

考试号：70-229 课程号：2071, 2073

SQL Server 2000 数据库编程

梁方明 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

微软认证高级技术培训教材系列

考试号: 70-229 课程号: 2071, 2073

SQL Server 2000 数据库编程

梁方明 编写



北京希望电子出版社
Beijing Hope Electronic Press
www.bhp.com.cn

内 容 简 介

本书是微软认证高级技术培训教材系列之一，对应考试号为 70-229，课程号为 2071，2073。

SQL Server 2000 是 SQL Server 家族中的最新版本，SQL Server 2000 扩展了其以前版本的性能、可靠性、质量和易用性。SQL Server 2000 还增加了几种新的功能，由此成为大规模联机事务处理（OLTP）和电子商务应用程序的优秀数据库平台。它也增加了对 XML 的支持，同时增强了数据仓库的功能。

本书从第 1 至第 8 章讲述 SQL Server 的基础知识，介绍了如何使用 Transact-SQL 语句编程，表、索引等的创建及规划。第 9 章至第 16 章主要是对前面知识的灵活运用，包括复杂查询技术、视图、存储过程、触发器、自定义函数及事务和锁等。附录部分包括 SQL Server 2000 中用到的重点词汇，一些常用的命令及实例。

本书内容翔实，结构合理，示例丰富，图文并茂，语言流畅。是微软认证高级技术培训中心的培训教材，还可作为各种数据库培训班的教材、大中专院校的数据库课程教材和数据库管理和开发人员的参考书。

需要本书或需要得到技术支持的读者，请与北京市海淀区知春路甲 63 号卫星大厦三层北京希望电子出版社（邮编 100080）联系，电话：010-62630301，82675588（总机），传真：010-62520573，E-mail: luandc@bhp.com.cn。

系 列 书： 微软认证高级技术培训教材系列

书 名： SQL Server 2000 数据库编程

文 本 著 者： 梁方明

责 任 编 辑： 范二朋

出版、发行者： 北京希望电子出版社

地 址： 北京市海淀区知春路甲 63 号卫星大厦三层 100080。网址: www.bhp.com.cn,
E-mail: luandc@bhp.com.cn。电 话： 010-62630301, 62520290, 62521724,
62528991, 62524940, 62521921, 82610344（发行） 010-82675588-501,
82675588-201（编辑部）

经 销： 各地新华书店、软件连锁店

排 版： 希望图书输出中心 张月岭

文 本 印 刷 者： 北京双青印刷厂

规格 / 开本： 787 毫米×1092 毫米 1/16 26.375 印张 615 千字

版次 / 印次： 2002 年 11 月第 1 版 2002 年 11 月第 1 次印刷

印 数： 0001~5000 册

本 版 号： ISBN 7-900118-72-1

定 价： 49.00 元

说明：凡我社产品如有残缺，可执相关凭证与本社调换。

出版说明

为了配合微软高级技术培训的教学与考试,进一步推广微软认证系统工程师(MCSE)、微软认证数据库管理员(MCDBA)、微软认证系统管理员(MCSA)和微软认证产品专家(MCP)的培训和考试,特组织优秀的微软认证讲师(MCT)编写了本套微软认证高级技术培训教材。

全套教材共 7 本:

序号	书 名	对应考试号
1	网络与操作系统基础	无
2	实现 MS Windows 2000 Professional 和 Server	70-210, 70-215
3	实现 MS Windows 2000 网络基础结构	70-216
4	MS Windows 2000 目录服务基础结构设计和管理	70-217, 70-219
5	MS SQL Server 2000 数据库管理	70-228
6	MS SQL Server 2000 数据库编程	70-229
7	管理 MS Windows 2000 环境	70-218

本套培训教材都由第一线的微软认证高级技术培训中心讲师(MCT)编写,凝聚了 MCT 们多年教学经验,符合中国人阅读习惯,教材的每章都有学习重点,在必要章节附有实验,供学员练习。本套教材还包括大量的模拟试题,所有模拟试题都加入了试题分析和知识点解析以适应不同考生考证使用。力求通过学习本套教材,即可通过 MCSE, MCDBA, MCSA 或 MCP 的考试。

学员通过学习课程 1, 2, 3, 4, 5, 6, 通过对应的考试,可以获得 MCSE 和 MCDBA 两种证书;通过学习课程 1, 2, 3 或 4, 7, 通过对应的考试,可获得 MCSA 证书;学习任何一门课程,通过对应的考试,可获得 MCP 证书。

本套教材既可作为微软认证高级技术培训教材,微软高级技术认证的自学教材,也可供广大网络、数据库技术人员和爱好者学习、参考使用。

编 者

前 言

关系型数据库管理系统(RDBMS)是管理关系型数据的系统软件,对数据的一切操作,包括定义、查询、更新以及其他各种控制,都是通过 RDBMS 进行的。当今 RDBMS 软件领域可谓群雄争霸,一大批优秀的 RDBMS 软件,正在各个领域起着十分重要的作用,Microsoft SQL Server 是其中的代表作之一,它几乎垄断了整个中型数据库管理系统领域,并且从 SQL Server 2000 开始微软的这套系统软件正在逐步蚕食大型数据库管理系统领域的市场份额。

