程 (第4版)

(SQL Server 2014版)

© 郑阿奇 主编



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
<http://www.phei.com.cn>

普通高等教育“十一五”国家级规划教材
高等学校计算机教材

SQL Server 实用教程

(第4版) (SQL Server 2014版)

郑阿奇 主编

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书以 Microsoft SQL Server 2014 中文版为平台,系统地介绍 SQL Server 2014 基础、实验和综合应用等内容。SQL Server 2014 通过界面和命令两种方式同步操作数据库,功能和实例互相补充,并形成系统。实验通过教程实例训练 SQL Server 基本操作和基本命令。综合应用介绍 PHP 5、Java EE 和 ASP.NET 4.5 等平台对 SQL Server 2014 学生成绩数据库的典型操作方法,同时实现学生成绩管理系统部分功能,更方便教学和学生模仿。

本书配有教学课件和配套的客户端/SQL Server 2014 应用系统数据库和所有源程序文件,需要者请到华信教育资源网 (<http://www.hxedu.com.cn>) 免费注册下载。

本书是普通高等教育“十一五”国家级规划教材,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。

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 实用教程: SQL Server 2014 版 / 郑阿奇主编. —4 版. —北京: 电子工业出版社, 2015.8
普通高等教育“十一五”国家级规划教材. 高等学校计算机教材
ISBN 978-7-121-26623-2

I. ①S… II. ①郑… III. ①关系数据库系统—高等学校—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2015)第 161430 号

策划编辑: 程超群

责任编辑: 郝黎明 特约编辑: 张燕虹

印 刷: 三河市双峰印刷装订有限公司

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: 19.75 字数: 568 千字

版 次: 2002 年 8 月第 1 版

2015 年 8 月第 4 版

印 次: 2015 年 8 月第 1 次印刷

定 价: 49.00 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

Microsoft SQL Server 是目前最流行的大中型关系型数据库管理系统 (DBMS), 为了适应市场的需要, 我国高校的许多专业都开设了介绍 SQL Server 数据库管理系统的课程。

2002 年, 我们结合教学和应用开发实践, 编写了《SQL Server 实用教程》; 2005 年, 我们推出了《SQL Server 实用教程 (第 2 版)》; 2009 年, 《SQL Server 实用教程 (第 3 版)》升级 SQL Server 到 2005 版; 《SQL Server 实用教程 (第 4 版)》在继承前面 3 个版本大获成功的经验基础上, 结合近年来教学与应用开发的实践, 对 SQL Server 部分进行了全面整合、优化和完善, 同时更新和重新设计了综合应用实习, 更加有利于理解和教学。根据目前市场应用和我国高校的实际情况, 分别以 SQL Server 2008、SQL Server 2012 和 SQL Server 2014 为平台推出, 除了基本内容适应不同平台的需要外, 综合应用开发实例则有不同侧重, 在应用内容系统性的基础上, 突出 SQL Server 的基本操作方法。

本书以 SQL Server 2014 中文版为平台, 系统地介绍 SQL Server 2014 基础、实验和 SQL Server 2014 综合应用等内容。SQL Server 2014 基础部分在介绍数据库的基础知识后, 系统介绍数据库创建、表的创建和操作、数据库的查询和视图、T-SQL 语言、索引与数据完整性、存储过程和触发器、备份与恢复、系统安全管理、其他概念等。实验部分先训练教程部分的实例, 然后要求自己实践完成, 其数据库操作自成系统。综合应用部分通过创建 SQL Server 2014 实习数据库及其数据库常用对象, 将数据库、表、视图、游标、完整性、触发器、存储过程等包含在其中。此后, 介绍 PHP 5、Java EE 和 ASP.NET 4.5 等平台对 SQL Server 2014 学生成绩数据库的典型操作方法, 同时实现学生成绩管理系统部分功能, 更方便教学和学生模仿。

本书配有教学课件和配套的客户端/SQL Server 2014 应用系统数据库和所有源程序文件, 需要者请到华信教育资源网 (<http://www.hxedu.com.cn>) 免费注册下载。

