



21世纪高职高专系列规划教材 · 计算机类专业



高职高专“十二五”规划教材

数据库技术与应用——

SQL Server 2005



SHUJUKU JISHU YU YINGYONG
SQL SERVER 2005

主 编◎陈 洁
副主编◎甄爱军 潘红改
马广慧



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

21世纪高职高专系列规划教材·计算机类专业
高职高专“十二五”规划教材

数据库技术与应用——

SQL Server 2005



SHUJUKU JISHU YU YINGYONG
SQL SERVER 2005

主 编◎陈 洁
副主编◎甄爱军
马广慧
参 编◎刘鲲鹏

潘红改



北京师范大学出版集团
BEIJING NORMAL UNIVERSITY PUBLISHING GROUP
北京师范大学出版社

图书在版编目(CIP)数据

数据库技术与应用——SQL Server 2005 / 陈洁主编. —北京:
北京师范大学出版社, 2013.1
21世纪高职高专系列规划教材
ISBN 978-7-303-15638-2

I. ①数… II. ①陈… III. ①关系数据库—数据库管理
系统—高等职业教育—教材 IV. ①TP311.138

中国版本图书馆CIP数据核字(2012)第269266号

营销中心电话 010-58802755 58800035
北师大出版社职业教育分社网 <http://zjfs.bnup.com.cn>
电子信箱 bsdzyjy@126.com

出版发行: 北京师范大学出版社 www.bnup.com.cn
北京新街口外大街19号
邮政编码: 100875

印刷: 北京中印联印务有限公司
经销: 全国新华书店
开本: 184 mm × 260 mm
印张: 17.5
字数: 330千字
版次: 2013年1月第1版
印次: 2013年1月第1次印刷
定价: 28.00元

策划编辑: 周光明	责任编辑: 周光明
美术编辑: 高霞	装帧设计: 国美嘉誉
责任校对: 李菡	责任印制: 孙文凯

版权所有 侵权必究

反盗版、侵权举报电话: 010-58800697

北京读者服务部电话: 010-58808104

外埠邮购电话: 010-58808083

本书如有印装质量问题, 请与印制管理部联系调换。

印制管理部电话: 010-58800825

前 言

数据库技术是计算机科学技术领域中应用最广泛、发展最迅速的技术，它已经成为计算机信息管理系统与应用系统的核心技术和重要基础。

SQL Server 2005 是微软公司的产品，由于它具有对数据集中进行管理、保证数据的安全性和完整性、可以优化系统性能等优点，在许多行业中得到广泛应用。

高等职业技术教育培养具有实际动手能力，面向生产第一线的应用型高级技术人才。在高职教学过程中，应注重学生职业岗位能力的培养，有针对性地进行职业技能的训练，以及学生解决问题的能力培养。为了加强学生动手能力，打破传统学科体系结构，突出高等职业教育的特色，本书按照高职示范校精神(教高[2006]16号文件)，结合国家首批高职示范校建设成果，以工作过程为导向，采用学习情境的形式组织教学内容，以任务驱动的教学方式进行编写，将数据库技术与应用技能和知识点融入到学生管理系统数据库实际工作任务之中。全书共5个学习情境，包括关系数据库的设计、创建和管理数据库与数据表、数据的操纵和完整性、数据库的安全管理与维护以及 T-SQL 语言及其应用。

本教材具有理论与实践一体化、教学过程与实际工作流程一致、突出实践能力培养等特点。根据高职教育的特点，对需求分析和数据库设计工作进行裁减，将教学重点放在创建和管理数据库与数据表，实现数据完整性，数据的增加、删除、修改和查询，创建视图，编写存储过程和触发器，游标使用，事务处理与并发控制，数据库安全性管理与维护等数据库技术应用上。通过学习，学生可以承担数据库开发工程师、数据库管理员等工作，并且为胜任软件开发等工作岗位奠定基础。

本书由天津职业大学陈洁统稿。情境1、情境2、情境3的任务4、情境5的任务5和任务6由天津职业大学陈洁编写；情境3的任务1、任务2、任务3，情境4的任务4由河南漯河职业技术学院潘红改、北京师范大学刘鲲鹏编写；情境4的任务1、任务2、任务3由天津农学院职业技术学院马广慧编写；情境5的任务1、任务2、任务3和任务4由天津农学院职业技术学院甄爱军编写。陈洁担任主编，甄爱军、潘红改、马广慧担任副主编。

