

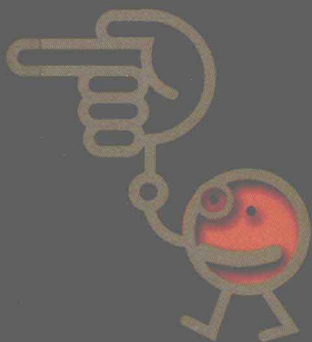
· 高等职业技术教育规划教材 ·



SQL Server 2008

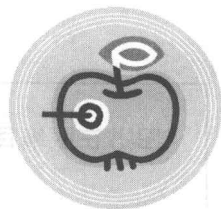
数据库技术与应用

- 主 编 刘琦曙
- 副主编 朱军涛
- 主 审 贾振刚



华中科技大学出版社

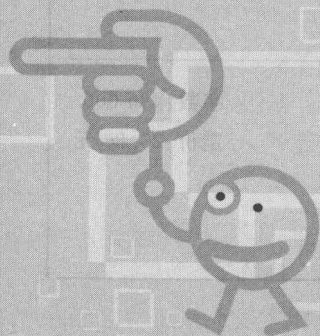
· 高等职业技术教育规划教材 ·



SQL Server 2008

数据库技术与应用

常州大学图书馆
藏书章



华中科技大学出版社
中国·武汉

图书在版编目(CIP)数据

SQL Server 2008 数据库技术与应用/刘琦曙 主编. —武汉:华中科技大学出版社,2010年1月

ISBN 978-7-5609-5868-2

I. S… II. 刘… III. 关系数据库-数据库管理系统,SQL Server 2008-高等学校:技术学校-教材 IV. TP311.138

中国版本图书馆 CIP 数据核字(2009)第 219067 号

SQL Server 2008 数据库技术与应用

刘琦曙 主编

策划编辑:沈旭日 杨 鸥

责任编辑:熊 慧

责任校对:朱 玟

封面设计:刘 卉

责任监印:周治超

出版发行:华中科技大学出版社(中国·武汉)

武昌喻家山 邮编:430074 电话:(027)87557437

录 排:武汉众心图文激光照排中心

印 刷:湖北新华印务有限公司

开本:787mm×1092mm 1/16

印张:19.5

字数:480 000

版次:2010年1月第1版

印次:2010年1月第1次印刷

定价:29.80元

ISBN 978-7-5609-5868-2/TP·711

(本书若有印装质量问题,请向出版社发行部调换)

● — 内 容 简 介

A b s t r a c t

SQL Server 2008 是 Windows Server 的最新版本,它推出了许多新的特性和关键的改进,是一款强大而全面的 SQL Server 版本。本书将详细介绍 SQL Server 2008 中的特性和功能。全书主要包括 SQL Server 基础知识、SQL Server 2008 新增特性、安装、数据库安全、基于 C2C 模式的多用户网上购物系统分析与设计、数据库管理、表的创建与管理、数据查询和管理、使用 Transact-SQL 编程、存储过程与触发器、索引和视图、事务、数据库的备份、还原和维护、使用 .NET 访问 SQL Server 2008、电子商铺系统实现等内容。

● — 前 言

F o r e w o r d

SQL Server 2008 是一款优秀的关系型数据库管理系统产品,具有开发速度快、功能强大和操作灵活等优点。

本书采用案例式教学法介绍了 SQL Server 2008 的基本性能、安装、配置及基本应用,共 13 章。全书以理论够用、实用第一为原则,使读者能够快速、轻松地掌握 SQL Server 2008 数据库技术与应用。本书内容丰富,结构清晰,语言简练,图文并茂,具有很强的实用性和可操作性,主要为刚接触数据库系统的初级用户提供学习 SQL Server 2008 所必需的基础知识。本书适合作为高等职业技术学院、高等专科学校学生阅读的教材,也可以作为 SQL Server 培训学员和 SQL Server 数据库爱好者的自学教材。

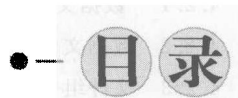
本书由刘琦曙担任主编,朱军涛担任副主编,贾振刚担任主审。任佳、左荣欣、宋继军、何振华、陈婕、易斌、周庞荣、周益卫编著。其中第 1 章、第 2 章由朱军涛编写,第 3 章、第 4 章由任佳编写,第 5 章、第 6 章由刘琦曙编写,第 7 章由左荣欣编写,第 8 章由宋继军编写,第 9 章、第 10 章由何振华编写,第 11 章由陈婕、易斌编写,第 12 章由周庞荣编写,第 13 章由周益卫编写。