本书由郑阿奇 (南京师范大学) 主编。

参加本书编写的还有刘启芬、顾韵华、梁敬东、陆文周、丁有和、曹弋、陈瀚、徐文胜、殷红先、张为民、姜乃松、钱晓军、彭作民、高茜、陈冬霞、徐斌、王志瑞、孙德荣、周怡明、刘博宇、郑进、周何骏、陶卫冬、严大牛、邓拼搏、俞琰、周怡君、吴明祥、于金彬、马骏等。此外, 还有许多同志对本书提供了很多帮助, 在此一并表示感谢!

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

第 1 部分 实用教程

第 0 章 数据库基础	1	3.1.3 表结构设计	39
0.1 数据库基本概念	1	3.2 以界面方式创建表	41
0.1.1 数据库系统	1	3.2.1 创建表	41
0.1.2 数据模型	2	3.2.2 修改表结构	42
0.1.3 关系型数据库语言	4	3.2.3 删除表	43
0.2 数据库设计	4	3.2.4 以界面方式操作表数据	43
0.2.1 概念模型	5	3.3 以命令方式创建表	46
0.2.2 逻辑模型	6	3.3.1 创建表: CREATE TABLE	46
0.2.3 物理模型	7	3.3.2 修改表结构: ALTER TABLE	48
0.3 数据库应用系统	8	3.3.3 删除表	49
0.3.1 应用系统的数据接口	8	3.4 以命令方式操作表数据	49
0.3.2 C/S 架构的应用系统	9	3.4.1 插入记录	49
0.3.3 B/S 架构的应用系统	10	3.4.2 修改记录	51
第 1 章 SQL Server 2014 简介	11	3.4.3 删除记录	52
1.1 SQL Server 2014 之前版本	11	第 4 章 数据库的查询和视图	54
1.2 SQL Server 2014 服务器组件 和管理工具	12	4.1 数据库的查询	54
1.3 SQL Server 2014 的不同版本	13	4.1.1 选择查询结果输出列	54
1.4 SQL Server 2014 的安装步骤	13	4.1.2 选择查询条件: WHERE 子句	60
1.5 SQL Server 2014 的运行	18	4.1.3 指定查询对象: FROM 子句	66
1.6 SQL Server 2014 服务器	20	4.1.4 连接	67
第 2 章 数据库创建	22	4.1.5 指定查询结果分组方法: GROUP BY 子句	72
2.1 SQL Server 数据库及其数据库对象	22	4.1.6 指定查询结果分组后筛选条件: HAVING 子句	73
2.2 以界面方式创建数据库	24	4.1.7 指定查询结果排序顺序: ORDER BY 子句	74
2.2.1 数据库的创建	24	4.1.8 SELECT 语句的其他语法	75
2.2.2 数据库的修改和删除	26	4.2 视图	78
2.3 以命令方式创建数据库	28	4.2.1 视图概念	78
2.3.1 创建数据库	28	4.2.2 创建视图	78
2.3.2 修改数据库	31	4.2.3 更新视图	82
2.3.3 删除数据库	34	4.2.4 修改视图的定义	84
第 3 章 表的创建和操作	35	4.2.5 删除视图	85
3.1 表结构和数据类型	35		
3.1.1 表和表结构	35		
3.1.2 数据类型	36		

