

◎ 微软MVP、51CTO专家讲师**韩立刚**老师呕心著作



跟韩老师学

SQL Server 数据库设计与开发

主 编 韩立刚
副主编 马龙帅 王艳华 韩利辉

韩老师教学支持网站：
<http://www.91xueit.com>

韩老师视频课程链接：
http://edu.51cto.com/lecturer/user_id-400469.html

韩老师微信公众号：
han_91xueit

韩老师QQ：
458717185



技术支持

- ◇ 范式、多表联接、联接查询、存储过程、索引、触发、事务、锁……这些核心概念你真的懂吗？
- ◇ 用最好懂的方式水煮复杂概念，所有似懂非懂一剑封喉！
- ◇ 超10年企业实战经验与多年大学授课经验完美结合！
- ◇ 课本知识与企业需求无缝衔接，项目教学+案例教学+校企合作完美实践！

跟韩老师学 SQL Server 数据库 设计与开发

主 编 韩立刚

副主编 马龙帅 王艳华 韩利辉



中国水利水电出版社
www.waterpub.com.cn

·北京·

内 容 提 要

在软件设计和开发过程中,需求分析确定后,在开发软件前就需要先设计数据库,然后再编写代码。本书讲解的内容是软件开发人员必须要掌握的技术,先讲解如何根据项目的需求按关系型数据库设计规范设计数据库中的表,然后使用自己设计的数据库来学习 SQL 语句。

本书完整地讲述了 SQL 语句中使用的变量、数据类型、注释行、运算符、常用函数,以及流程控制语句。从单表查询讲起,讲解使用 WHERE 筛选行将查询的结果分组、排序,然后讲解多表查询和数据汇总,讲解多表交叉联结、内联结、外联结、自联结,使用 UNION 合并查询结果集,使用聚合函数汇总查询结果,接着讲解了子查询、开窗函数和行列转换技术,实现数据完整性,创建和使用索引优化性能,在数据库中创建和使用视图简化查询,使用存储过程操作数据库,创建用户自定义函数,在 SQL 语句中使用自定义函数,使用触发器实现复杂的业务逻辑,使用事务实现数据的完整性和一致性。

图书在版编目(CIP)数据

跟韩老师学SQL Server数据库设计与开发 / 韩立刚
主编. — 北京:中国水利水电出版社, 2017.2
ISBN 978-7-5170-5182-4

I. ①跟… II. ①韩… III. ①关系数据库系统 IV.
①TP311.138

中国版本图书馆CIP数据核字(2017)第027436号

责任编辑:周春元 张玉玲

封面设计:李 佳

书 名	跟韩老师学 SQL Server 数据库设计与开发 GEN HANLAOSHI XUE SQL Server SHUJUKU SHEJI YU KAIFA
作 者	主 编 韩立刚 副主编 马龙帅 王艳华 韩利辉
出版发行	中国水利水电出版社 (北京市海淀区玉渊潭南路1号D座 100038) 网址: www.waterpub.com.cn E-mail: mchannel@263.net (万水) sales@waterpub.com.cn
经 售	电话: (010) 68367658 (营销中心)、82562819 (万水) 全国各地新华书店和相关出版物销售网点
排 版	北京万水电子信息有限公司
印 刷	北京瑞斯通印务发展有限公司
规 格	184mm×240mm 16开本 18印张 398千字
版 次	2017年2月第1版 2017年2月第1次印刷
印 数	0001—3000册
定 价	58.00元

凡购买我社图书,如有缺页、倒页、脱页的,本社营销中心负责调换
版权所有·侵权必究

前 言

我最初学习数据库是 2001 年,感觉学习 SQL 语句很难,而且学完后时间稍微一长就忘了。现在回想起来,当时学习的难点非常关键的一点就是使用微软 SQL Server 2000 的示例数据库来学习,作为初学者,根本不知道 Northwind 数据库是用来做什么的,更不知道其中各个表之间的关系,也不知道为什么要把数据放到多张表,查询时再连接成一张表。使用陌生的数据库学习新知识自然带来不少困难,不理解数据库为什么要设计出多张表来存放数据以及各表之间的关系也就弄不清如何写多表联合查询的条件。

独立为几家企业开发几款软件后熟悉了整个软件开发流程,从客户的需求分析到数据库设计,再到软件开发、测试部署和维护。在此期间使用自己设计的数据库编写 SQL 语句真是得心应手,也因此找到了学习 SQL 的正确方法,即先学习数据库设计,再学习对这些表增删改查的 SQL 语句。