使用本教材时,建议理论教学授课 46 学时,实验 46 学时。

由于作者水平有限,加之时间仓促,书中如果出现错漏之处,恳请广大读者批评指正,以便再版时修订完善。E-mail:lqs_8@126.com。

编 者

2009 年 7 月



C o n t e n t s

第 1 章 SQL Server 2008 简介	(1)	2.3.2 创建 SQL Server 账户登录	...	(22)
1.1 数据库基础知识	(1)	2.3.3 删除登录账户	(22)	
1.1.1 基本概念	(1)	2.4 权限	(23)	
1.1.2 数据库系统的发展历史	(2)	2.4.1 权限的种类	(23)	
1.1.3 数据库系统的特点	(3)	2.4.2 权限的管理	(24)	
1.2 关系数据库	(3)	2.5 数据库用户	(26)	
1.2.1 数据库的数据模型	(3)	2.5.1 数据库用户的创建	(26)	
1.2.2 关系数据库的定义	(5)	2.5.2 修改数据库用户	(28)	
1.3 范式理论和 E-R 模型	(5)	2.5.3 删除数据库用户	(28)	
1.3.1 范式理论	(5)	2.6 角色	(28)	
1.3.2 E-R 模型	(6)	2.6.1 服务器角色	(29)	
1.4 SQL Server 2008 概述	(7)	2.6.2 数据库角色	(30)	
1.4.1 随时随地管理数据	(7)	2.6.3 应用程序角色	(31)	
1.4.2 新功能	(8)	2.6.4 用户自定义角色	(32)	
1.5 安装与配置	(9)	本章小结	(33)	
1.5.1 安装环境	(9)	习题	(33)	
1.5.2 安装 SQL Server 2008	(9)	第 3 章 多用户网上购物系统分析与设计	(34)	
1.5.3 配置 SQL Server 2008	(14)	3.1 系统设计	(34)	
1.6 SQL Server 2008 管理工具	(16)	3.1.1 系统需求描述	(34)	
本章小结	(17)	3.1.2 功能模块划分	(35)	
习题	(17)	3.1.3 系统流程分析	(35)	
第 2 章 数据库安全	(18)	3.2 数据库设计	(36)	
2.1 安全机制	(18)	3.2.1 数据库需求分析	(36)	
2.1.1 安全配置	(18)	3.2.2 数据库概念结构设计	(37)	
2.1.2 所有活动审查	(19)	3.2.3 数据库逻辑结构设计	(38)	
2.1.3 独立的用户与架构机制	(19)	本章小结	(40)	
2.1.4 加密密钥和内置加密函数	(19)	习题	(40)	
2.2 SQL Server 2008 验证模式	(20)	第 4 章 数据库管理	(41)	
2.3 SQL Server 2008 登录	(21)	4.1 数据库	(41)	
2.3.1 创建 Windows 账户登录	(21)			

4.2 数据文件和日志..... (42)	6.1.5 GROUP BY 子句 (105)
4.2.1 数据文件 (42)	6.1.6 HAVING 子句 (106)
4.2.2 数据文件的结构 (42)	6.2 SELECT 操作多表数据 (107)
4.2.3 文件组 (43)	6.2.1 JOIN 连接 (107)
4.2.4 日志文件的结构 (44)	6.2.2 内连接 (108)
4.3 数据库管理基础..... (44)	6.2.3 外连接 (109)
4.3.1 系统数据库 (44)	6.2.4 交叉连接 (110)
4.3.2 数据库快照 (46)	6.2.5 自连接 (111)
4.4 数据库状态和文件状态..... (47)	6.2.6 联合查询 (111)
4.5 创建数据库..... (49)	6.2.7 使用子查询 (112)
4.5.1 使用图形化方法创建 (49)	6.3 插入 SQL 数据..... (113)
4.5.2 使用 Transact-SQL 创建 (51)	6.3.1 INSERT...VALUES 语句 ... (114)
4.6 数据库操作..... (54)	6.3.2 INSERT...SELECT 语句..... (114)
4.6.1 查看数据库信息 (54)	6.3.3 SELECT...INTO 语句创建表 (115)
4.6.2 修改数据库 (56)	6.4 更新 SQL 数据..... (115)
4.6.3 删除数据库 (58)	6.4.1 根据表中的数据更新行 (116)
4.6.4 其他数据库操作 (58)	6.4.2 根据其他表更新行 (116)
本章小结 (61)	6.4.3 使用 TOP 表达式 (117)
习题 (62)	6.5 删除 SQL 数据..... (117)
第 5 章 表的创建与管理 (63)	6.5.1 使用 DELETE 语句 (117)
5.1 表概述..... (63)	6.5.2 使用 TRUNCATE TABLE 语句 (118)
5.1.1 表的含义 (63)	6.5.3 删除基于其他表中的行 (119)
5.1.2 临时表和系统表 (63)	本章小结..... (119)
5.1.3 创建表 (63)	习题..... (120)
5.2 列数据类型..... (66)	第 7 章 使用 Transact-SQL 编程 (121)
5.3 列的其他属性..... (73)	7.1 Transact-SQL 概述 (121)
5.4 维护数据表..... (76)	7.1.1 Transact-SQL 简介 (121)
5.5 数据库完整性..... (79)	7.1.2 Transact-SQL 的特点 (121)
5.5.1 约束类型..... (80)	7.1.3 Transact-SQL 的语法规范..... (122)
5.5.2 使用默认值维护数据完整性 ... (86)	7.2 Transact-SQL 的语言基础..... (124)
5.5.3 使用规则维护数据完整性..... (88)	7.2.1 常量与变量 (124)
5.6 表关系图..... (90)	7.2.2 表达式 (127)
本章小结 (92)	7.2.3 运算符 (128)
习题 (92)	7.2.4 控制语句 (132)
第 6 章 数据查询和管理 (93)	7.3 常用函数 (137)
6.1 SELECT 获取简单数据 (93)	7.3.1 字符串函数 (137)
6.1.1 SELECT 语法 (94)	7.3.2 数学函数 (138)
6.1.2 FROM 子句..... (98)	
6.1.3 WHERE 子句 (99)	
6.1.4 ORDER BY 子句 (104)	