以下几个原因说明了 Microsoft SQL Server 是众多领域中最终用户和程序员用于架构商业应用程序模式的最佳选择。

首先,因为其 Windows9x、WindowsNT/2000 的无缝连结,在 Windows 操作系统上安装的数据库管理系统 SQL Server 无疑是无人能比的,而随着 Windows9x、WindowsNT/2000 操作系统安装数量的快速增长,Microsoft SQL Server 被推向更加重要的地位是必然的。

其次,其价格相对十分低廉,同等性能、实现同等功能的情况下,Microsoft SQL Server 几乎只是其同类产品价格的 1/3。

第三,Microsoft SQL Server 是十分易于使用的数据库管理系统,它有极其友好的用户界面。做界面是微软的强项,这在业界是人所共知的,SQL Server 仍然没让大家失望。而且 Microsoft 还提供了多种不同的工具用以帮助用户创建数据库对象,调试数据库程序,以求快速完成系统的管理任务。例如:SQL Server 2000 中包含了几十种向导,几乎覆盖了所有操作(数据库的创建、维护、管理、备份、恢复、数据复制等等许多工作)。

SQL Server 2000 是 SQL Server 家族中的最新版本,Microsoft SQL Server 2000 扩展了其以前版本的性能、可靠性、质量和易用性。Microsoft SQL Server 2000 增加了几种新的功能,由此成为大规模联机事务处理(OLTP)和电子商务应用程序的优秀数据库平台。它也增加了对 XML 的支持,同时增强了数据仓库的功能。

本书从第 1 章到第 8 章介绍了 SQL Server 2000 的一些基础知识。从第 9 章开始讲述 SQL Server 2000 的一些实用知识。其后的每个章节都是互相独立的,可以任意顺序阅读。这里没有详细罗列所有的技术,而只是集中讲述了在应用中会经常用到的知识。另外对于 XML 和数据仓库的功能限于篇幅也不做讲解,如有需要请查阅相关书籍。

本书建议的学习时间为 70 学时。在学习过程中,首先通读全文,然后做课后的实验。实验的内容非常简单,所以在做的过程中应该多问自己为什么这么做。从第 2 章开始每章课后都有模拟试题,这一部分主要是为希望通过 Microsoft 认证考试的同学设计的,有一定的难度,如果每道题都弄懂的话,通过认证应该不成问题。

感谢李孟召、李辉、刘亮、贺来平等帮助审校,同时感谢汤梦希、陈杰磊、俞江等帮助整理资料 and 提出宝贵意见。由于时间仓促及编者水平有限,疏漏之处欢迎读者批评指正。如想进一步了解本书信息,请访问我的主页:<http://www16.brinkster.com/sqlserver1fm/>。

目 录

✓ 第 1 章 数据库概念简述	1	4.6 模拟试题分析	105
1.1 数据库术语	1	第 5 章 表的规划和创建	110
1.2 正规化的概念	3	5.1 表的规划	110
1.3 什么是 SQL	10	5.2 表的创建	112
1.4 什么是 C/S 结构	12	5.3 表的修改	117
1.5 SQL Server 应该解决的问题	13	5.4 小结	122
1.6 小结	14	5.5 实验	122
1.7 实验	14	5.6 模拟试题分析	124
1.8 复习题	15	✓ 第 6 章 实现数据的完整性	131
✓ 第 2 章 SQL Server 2000 概述	17	6.1 实体完整性 (Entity Integrity)	131
2.1 什么是 SQL Server 2000	17	6.2 域完整性 (Domain Integrity)	133
2.2 SQL Server 2000 构成概述	20	6.3 参照完整性 (Referential Integrity) ..	137
2.3 引用 SQL Server 数据库对象	23	6.4 用户定义的完整性	
2.4 SQL Server 的安全策略	24	(User-defined Integrity)	142
2.5 小结	29	6.5 使用约束时应该注意的问题	142
2.6 实验	30	6.6 小结	144
2.7 模拟试题分析	32	6.7 实验	144
第 3 章 用 Transact-SQL 编程	38	6.8 模拟试题分析	149
3.1 什么是 Transact-SQL	39	✓ 第 7 章 规划、创建并维护索引	154
3.2 Transact-SQL 语法规则	41	7.1 索引概述	155
3.3 如何执行一个 Transact-SQL 语句	43	7.2 SQL Server 索引的体系结构	156
3.4 简单查询	49	7.3 索引的选择技术	161
3.5 数据类型	54	7.4 生成索引	162
3.6 统配符和运算符	57	7.5 如何维护一个索引	168
3.7 变量	60	7.6 决定哪一列使用索引	172
3.8 函数	62	7.7 索引优化建议	172
3.9 数据的汇总	68	7.8 小结	173
3.10 流程控制语句	72	7.9 实验	174
3.11 小结	78	7.10 模拟试题分析	176
3.12 实验	78	第 8 章 全文索引查询	181
3.13 模拟试题分析	82	8.1 全文索引和查询概念	181
✓ 第 4 章 数据库的创建和管理	87	8.2 创建全文检索目录	182
4.1 SQL Server 数据库的基础知识	87	8.3 全文索引查询	186
4.2 创建数据库	91	8.4 小结	189
4.3 数据库的修改	95	8.5 实验	189
4.4 小结	103	8.6 模拟试题分析	191
4.5 实验	103	第 9 章 复杂查询技术	192