本书使用微软 SQL Server 2008 搭建学习环境,以软件开发过程中的需求分析文档为基础设计学习数据库,然后使用自己设计的数据库来学习 SQL 语句。

本书提供了生成学习环境的数据表结构和插入记录的 SQL 语句,为你的学习准备好了数据。

本书主要内容

- 数据的结构化集合称为数据库。数据库模型可分为层次模型、网状模型和关系模型。关系模型是目前最普遍使用的一种数据模型,它采用人们所熟知的二维表格来描述现实世界中的实体以及各实体之间的各种联系,概念清晰、使用方便。在其被提出之后,就得到数据库开发商的积极支持,使其迅速成为继层次模型和网状模型之后一种崭新的数据库模型,如今在数据库技术领域占据了统治地位,市场上流行的数据库产品几乎都支持关系模型。本章通过一张表和一个简单的教学管理系统案例来讲解关系模型,以及如何利用关系模型来设计数据库。
- SQL 的全称是结构化查询语言,是 Structured Query Language 的缩写。随着关系数据库理论的提出,美国国家标准协会(ANSI)首次定义了 SQL 标准,随后国际标准化组织(ISO)将 SQL 标准采纳为国际标准。SQL 语言在不断发展的过程中经历了多次修改和完善。各大数据库厂商在遵循 ANSI 标准的同时,根据自己的产品特点对 SQL 进行再开发和扩展,于是就有了 SQL Server 的 Transact-SQL(简称 T-SQL)、Oracle 的 PL/SQL 等语言。本书以 T-SQL 为基础进行 SQL 语言介绍,大部分 SQL 语言内容同样适用于其他数据库产品。本章讲述最基本的 SQL 内容,这些知识虽然基础简单,

但是对学好本书的后续内容起着至关重要的作用。

- 从本章开始将踏上对数据的操作之旅，这正是 SQL 的目的，也是学习 SQL 语句乐趣的开始。数据的查询，是 SQL 中使用最频繁的操作，也是最重要的知识之一。在 SQL Server 中，大量的组件和技术都是为了查询而存在，这正说明了查询的重要性。在本章的数据查询中会大量使用上一章中介绍的 T-SQL 语法、运算符等基础知识，这些基础知识还会贯穿在后续的章节中。
- 第 3 章中讲述了基础查询，其涉及的都是单个表上的查询。但是单表查询常常不能满足查询需求。在关系数据库设计时，出于规范化，需要将数据分散在多个表中。T-SQL 因关系型数据库而生，因此在使用 T-SQL 进行查询时会经常涉及两表或多表中的数据，这就需要使用表的联接来实现多个表数据之间的联接查询。联接查询是关系型数据库的一个主要特点，同时也是区别于其他类型数据库管理系统的一个主要标志。本章将详细讲述多表联接的类型，UNION 合并结果集，增强型分组函数 CUBE、ROLLUP 和 GROUPING SETS 的使用方法。
- 如果说多表联接使查询语句变得复杂，那么子查询则使得查询变得更灵活、更有逻辑。所谓子查询，就是一个嵌套在 SELECT、INSERT、UPDATE、DELETE 语句或其他子查询中的查询。在很多情况下，查询并不是一蹴而就的，往往需要其他载体充当中间角色，而子查询正是充当着中间集这样一种角色。子查询是内部查询，返回的结果集供外部查询使用。子查询根据返回结果集中值的数量可以分为标量子查询、多值子查询和表子查询，也可以根据对外部查询的依赖性分为独立子查询和相关子查询。本章将按照这两种分类方式穿插介绍子查询。
- 在设计表的时候，很多情况下都以方便数据存储为目的，导致有时候直接阅读这些表数据比较困难。前面几章介绍的 SQL 查询，数据的检索都是针对一张表或多张表进行的，而窗口计算则是在表内按用户自定义规则进行每组数据的检索和计算。在使用 GROUP BY 子句时，总是需要将筛选的所有数据进行分组操作，它的分组作用域是整张表，分组以后查询为每个组只返回一行，而使用基于窗口的操作则是对表中的一个窗口进行操作。行列转换是通过语句将行数据转换成列或列数据转换成行显示出来。
- 在前面各章中详细地讲述了有关数据查询 SELECT 的知识，也在必要的地方略有涉及数据修改语句。在本章中，将会详细介绍数据操作语言（DML）中的 INSERT、UPDATE 和 DELETE 语句以及使用 MERGE 进行数据合并。
- 数据完整性是用来保证数据库中数据一致性的一种物理机制，主要用于防止非法、不合理或赘余数据存在于数据库中。SQL Server 提供了诸如数据类型、主键、外键、默认值约束、检查约束和规则等措施来实现这种机制。数据类型的实现机制不言自明，规则已经逐渐被淘汰，并且可以使用约束来替代它，因此本章主要介绍数据完整性的基础知识和实现数据完整性的各种约束。