由于作者水平、时间、精力所限，本书难免存在不足之处，恳请广大读者指正。
编者 E-mail: xchenjie@126. com。

编者
2012年12月

目 录

学习情境 1 关系数据库的设计	(1)	操作与练习	(158)
任务 1 认识数据库的基本概念	(1)	学习情境 4 数据库的安全管理与	
任务 2 认识关系数据库	(6)	维护	(160)
任务 3 设计关系数据库	(12)	任务 1 登录和用户管理	(160)
操作与练习	(22)	任务 2 权限和角色管理	(167)
学习情境 2 创建和管理数据库与		任务 3 数据库的备份与还原	(176)
数据表	(23)	任务 4 数据的导入与导出	(183)
任务 1 安装 SQL Server 2005	(23)	操作与练习	(193)
任务 2 创建和管理数据库	(46)	学习情境 5 T-SQL 语言及其应用	
任务 3 创建和管理数据表	(60)		(194)
任务 4 实现数据的完整性	(68)	任务 1 流程控制语句应用	(194)
操作与练习	(90)	任务 2 SQL server 函数的应用	
学习情境 3 数据的操纵和完整性			(204)
维护	(93)	任务 3 游标的使用	(218)
任务 1 数据的增加、修改和删除		任务 4 编写、应用与管理存储	
.....	(93)	过程	(224)
任务 2 数据的简单查询	(97)	任务 5 事务处理与并发控制	
任务 3 数据的高级查询	(120)		(238)
任务 4 创建和管理视图	(136)	任务 6 编写、应用与管理触发器	
任务 5 建立索引提高检索效率			(242)
.....	(145)	操作与练习	(269)
		参考文献	(271)

学习情境 1 关系数据库的设计

学习目标

1. 认识数据库的基本概念，数据库系统的组成和数据库系统的体系结构。
2. 认识关系数据库，了解关系模型、关系数据库标准语言 SQL 和关系数据库理论。
3. 关系数据库设计的内容和步骤，需求分析、概念结构设计和逻辑结构设计的方法。

技能要求

1. 关系数据库设计的内容和步骤。
2. 需求分析、概念结构设计和逻辑结构设计的方法。

任务 1 认识数据库的基本概念

情境描述

数据库技术研究如何科学地组织和存储数据，如何高效地获取和处理数据。本任务认识数据库的基本概念。主要内容有数据、信息、数据库管理系统、数据库系统等基本概念；数据库技术的产生和发展；数据库系统的组成；数据库系统的体系结构；概念模型与数据模型。

相关知识与技能点

1. 数据、信息、数据库管理系统、数据库系统等基本概念。
2. 数据库技术的产生和发展。
3. 数据库系统的组成和数据库系统的体系结构。
4. 概念模型与数据模型表示方法。

1.1 数据与信息、数据库、数据库管理系统、数据库系统

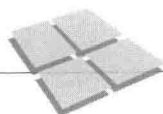
1. 数据(Data)

数据是数据库中存储的基本对象，是描述事物的符号记录。描述事物的符号可以是数字，也可以是文字、图表、图像、声音等。人们通过数据来认识世界，交流信息。

2. 信息(Information)

数据经过加工处理后所获得的有用的知识称为信息。信息是以某种数据形式表现的。

数据和信息是既相互联系又相互区别的概念。数据是信息的具体表现形式，信息



是数据有意义的表现。

3. 数据处理(Data Handle)

数据处理指对各种形式的数据进行分类、组织、编码、存储、检索和维护的一系列活动。

4. 数据库(Database, DB)

顾名思义,是存放数据的仓库。数据库是长期存储在计算机内、有组织的、统一管理的相关数据的集合。数据库中的数据是长期存储在计算机内按一定的数据模型组织、描述和存储,具有较小的冗余度,数据间联系紧密又有较高的数据独立性和易扩展性,并可为各种用户共享。

5. 数据库管理系统(Database Management System, DBMS)