9.1	多表查询	192
9.2	子查询	202
9.3	应该使用连接还是子查询	208
9.4	存储查询结果	209
9.5	小结	210
9.6	实验	210
9.7	模拟试题分析	215
第 10 章	实现视图	218
10.1	视图简介	218
10.2	创建和修改视图	220
10.3	通过视图更改数据	228
10.4	索引视图	230
10.5	使用信息架构视图	232
10.6	调用链	234
10.7	小结	236
10.8	实验	237
10.9	模拟试题分析	240
第 11 章	实现存储过程	244
11.1	存储过程简介	244
11.2	创建存储过程	247
11.3	管理存储过程	254
11.4	系统和扩展存储过程	260
11.5	处理错误信息	263
11.6	小结	265
11.7	实验	265
11.8	模拟试题分析	268
第 12 章	实现触发器	277
12.1	触发器简介	277
12.2	创建触发器	280
12.3	修改和删除触发器	284
12.4	触发器的工作原理	285
12.5	触发器实例	292
12.6	使用触发器的限制和推荐实 现的方案	295
12.7	小结	296
12.8	实验	296
12.9	模拟试题分析	301

第 13 章	实现用户自定义函数	307
13.1	创建用户定义函数	307
13.2	用户定义函数调用	314
13.3	修改、查看和删除用户 自定义函数	315
13.4	推荐使用方案	317
13.5	小结	318
13.6	实验	318
13.7	模拟试题分析	321
第 14 章	管理事务和锁	323
14.1	事务和加锁	323
14.2	管理事务	325
14.3	SQL Server 加锁	332
14.4	推荐的实现方案	341
14.5	小结	341
14.6	实验	341
14.7	模拟试题分析	343
第 15 章	优化查询性能	351
15.1	查询优化器简介	351
15.2	获得执行计划的信息	354
15.3	索引提示	362
15.4	优化查询时要注意的问题	363
15.5	高级查询优化	364
15.6	查询优化建议	365
15.7	小结	366
15.8	实验	366
15.9	模拟试题分析	367
第 16 章	分布式查询	373
16.1	什么是分布式查询	373
16.2	建立链接服务器环境	376
16.3	在链接服务器上执行查询	381
16.4	小结	382
16.5	实验	382
16.6	模拟试题分析	382
附录 A	重点词汇	384
附录 B	一些常用的命令	404
附录 C	一个简单的案例	411

第 1 章 数据库概念简述

- 数据库术语
- 正规化的概念
- 什么是 SQL
- 什么是 C/S 结构
- SQL Server 应该解决的问题

SQL Server 2000 是微软最新推出的一款数据库管理系统。SQL Server 2000 以前的版本以其低廉的价格, 优质的性能, 几乎垄断了中低档数据库管理系统的市场。SQL Server 2000 的推出瞄准了高端, 在性能方面比以前的版本有较大的提高, 很多方面可以和大型数据库管理系统媲美, 而在易用性和价格上是无人能敌的。本书是面向微软认证考试的, 但无论你是准备应考的人员, 还是有经验的开发人员, 在本书中都会找到你感兴趣的东西。

学习目标

- **数据库术语** 学习任何一门知识必须首先要了解它的相关术语, 不然根本无从下手。如果你是数据库新手, 应该详细阅读本节。
- **正规化的概念** 创建一个好的数据库结构对一个数据库开发人员来说是十分重要的。正规化是一个与数据库结构相关的知识。正规化的概念就是通过几个范式将数据库中的数据分解到各个表中去。
- **什么是 SQL** SQL 即结构化查询语言 (Structured Query Language)。本小节中简要地叙述了 SQL 语言的发展历史、SQL 语言的组成与结构。
- **什么是 C/S 结构** C/S 结构是当前最为流行的一种体系结构, 就是将一个应用的处理分前端和后端两个部分, 前端为用户接口 (User Interface), 后端为服务处理。SQL Server 是一个典型的 C/S 结构的应用程序。
- **SQL Server 应该解决的问题** 这里所列的问题不仅是 SQL Server 应该解决的, 其他的大型数据库管理系统也都应该解决。这些问题是: 数据冗余和不一致、数据的访问困难、数据完整性问题、并发性问题和安全性问题。

1.1 数据库术语

介绍 SQL Server 之前, 首先介绍一些数据库的基本概念。也许你已经熟悉本章的内容, 但不妨浏览一遍, 把你琐碎的记忆, 系统地串成线。如果你从没接触过数据库概念, 这里所讲述的是数据库入门的必备知识。

