



高等学校计算机教材

# SQL Server

## 实用教程 (第4版)

### (SQL Server 2012版)

◎ 郑阿奇 主编

◎ 刘启芬 顾韵华 编著



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>

高等学校计算机教材

# SQL Server 实用教程

(第4版)

(SQL Server 2012 版)

郑阿奇 主编

刘启芬 顾韵华 编著

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 2012 中文版为平台,系统地介绍数据库基础、SQL Server 2012 和综合应用等内容。SQL Server 2012 内容主要包括数据库的创建,表的创建和操作,数据库的查询、视图和游标,T-SQL 语言,索引和数据完整性,存储过程和触发器,备份与恢复,系统安全管理和 SQL Server 其他功能等。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。实习部分通过创建 SQL Server 2012 实习数据库及其数据库常用对象,将数据库、表、视图、触发器、完整性、存储过程等进行综合应用;之后,通过学生成绩管理系统,介绍目前流行的 4 种开发平台操作 SQL Server 2012 数据库,包括 PHP 5.3.29、Java EE (8/8/2014)、ASP.NET 4.5、Visual C#2013,统一开发学生成绩管理系统。

本书免费提供教学课件和配套的客户端(SQL Server 2012 应用系统数据库)和所有源程序文件。

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。

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 2012 版 / 郑阿奇主编; 刘启芬, 顾韵华编著. —4 版. —北京: 电子工业出版社, 2015.7

ISBN 978-7-121-26038-4

I. ①S… II. ①郑… ②刘… ③顾… III. ①关系数据库系统—教材 IV. ①TP311.138

中国版本图书馆 CIP 数据核字(2015)第 097800 号

策划编辑: 袁 玺

责任编辑: 郝黎明

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

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: 25.25 字数: 646.4 千字

版 次: 2009 年 12 月第 1 版

2015 年 7 月第 2 版

印 次: 2015 年 7 月第 1 次印刷

定 价: 52.00 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](mailto:zltz@phei.com.cn), 盗版侵权举报请发邮件至 [dbqq@phei.com.cn](mailto:dbqq@phei.com.cn)。

服务热线: (010)88258888。

# 前 言

20 世纪 80 年代后期, Microsoft、Sybase 和 Ashton-Tate 这 3 家公司共同开发了最初的 SQL Server, 1988 年, 把该产品移植到 OS/2 上, 1992 年, 将 SQL Server 移植到了 Windows NT 平台上。1993 年, SQL Server 4.2 面世, 它是个桌面数据库系统, 虽然其功能相对有限, 但是采用 Windows GUI, 向用户提供了易于使用的用户界面。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 版和 2012 版。目前 SQL Server 已经是市场上最流行的大中型关系型数据库管理系统。

为了适应市场的需要, 我国高校的许多专业都开设介绍 SQL Server 数据库管理系统的课程。本书以 SQL Server 2012 为平台, 结合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, 然后是 Server 综合应用。SQL Server 2012 部分内容包括数据库的创建, 表的创建和操作, 数据库查询、视图和游标, T-SQL 语言, 索引和数据完整性, 存储过程和触发器, 备份与恢复, 系统安全管理和 SQL Server 其他功能等。这部分内容的介绍不再强调命令的格式, 而是突出主要功能介绍和配套举例。而详细格式和功能说明可参考有关文档。

SQL Server 综合应用以当前最流行的数据库应用开发工具为平台开发 SQL Server 数据库应用系统, 平台包括 PHP5.3.29、Java EE (8/8/2014)、ASP.NET4.5 和 Visual C#2013 等, 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, 教和学将非常方便。

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。

本书配有教学课件和配套的客户端 (SQL Server 2012 应用系统数据库) 和所有源程序文件。需要者请到出版社网站 (<http://www.hzbook.com>) 免费注册下载。

本书由南京师范大学郑阿奇担任主编, 刘启芬、顾韵华编著。参加本书编写的还有丁有和、徐文胜、彭作民、孙德荣、周怡明、刘博宇、郑进、周何骏、陶卫冬、严大牛、周怡君、吴明祥、王海娇、韩翠青、孙承龙等。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。