- 无论是数据库的开发人员、管理人员还是用户，他们都希望能够快速地从数据库中查询到期望的数据。如果表中数据量小，可能都能立即得到结果，但随着数据量的不断增大，查询所花费的时间也在急剧增加。使用合理的索引可以对查询进行优化，但是不合理的索引也可能产生很大的副作用。
- 在本章中将详细介绍另一种表表达式：视图（VIEW）。表表达式的基本目的是一样的，都是通过查询语句将满足条件的数据筛选出来作为中间结果集以作他用。相比前面介绍的两种表表达式，视图和表一样是数据库的一种对象，它实实在在存储在数据库中，只要不显式删除它们就可以重复使用。
- 存储过程（Stored Procedure）是“过程”的，和其他编程语言的过程类似。存储过程是一段可执行的服务端程序，一个存储过程可以集合多条 SQL 语句，当它和服务器进行数据交互时，不管存储过程中包含有多少条 SQL 语句，服务器都会将它们作为一个事务进行处理并缓存它的执行计划。存储过程封装了语句，在编程角度考虑，它提高了复用性。和视图类似，它隐藏了数据库的复杂性，通过和视图类似的授权方式它还提高了数据库的安全性。同时，作为一种数据库对象，存储过程可以在需要时直接调用。本章主要介绍存储过程的创建、修改和注意事项，以及存储过程的特性。
- 在第 2 章中介绍了一些内置的系统函数，如聚合函数 AVG()和 SUM()、字符串函数 SUBSTRING()和 LTRIM()等，这些函数的作用就是实现特定的功能，简化并封装频繁执行的逻辑操作。除了这些内置函数，SQL Server 还允许创建用户自定义函数（User-Defined Functions, UDF）来扩展 SQL 语句的功能。
- 触发器实现的是一种通过某些操作的触发来完成另一个操作过程的功能。例如，当删除一张表中的某些记录时同时希望删除另一张表中的某些相关记录，当然这可以通过外键引用来实现，但是对有些表没有必要定义外键，这时候就可以通过触发器来触发实现。触发器是被动触发执行的，它不像函数、存储过程一样可以被显式调用。本章将详细讲述常用的 AFTER 触发器及其工作原理，在介绍过程中会涉及简单的事务概念。
- 多个用户对同一数据进行交互称为并发。如果不加以控制，并发可能引起很多问题。数据库提供了可以合理解决并发问题的方案。在本章中，从事务开始介绍，然后介绍并发带来的问题，最后详细介绍锁的机制和事务隔离级别。本章内容贯穿整个数据库系统，理解本章内容对数据库其他方面的学习有极大帮助，由于涉及较多理论，因此这些内容应当着重理解并通过实验进行验证。

本书适用于

- 数据库管理员

- 软件开发人员
- 企业 IT 运维人员

学生评价

韩立刚老师数据库开发视频教程：

http://edu.51cto.com/course/course_id-904.html

http://edu.51cto.com/course/course_id-926.html

下面是 51CTO 学院学生听完韩老师“数据库设计与开发”课程后的评价。

课程目录

课程介绍

课程问答

学员笔记

课程评价

资料下载

★★★★★ 5分

韩老师的课程说得非常到位、非常详细，不仅仅是教你如何做，更多的是直接将原理论告诉你，告诉你为什么要这样做，非常好，看完韩老师的视频，你也可以受教了。

★★★★★ 5分

韩老师的课总是通俗易懂、由浅入深、循循善诱，个人受益匪浅！由于时间和精力有限的原因，我要一课一课慢慢学习。

★★★★★ 5分

学习了九章，感觉韩老师的讲课风格非常好，浅显易懂，适合初学者。

★★★★★ 5分

老师很有教学经验，很了解学生，所以讲的内容很适合学生，老师讲得很有条理性，听了很有收获。

★★★★★ 5分

好棒啊，韩老师讲的我都能理解呢，通俗易懂。

★★★★★ 5分

听了两个视频，理论与实践结合，非常好，比看书快，也容易懂。

★★★★★ 5分

韩立刚老师讲得很细，表达很清晰，通俗易懂。因为一门课，通过学习达到最起码的入门，然后在工作中或者以后的学习过程中进一步掌握才有意义，有的老师讲个不停，学生却无从下手。感谢韩立刚老师。