从 20 世纪 50 年代中期开始, 计算机的应用由科学研究部门逐步扩展到企业、行政部门。至 60 年代, 数据处理成为计算机的主要任务。数据库技术作为数据管理技术, 是计算机软件领域的一个重要分支, 产生于 60 年代末。现在已经形成相当规模的理论体系和一整套的技术应用。

计算机的数据处理应用，首先要把大量的信息以数据形式存放在存储器中，而如何有效的利用和管理这些数据便是数据库管理系统（DBMS）的任务了。而本书就是讲解一种数据库管理系统（DBMS）的用法的。现在我们先介绍一下与 DBMS 相关的一些术语。

数据库（Database）

Database 是统一管理的相关数据的集合。Database 能为各种用户共享，具有最小的冗余，数据间联系密切，而又有较高的数据独立性。

数据库管理系统（DBMS）

DBMS（Database Management System）可以这样理解：管理数据库的系统软件。实际上 DBMS 是用户与操作系统之间的一层管理软件，为用户或应用程序提供访问 Database 的方法，包括 Database 的创建、维护、管理、备份、恢复、数据复制等等许多工作。

关系型数据库管理系统（RDBMS）

可以把关系理解成二维表。可以这样讲，一个关系就是一张二维表格，而关系型数据库管理系统所管理的就是这些二维表格的集合。也就是说，RDBMS 就是管理二维表的 DBMS。与早期的数据库系统相比（网络数据库和层次数据库），关系数据库系统是基于浓厚数学背景的关系数据模型。所以关系型数据库管理系统是现在最为流行的数据库系统。本书中主要讲解的就是 SQL Server 如何实现关系数据库管理的功能。

在这里我们阐述几个与关系型数据库管理系统有关的概念。

- **实体(Entity)** 客观存在可以相互区别的事物称为实体。实体可以是具体的对象，如：桌子，电脑等具体的东西；也可以是抽象的事件，如：一次编程比赛，修理电脑等抽象的事情。
- **属性(Attribute)** 实体有很多的特性，每一个特性称为属性。每一个属性有一个值域，其类型可以是整数型，字符型，日期型。例如：年龄，姓名，生日等。而属性在二维表中又叫字段，也就是二维表中的列，如图 1-1 所示。
- **记录(Record)** 字段的有序集合称为记录。一般用记录描述一个实体，所以记录又可以定义为能完整地描述一个实体的字段集。例如：一个职工记录，由有序的字段的组成：（员工姓名，员工编号，性别，年龄）。在二维表格中也就是行。

EmployeeID	LastName	FirstName	Title	TitleOfCourtesy
1	Davolio	Nancy	Sales Representative	Ms.
2	Fuller	Andrew	Vice President, Sales	Dr.
3	Leverling	Janet	Sales Representative	Ms.
4	Peacock	Margaret	Sales Representative	Mrs.
5	Buchanan	Steven	Sales Manager	Mr.
6	Suyama	Michael	Sales Representative	Mr.
7	King	Robert	Sales Representative	Mr.
8	Callahan	Laura	Inside Sales Coordinator	Ms.

图 1-1 查询结果

1.2 正规化的概念

对于一个数据库软件的设计来说,除了要设计一个易用的界面,还要设计数据库中存放数据的逻辑结构,这一步需要在界面设计之前完成,一个数据库设计成败的关键就在于数据结构设计的是否合理。

设计优化数据组织的过程称为正规化(Normalization),其实说白了就是将数据库中的数据分解到各个表中去。除非没办法分了,否则就应该把数据分解开来,一会儿来讲解为什么要这样做。正规化的过程要遵循几个范式,正规化能够帮助你得到一个更为有效的存储结构。

1.2.1 和正规化有关的一些概念

数据库的功能在于将数据整理得有条不紊,随着不同设计者规划的差异,创建的数据库结构可能并不相同。如何能使整理出来的数据表尽量不发生数据重复的情况,对于数据库大小及访问效率所造成的影响是相当大的。为了达到这个目的,通常必须将数据分成独立的表,这种数据分离的技巧,称为正规化程序。

正规化实际上就是为了保持数据的完整性。无论对数据进行什么操作,都应该避免插入和生成无效的操作。

下面先介绍一些与正规化有关的概念:

实体完整性

刚才讲过了,实体就是现实中的一个对象或是一个概念。数据库中存放的正是实体的信息。实体可以是物体(桌子),也可以是一件抽象的事件,如老师批改作业。将数据组织到数据库中的步骤之一就是确定数据库涉及的实体。例如:

- ✎ **对象** 如公司销售的产品。
- ✎ **主体** 如公司涉及的客户或厂家。
- ✎ **事件** 如向客户销售产品。

例如,假设开发一个数据库,是为了描述一个公司员工信息,则数据库包含下列实体:

- ✎ 各个员工
- ✎ 员工与库房的关系

确定实体是非常重要的,拿到一个数据库项目,首先就要确定实体是什么。这是需要一些经验的,只有正确地确定实体才能开始工作,这一步花些时间也是十分必要的,这有点儿像软件工程中的需求分析。

数据库中存放的每个实体都要有主关键字。什么是主关键字呢?就是该实体区别于其他同类实体的惟一特性,也就是用来保证数据库中的记录的惟一性。例如:员工表中,可以选择员工号作为主关键字来惟一确定每一个员工。

如果能用表中的一个字段也就是一列来惟一标识记录,则称为简单主关键字。如果要多列才能惟一标识记录,则称为复合主关键字。例如,员工表中:

- ✎ 乔丹,男,2000-5 进入公司,编号 001

- 奥尼尔, 男, 2001-2 进入公司, 编号 002
- 乔丹, 女, 2001-7 进入公司, 编号 003
- 加内特, 男, 2002-3 进入公司, 编号 004

这里可以选择编号作为主关键字, 也可以选择名字和性别组成复合主关键字。但是用哪个更好呢, 当然用编号肯定是没有问题的了。但是你可能认为编号实在是很难记忆, 这时就可以选择名字和性别组成复合主关键字。但是要有个前提, 就是名字和性别组成复合主关键字必须在表中是惟一的, 只有这样才能构成复合主关键字。

为什么要对每个记录标识惟一主关键字呢?

因为数据库服务器用主关键字作为取得所要信息的关键句柄。通过标识主关键字, 可以告诉数据库服务器要取哪条记录的信息。如果主关键字不惟一, 则数据库不知道要返回哪条记录。主关键字是数据库实现数据完整性的主要机制, 是检索的基础。数据库中的每张表最好都要设置主关键字。

域完整性

实体完整性的目的是为了便于读取数据库中存放的信息。而域完整性则是限制数据库中可以存放的信息。也可以把域看成一组业务规则, 控制表中每一列的允许值。这里有点像数学中的值域的概念。

最简单的例子, 由列指定的数据类型来实现域完整性, 如: 不可以在定义为日期型的数据的字段中输入中文。对字段可输入的内容限制越多, 数据库中输入不良的数据的几率就越小。比方说, 上面提到的员工表中, 你就应该为员工进入公司的日期设置值域, 这个日期必须不能是今天后的某一天。利用这些常识就可以大大的降低数据录入员错误输入的可能性。所以在设计表时一定要根据用户的业务规则, 尽可能的限制数据的值域。

SQL Server 2000 中提供了各种实现域完整性的工具, 如:

- 数据类型
- 用户定义数据类型
- 默认限制
- 约束
- 规则
- 外键

在第 6 章将详细介绍这些工具。

参照完整性

举个例子来说明参照完整性, 见表 1-1 和表 1-2。

表 1-1 员工表

员工编号	姓名	性别	进入公司时间
001	乔丹	男	2000-5
002	奥尼尔	男	2001-2
003	邓肯	男	2001-7

004	加内特	男	2002-3
...	...	...	...

表 1-2 仓库借物表

借物员工编号	所借物品	借物时间	所借数量
001	移动硬盘	2001-1	1
003	万用表	2001-2	1
001	万用表	2002-2	1
002	笔记本电脑	2002-3	1
...	...	...	...

此两张表格中的“员工编号”和“借物员工编号”是有联系的，仓库借物表中的“借物员工编号”人员必须在员工表中存在。而如果公司的某个员工要辞职走人的时候想把他从员工表中删除，则其必须在仓库借物表中没有记录。所以这两张表是按照共享列（员工编号或借物员工编号）相互关联。

参照完整性是保证表与表之间语意上的完整性，既当一个表引用在另一个表中定义的实体时，要保证这个实体的有效性。

参照完整性的目的就是保证一对表中的相关行即使在数据发生改变时也保持相互联系。数据库执行参照完整性时，能防止用户错误的操作部分数据。上例中为了保持员工表和仓库借物表之间的参照完整性，数据库需要限制几个操作：

- ✎ 用户不能在员工表中还不存在员工时在仓库借物表中插入数据。这是显然的，你还没进入公司，怎么能让你借东西呢。
- ✎ 用户不能把仓库借物表还有记录的员工从员工表中删除。就是说，如果辞职人借的东西还没有归还给仓库的话，他不能办理离职手续。
- ✎ 用户不能对员工表中的员工编号内容随便改动。如果数据库允许用户破坏这个规则，则仓库借物表中可能出现孤立数据，进而不能引用员工表中数据。