编 者

# 目 录

第 1 章 数据库基础和 SQL Server 2012 环境的构建	1
1.1 数据库基本概念	1
1.1.1 数据库、数据库管理系统与数据库系统	1
1.1.2 数据模型	2
1.1.3 关系数据库语言	3
1.2 数据库设计	4
1.2.1 概念数据模型	4
1.2.2 逻辑数据模型	5
1.2.3 物理数据模型	7
1.3 数据库应用系统	7
1.3.1 应用系统的数据接口	7
1.3.2 C/S 架构的应用系统	9
1.3.3 B/S 架构的应用系统	9
1.4 SQL Server 2012 简介	10
1.4.1 SQL Server 的演变	10
1.4.2 SQL Server 2012 的版本及支持功能	11
1.4.3 服务器组件和管理工具	12
1.5 SQL Server 2012 的安装和运行	14
1.5.1 SQL Server 2012 的安装环境	14
1.5.2 SQL Server 2012 的安装和运行	15
第 2 章 数据库的创建	20
2.1 数据库及其数据库对象	20
2.2 通过界面方式创建数据库	22
2.2.1 数据库的创建	22
2.2.2 数据库的修改和删除	24
2.3 通过命令方式创建数据库	28
2.3.1 创建数据库: CREATE DATABASE	28
2.3.2 修改数据库: ALTER DATABASE	31
2.3.3 删除数据库: DROP DATABASE	34
2.3.4 数据库快照: AS SNAPSHOT	35
习题	36
第 3 章 表的创建和操作	37
3.1 表结构和数据类型	37
3.1.1 表和表结构	37
3.1.2 数据类型	38
3.1.3 表结构设计	43

3.2	通过界面方式创建表	44
3.2.1	创建表	44
3.2.2	修改表结构	45
3.2.3	删除表	47
3.3	通过命令方式创建表	47
3.3.1	创建表: CREATE TABLE	47
3.3.2	修改表结构: ALTER TABLE	49
3.3.3	删除表: DROP TABLE	51
3.4	通过界面方式操作表数据	51
3.5	通过命令方式操作表数据	53
3.5.1	插入记录: INSERT	53
3.5.2	修改记录: UPDATE	56
3.5.3	删除记录: DELETE/TRUNCATE TABLE	56
3.5.4	同步记录: MERGE	58
	习题	59
<b>第 4 章</b>	<b>数据库的查询、视图和游标</b>	<b>60</b>
4.1	关系运算	60
4.2	数据库的查询: SELECT	62
4.2.1	选择查询结果输出列: */列表	62
4.2.2	选择查询条件: WHERE	67
4.2.3	指定查询对象: FROM	76
4.2.4	连接: =/JOIN	81
4.2.5	指定查询结果分组方法: GROUP BY	84
4.2.6	指定查询结果分组后的筛选条件: HAVING	86
4.2.7	指定查询结果排序顺序: ORDER BY	87
4.2.8	其他: INTO/UNION/EXCEPT/INTERSECT/ CTE	88
4.3	视图	91
4.3.1	视图简介	91
4.3.2	创建视图: CREATE VIEW	91
4.3.3	查询视图: SELECT	95
4.3.4	更新视图: INSERT/UPDATE/DELETE	96
4.3.5	修改视图定义: ALTER VIEW	97
4.3.6	删除视图: DROP VIEW	98
4.4	游标	99
4.4.1	声明游标: DECLARE CURSOR	99
4.4.2	打开游标: OPEN	102
4.4.3	读取数据: FETCH	103
4.4.4	关闭与删除游标: CLOSE/DEALLOCATE	105
	习题	106
<b>第 5 章</b>	<b>T-SQL 语言</b>	<b>107</b>
5.1	T-SQL 语言简介	107