★★★★★ 5分

我从来没听过这么好的老师讲课，讲得很细，而且有耐心，让我们在深入了解具体 SQL 操作的同时拓宽思路，让数据库学习充满乐趣，而不再枯燥，学得很快，给老师一个大大的赞。

技术支持

技术交流和资料索取请联系韩老师。

韩老师 QQ: 458717185。

韩立刚 IT 技术交流群: 301678170。

韩老师视频教学网站: <http://www.91xueit.com>。

韩老师微信: hanligangdongqing, 微信支付书费, 韩老师签名寄书。

韩老师微信公众号: han_91xueit。



致谢

河北师范大学软件学院采用“校企合作”的办学模式: 在课程体系设计上与市场接轨; 在教师的使用上, 大量聘用来自企业一线的工程师; 在教材及实验手册建设上, 结合国内优秀教材的知识体系, 大胆创新, 开发了一系列理论与实践相结合的教材(本教材即是其中一本)。在学院新颖模式的培养下, 百余名学生进入知名企业实习或已签订就业合同, 得到了用人企业的广泛认可。这些改革及成果的取得, 首先要感谢河北师范大学校长蒋春澜教授的大力支持和鼓励, 同时还要感谢河北师范大学校党委对这一办学模式的肯定与关心。

在此对河北师范大学数学与信息科学学院院长邓明立教授、软件学院副院长赵书良教授和李文斌副教授表示真诚的谢意, 是他们为本书的写作提供了良好的环境, 为本书内容的教学实践保驾护航, 他们与编著者关于教学的沟通与交流为本书提供了丰富的案例和建议。感谢河北师范大学软件学院教学团队的每一位成员, 感谢河北师范大学软件学院的每一位学生, 是他们的友好、热情、帮助和关心促成了本书。

最后, 感谢我的家人在本书创作过程中给予我的支持和理解。

韩立刚

2016年12月

目 录

前言	
第 1 章 关系数据库	1
1.1 关系模型基本概念	1
1.2 E-R 模型到关系模型的转化	3
1.3 数据库设计规范	6
1.3.1 第一范式 (1NF)	6
1.3.2 第二范式 (2NF)	7
1.3.3 第三范式 (3NF)	8
第 2 章 Transact-SQL 概述	10
2.1 启动 SQL Server Management Studio (SSMS)	10
2.2 T-SQL 语句的类型	12
2.2.1 DDL 语句	12
2.2.2 DCL 语句	13
2.2.3 DML	14
2.3 T-SQL 语法要素	16
2.3.1 SQL 语句的批处理符号 GO	16
2.3.2 EXEC	17
2.3.3 注释符	17
2.3.4 标识符	18
2.4 变量	19
2.5 数据类型	21
2.5.1 字符串类型	21
2.5.2 Unicode 字符串类型	23
2.5.3 日期时间类型	23
2.5.4 数值类型	23
2.6 运算符	25
2.6.1 算术运算符	25
2.6.2 比较运算符	25
2.6.3 逻辑运算符	25
2.6.4 连接运算符	26
2.7 常用函数	26
2.7.1 聚合函数	27
2.7.2 数值函数	28
2.7.3 字符串函数	29
2.7.4 日期时间函数	30
2.7.5 数据类型转换函数	31
2.7.6 控制 NULL 的常用函数	32
2.8 流程控制语句	32
2.8.1 条件判断语句 IF...ELSE 和 CASE	33
2.8.2 循环语句	35
第 3 章 查询基础	36
3.1 简单查询逻辑处理过程	36
3.2 数据库对象的引用规则	39
3.3 指定表的返回列来筛选列	40
3.4 使用 WHERE 筛选行	41
3.4.1 使用比较运算符过滤数据	41
3.4.2 搜索范围内的值	41
3.4.3 使用 IN 指定列表搜索条件	42
3.4.4 使用 LIKE 关键字进行模糊匹配	42
3.4.5 使用 NULL 比较搜索条件	43
3.4.6 筛选时影响性能的注意事项	44
3.5 格式化结果集	44