7.3.3 聚合函数	(138)	本章小结	(188)
7.3.4 数据类型转换函数	(139)	习题	(188)
7.3.5 日期和时间函数	(140)	第 10 章 事务	(189)
7.4 高级操作	(141)	10.1 事务	(189)
7.4.1 锁	(141)	10.1.1 事务的设计	(189)
7.4.2 游标	(143)	10.1.2 显式事务	(190)
本章小结	(145)	10.1.3 自动提交事务	(191)
习题	(145)	10.1.4 隐式事务	(192)
第 8 章 存储过程与触发器	(146)	10.2 SQL Server 2008 本地事务支持	(194)
8.1 存储过程概述	(146)	10.3 隔离级别	(197)
8.1.1 什么是存储过程	(146)	10.4 分布式事务	(199)
8.1.2 存储过程的特点	(148)	10.5 高级事务	(202)
8.2 使用存储过程	(149)	本章小结	(208)
8.2.1 创建普通存储过程	(149)	习题	(208)
8.2.2 使用存储过程参数	(151)	第 11 章 数据库的备份、还原和维护	(210)
8.2.3 执行存储过程	(153)	11.1 备份概述	(210)
8.2.4 管理存储过程	(155)	11.2 备份数据	(212)
8.3 触发器概述	(157)	11.2.1 创建备份设备	(212)
8.3.1 触发器定义	(157)	11.2.2 管理备份设备	(214)
8.3.2 触发器的分类	(157)	11.2.3 完整备份	(214)
8.4 创建触发器	(158)	11.2.4 差异备份	(217)
8.4.1 数据操纵语言触发器的创建	(158)	11.2.5 事务日志备份	(219)
8.4.2 数据定义语言触发器	(162)	11.2.6 文件组备份	(221)
8.4.3 嵌套触发器	(163)	11.2.7 备份到多个设备上	(223)
8.4.4 递归触发器	(164)	11.2.8 备份压缩	(224)
8.5 管理触发器	(165)	11.3 恢复数据	(225)
本章小结	(167)	11.3.1 还原差异数据库备份	(225)
习题	(167)	11.3.2 还原事务日志备份	(228)
第 9 章 索引和视图	(168)	11.4 复制数据库	(231)
9.1 索引	(168)	本章小结	(232)
9.2 索引类型	(168)	习题	(232)
9.3 索引操作	(170)	第 12 章 使用 .NET 访问 SQL Server 2008	(233)
9.4 视图的概述	(177)	12.1 ADO.NET	(233)
9.4.1 视图的分类	(177)	12.1.1 ADO.NET 简介	(233)
9.4.2 视图的使用范围	(177)	12.1.2 连接对象	(234)
9.4.3 视图的创建	(178)	12.1.3 命令	(235)
9.4.4 修改视图定义	(181)		
9.4.5 视图加密	(183)		
9.4.6 使用视图加强数据的安全	(184)		
9.4.7 通过视图修改数据	(187)		



12.1.4	数据读取器对象	(236)	13.4.1	系统首页界面设计	(266)
12.1.5	数据集对象	(237)	13.4.2	会员登录模块界面设计	(266)
12.1.6	数据适配器对象	(237)	13.4.3	商品查找模块界面设计	(267)
12.2	数据绑定和访问数据	(241)	13.4.4	商品分类列表模块界面设计	(267)
12.2.1	数据绑定	(241)	13.4.5	最受欢迎商品模块界面设计	(267)
12.2.2	数据导航	(243)	13.4.6	商品详细信息模块界面设计	(267)
12.2.3	数据的过滤	(246)	13.4.7	购物车模块界面设计	(268)
12.3	更新数据	(250)	13.4.8	订单查询模块界面设计	(270)
12.4	多表操作	(252)	13.5	模块功能设计与代码实现分析	(270)
本章小结		(256)	13.5.1	系统登录模块	(270)
习题		(256)	13.5.2	商品查找模块	(276)
第 13 章 电子商铺系统实现		(257)	13.5.3	商品分类列表模块	(277)
13.1	系统设计	(257)	13.5.4	最受欢迎商品模块	(278)
13.1.1	系统功能描述	(257)	13.5.5	商品详细信息模块	(280)
13.1.2	功能模块划分	(257)	13.5.6	购物车模块	(282)
13.1.3	系统流程分析	(258)	13.5.7	订单查询模块	(295)
13.2	数据库设计	(258)	13.6	实例演示	(300)
13.2.1	数据库需求分析	(258)	参考文献		(302)
13.2.2	数据库概念结构设计	(259)			
13.2.3	数据库逻辑结构设计	(260)			
13.3	连接数据库	(263)			
13.4	界面设计	(266)			