5.2	常量、变量与数据类型	108
5.2.1	常量	108
5.2.2	变量: DECLARE/SET/SELECT	109
5.2.3	数据类型	113
5.3	运算符与表达式	116
5.4	流程控制语句	121
5.4.1	语句块: BEGIN...END	121
5.4.2	条件语句: IF...ELSE	122
5.4.3	分支语句: CASE	124
5.4.4	无条件转移语句: GOTO	125
5.4.5	循环语句: WHILE/BREAK/CONTINUE	125
5.4.6	返回语句: RETURN	126
5.4.7	等待语句: WAITFOR DELAY/TIME	126
5.4.8	错误处理语句: TRY...CATCH	127
5.5	系统内置函数	127
5.5.1	系统内置函数介绍	127
5.5.2	常用系统标量函数	128
5.6	用户定义函数: CREATE/ALTER/ DROP FUNCTION	135
5.6.1	标量函数: SELECT/EXEC	135
5.6.2	内嵌表值函数: FROM 表(...)	137
5.6.3	用户定义函数的删除: DROP FUNCTION	140
	习题	140
<b>第 6 章</b>	<b>索引和数据完整性</b>	<b>141</b>
6.1	索引	141
6.1.1	索引的分类	141
6.1.2	索引的创建: CREATE INDEX	142
6.1.3	重建索引: ALTER INDEX	146
6.1.4	索引的删除: DROP INDEX	146
6.2	数据完整性	147
6.2.1	数据完整性分类	147
6.2.2	实体完整性: PRIMARY KEY/UNIQUE	148
6.2.3	域完整性: CHECK/CREATE RULE	151
6.2.4	参照完整性: FOREIGN KEY	156
	习题	159
<b>第 7 章</b>	<b>存储过程和触发器</b>	<b>160</b>
7.1	存储过程	160
7.1.1	存储过程的类型	160
7.1.2	存储过程的创建与执行: CREATE PROCEDURE	161
7.1.3	存储过程的修改: ALTER PROCEDURE	166
7.1.4	存储过程的删除: DROP PROCEDURE	167

7.1.5	通过界面方式操作存储过程	167
7.2	触发器	168
7.2.1	触发器的类型	168
7.2.2	触发器的创建: CREATE TRIGGER	169
7.2.3	触发器的修改: ALTER TRIGGER	175
7.2.4	触发器的删除: DROP TRIGGER	176
7.2.5	通过界面方式操作触发器	177
	习题	177
<b>第 8 章</b>	<b>备份与恢复</b>	<b>178</b>
8.1	备份和恢复概述	178
8.1.1	备份和恢复需求分析	178
8.1.2	数据库备份简介	178
8.1.3	数据库恢复简介	180
8.2	数据库备份	181
8.2.1	创建备份设备	181
8.2.2	通过命令方式备份数据库: BACKUP DATABASE	183
8.2.3	通过界面方式备份数据库	187
8.3	数据库恢复	188
8.3.1	检查点	188
8.3.2	通过命令方式恢复数据库: RESTORE DATABASE	189
8.3.3	界面方式恢复数据库	192
8.4	附加数据库和数据库复制	193
8.4.1	附加数据库	193
8.4.2	复制数据库	194
	习题	194
<b>第 9 章</b>	<b>系统安全管理</b>	<b>195</b>
9.1	SQL Server 2012 的安全机制	195
9.1.1	SQL Server 身份验证模式	195
9.1.2	SQL Server 安全性机制	195
9.1.3	SQL Server 数据库安全验证	196
9.2	建立和管理用户账户	196
9.2.1	通过界面方式管理用户账户	197
9.2.2	通过命令方式管理用户账户: CREATE LOGIN/USER	200
9.3	角色管理	203
9.3.1	固定服务器角色	203
9.3.2	固定数据库角色	204
9.3.3	自定义数据库角色: CREATE/DROP ROLE	206
9.3.4	应用程序角色	208
9.4	数据库权限的管理	210
9.4.1	授予权限: GRANT	210