数据库管理系统指位于用户与操作系统之间的数据管理软件。如 SQL Server 2005 就是一个 DBMS。数据库在建立、运用和维护时由数据库管理系统统一管理、统一控制。数据库管理系统使用户能方便地定义数据和操纵数据,并能够保证数据的安全性、完整性、多用户对数据的并发使用及发生故障后的系统恢复。

6. 数据库系统(Database System, DBS)

数据库系统指在计算机系统中引入数据库后构成的系统,一般由数据库、数据库管理系统(及其开发工具)、应用系统、数据库管理员和用户构成。

1.2 数据库技术的产生与发展

随着计算机硬件和软件不断发展,数据处理经历了 3 个阶段:人工管理阶段、文件系统阶段和数据库系统阶段。

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。从硬件看,外存只有纸带、卡片、磁带,没有直接存取设备;从软件看,没有操作系统及管理数据的软件。因此称这个阶段的数据管理方式为人工管理阶段。这个阶段的特点是:

- (1)数据不保存;
- (2)应用程序管理数据,数据与程序结合在一起;
- (3)数据不共享,数据是面向应用的,一个程序对应一组数据;
- (4)数据不具有独立性。

2. 文件系统阶段

20 世纪 50 年代后期到 60 年代中期,硬件有了磁鼓、磁盘等直接存存储设备;软件有操作系统,操作系统中已经有了专门的数据管理软件——文件系统。这个阶段的特点是:

- (1)数据以文件的形式长期保存在计算机里;
- (2)操作系统的文件管理提供了对数据的输入和输出管理;
- (3)数据可以共享,一个数据文件可以被多个应用程序使用;
- (4)数据文件之间彼此独立,不能反映数据之间的联系,数据大量冗余。

3. 数据库系统阶段

数据库系统阶段从 20 世纪 60 年代后期开始。随着计算机系统性价比的持续提高,软件技术的不断发展,为了解决多用户、多应用共享数据的要求,使数据为尽可能多

的应用服务,人们克服了文件系统的不足,开发了对数据进行统一管理和控制的数据库管理系统(Database Management System, DBMS)。在计算机领域逐步形成了数据库技术这个独立的分支。数据库系统阶段的特点是:

- (1)数据结构化;
- (2)数据的共享性高,冗余度低,容易扩充;
- (3)数据独立性高;
- (4)数据由 DBMS 统一管理和控制。

1.3 数据库系统的组成

1. 硬件平台及数据库

(1)要有足够大的内存,存放操作系统和 DBMS 的核心模块、数据库缓冲区和应用程序;

(2)有足够大的磁盘等直接存取设备存放数据库数据,有足够的光盘、磁盘、磁带等作为数据备份介质;

- (3)要求连接系统的网络有较高的数据传送率;
- (4)有较强处理能力的中央处理器来保证数据处理的速度。

2. 软件

- (1)数据库管理系统 DBMS,为数据库的建立、使用和维护配置的软件;
- (2)支持 DBMS 运行的操作系统;
- (3)与数据库通信的高级程序语言及编译系统;
- (4)为特定应用环境开发的数据库应用系统。

3. 数据库管理员及其他相关人员

(1)数据库管理员(Database Administrator, DBA)。数据库管理员负责管理和监控数据库系统,负责为用户解决应用中出现的系统问题。为了保证数据库能够高效正常运行,大型数据库系统都设有专人负责数据库系统得管理和维护。

(2)系统分析员和数据库设计人员。系统分析员负责应用系统的需求分析和规范说明,与用户及 DBA 配合,确定系统的硬件、软件配置,并参与数据库系统概要设计。

(3)应用程序员。应用程序员是负责设计、开发应用系统功能模块的软件编程人员,根据数据库结构编写特定的应用程序,并进行调试和安装。

(4)用户。这里的用户是指最终用户。最终用户通过应用程序的用户接口使用数据库。常用的接口方式有浏览器、菜单驱动、表格操作、图形显示和报表等。

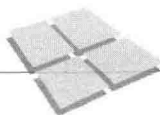
1.4 数据库系统的体系结构

数据库系统的结构可以从不同的层次或角度来考察。

从数据库管理系统角度看,数据库系统通常分为三级模式,这是数据库管理系统内部的体系结构。

从数据库最终用户角度看,数据库系统的结构分为单用户结构、主从式结构、分布式结构、客户/服务器结构等。这是数据库系统外部的体系结构。

尽管实际数据库管理系统种类繁多,支持不同的数据模型,使用的数据库语言各异,建立在不同的操作系统之上,采用的数据存储结构也各不相同,但是体系结构上都采用三级模式结构并提供两层映像功能。