SQL Server 2008 简介

2008 年,微软(Microsoft)公司发布了数据库软件 SQL Server 2008,这是一个重要的数据库产品版本,它推出了许多新的特性和关键的改进,成为至今为止的最强大和最全面的 SQL Server 版本。

本章首先介绍数据库基础知识,然后介绍 SQL Server 2008 的安装与配置,最后介绍 SQL Server 2008 的主要管理工具。

1.1 数据库基础知识

1.1.1 基本概念

1. 数据

数据(data)是数据库中存储的基本对象,是描述事物的符号。它与传统意义上理解的数据不同,数据在这里可以是数字、文字、图形、图像和声音等,即数据有多种形式,它们都可以经过数字化后存入计算机。

2. 数据库

数据库(database,简称 DB)可以直观地理解为存放数据的仓库。只不过这个仓库是建立在计算机上的,在计算机上需要有存储空间和一定的存储格式。严格的定义是:数据库是被长期存储在计算机内的、有组织的、统一管理的相关数据的集合,它能为用户共享,具有最小冗余度,数据间联系密切,有较高的独立性。

3. 数据库管理系统

数据库管理系统(database management system,简称 DBMS)是位于用户和操作系统之间的一种操纵和管理数据库的大型软件,用于建立、使用和维护数据库。它对数据库进行统一的管理和控制,以保证数据库的安全性和完整性。用户通过数据库管理系统访问数据库中的数据,数据库管理员(database administrator,简称 DBA)也通过数据库管理系统进行数据库的维护工作。

常用的数据库管理系统有 Oracle、Sybase、Informix、Microsoft SQL Server、Microsoft Access 和 Visual FoxPro 等。

4. 数据库系统

数据库系统(database system,简称 DBS)是指在计算机系统中引入数据库后构成的系统,是一个实际可以运行的、按照数据库方法存储、维护并向应用系统提供数据支持的系统,它是由计算机硬件、操作系统、数据库管理系统,以及在它支持下建立起来的数据库、应用程序、用户和数据库管理员组成的一个整体。数据库系统简图如图 1-1 所示。

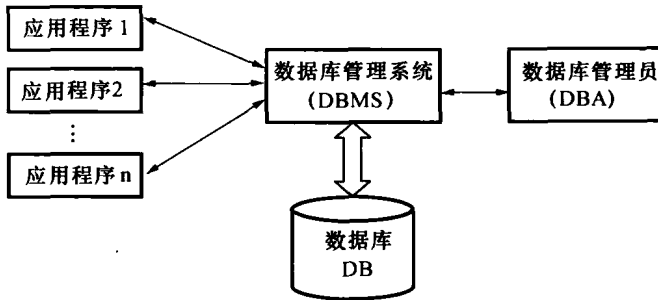


图 1-1 数据库系统简图

5. 数据库管理员

数据库管理员是负责数据库系统正常运行的高级人员,决定数据库的数据内容和存储结构、定义数据的安全性、完整性、监控数据库的运行与数据的重组恢复。

1.1.2 数据库系统的发展历史

1. 人工管理阶段

20 世纪 50 年代中期以前,计算机主要用于科学计算。硬件方面只有卡片、纸带、磁带等,没有可以直接访问、直接存取的外部存取设备。数据与程序不能独立,数据不能长期保存,如图 1-2 所示。

人工管理阶段的特点为:数据不进行保存,没有专门的数据管理软件,数据面向应用,基本上没有文件的概念。

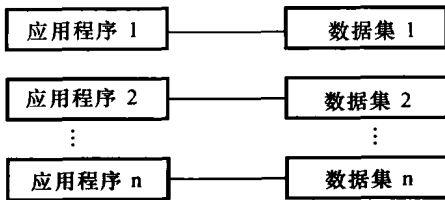


图 1-2 人工管理数据与程序的关系

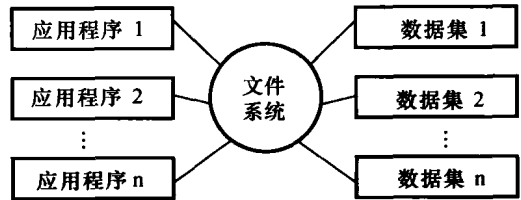


图 1-3 文件系统数据与程序的关系

2. 文件系统阶段

20 世纪 50 年代中期到 60 年代中后期,计算机主要用于数据处理。硬件出现了直接存取的磁盘、磁鼓。软件则出现了高级语言和操作系统,以及专门管理外存的数据管理软件,实现了按文件访问的管理技术,如图 1-3 所示。在这个阶段,程序与数据有了一定的独立性,程序与数据分开,有了程序文件与数据文件的区别。数据文件可以长期保存在外存上多次存取,进行诸如查询、修改、插入、删除等操作。但数据冗余度大,缺乏数据独立性,数据无法集中管理。