第 5 章 游标	86	7.2 数据完整性	130
5.1 声明游标	86	7.2.1 创建实体完整性:	
5.2 打开游标	88	PRIMARY KEY/UNIQUE	131
5.3 读取数据	89	7.2.2 创建域完整性:	
5.4 关闭和删除游标	92	CHECK/CREATE RULE	134
第 6 章 T-SQL 语言	93	7.2.3 创建参照完整性:	
6.1 常量、变量与用户定义类型	93	FOREIGN KEY	138
6.1.1 常量	93	第 8 章 存储过程和触发器	143
6.1.2 用户定义类型	95	8.1 存储过程	143
6.1.3 变量: DECLARE/SET/ SELECT	97	8.1.1 存储过程的类型	143
6.2 运算符与表达式	101	8.1.2 存储过程的创建与执行:	
6.3 流程控制语句	105	CREATE PROCEDURE	144
6.3.1 语句块: BEGIN...END	106	8.1.3 存储过程的修改:	
6.3.2 条件语句: IF...ELSE	106	ALTER PROCEDURE	148
6.3.3 多重分支: CASE	108	8.1.4 存储过程的删除:	
6.3.4 无条件转移语句: GOTO	109	DROP PROCEDURE	149
6.3.5 循环语句: WHILE/BREAK/ CONTINUE	109	8.1.5 以界面方式操作存储过程	149
6.3.6 返回语句: RETURN	110	8.2 触发器	151
6.3.7 等待语句: WAITFOR DELAY/ TIME	110	8.2.1 触发器的类型	151
6.3.8 错误处理语句: TRY...CATCH	111	8.2.2 触发器的创建:	
6.4 系统内置函数	111	CREATE TRIGGER	151
6.4.1 系统内置函数介绍	111	8.2.3 触发器的修改:	
6.4.2 常用系统标量函数	112	ALTER TRIGGER	157
6.5 用户定义函数: CREATE/ALTER/ DROP FUNCTION	119	8.2.4 触发器的删除:	
6.5.1 标量函数: SELECT/EXEC	119	DROP TRIGGER	158
6.5.2 内嵌表值函数: FROM 表(...)	121	8.2.5 以界面方式操作触发器	158
6.5.3 用户定义函数的删除: DROP FUNCTION	124	第 9 章 备份与恢复	160
第 7 章 索引和完整性	125	9.1 备份和恢复概述	160
7.1 索引	125	9.1.1 数据库备份概念	160
7.1.1 索引的分类	125	9.1.2 数据库恢复概念	162
7.1.2 索引的创建:		9.2 数据库备份	163
CREATE INDEX	125	9.2.1 创建备份设备	163
7.1.3 重建索引: ALTER INDEX	129	9.2.2 以命令方式备份数据库:	
7.1.4 索引的删除: DROP INDEX	129	BACKUP DATABASE	165
		9.2.3 以界面方式备份数据库	168
		9.3 数据库恢复	169
		9.3.1 以命令方式恢复数据库:	
		BESTORE DATABASE	169
		9.3.2 以界面方式恢复数据库	172
		9.4 附加数据库和数据库复制	173
		9.4.1 附加数据库	173

9.4.2	数据库复制	174	11.1.3	事务处理: BEGIN/COMMIT/ ROLLBACK/SAVE/ROLLBACK TRANSACTION	199
第 10 章	系统安全管理	175	11.1.4	事务隔离级: SET TRANSACTION ISOLATION LEVEL	202
10.1	SQL Server 2014 的安全机制	175	11.2	锁定	203
10.1.1	SQL Server 身份验证模式	175	11.2.1	锁定粒度	204
10.1.2	SQL Server 安全性机制	176	11.2.2	锁定模式	204
10.1.3	SQL Server 数据库安全 验证过程	177	11.3	自动化管理	205
10.2	建立和管理用户账户	177	11.3.1	SQL Server 代理	205
10.2.1	以界面方式管理用户账户	177	11.3.2	操作员	206
10.2.2	以命令方式管理用户账户: CREATE LOGIN/USER	181	11.3.3	作业	206
10.3	角色管理	184	11.3.4	警报	206
10.3.1	固定服务器角色	184	11.3.5	数据库邮件	206
10.3.2	固定数据库角色	185	11.3.6	维护计划向导	207
10.3.3	自定义数据库角色: CREATE/DROP ROLE	187	习题		208
10.3.4	应用程序角色	189	第 0 章	数据库基础	208
10.4	数据库权限的管理	190	第 1 章	SQL Server 2014 简介	209
10.4.1	授予权限: GRANT	190	第 2 章	数据库创建	209
10.4.2	拒绝权限	193	第 3 章	表的创建和操作	210
10.4.3	撤销权限	194	第 4 章	数据库的查询和视图	212
10.5	数据库架构的定义和使用	195	第 5 章	游标	213
10.5.1	以界面方式创建架构	195	第 6 章	T-SQL 语言	213
10.5.2	以命令方式创建架构: CREATE SCHEMA	196	第 7 章	索引和完整性	214
第 11 章	其他概念	198	第 8 章	存储过程和触发器	215
11.1	事务	198	第 9 章	备份与恢复	216
11.1.1	事务与 ACID 属性	198	第 10 章	系统安全管理	217
11.1.2	多用户使用问题	199	第 11 章	其他概念	217