1. 数据库系统的三级模式结构

数据库系统的三级模式结构是指数据库系统是由外模式、模式和内模式三级组成。

(1)外模式。外模式也称子模式或用户模式，它是数据库用户(包括应用程序员和最终用户)看见和使用的局部数据的逻辑结构和特征的描述，是数据库用户的数据视图，是与某一应用有关的数据的逻辑表示。一个数据库可以有多个外模式。

(2)模式。模式也称逻辑模式，是数据库中全体数据的逻辑结构和特征的描述，是所有用户的公共数据视图。一个数据库只有一个模式。

(3)内模式。内模式也称存储模式，它是数据物理和存储结构的描述，是数据在数据库内部的表示方式。一个数据库只有一个内模式。

2. 数据库的两层级映像功能与数据独立性

数据库系统在这三级模式之间提供了两层映像：外模式/模式映像和模式/内模式映像。正是这两层映像保证了数据库系统的数据能够具有较高的逻辑独立性和物理独立性。

(1)外模式/模式的映像及作用。

模式描述的是数据的全局逻辑结构，外模式描述的是数据的局部逻辑结构。对应于同一个模式可以有任意多个外模式。对于每一个外模式，数据库系统都有一个外模式/模式映像，它定义了该外模式与模式之间的对应关系。当模式改变时(例如，增加新的数据类型、新的数据项、新的关系等)，由数据库管理员对各个外模式/模式的映像作相应改变，可以使外模式保持不变，从而应用程序不必修改，保证了数据的逻辑独立性。

(2)模式/内模式的映像及作用。

数据库中只有一个模式，也只有一个内模式，所以模式/内模式映像 is 唯一的，它定义了数据全局逻辑结构与存储结构之间的对应关系。当数据库的存储结构改变时(例如，采用了更先进的存储结构)，由数据库管理员对模式/内模式映像作相应改变，可以使模式保持不变，从而保证了数据的物理独立性。

(3)数据库中的数据独立性。

数据的物理独立性是指应用程序对数据存储结构(也称物理结构)的依赖程度。数据物理独立性高是指当数据的物理结构发生变化时，应用程序不需要修改也可以正常工作。

数据库中的数据逻辑结构分全局逻辑结构和局部逻辑结构两种：数据全局逻辑结构指全系统总体的数据逻辑结构，它是按全系统使用的数据、数据的属性及数据联系来的组织的。数据局部逻辑结构是指一个用户或程序使用的数据逻辑结构，它是根据用户自己对数据的需求进行组织的。局部逻辑结构仅涉及与该用户(或程序)相关的数据结构。数据局部逻辑结构与全局逻辑结构之间是不完全统一的，两者间可能会有较大的差异。

数据的逻辑独立性是指应用程序对数据全局逻辑结构的依赖程度。数据逻辑独立性高是指当数据库系统的数据全局逻辑结构改变时，它们对应的应用程序不需要改变仍可以正常运行。当新增加一些数据和联系时，不影响某些局部逻辑结构的性质。

1.5 概念模型

根据模型应用目的不同,可以将模型分为两类:一类是概念模型,它是按用户的观点对数据和信息进行抽象。另一类是数据模型,它是按计算机系统的观点对数据建模,主要用于 DBMS 的实现。概念模型是现实世界到信息世界的第一次抽象,是数据库设计人员与用户之间交流的语言。现实世界的事物反映到人的脑子中来,人们把这些事物抽象为一种既不依赖于具体的计算机系统又不为某一 DBMS 支持的概念模型,然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型。

1. 基本概念

(1)实体(Entity):客观存在并相互区别的事物及其事物之间的联系。例如,一个学生、一门课程、学生的一次选课等都是实体。

(2)属性(Attribute):实体所具有的某一特性。例如,学生的学号、姓名、性别、出生年份、系、入学时间等。

(3)码(Key):唯一标志实体的属性集。例如,学号是学生实体的码。

(4)域(Domain):属性的取值范围。例如,年龄的域为大于 15 小于 35 的整数,性别的域为(男,女)。

(5)实体型(Entity Type):用实体名及其属性名集合来抽象和刻画同类实体,称为实体型。例如,学生(学号,姓名,性别,出生年份,系,入学时间)就是一个实体型。