文件系统阶段的特点为:数据可以长期保存在磁盘上;文件系统提供数据与程序之间的存取方法;数据冗余量大;文件之间缺乏联系,相互孤立,仍然不能反映现实世界各种事物之间错综复杂的联系。

3. 数据库系统阶段

从 20 世纪 60 年代后期开始,由于实际需要,数据库技术得到了发展。数据库是通用化的相关数据集,它不仅包括数据本身,而且包括数据之间的联系。为了让多种应用程序并发地使用数据库中具有最小冗余的共享数据,必须使数据与程序具有较高的独立性。这就需要一

个软件系统对数据实行专门的管理,提供安全性和完整性等统一控制,方便用户以交互命令或程序方式对数据库进行操作。为数据库的建立、使用和维护而配置的软件成为数据库管理系统。

数据库系统阶段的特点为:数据结构化;数据共享性好;数据独立性好;数据存取粒度小;数据库管理系统对数据进行统一的管理和控制,为用户提供了友好的接口。

4. 分布式数据库系统阶段

分布式数据库系统在逻辑上像一个集中式数据库系统,实际上,数据存储在网络的不同地域的节点上。每个节点有自己的局部数据库管理系统,它有很高的独立性。用户可以通过分布式数据库系统通过网络相互传输数据。

1.1.3 数据库系统的特点

数据库系统能被广泛地应用,原因是它有很多特点,或者应该说是优点。下面列出了数据库系统的一些特点。

(1) 数据冗余度低。可以最大限度地减少数据库系统中的重复数据,使存取速度更快,在有限的存储空间内存放更多的数据。

(2) 避免数据的不一致,保证较高的数据独立性。数据和程序彼此独立,数据存储结构的变化尽量不影响用户程序的使用,使应用程序的开发更加自由。

(3) 数据实现共享。可以使更多的用户更充分地利用已有数据资源,减少资料收集、数据采集等重复劳动和相应费用,降低了系统开发的成本,使用户提高了工作效率。

(4) 具有数据安全性和完整性保障。数据库系统具有数据的安全性,可以防止数据丢失和被非法使用;具有数据的完整性,可以保证数据的正确、有效和相容。

(5) 实现并发控制和数据恢复。数据可以并发控制,避免并发程序之间的相互干扰,多用户操作可以进行并行调度;具有数据的恢复功能,在数据库被破坏或数据不可靠时,系统有能力把数据库恢复到最近某个正确的状态。

(6) 易于使用、便于扩展。数据库系统的使用简单,开发的应用软件便于用户掌握。

1.2 关系数据库

1.2.1 数据库的数据模型

数据模型是指数据库中数据的组织形式和联系方式。数据库中的数据是按照一定的结构存放的,以反映数据之间的联系。按照数据库中数据采取的不同联系方式,数据模型可分为三种:层次模型、网状模型和关系模型,相应的数据库分别称为层次数据库、网状数据库和关系数据库。其中层次数据库和网状数据库统称为非关系数据库。

1. 层次模型

层次模型是数据库系统中最早出现的数据模型,它用树形结构表示各类实体及实体间的联系。若用图来表示,则层次模型是一棵倒立的树。节点层次从根开始定义,根为第一层,根的孩子称为第二层,根称为其孩子的双亲,同一双亲的孩子称为兄弟。图 1-4 所示的是一个简单的层次模型。

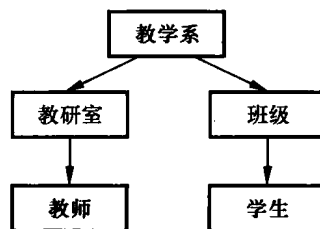


图 1-4 层次模型示例

在数据库中,满足以下两个条件的数据模型称为层次模型:

- (1) 有且仅有一个节点无双亲,这个节点称为根节点;
- (2) 其他节点有且仅有一个双亲。

层次模型具有层次清晰、构造简单、易于实现等优点。但由于受到如上所述两个条件的限制,它可以比较方便地表示一对一和一对多的实体联系,而不能直接表示多对多的实体联系。对于多对多的联系,必须先将其分解为几个一对多的联系才能表示出来。因而对于复杂的数据关系,实现起来比较困难,这是层次模型的局限性。采用层次模型设计的数据库称为层次数据库。层次数据库系统的典型代表是 IBM 公司的 IMS(Information Management Systems)数据库管理系统,这是一个最早出现的大型数据库管理系统。