第 2 部分 实 验

第 2 章	数据库创建	219	第 7 章	索引和完整性	222
第 3 章	表的创建和操作	219	第 8 章	存储过程和触发器	222
第 4 章	数据库的查询和视图	221	第 9 章	备份与恢复	222
第 5 章	游标	221	第 10 章	系统安全管理	223
第 6 章	T-SQL 语言	221			

第3部分 SQL Server 2014 综合应用

实习 0 创建实习数据库	224	实习 2.2.3 连接 SQL Server 2014	259
实习 0.1 创建数据库及其对象	224	实习 2.3 系统主页设计	262
实习 0.2 功能和界面	229	实习 2.3.1 创建 PHP 环境	262
实习 1 PHP 5/SQL Server 2014		实习 2.3.2 功能导航	263
学生成绩管理系统	232	实习 2.4 学生管理	265
实习 1.1 PHP 开发平台搭建	232	实习 2.4.1 界面设计	265
实习 1.1.1 创建 PHP 环境	232	实习 2.4.2 功能实现	268
实习 1.1.2 Eclipse 安装与配置	235	实习 2.5 成绩管理	274
实习 1.2 PHP 开发入门	237	实习 2.5.1 界面设计	274
实习 1.2.1 PHP 项目的建立	237	实习 2.5.2 功能实现	277
实习 1.2.2 PHP 项目的运行	238	实习 3 ASP.NET 4/SQL Server 2014	
实习 1.2.3 PHP 连接		学生成绩管理系统	283
SQL Server 2014	238	实习 3.1 ADO.NET 架构原理	283
实习 1.3 系统主页设计	239	实习 3.2 创建 ASP.NET 项目	284
实习 1.3.1 主界面	239	实习 3.2.1 ASP.NET 项目的建立	284
实习 1.3.2 功能导航	240	实习 3.2.2 ASP.NET 4 连接	
实习 1.4 学生管理	242	SQL Server 2014	285
实习 1.4.1 界面设计	242	实习 3.3 系统主页设计	285
实习 1.4.2 功能实现	245	实习 3.3.1 主界面	285
实习 1.5 成绩管理	247	实习 3.3.2 功能导航	287
实习 1.5.1 界面设计	247	实习 3.4 学生管理	289
实习 1.5.2 功能实现	249	实习 3.4.1 界面设计	289
实习 2 Java EE 7/SQL Server 2014		实习 3.4.2 功能实现	291
学生成绩管理系统	251	实习 3.5 成绩管理	296
实习 2.1 Java EE 开发平台搭建	251	实习 3.5.1 界面设计	296
实习 2.1.1 安装软件	251	实习 3.5.2 功能实现	298
实习 2.1.2 环境整合	255	附录 A 学生成绩 (pxscj) 数据库	
实习 2.2 创建 Struts 2 项目	257	表样本数据	304
实习 2.2.1 创建 Java EE 项目	257		
实习 2.2.2 加载 Struts 2 包	258		

第1部分 实用教程

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。如果学习过数据库原理，那么本章数据库原理部分仅仅作为一个参考。

0.1 数据库基本概念

0.1.1 数据库系统

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内。例如，把一个学校的学生、课程、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。

2. 数据库管理系统

数据库管理系统 (DBMS) 按一定的数据模型组织数据形成数据库，并对数据库进行管理。简单地说，DBMS 就是管理数据库的系统 (软件)。数据库系统管理员 (DBA, DataBase Administrator) 通过 DBMS 对数据库进行管理。