这些规则不是随意的，基本思想是无论对数据进行什么操作，都要对员工表和仓库借物表中的相应行进行匹配。参照完整性要求数据库中要执行不可变的表间关系。

SQL Server 2000 中提供了几种保证参照完整性的工具：

- ✎ 外键约束
- ✎ 约束限制
- ✎ 触发器
- ✎ 存储过程

这些工具在以后的章节中叙述。

用户自定义完整性

上述 3 种完整性是正规的数据库概念，这些完整性在每个数据库中都有，可能有些实用的工具与 SQL Server 不同，但是对于这些完整性肯定是要支持的。

用户自定义完整性则是执行不适于上述概念的一些特殊的商业规则。例如：在图书馆的借阅系统中，如果客户想要的书已经被借光了，系统要推荐一些相关的书。这个案例就无法用上述的完整性规则表示，我们只能利用存储过程、触发器等工具自己来设计这种规则。这就是一个用户自定义完整性的过程。

实际上用户自定义完整性就是针对某一具体数据的约束条件，是由应用环境决定的。用户定义的完整性规则反映某一具体应用涉及的数据必须满足的语义要求。系统提供定义和检验这类完整性的机制，以便以统一的方法处理他们，不再由应用程序承担这项工作。我们再举一个简单的例子：如果你正在给某学校做一个管理系统，学生的年龄不会太大也不会太小，你就可以自己来写一条规则，把年龄限制在 15~30 岁之间。

对于用户自定义完整性，各种数据库的支持程度是各不相同的。SQL Server 比较全面的支持用户自定义完整性，让你可以充分的自由发挥。

1.2.2 第一范式

在我们了解了各种不同类型的数据库完整性之后，我们现在开始学习范式。范式其实就是组织数据的规则，主要用于数据库设计的数据规范化过程中。不同的范式用编号表示，如第一范式、第二范式等等，理论上，至少有 5 种不同的范式，其中前三种在实际应用中最为重要。后一种范式建立在前一种范式的基础上的，就是说如果想要符合第二范式就必须符合第一范式。因而，如果你设计一个数据库中的所有表格都是符合第三范式的，那么就可以说你设计了一个良好的数据库。

第一范式的规则很简单：指一张表中没有多值属性或是复合属性（复合属性中包含了其他的属性，因而可被分为多个小部分），每个数据只在一个地方保存。所有的关系表格都是用第一范式定义的，任何一列的数值必须是原子性的，也就是说每个字段都是不可分割的。比如表 1-3 所示的学生选课表。

表 1-3 学生选课表

学号	姓名	性别	授课教师	学分
001	王文	男	马天才，操作系统	5
003	李丽	男	李立功，数学	5
...	...	...	...	...

表 1-3 所示的学生选课表就是一个非正规化的数据，它不符合第一范式，因为这张表中授课教师这一列有多值属性，在设计表时我们应该绝对避免这种情况的发生。如果已经确定我们的数据表符合第一范式，我们就应该为这张表标识主关键字了。

标识主关键字

这是检查完数据表是否符合第一范式后，要做的第一件事。如表 1-3 可以选学号作为主关键字，因为它可以使各行惟一，也可选姓名加性别组成复合主关键字。这要看你的需要了。选择关键字是一门艺术，需要知道如何选择主关键字和如何选择最佳主关键字。

在选择最佳主关键字时，应该考虑下列因素：

✎ **稳定性** 如果列中的值可能变化，则不适宜作为最佳主关键字。因为表格相互关

联时，后面要通过查阅主关键字值跟踪这个关系，比如：表中客户的电话就不适合作为主关键字，因为客户的电话随时都可能变化。

- ✎ **最简性** 主关键字的列越少越好，如非必要最好一列，因为多列会使涉及主关键字的操作变慢，其实很多情况下增加一列并不能增加关键字的独特性，所以应该少用复合主关键字。
- ✎ **熟悉性** 如果数据库用户熟悉实体的特定标识，则适合用其做主关键字。因为用户在做查询时喜欢用自己熟悉的字段作为条件，如果你设计数据库时使用用户熟悉的字段作为主关键字，有利于提高用户的查询速度。

人为关键字的使用

有时表中的自然数据不太适合作为最佳主关键字。例如，一个表中包含姓名，性别，电话，家庭住址。这些字段都不是很适合于作为主关键字，这种情况下，应该对表格生成人为主关键字。人为主关键字原先不属于表中的数据，但是为了惟一标识数据行，而人为的引入一列，比方说用户编号。这种情况在现实中有很多的应用。

1.2.3 第二范式

为了实现第二范式，要保证表格符合第一范式。

第二范式的定义是：在保证符合第一范式的基础上，表中必须能标识主关键字，所有的非主关键字段，只依赖于这个主关键字，而不依赖表中其他字段。

如果选主关键字应该是选学生编号，但是课程字段不仅依赖于学号字段还要依赖于授课教师字段。就是说不同的人可能选不同的课，这时可以说课程是依赖于学号的，但是课程也是随着教师的不同而不同，所以说课程同时依赖于授课教师。所以图 1-2 的例子是不符合第二范式的，存在问题如下。