2. 网状模型

网状模型用以实体为节点的有向图来表示各实体及其之间的联系。自然界中实体模型间的联系更多的是非层次关系,用层次模型难以描述非层次关系,用网状模型则可以。网状模型是一个网络,如图 1-5 所示。

在数据库中,网状模型必须满足以下两个条件:

- (1) 允许一个以上的节点无双亲;
- (2) 一个节点可以有多个的双亲。

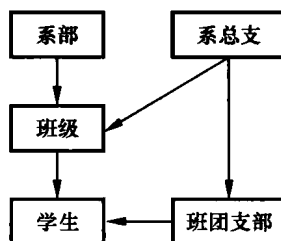


图 1-5 网状模型示例

网状数据模型的典型代表是 DBTG 系统,也称为 CODASYL 系统,该系统是 20 世纪 70 年代数据系统语言研究会下属的数据库任务组(database Task Group,简称 DBTG)提出的一个数据模型方案。

在以上两种数据模型中,各实体之间的联系是用指针实现的,其优点是查询速度快。但是,当实体集和实体集中实体的数目都较多时,众多的指针使得管理工作变得相当复杂,对用户来说使用也比较麻烦。

3. 关系模型

关系模型与层次模型和网状模型相比有着本质的差别,它用二维表格来表示实体及其相互之间的联系。在关系模型中,把实体集看成一张二维表,每一张二维表称为一个关系。每个关系均有一个名字,称为关系名。表 1-1 所示的是一张学生信息表,该表记录了学生的学号、班级编号、班级人数和姓名等信息。

表 1-1 学生信息表

学号	班级编号	班级人数	姓名	性别	民族	出生日期	政治面貌	籍贯
001	04	45	王天	男	汉族	1990-3-12	团员	河南省郑州市
002	04	45	李晓旭	女	汉族	1991-1-9	团员	安徽省巢湖市
003	04	45	杨峰	男	汉族	1989-12-25	党员	河南省洛阳市
004	04	45	王媛媛	女	瑶族	1990-5-18	团员	湖南省怀化市

一个关系就是没有重复行和重复列的二维表,二维表的每一行在关系中称为元组,每一列在关系中称为属性。学生关系的每一行代表一个学生的记录,每一列代表学生记录的一个字段。

关系模型的特点如下。

- (1) 每一列必须是基本数据项(不可再分解)。
- (2) 表中每一列必须具有相同的数据类型。

- (3) 表中每一列的名字必须是唯一的。
- (4) 表中不应有内容完全相同的行。
- (5) 行的次序与列的次序不影响表格中所表示的信息的含义。

虽然关系模型与层次模型和网状模型相比发展较晚,但是因为它建立在严格的数学理论基础上,采用了人们习惯使用的表格形式作为存储结构,易学易用,因而成为使用最广泛的数据库模型。自 20 世纪 80 年代以来,新推出的数据库管理系统几乎都支持关系模型。本书讨论的 SQL Server 2008 就是一种关系数据库管理系统。

1.2.2 关系数据库的定义

关系数据库是一些相关的表和其他数据库对象的集合。该定义有以下三层含义。

- (1) 在关系数据库中,信息存放在二维表格结构的表(table)中,一个关系对应一张二维表。一个关系数据库可以包含多张表,每一张表都由行(又称为记录)和列(又称为字段)组成。
- (2) 实体的属性中,能唯一标识实体集中每个实体的一个或几个属性,称为实体的关键字。在关系数据库中,关键字称为主键。关系数据库的表之间是相互关联的,表之间的这种关联性是由主键和外键所体现的参照关系实现的。
- (3) 关系数据库不仅包括表,还包括其他数据库对象,如视图、存储过程、索引等。

以下是一些相关术语。

键码(key):键码用来标识行的一列或多列。

超关键字:二维表中,能够唯一确定记录的一个字段或几个字段的组合称为超关键字,超关键字虽然能唯一确定记录,但是它所包含的字段可能是有多余的。

主关键字(primary key):主关键字是被挑选出来,用于表的行的唯一标识的候选关键字。一张表只有一个主关键字。主关键字又称为主键。主键可以由一个字段,也可以由多个字段组成,这个(些)字段分别成为单字段主键或多字段主键。

候选关键字(candidate key):如果一个超关键字在去掉其中任何一个字段后就不再能唯一地确定记录,则称为候选关键字。候选关键字能唯一地确定记录,它包含的字段有的是最精练的。也就是说,候选关键字是最简单的超关键字。

外关键字(foreign key):如果公共关键字在一个关系中是主关键字,那么这个公共关键字称为另一个关系的外关键字。由此可见,外关键字表示了两个关系之间的联系。以另一个关系的外关键字为主关键字的表称为主表,具有此外关键字的表称为主表的从表。外关键字又称为外键。例如,当一张二维表(如表 A)的主关键字被包含在另一张二维表(如表 B)中时,表 A 中的主关键字便成为表 B 的外关键字。