目前，比较流行的 DBMS 有：SQL Server、Oracle、MySQL、Sybase、DB2、Access、Visual FoxPro 等。其中，SQL Server 是目前最流行的中大型关系数据库管理系统。本书介绍的是最新的 SQL Server 2014 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。如图 0.1 所示描述了数据库系统的构成。

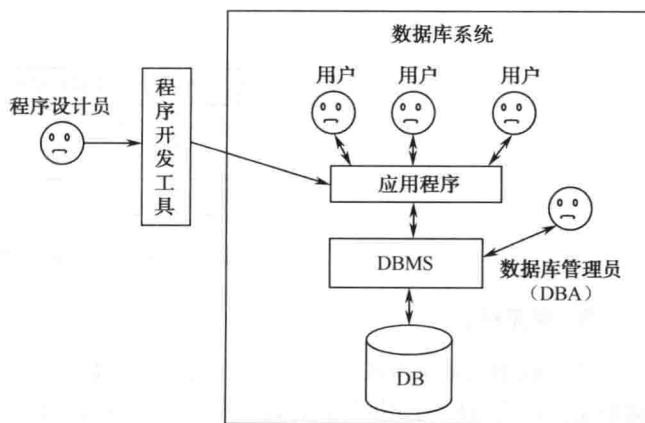


图 0.1 数据库系统的构成

图 0.1 所示描述了数据库系统的构成。

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。

如图 0.2 所示为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。但网状结构的复杂,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,使得数据量增大;数据的修改不方便。

如图 0.3 所示为按网状模型组织的数据示例。

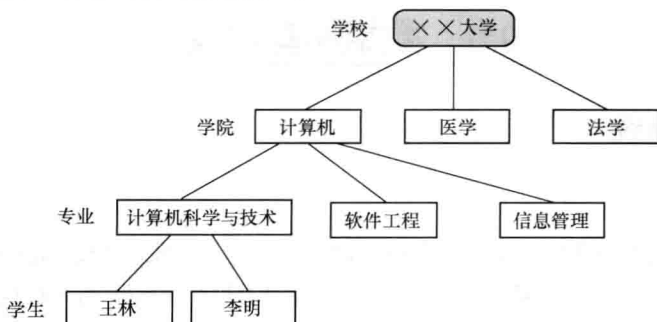


图 0.2 按层次模型组织的数据示例

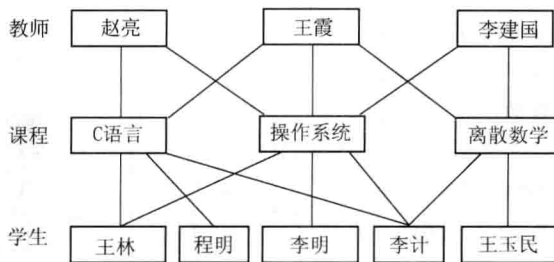


图 0.3 按网状模型组织的数据示例

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,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、更好的安全保密性。但数据库大时,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所涉及的“学生”、“课程”和“成绩”三个表中,“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。如表 0.1、表 0.2 和表 0.3 所示分别描述了学生成绩管理系统中“学生”、“课程”和“成绩”三个表的部分数据。

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
081101	101	80	081108	101	85
081101	102	78	081108	102	64
081101	206	76	081108	206	87
081103	101	62	081202	101	65
081103	102	70	081204	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字，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。表 0.3 中的“学号”和“课程号”可唯一标识每一个学生一门课程的成绩。

有时，一个表可能有多个码，比如表 0.1 中，姓名不允许重名，则“学号”、“姓名”均是学生信息表码。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 的名字为 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 的名字为 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 的名字为 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。SQL Server

是目前最流行的关系数据库管理系统。

0.1.3 关系型数据库语言

SQL（Structured Query Language，结构化查询语言）是用于关系数据库查询的结构化语言。SQL 的功能包括数据定义语言（DDL）、数据操纵语言（DML）、数据查询语言（DQL）和数据控制语言（DCL）。