9.4.2	拒绝权限	213
9.4.3	撤销权限	214
9.5	数据库架构的定义和使用	215
9.5.1	通过界面方式创建架构	215
9.5.2	通过命令方式创建架构: CREATE SCHEMA	217
	习题	218
<b>第 10 章</b>	<b>SQL Server 其他功能</b>	<b>219</b>
10.1	事务	219
10.1.1	事务与 ACID 属性	219
10.1.2	多用户使用问题	220
10.1.3	事务处理: BEGIN/ COMMIT/ROLLBACK/SAVE/ROLLBACK TRANSACTION	220
10.1.4	事务隔离级: SET TRANSACTION ISOLATION LEVEL	223
10.2	锁定	225
10.2.1	锁定粒度	225
10.2.2	锁定模式	225
10.3	自动化管理	226
10.3.1	SQL Server 代理	227
10.3.2	操作员	227
10.3.3	作业	227
10.3.4	警报	228
10.3.5	数据库邮件	228
10.3.6	维护计划向导	228
	习题	228
<b>实验 1</b>	<b>SQL Server 2012 环境</b>	<b>230</b>
<b>实验 2</b>	<b>创建数据库和表</b>	<b>234</b>
<b>实验 3</b>	<b>表数据的插入、修改和删除</b>	<b>238</b>
<b>实验 4</b>	<b>数据库的查询和视图</b>	<b>242</b>
T4.1	数据库的查询	242
T4.2	视图的使用	248
<b>实验 5</b>	<b>T-SQL 编程</b>	<b>251</b>
<b>实验 6</b>	<b>索引和数据完整性的使用</b>	<b>256</b>
T6.1	索引	256
T6.2	数据完整性	257
<b>实验 7</b>	<b>存储过程和触发器的使用</b>	<b>260</b>
T7.1	存储过程	260
T7.2	触发器	262
<b>实验 8</b>	<b>备份恢复与导入导出</b>	<b>265</b>
T8.1	数据库的备份	265

T8.2 数据库的恢复 .....	266
<b>实验 9 数据库的安全性 .....</b>	<b>268</b>
T9.1 数据库用户的管理 .....	268
T9.2 服务器角色的应用 .....	269
T9.3 数据库权限管理 .....	270
<b>实验 10 SQL Server 与 XML .....</b>	<b>272</b>
<b>实习 0 SQL Server 2012 实习数据准备 .....</b>	<b>276</b>
P0.1 数据库 .....	276
P0.2 基本表 .....	276
P0.3 视图 .....	280
P0.4 完整性约束 .....	281
P0.5 存储过程 .....	284
P0.6 触发器 .....	287
P0.7 系统功能 .....	289
<b>实习 1 PHP/SQL Server 2012 图书管理系统 .....</b>	<b>291</b>
P1.1 开发环境的搭建 .....	291
P1.2 创建 PHP 项目 .....	295
P1.3 模块 1: 借书还书 .....	298
P1.4 模块 2: 读者管理 .....	304
<b>实习 2 Java EE/SQL Server 2012 图书管理系统 .....</b>	<b>315</b>
P2.1 开发环境的搭建 .....	315
P2.2 创建 Struts 2 项目 .....	320
P2.3 模块 1: 借书还书 .....	324
P2.4 模块 2: 读者管理 .....	336
<b>实习 3 ASP.NET 4.5(C#)/SQL Server 2012 图书管理系统 .....</b>	<b>348</b>
P3.1 创建图书管理网站 .....	348
P3.2 模块 1: 借书还书 .....	352
P3.3 模块 2: 读者管理 .....	357
<b>实习 4 Visual C# 2013/SQL Server 2012 图书管理系统 .....</b>	<b>368</b>
P4.1 创建图书管理系统 .....	368
P4.2 功能实现: 借书还书 .....	371
<b>附录 A 学生成绩(pxscj)数据库表样本数据 .....</b>	<b>377</b>
<b>附录 B SQL Server 与 XML .....</b>	<b>379</b>
B.1 XML 类型 .....	379
B.2 在 SQL Server 中导入 XML 数据 .....	379
B.3 XQuery 的基本用法 .....	381
B.4 FOR XML 子句的使用 .....	387

# 第 1 章

## 数据库基础和

## SQL Server 2012 环境的构建

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作为参考。

### 1.1 数据库基本概念

#### 1.1.1 数据库、数据库管理系统与数据库系统

##### 1. 数据库

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

- (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中的数据。
- (6) 数据库系统的故障恢复功能：使数据库在运行出现故障时进行数据库恢复，以保证数据库的可靠运行。
- (7) 在网络环境下访问数据库的功能。

(8) 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，比较流行的数据库管理系统有 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 (Relational Database Management System, RDBMS)，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。如图 1.1 所示描述了数据库系统的构成。

### 1.1.2 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。但其结构不够灵活;同一属性的数据要存储多次,数据冗余大;不适合于拓扑空间数据的组织。

如图 1.2 所示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,且数据冗余小。但网状结构的复杂,增加了用户查询和定位的困难度;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 1.3 所示为按网状模型组织的数据示例。