(6)实体集(Entity Set):同型实体的集合称为实体集。例如,全体学生就是一个实体集。

(7)联系(Relationship):实体与实体之间以及实体与组成它的各属性间的关系。

联系有三种情况:一对一联系,一对多联系,多对多联系。

①一对一联系(1:1)。实体集 A 中的一个实体至多与实体集 B 中的一个实体相对应,反之亦然,则实体集 A 与实体集 B 为一对一的联系。记作 1:1。如:学校与校长,观众与坐席。

②一对多联系(1:n)。实体集 A 中的一个实体与实体集 B 中的多个实体相对应,实体集 B 中的一个实体至多与实体集 A 中的一个实体相对应。记作 1:n。如,学校与教师,班级与学生。

③多对多联系(m:n)。实体集 A 中的一个实体与实体集 B 中的多个实体相对应,实体集 B 中的一个实体与实体集 A 中的多个实体相对应。记作 m:n。如:教师与学生、学生与课程。

2. 概念模型的表示方法

概念模型的表示方法很多,最常用的是实体——联系方法(Entity-Relationship)。该方法用 E-R 图来描述现实世界的概念模型。E-R 图提供了表示实体、属性和联系的方法。

(1)实体:用矩形表示,矩形框内写明实体名。

(2)属性:用椭圆形表示,并用无向边将其与相应的实体连接起来。

(3)联系:用菱形表示,菱形框内写明联系名,并用无向边分别与有关实体连接起来,同时在无向边旁标上联系的类型(1:1, 1:n 或 m:n)。

如图 1.1 所示就是一个班级的概念模型的 E-R 图。

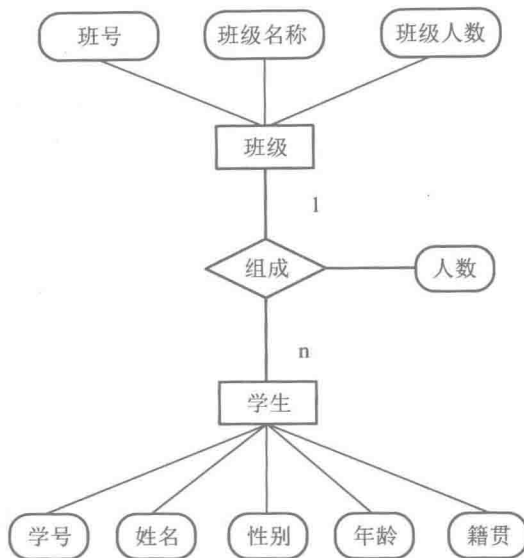
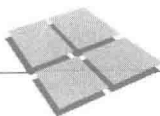


图 1.1 班级的 E-R 图

1.6 数据模型

数据模型是描述数据库中数据的逻辑结构，它是用于机器世界的第二次抽象。数据模型通常由 3 部分组成，也称为数据模型的三大要素，分别是数据结构、数据操纵和完整性约束。

目前，常用的数据模型有四种，它们是：

- (1) 层次模型(Hierarchical Model)，用“树结构”表示数据之间的联系；
- (2) 网状模型(Network Model)，用“图结构”表示数据之间的联系；
- (3) 关系模型(Relational Model)，用“二维表”表示数据之间联系；
- (4) 面向对象模型(Object Oriented Model)，用对象表示数据之间的联系。

关系模型是目前最重要的一种数据模型。关系数据库系统采用关系模型为数据的组织方式，SQL Server 2005 数据库就是基于关系模型建立的。关系模型具有如下优点：

- (1) 关系模型是建立在严格的数学概念基础上的；
- (2) 关系模型的概念单一，无论是实体还是实体之间的联系都用关系表示，对数据的检索结果也是关系；
- (3) 关系模型的存取路径对用户透明。

任务 2 认识关系数据库

情境描述(任务背景、主要内容等)

关系数据库是当前信息管理系统中最常用的数据库，关系数据库采用关系模型，应用关系代数的方法来处理数据库中的数据。本任务介绍关系模型、关系数据库标准语言 SQL 和关系数据库理论。