（1）数据定义语言（DDL）。DDL 用于执行数据库的任务，对数据库以及数据库中的各种对象进行创建、删除、修改等操作。如前所述，数据库对象主要包括表、默认约束、规则、视图、触发器、存储过程。DDL 包括的主要语句及功能如表 0.4 所示。

表 0.4 DDL 主要语句及功能

语 句	功 能
CREATE	创建数据库或数据库对象
ALTER	对数据库或数据库对象进行修改
DROP	删除数据库或数据库对象

（2）数据操纵语言（DML）。DML 用于操纵数据库中的各种对象，检索和修改数据。DML 包括的主要语句及功能如表 0.5 所示。

表 0.5 DML 主要语句及功能

语 句	功 能
SELECT	从表或视图中检索数据
INSERT	将数据插入到表或视图中
UPDATE	修改表或视图中的数据
DELETE	从表或视图中删除数据

（3）数据控制语言（DCL）。DCL 用于安全管理，确定哪些用户可以查看或修改数据库中的数据。DCL 包括的主要语句及功能如表 0.6 所示。

表 0.6 DCL 主要语句及功能

语 句	功 能
GRANT	授予权限
REVOKE	收回权限
DENY	收回权限，并禁止从其他角色继承许可权限

（4）数据查询语言（DQL）。主要通过 SELECT 语言实现各种查询功能。

T-SQL 是在 SQL Server 使用的 SQL 语言。它是 ANSI SQL 的扩展加强版 SQL 语言，除了提供标准的 SQL 命令之外，T-SQL 还对 SQL 做了许多补充。

0.2 数据库设计

数据模型按不同的应用层次分成三种类型：分别是概念数据模型、逻辑数据模型、物理数据模型。

0.2.1 概念模型

概念模型 (Conceptual Data Model) 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, 它使数据库的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。概念数据模型必须换成逻辑数据模型, 才能在 DBMS 中实现。

概念模型用于信息世界的建模: 一方面, 应该具有较强的语义表达能力, 能够方便直接表达应用中的各种语义知识; 另一方面, 它还应该简单、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, 在学生成绩管理系统中主要涉及“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。就“学生”实体集而言, 它的属性有: 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分的。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, 用带半圆的矩形框表示属性, 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斜线, 则可以用如图 0.4 所示的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表示称为实体 (Entity)-联系 (Relationship) 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三种情况之一。

1. 一对一的联系 (1:1)

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, 一个正长只属于一个班级。“班级”与“班长”两个实体集的 E-R 模型如图 0.5 所示。

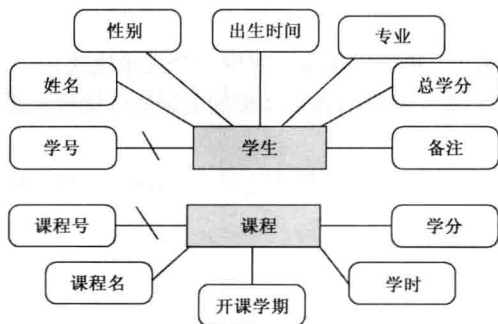


图 0.4 学生和课程实体集属性的描述

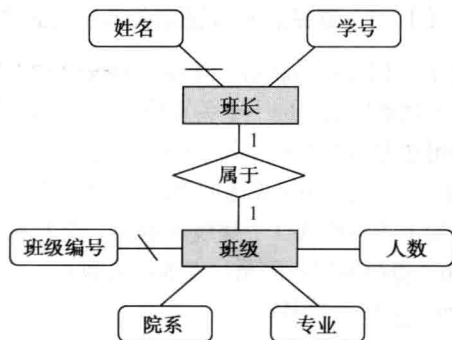


图 0.5 “班级”与“班长”两个实体集的 E-R 模型

2. 一对多的联系 (1:n)

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)