公共关键字(common key):在关系数据库中,关系之间的联系是通过相容或相同的属性或属性组来表示的,如果两个关系具有相容或相同的属性或属性组,那么这个属性或属性组称为这两个关系的公共关键字。

1.3 范式理论和 E-R 模型

1.3.1 范式理论

构造数据库必须遵循一定的规则,在关系数据库中,这种规则就是范式。范式是符合某一

种级别的关系模式的集合。关系数据库中的关系必须满足一定的要求,即满足不同的范式。目前关系数据库有六种范式:第一范式(1NF)、第二范式(2NF)、第三范式(3NF)、BCNF 范式、第四范式(4NF)和第五范式(5NF)。满足最低要求的范式是第一范式。在第一范式的基础上进一步满足更多要求的称为第二范式,其余范式以此类推。一般来说,数据库只需满足第三范式即可。下面介绍第一范式、第二范式和第三范式。

1. 第一范式

在任何一个关系数据库中,第一范式是对关系模式的基本要求,不满足第一范式的数据库就不是关系数据库。

所谓第一范式是指数据库表的每一列都是不可分割的基本数据项,同一列中不能有多个值,即实体中的某个属性不能有多个值或者不能有重复的属性。如果出现重复的属性,就可能需要定义一个新的实体,新的实体由重复的属性构成,新实体与原实体之间为一对多的关系。在第一范式中,表的每一行只包含一个实例的信息。简而言之,第一范式就是无重复的列。

2. 第二范式

第二范式是在第一范式的基础上建立起来的,即满足第二范式必须先满足第一范式。第二范式要求数据库表中的每个实例或行必须可以被唯一地区分。为实现区分通常需要为表加上一个列,以存储各个实例的唯一标识。例如,在表 1-1 所示的学生信息表中加上学生编号(学号)列,因为每个学生的学号是唯一的,因此每个学生可以被唯一区分。这个唯一属性列称为主关键字或主键、主码。

第二范式要求实体的属性完全依赖于主关键字。完全依赖是指不能存在仅依赖主关键字一部分的属性,如果存在,就应该将这个属性和主关键字的这一部分分离出来,形成一个新的实体,新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列,以存储各个实例的唯一标识。简而言之,第二范式就是非主属性非部分依赖于主关键字。

3. 第三范式

满足第三范式必须先满足第二范式。也就是说,第三范式要求一个数据库表中不包含已在其他表中包含的非主关键字信息。例如,存在一个班级信息表,其中每个班级有班级编号、班级名称、班级人数等,那么在表 1-1 所示的学生信息表中列出班级编号后,就不能再将班级名称等与班级有关的信息加入学生信息表中。如果不存在班级信息表,则根据第三范式也应该构建它,否则就会有大量的数据冗余。简而言之,第三范式就是属性不依赖于其他非主属性。

设计关系数据库时一定要遵守一定的规则,尤其是数据库设计范式。

1.3.2 E-R 模型

1. E-R 模型的结构

实体-联系模型(简称 E-R 模型)是由 P. P. Chen 于 1976 年首先提出的。它提供不受任何数据库管理系统约束的面向用户的表达方法,在数据库设计中被广泛用作数据建模的工具。E-R 数据模型问世后,经历了许多修改和扩充,这里仅介绍基本的 E-R 模型。

E-R 模型的构成成分是实体集、属性和联系集,其表示方法如下。

- (1) 实体集用矩形框表示,矩形框内写上实体名。
- (2) 实体的属性用椭圆框表示,框内写上属性名,并用无向边与其实体集相连。
- (3) 实体间的联系用菱形框表示,联系以适当的含义命名,名字写在菱形框中,用无向连

线将参加联系的实体矩形框分别与菱形框相连,并在连线上标明联系的类型,即 1:1(1 对 1)、1:m(1 对多)或 m:m(多对多)。

因此,E-R 模型也称为 E-R 图。图 1-6 所示的是某企业生产和库存系统的 E-R 模型。

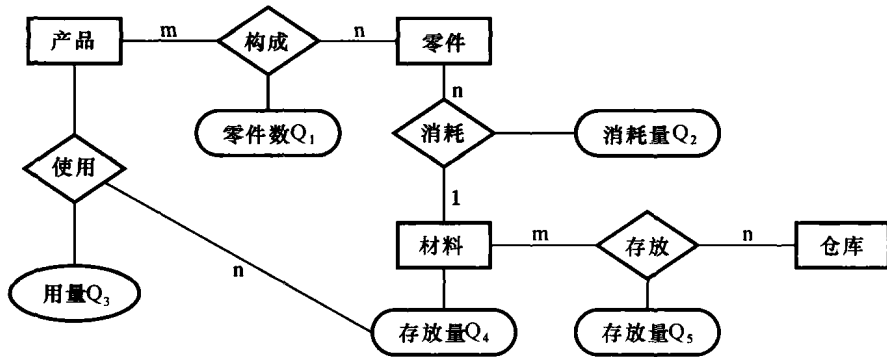


图 1-6 生产和库存的 E-R 模型

从图 1-6 中可以看出,一个产品可以使用多种材料,一种材料可以用在多种产品中;一个产品可以使用多个零件,一个零件可以用在多个产品中;一个零件只能使用一种材料,一种材料可用在多个零件中;一种材料可存放在多个仓库中,一个仓库可以存放多种材料。