3.5.1	在选择列表中使用常量、函数和 表达式	44	4.9.2	CUBE	70
3.5.2	使用别名	45	4.9.3	使用 GROUPING SETS	70
3.5.3	使用 ORDER BY 子句对结果排序	46	4.9.4	GROUPING 查看汇总行并区分 NULL	71
3.5.4	使用 TOP 限制结果集	48	第 5 章 子查询	73	
3.5.5	使用 DISTINCT 消除重复行	50	5.1	独立子查询	73
3.6	使用 GROUP BY 子句和聚集函数进行 分组计算	51	5.1.1	独立标量子查询	74
3.6.1	在查询中使用聚集函数	51	5.1.2	独立多值子查询	74
3.6.2	GROUP BY 使用基础	52	5.2	相关子查询	75
3.6.3	使用 HAVING 子句筛选分组后 的数据	53	5.3	使用 IN (NOT IN) 和 EXISTS (NOT EXISTS) 谓词	76
第 4 章 多表联接查询和数据汇总	55		5.3.1	IN 和 NOT IN	77
4.1	联接基础知识	55	5.3.2	EXISTS 和 NOT EXISTS	77
4.1.1	在 FROM 子句中联接	56	5.4	表表达式	81
4.1.2	在 WHERE 子句中联接	56	5.4.1	派生表	81
4.2	交叉联接	57	5.4.2	公用表表达式 (CTE)	84
4.3	内联接	57	第 6 章 开窗函数和行列转换	89	
4.4	外联接	59	6.1	窗口和开窗函数	89
4.4.1	左外联接	59	6.2	排名窗口函数	90
4.4.2	右外联接	60	6.2.1	使用 ROW_NUMBER()进行 分区编号	91
4.4.3	完全外部联接	60	6.2.2	使用 RANK()和 DENSE_RANK() 进行分区排名	94
4.5	自联接	62	6.2.3	使用 NTILE()进行数据分组	96
4.5.1	使用同一列进行自联接	62	6.3	聚合窗口函数	97
4.5.2	使用不同列进行自联接	62	6.4	行列转换	100
4.6	联接查询的逻辑处理过程	64	6.4.1	行转列	100
4.7	多表联接查询	64	6.4.2	列转行	105
4.8	UNION 合并结果集	64	第 7 章 数据修改	109	
4.8.1	UNION 和 UNION ALL	65	7.1	插入数据	109
4.8.2	对合并结果集进行排序	66	7.1.1	使用 INSERT 和 VALUES 插入 数据	109
4.8.3	结果集的合并顺序	66	7.1.2	使用 SELECT INTO 插入数据	110
4.9	汇总数据	67			
4.9.1	ROLLUP	67			

7.1.3 使用 INSERT 和 SELECT 插入数据	111	8.7.4 设置默认值约束	144
7.2 删除数据	112	8.7.5 设置外键	145
7.2.1 使用 DELETE 删除行	112	8.7.6 禁用约束	147
7.2.2 使用 TRUNCATE TABLE 删除所有行	113	第 9 章 索引	148
7.3 更新数据	113	9.1 数据的存储方式	149
7.3.1 使用 SET 和 WHERE 更新数据	113	9.1.1 页 (Page)	149
7.3.2 更新基于其他表的数据	113	9.1.2 区 (Extent)	152
7.4 使用 MERGE 合并数据	114	9.1.3 索引分配映射页 (IAM 页)	152
7.5 通过表表达式修改数据	118	9.1.4 估算表的大小	153
第 8 章 数据完整性	121	9.2 B 树 (Balanced Tree)	154
8.1 数据完整性的类型和实现方式	121	9.3 查看页内容的工具	156
8.1.1 域完整性及实现方式	122	9.3.1 DBCC IND	156
8.1.2 实体完整性及实现方式	122	9.3.2 DBCC PAGE	158
8.1.3 参照完整性及实现方式	122	9.4 堆 (Heaps)	160
8.2 实现实体完整性	124	9.5 聚集索引	162
8.2.1 使用主键	124	9.5.1 约束与索引的关系	162
8.2.2 使用 UNIQUE 约束	127	9.5.2 唯一聚集索引的物理结构	163
8.2.3 使用自增列	127	9.5.3 不唯一聚集索引的物理结构	167
8.3 实现域完整性	131	9.5.4 使用聚集索引查找数据的 执行计划	168
8.3.1 CHECK 约束	131	9.6 非聚集索引及其物理结构	169
8.3.2 DEFAULT 约束	132	9.6.1 在堆中的非聚集索引	169
8.3.3 NOT NULL 定义	133	9.6.2 在聚集表中的非聚集索引	172
8.4 参照完整性	134	9.6.3 使用非聚集索引查找数据的 执行计划	175
8.4.1 实现表间列的参照完整性	134	9.7 修改数据对索引结构的影响	176
8.4.2 实现表内列的参照完整性	137	9.7.1 页拆分和行移动现象	176
8.5 使用关系图实现参照完整性	138	9.7.2 插入行	178
8.6 使约束失效	141	9.7.3 删除行	179
8.7 使用图形界面实现数据完整性	141	9.7.4 更新行	180
8.7.1 添加主键	141	9.8 碎片和索引维护	180
8.7.2 设置 UNIQUE 约束	142	9.8.1 碎片	180
8.7.3 设置 CHECK 约束	142	9.8.2 维护索引	182