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。

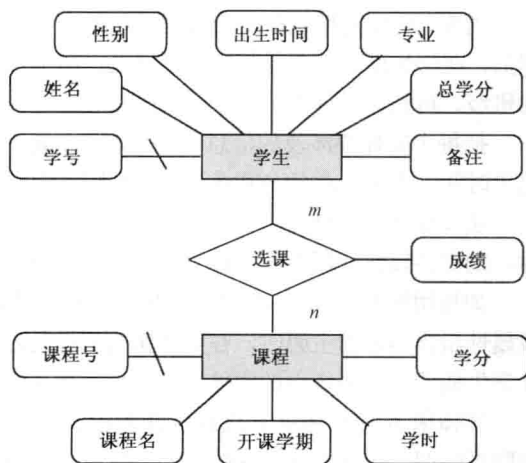
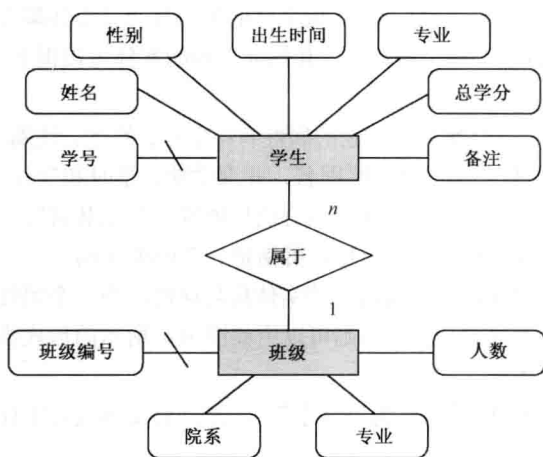


图 0.6 “班级”与“学生”两个实体集的 E-R 模型 图 0.7 “学生”与“课程”两个实体集的 E-R 模型

0.2.2 逻辑模型

逻辑模型 (Logical Data Model) 是用户从数据库所看到的模型, 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?

前面已介绍了实体集之间的联系, 下面将根据三种联系从 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

对于 (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) 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 (下横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) 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

对于 (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) 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) 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

对于 ($m:n$) 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,已介绍了根据 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。

在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 物理模型

物理模型 (Physical Data Model) 是面向计算机物理表示的模型,描述了数据在储存介质上的组织结构,它不但与具体的 DBMS 有关,而且还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

ODBC 即开放式数据库互连 (Open DataBase Connectivity), 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就可以通过 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 本质上是一组数据库访问 API (应用程序编程接口), 由一组函数调用组成, 核心是 SQL 语句。

在具体操作时, 首先必须用 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, 建立起 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 就能建立起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, 然后再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而对数据库服务器资源的占用。

3. ADO.NET 数据库接口

ADO.NET 数据模型由 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环境下工作。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, 始终做到保持与数据源连接, 不符合网站的要求, 不仅效率低, 付出代价高, 而且常会引发由于多个用户同时访问而带来的冲突。因此, ADO.NET 系统集中主要精力来解决在断开与数据源连接的环境下处理数据的问题。

ADO.NET 提供了面向对象的数据库视图, 并且在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一个数据库或者从另一个数据库获得数据等细节问题。图 0.8 显示了通过 ADO.NET 访问数据库的接口模型。

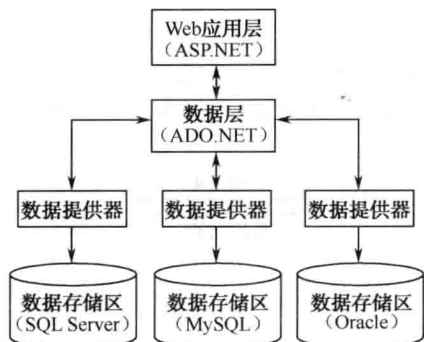


图 0.8 通过 ADO.NET 访问数据库的接口模型

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，以 Java 语言编写的用于数据库连接和操作的类和接口，可为多种关系数据库提供统一的访问方式。通过 JDBC 对数据库的访问包括 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 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，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：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 方式进行数据库连接，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 6.0、Visual C++6.0、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