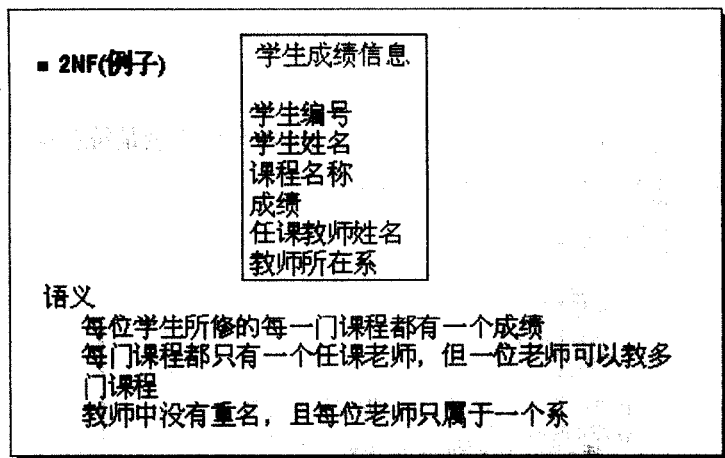


图 1-2 2NF 例子

✎ 数据冗余太大

一门课程的教师名必须在选这门的所有学生中重复一次；一个系名必须对选该系所开课程的所有学生重复一次。

更新异常问题 (Update anomalies)

- **修改异常 (Modification Anomaly)** 由于冗余, 在修改时会导致数据的不一致。如改变一门课的老师, 因为数据表中存在冗余, 所以改变一门课的老师时, 要在表中更改多处, 所以可能造成修改异常。
- **插入异常 (Insert Anomaly)** 由于主键不能为空, 如某系有位老师不教课; 或者所开的课暂时无人选, 则该位老师和所在的系名无法插入。
- **删除异常 (Deletion Anomaly)** 如果学生都退选一门课, 则有关这门课的其他数据 (教师, 开课系) 都将被删除。

因此上面的关系应该进一步分解, 形成两个实体, 如图 1-3 所示。

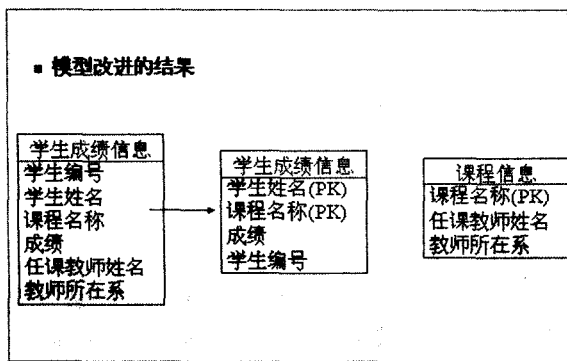


图 1-3 模型改进结果

外部关键字与关系：将表分解为两个表后, 就要知道这些表如何组合——重新生成原数据。这时可以通过匹配两张表中的课程名称列而实现。课程信息表中的课程名称是主关键字, 而学生成绩信息表中的课程编号称为外部关键字。通过标识主关键字及相应的外部关键字, 可以告诉数据库服务器两个表要维护的参照完整性。

1.2.4 第三范式

第三范式的定义：关系表如果是符合第三范式, 当且仅当表是符合第二范式, 并且所有非主关键字列都是非传递地依赖于主关键字。

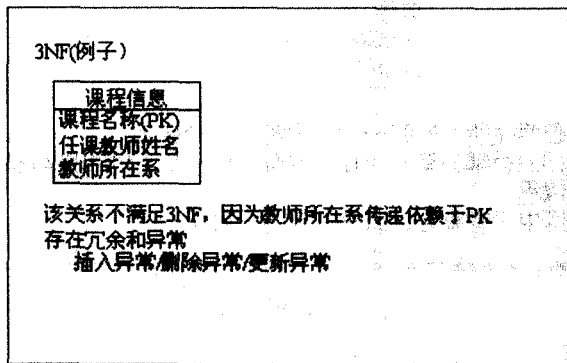


图 1-4 3NF 例子

此例中表面上看来“教师所在的系”似乎也是随着“课程名称”的变化而变化, 是依

赖于“课程名称”的。但实际上教师所在的系是通过“任课教师姓名”传递依赖于“课程名称”的，所以它不符合第三范式，需要分解，结果如图 1-5 所示。

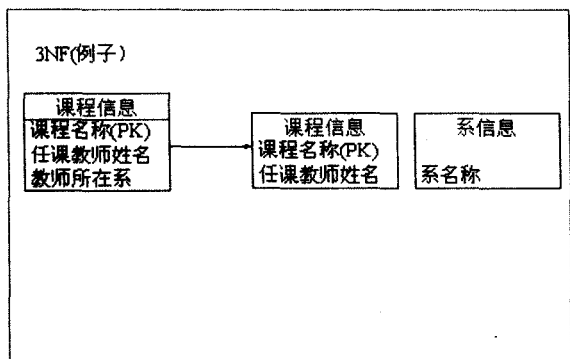


图 1-5 模型改进结果

注意：第三范式不是正规化的终结。数据库研究人员还提出了其他范式，如第四、第五范式。

但是，对大多数日常数据库而言，数据库采用第三范式就够了。实际上，对于比较简单的数据库而言，如果按照第三范式规范化，其可能就已经符合第五范式了，但是如果数据库很复杂的话，需要用到如第四、第五范式，则应该请教正规化专家来帮忙了。