9.9	复合索引	187	11.1.2	扩展存储过程	217
9.10	包含列索引	188	11.1.3	用户自定义存储过程	217
9.11	索引交叉	188	11.2	SQL 存储过程	218
9.12	筛选索引	189	11.2.1	创建无参数的存储过程	218
9.13	全文搜索	190	11.2.2	修改和删除存储过程	220
9.13.1	创建和修改全文目录	191	11.2.3	创建带参数的存储过程	221
9.13.2	创建全文索引	191	11.2.4	获取存储过程的执行结果	223
9.13.3	全文搜索的查询方法	192	11.3	存储过程示例分析	224
9.14	统计信息	194	11.4	存储过程的解析特点	225
9.14.1	了解统计信息的作用	194	11.5	存储过程的编译、重编译	226
9.14.2	查看和分析统计信息	195	11.6	使用存储过程的优缺点	227
9.14.3	创建统计信息	198	第 12 章	用户自定义函数	229
9.14.4	更新统计信息	198	12.1	UDF 简介	229
9.14.5	同步和异步统计信息更新	200	12.2	标量 UDF	230
9.15	基于索引设计的考虑	200	12.3	内联表值函数	232
第 10 章	视图	202	12.4	多语句表值函数	233
10.1	视图简介	202	12.5	UDF 的修改和删除	235
10.2	使用标准视图	204	12.6	UDF 的调用分析	235
10.2.1	创建标准视图	204	第 13 章	触发器	239
10.2.2	通过视图更新数据	206	13.1	触发器的概念和分类	239
10.2.3	修改和删除视图	208	13.2	DML 触发器	240
10.3	视图选项	209	13.2.1	两张特殊的临时表	240
10.3.1	使用 ENCRYPTION 选项	209	13.2.2	insert 触发器	241
10.3.2	使用 SCHEMABINDING 选项	210	13.2.3	delete 触发器	242
10.3.3	使用 CHECK OPTION 选项	211	13.2.4	update 触发器	244
10.4	使用索引视图	212	13.2.5	禁用和启用触发器	244
10.4.1	创建索引视图	213	13.2.6	检测基于列修改的触发器	244
10.4.2	索引视图的性能	213	13.2.7	FIRST 触发器和 LAST 触发器	245
10.4.3	更新索引视图	214	13.3	使用触发器实现审核跟踪	246
10.5	视图的优点	215	13.4	DML 触发器的工作原理和事务控制	246
第 11 章	存储过程	216	13.4.1	触发器的工作原理	247
11.1	存储过程的类型	216	13.4.2	DML 触发器的事务控制	248
11.1.1	系统存储过程	217	13.5	DDL 触发器	249

13.5.1 创建数据库级别的 DDL 触发器	249	14.3 锁	260
13.5.2 创建服务器级别的 DDL 触发器	250	14.3.1 锁的粒度和锁升级	260
第 14 章 事务和锁	251	14.3.2 锁的类型和查看锁	261
14.1 事务	251	14.3.3 锁的兼容性	265
14.1.1 显式事务处理模式	252	14.4 设置事务隔离级别	266
14.1.2 自动提交事务模式	253	14.4.1 SQL 标准定义的 4 种事务隔离 级别	266
14.1.3 隐式事务处理模式	253	14.4.2 行版本的事务隔离级别	271
14.1.4 嵌套事务的控制	254	14.5 隔离级别、锁和并发问题的关系	274
14.1.5 事务、事务日志和检查点	255	14.6 死锁	275
14.2 并发访问引起的问题	257		