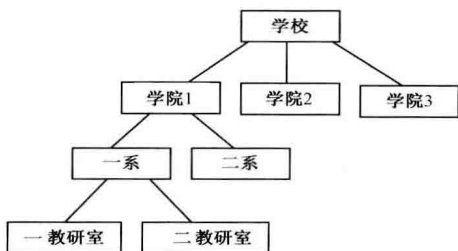


图 1.2 按层次模型组织的数据示例

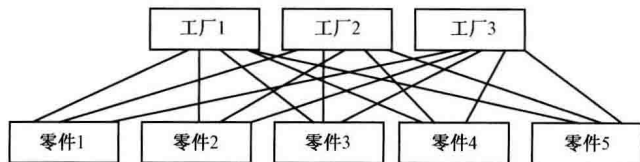


图 1.3 按网状模型组织的数据示例

#### 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和更好的安全保密性。但数据库大时,其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,学生成绩管理系统所涉及的“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。如表 1.1~表 1.3 所示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  $n$  个字段的表的结构可表示为:关系表名(字段名<sub>1</sub>, ..., 字段名 <sub>$n$</sub> ),通常把表的结构称为关系模式。

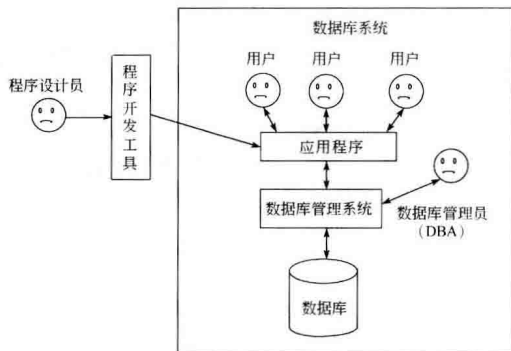


图 1.1 数据库系统的构成

表 1.1 “学生”表

学号	姓名	性别	出生时间	专业	总学分	备注
081101	王林	男	1990-02-10	计算机	50	
081103	王燕	女	1989-10-06	计算机	50	
081108	林一帆	男	1989-08-05	计算机	52	已提前修完一门课
081202	王林	男	1989-01-29	通信工程	40	有一门课不及格，待补考
081204	马琳琳	女	1989-02-10	通信工程	42	

表 1.2 “课程”表

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

表 1.3 “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

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 1.1 中的“学号”可唯一标识每一个学生，表 1.2 中的“课程号”可唯一标识每一门课程。表 1.3 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。例如，表 1.1 中，姓名不允许重名，而“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，一般用下划线标出主码。

设表 1.1 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 1.2 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 1.3 的名称为 cjb，关系模式可分别表示为：cjb (学号, 课程号, 成绩)。

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，主流的关系数据库管理系统包括 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。

### 1.1.3 关系数据库语言

SQL 语言的全名是结构化查询语言 (Structured Query Language, SQL)，是用于数据库中的标准数据查询语言，IBM 公司最早使用在其开发的数据库系统。1986 年 10 月，美国 ANSI 对 SQL 进行规范后，以此作为关系数据库管理系统的标准语言。

作为关系数据库的标准语言，它已被众多商用数据库管理系统产品采用，不过不同的数据库管理系统在其实践过程中都对 SQL 规范做了某些改变和扩充。所以，实际上，不同数据库管理系统之间的 SQL 语言不能完全相互通用。例如，Microsoft 公司的 SQL Server 支持的是 T-SQL，而甲骨文公司的 Oracle 数据库所使用的 SQL 语言则是 PL-SQL。

## 1.2 数据库设计

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

### 1.2.1 概念数据模型

概念数据模型 (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, 摆脱计算机系统及数据库管理系统的具体技术问题, 集中精力分析数据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, 才能在数据库管理系统中实现。

概念数据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、直接地表达应用中的各种语义知识, 另一方面还应该简单、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”, 而把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有好多, 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 1.4 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, 使用带半圆的矩形框表示属性, 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)

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 1.5 所示。

#### 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 1.6 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 1.7 所示。