2. 作 E-R 图的步骤

作 E-R 图的步骤如下:

- (1) 确定实体和实体的属性;
- (2) 确定实体之间的联系及联系的类型;
- (3) 给实体和联系加上属性。

3. E-R 模型向关系模型的转换

E-R 模型向关系模型转换的规则如下。

- (1) 一个实体型转换为一个关系模式,实体的属性就是关系的属性,实体的码就是关系的码。
- (2) 一个 1:1 联系可以转换为一个独立的关系模式,也可以与任意一端对应的关系模式合并。如果转换为一个独立的模式,则与该联系相连的各实体的码及联系本身的属性均转换为关系的属性,每个实体的码均是该关系的候选键。
- (3) 一个 1:n 联系可以转换为一个独立的关系模式,也可以与任意 n 端对应的关系模式合并。如果一个 1:n 联系转换为一个独立的模式,则与该联系相连的各实体的码及联系本身的属性均转换为关系的属性,而关系的码为 n 端实体的码。
- (4) 一个 m:n 联系转换为一个独立的关系模式,与该联系相连的各实体的码及联系本身的属性均转换为关系的属性,而关系的码为各实体码的组合。
- (5) 三个以上实体间的一个多元联系可以转换为一个独立的关系模式,与该联系相连的各实体的码及联系本身的属性均转换为关系的属性,而关系的码为各实体码的组合。

1.4 SQL Server 2008 概述

1.4.1 随时随地管理数据

SQL Server 2008 在 Microsoft 的数据平台上发布,有助于用户随时随地管理任何数据。

它可以将结构化、半结构化和非结构化文档的数据(如图像和音乐)直接存储到数据库中。SQL Server 2008 提供一系列丰富的集成服务,可以对数据进行查询、搜索、同步、报告和分析等操作。数据可以存储在各种设备上,从数据中心最大的服务器一直到桌面计算机和移动设备,用户可以控制数据而不用管数据存储在哪儿。

SQL Server 2008 允许用户在使用 Microsoft .NET 和 Visual Studio 开发的自定义应用程序中使用数据,在面向服务的架构(SOA)和通过 Microsoft BizTalk Server 进行的业务流程中使用数据。信息工作人员可以通过日常使用的工具(如 Microsoft Office 2007 系统)直接访问数据。SQL Server 2008 提供一个可信的、高效率智能数据平台,可以满足用户的数据需求。

1.4.2 新功能

SQL Server 2008 为业务决策应用提供高级别的安全性、可靠性和可伸缩性。

1. 对信息的保护

1) 透明数据加密

透明数据加密允许对整个数据库、数据文件或日志文件加密而不需要变更应用。其优势如下:范围及模糊搜索均可用于搜索加密数据,保护数据免受非法用户搜索,不必对现有的应用程序作出任何变动就可实现数据加密。

2) 可扩展的密钥管理

SQL Server 2008 为加密和密钥管理提供了一个综合解决方案。针对这方面日益增长的需求,SQL Server 2008 提供了能够兼容第三方的密钥管理和硬件安全模块(HSM)产品。

3) 审计

SQL Server 2008 能够通过动态链接库来创建并管理审计,同时通过提供更全面的数据审计来简化审计跟踪。

2. 确保业务的持续性

1) 增强的数据库镜像

SQL Server 2008 构建于 SQL Server 2005 之上,并提供更加可靠的带增强数据库镜像功能的平台,包括自动页修复、性能提升及增强的可支持性。

2) 数据页的自动恢复

通过显式地将镜像成员请求置疑页的一个刷新副本给终端用户及应用,SQL Server 2008 允许代理及镜像机显式地从 823/824 类数据页错误中恢复。

3) 日志流压缩

数据库镜像需要在各镜像执行参与者间进行数据传输。在 SQL Server 2008 中,各参与者之间的输出日志流能够压缩达到性能的优化,以及数据库镜像对网络带宽的最小化占用。

3. 开启对响应的可预见性

1) 资源管理器

简要地说,资源管理器给终端用户提供一致的和可预见的回馈,它允许组织为不同的负载指定资源限额及优先级,可使并行负载能为终端用户提供一致的性能。

2) 可预见的查询执行

SQL Server 2008 通过提供函数锁定查询规划来强化查询执行的稳定性和可预见性,允许组织在硬件服务器替代、服务器升级及产品部署等场景创设稳健的查询规划。

3) 数据压缩

SQL Server 2008 数据压缩功能允许数据更有效地存储并降低其存储要求,这能显著改