1

关系数据库



主要内容

- 了解关系型数据库
- 掌握 E-R 模型到关系模型的转化
- 掌握数据库设计的三个规范

数据的结构化集合称为数据库。数据库模型可分为层次模型、网状模型和关系模型。关系模型是目前最普遍的一种数据模型,它采用人们所熟知的二维表格来描述现实世界中的实体以及各实体之间的各种联系,概念清晰、使用方便。在其被提出之后,就得到数据库开发商的积极支持,使其迅速成为继层次模型和网状模型之后一种崭新的数据库模型,如今在数据库技术领域占据了统治地位,市场上流行的数据库产品几乎都支持关系模型。

本章通过一张表和一个简单的教学管理系统案例来讲解关系模型以及如何利用关系模型来设计数据库。

1.1 关系模型基本概念

关系模型是由美国 IBM 公司的 E.F.Codd 博士提出的,是一种用二维表的形式来描述现实世界中实体以及实体间联系的数据模型。

现在通过图 1-1 所示的学生信息二维表来介绍关系模型中的基本概念。

- (1) 关系: 一个关系对应一张没有重复行、重复列的二维表。
- (2) 表结构: 即表中的第一行,表示组成该表的各个字段的名称。在表中,还应该包括

各字段的数据类型等。

(3) 属性：二维表中垂直方向的每一列称为一个属性，在表中对应一个字段。每一列有一个属性名，即字段名，如表中的学号、姓名、性别、生日等。

(4) 关系模式：是指对关系结构的描述，一般表示如下：

关系名(属性 1,属性 2,属性 3,...,属性 n)

例如图 1-1 所示的关系模式可以描述为：

学生(学号,姓名,性别,生日,邮箱,班级)

(5) 属性值：表中行和列的交叉位置表示某个属性的值，如“郭力月”“男”。

(6) 空值：表示属性值“不确定”“不知道”，用 NULL 表示。例如图 1-1 中学号为 0000000003 的学生没有给出邮箱，属性值为 NULL，因为不确定该学生的邮箱是什么。

(7) 域：属性的取值范围，如性别的域是(男,女)。

学号	姓名	性别	生日	邮箱	班级
0000000001	马文霞	男	1982-02-01	mawenxia@91xueit.com	测试
0000000002	郭力月	男	1987-01-15	guoliyue@91xueit.com	开发
0000000003	崔霄晨	女	1985-11-04	NULL	网络
0000000004	钟霁翔	女	1986-08-06	zhongaixiang@91xueit.com	开发
0000000005	夏可炎	男	1989-03-25	xiakeyan@91xueit.com	网络
0000000006	常亮思	女	1984-01-26	changliangsi@91xueit.com	开发
0000000007	邓咏桂	男	1982-11-18	dengyonggui@91xueit.com	开发
0000000008	方伊家	女	1985-03-02	fangyijia@91xueit.com	测试

图 1-1 学生信息表

(8) 主键：在关系中能够唯一标识表中不同行的属性或属性组合，并且这些属性值不包括空值和重复值，用 Primary Key 表示。如果是属性组合的主键，则称为联合主键。一张表中只能有一个主键，联合主键虽然包含了两列甚至多列，但也是一个联合主键。

例如在学生信息表中，一个学号值可以唯一地确定一个学生的所有信息，而且学号不能为空，也不可能重复，因而可以作为主键。因为学生可能重名，因此“姓名”列不能作为主键。

(9) 外键：如果 A 表中的一个字段引用了 B 表中的字段，两表之间根据这个引用的字段建立联系，那么引用的这个字段称为 A 表的外键，用 Foreign Key 表示。一般作为外键的字段具有唯一性和非空性，因此某个表的主键常常被引用为另一个表的外键。

例如在图 1-2 所示的学生成绩表中，“学号”“课程号”和“分数”都不能唯一地标识每个学生的成绩信息，只有学号和课程号组合起来才能区分每个学生的每科成绩，因此该表的主键是属性组(学号,课程号)组成的联合主键。学号是学生信息表的主键，而不是学生成绩表的主键，在两表相关联的时候学号称为学生成绩表的外键。