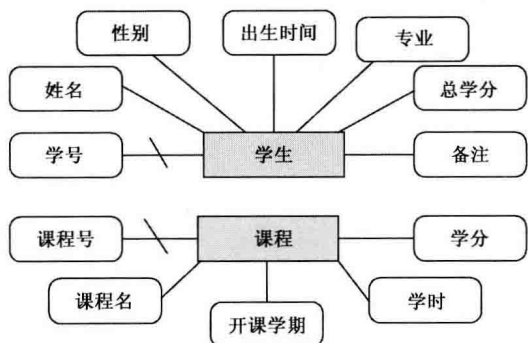


图 1.4 “学生”和“课程”实体集属性的描述

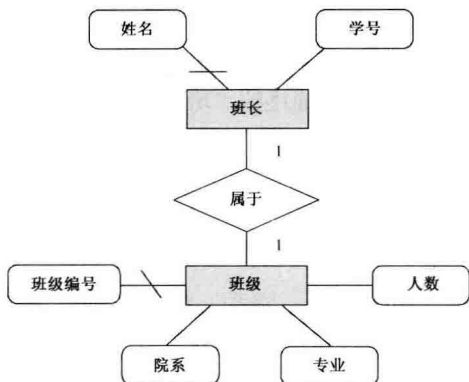


图 1.5 “班级”与“班长”两个实体集的 E-R 模型

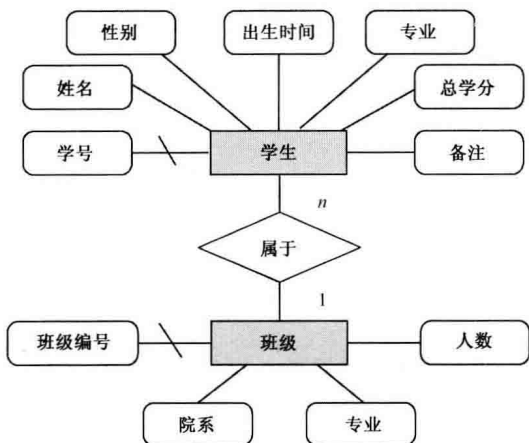


图 1.6 “班级”与“学生”两个实体集的 E-R 模型

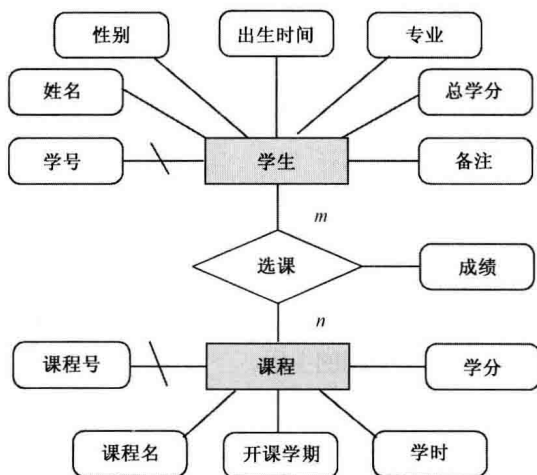


图 1.7 “学生”与“课程”两个实体集间的 E-R 模型

## 1.2.2 逻辑数据模型

逻辑数据模型 (Logical Data Model) 是用户从数据库所看到的模型，是具体的数据库管理系统所支持的数据模型。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，需要确定包含的表和每个表的结构。

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

### 1. 一对一联系的 E-R 图到关系模式的转换

对于一对一的联系，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：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，图 1.5 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（下画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, 图 1.5 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. 一对多联系的 E-R 图到关系模式的转换

对于一对多的联系, 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: 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, 图 1.6 描述的“班级 (bjb)”与“学生 (xsb)”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)

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: 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 1.6 描述的“班级 (bjb)”与“学生 (xsb)”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)

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

## 3. 多对多联系的 E-R 图到关系模式的转换

对于多对多的联系, 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 1.7 描述的“学生 (xsb)”与“课程 (kcb)”实体集通过成绩 (cgb) 联系 E-R 模型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, 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。

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。



### 1.2.3 物理数据模型

物理数据模型 (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, 它不但与具体的数据库管理系统有关, 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。数据库管理系统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。

## 1.3 数据库应用系统

### 1.3.1 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。虽然现有数据库管理系统几乎都遵循 SQL 标准, 但不同厂家开发的数据库管理系统有差异, 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数据库管理系统的通用方法、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 Microsoft 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 Microsoft 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