1.2.5 非规范化

正规化过程将数据库中的数据尽量减少冗余，而非规范化则是在数据库中故意引用一些冗余数据。当然，理论上说数据不能非规范化。但是实际情况并不是这么简单的，有时可能需要将数据库非规范化以提高性能。过分正规化的数据库由于把数据分成了很多的表，当取数据需要多个联结时，可能会使系统性能下降。这就要根据具体情况，权衡利弊来做出决策了。

注意：需要将数据非规范化以提高性能时，一定要记录这个决策，使别的开发人员不会认为是你的错误，这一点非常重要。

1.2.6 SQL Server 中的正规化工具

SQL Server 中有几个工具可以帮助我们使数据库符合范式。这些工具保证表中只插入合法数据，只能进行合理地改变。任何时候可以直接对服务器执行正规化时，而不必编写应用程序代码执行正规化。这对大多数数据库都是大有好处的。

SQL Server 中有以下工具用于正规化过程：

- 标识列 (Identity Columns)
- 约束 (Constraint)
- 规则 (Rules)
- 声明参照完整性 (Declarative referential integrity)
- 触发器 (Triggers)
- 数据库框图 (Database Diagrams)

这些工具将在本书以后的章节中详细介绍。

1.3 什么是 SQL

1.3.1 SQL 语言的发展历程

1970 年, 美国 IBM 研究中心的 E.F.Codd 连续发表多篇论文, 提出关系数据库模型。1972 年, IBM 公司开始研制试验型关系数据库管理系统 System R, 查询语言称为 SQUARE (Specifying Queries As Relational Expression) 语言, 在语言中使用了较多的数学符号。1974 年, Boyce 和 Chamberlin 把 SQUARE 语言修改为 SEQUEL (Structured English Query Language) 语言。这两个语言基本上是相同的, 但 SEQUEL 去掉了一些数学符号, 并采用英语单词表示, 使用结构式的语言规则, 看起来很像英语句子, 用户比较欢迎这种形式的语言。后来 SEQUEL 简称 SQL (Structured Query Language), 即“结构化查询语言”。SQL 虽然名为查询语言, 但实际上具有定义、查询、更新和控制等多种功能。你可以把 SQL 读作“sequel”, 也可以按单个字母的读音读作 S-Q-L。两种发音都是正确的, 每种发音各有大量的支持者。在本书里, 认为 SQL 读作“sequel”。

SQL 语言对关系模型的发展和商用 DBMS 的研究起着重要的作用, 是关系数据库的标准语言。由于 SQL 使用方便、功能丰富、语言简洁易学, 很快就得到非常广泛的应用。

在 System R 获得成功之后, 许多新公司选择了 SQL 来构建他们自己的关系型数据库系统。所有的公司都是在 SQL 的原有基础上作了新的扩充, 而这些扩充都是这一语言的分支。因此, 美国国家标准协会 (ANSI) 和国际标准化组织 (ISO) 在 1982 年成立了一个委员会, 目标是开发一套标准版本的 SQL。SQL 的第一个标准颁布于 1986 年, 主要是基于 IBM 公司的那个语言的分支。1989 年颁布了一个中间标准后, 一个更为庞大的标准 SQL92 (又叫 SQL2) 被开发并于 1992 年 12 月公布于众。在最近几年里, 两个标准化组织又共同开发了一个新的标准——SQL3, 它包含了几个关于数据库的最新概念, 例如存储过程、触发器、及众多面向对象的概念。SQL3 标准的主要部分于 1999 年 9 月颁布 (因而 SQL3 也称为 SQL99)。大家可以看出从 1986 年至今, 也不过才出现了三个版本, 而且这三个版本之间的变化也不是很大, 这样大大地节约了用户的学习成本。所以就有了学会了 SQL 可以终身受用的说法。

与传统的编程语言例如 C, C++ 及 Basic 不同, SQL 是面向集合的语言 (而前者们是面向记录的语言或叫做只处理记录的语言)。也就是说, SQL 用一个语句就可以从一个或几个表中查询多行数据。而且 SQL 更加接近于自然语言, 是比 C、C++、Basic 更为高级的语言, 十分地易学易用, 举个简单的例子: `SELECT title FROM northwind..employees`, 这段 SQL 语句的意思是显示 Northwind 数据库中的 Employees 表 Title 字段的内容。我们通过英文的字面上也差不多明白这段 SQL 语句的意思了。

SQL 的另一个重要特性是它的非过程特性。任何一门过程化语言 (C, C++, Basic) 要描述的是应该如何一步一步的去完成一项任务。而 SQL 作为一种非过程化的语言, 描述的是用户需要的是什么。因而, 系统就要负担起寻求合适的方法来解决用户需要的任务。其实说白了就是你用 SQL 语言和系统对话, 让系统干什么, 至于怎么干就是系统的事了,