(10) 主表和从表：是指通过外键相关联的两个表，其中被引用字段所在的表称为主表，另一张表称为从表。

例如图 1-2 中，学生信息表以学号作为主键，而学号是学生成绩表的外键，所以学生信息表是主表，学生成绩表是从表。

学号	姓名	性别	生日	班级
0000000001	马文霞	男	1982-02-01	测试
0000000002	郭力月	男	1987-01-15	开发
0000000003	崔霄晨	女	1985-11-04	网络
0000000004	钟遥翔	女	1986-08-06	开发
0000000005	夏可炎	男	1989-03-25	网络
0000000006	常亮思	女	1984-01-26	开发

学号	课程号	分数
0000000001	0001	59
0000000001	0002	55
0000000001	0003	70
0000000002	0001	58
0000000002	0002	77
0000000002	0003	63

图 1-2 学生信息表和学生成绩表

(11) 关系数据库：是指一些相关的表和其他数据库对象的集合。一个关系数据库包含多个数据表，这些表之间的关联性是由主键和外键所体现的参照关系来实现的。数据库不仅包含表，而且包含了其他数据库对象，如视图、存储过程、索引等。

1.2 E-R 模型到关系模型的转化

数据库设计案例：某学校设计学生教学管理系统，学生实体包括学号、姓名、性别、生日、民族、籍贯、登记照，每名选择选择一个主修专业，专业包括专业号、专业名和专业类别，一个专业属于一个学院，一个学院可以有若干个专业。学院信息要存储学院号、学院名、院长。教学管理还要管理课程表和学生成绩。课程表包括课程号、课程名、学分，每门课程由一个学院开设。学生学习每门课程获得一个成绩。

如图 1-3 所示，有学生、专业、学院和课程 4 个实体和每个实体对应的属性，实体用方框表示，属性用椭圆表示，这种模型称为 E-R 模型。所谓 E-R 模型，它是一种工具，将实体与属性转化为关系模型，作为连接实际对象与数据库的桥梁。



图 1-3 4 个实体的 E-R 图



图 1-3 4 个实体的 E-R 图 (续图)

针对案例,可以初步设计成 4 张二维表,分别为学生表、专业表、学院表和课程表。每张表都有一个主键,分别是学号、专业号、学院号和课程号。

根据上面的案例需求,一个学生只能主修一个专业,一个专业可以由多名学生主修,因此专业和学生是一对多的关系;一个学院可以设置多个专业,一个专业只能属于一个学院,因此学院和专业是一对多的关系;一个学院可以开设多门课程,一门课程只能由一个学院开设,因此学院和课程是一对多的关系;一个学生可以学习多门课程,一门课程可以由多名学生学习,因此课程和学生的关系是多对多的关系。

1. E-R 模型到关系模型的转化

(1) E-R 模型中的实体映射为关系表。

(2) E-R 模型中的属性映射为关系属性,即字段。

2. 一对多 (1:n) 到关系模型的转化

要转化 1:n 的关系,需要在 n 方实体表中增加一个属性,该属性对应 1 方的主键属性。例如专业和学生两个实体,学生实体是 n 方,专业实体是 1 方,需要在学生实体表中增加专业表中的主键,即专业号。通过增加该属性可以唯一确定双方之间的联系,知道了学号可以立即对应到该学生所修的专业。

图 1-4 所示的模型转化为关系模型为:

学生(学号,姓名,性别,生日,民族,籍贯,登记照,专业号)

专业(专业号,专业名,专业类别)

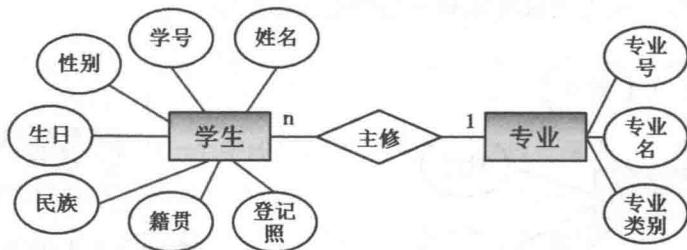


图 1-4 1:n 到关系模型的转化

3. 多对多 (m:n) 到关系模型的转化

转化一个 m:n 的关系,除了对两个实体分别转化以外,还需要单独建立一个关系模式,该关系模式以两个实体的主键作为它的外键,以表示两个实体间的 m:n 关系。例如,学生和