相关知识与技能点

1. 关系模型的基本术语、关系模型的数据结构、关系操作和关系模型的数据完整性。
2. 关系数据库语言 SQL(又称结构化查询语言 SQL)的基本知识,包括数据定义、数据操作和数据控制功能。

3. 数据依赖与关系数据库规范化理论。

2.1 关系模型

关系模型由数据结构、关系操作和关系的完整性三个部分组成。

1. 关系模型的基本术语

一个关系模型的逻辑结构是一张二维表,它由行和列组成。例如,表 1.1 中的学生记录就是一个关系模型,它涉及下列概念。

(1)关系(Relation):一个关系对应一张二维表,二维表名就是关系名。表 1.1 中的这张学生记录表就是一个关系。

(2)元组(Tuple):二维表中的一行,即每一条记录的值称为一个元组。如表 1.1 有 20 行,就有 20 个元组。

(3)属性(Attribute):二维表中的列(字段)称为属性,如表 1.1 有 5 列,对应 5 个属性(学号、姓名、性别、年龄和所在系)。

(4)域(domain):属性的取值范围,如表 1.1 中学生年龄的域应是(16~28),性别的域是(男,女),系别的域是一个学校所有系名的集合。

(5)键或码(key):由一个或多个属性组成。

(6)候选键(Candidate Key):如果一个关系中存在多个属性,都能用来唯一标志该关系的元组,则这些属性称为该关系的候选键或候选码。

(7)主键(Primary Key):在一个关系中的若干候选码中指定为关键字的属性称为该关系的主键或主码。

(8)外键(Foreign key):如果关系中的某个属性不是该关系的主键,但却是另一个关系的主键,称该属性为这个关系的外键。

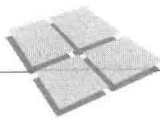
(9)关系模式(Relation Schema):对关系的描述,一般表示为:

关系名(属性 1,属性 2,...,属性 n)

例如,上面的关系可描述为:学生(学号,姓名,性别,年龄,所在系)

表 1.1 学生记录表

学号	姓名	性别	年龄	所在系
0000101	王萧	男	17	经济系
000207	李云虎	男	18	机械系
010302	郭敏	女	18	信息系
010408	高红	女	20	土木系
...	...	...	...	...
020309	王睿	男	19	信息系
020506	路旭青	女	21	管理系



2. 关系模型的数据结构

关系模型的数据结构是一种二维表格结构。在关系模型中,现实世界的实体与实体之间的联系均用二维表格表示,如表 1.1 所示。

3. 关系操作

关系模型中给出了关系操作的能力与特点。关系操作的特点是集合操作,即操作的对象和结果都是集合。关系操作的能力有选择操作(Select)、投影(Project)、连接(Join)、除(Divide)、并(Union)、交(Intersection)、差(Difference)、查询(Query)和插入(Insert)、删除>Delete)、修改(Update)操作。

关系操作的能力可以用关系代数来表示。关系代数是一种抽象的查询语言,这些抽象的语言与具体的 DBMS 中实现语言并不完全一致。

4. 关系模型的数据完整性

关系模型的完整性规则是对关系的某种约束条件。关系模型中有三类完整性约束:实体完整性、参照完整性和用户定义的完整性。关系数据库提供了对实体完整性和参照完整性约束的自动支持,也就是在插入、修改、删除操作时,数据库系统自动保证数据的正确性与一致性。

(1) 实体完整性规则(Entity Integrity Rule)。

关系的实体完整性规则为:若属性 A 是基本关系 R 的主属性,则属性 A 的值不能为空值。例如,学生表中的学号属性不能为空。学生的其他属性可以是空值,若“年龄”值或“性别”值如果为空,则表明不清楚该学生的这些特征值。

(2) 关系模型的参照完整性(Reference Integrity Rule)。

设 F 是基本关系 R 的一个或一组属性,但不是关系 R 的主码(或候选码)。如果 F 与基本关系 S 的主码 K_s 相对应,则称 F 是基本关系 R 的外码(Foreign Key),并称基本关系 R 为参照关系(Referencing Relation),基本关系 S 为被参照关系(Referenced Relation)或目标关系(Target Relation)。

参照完整性规则定义了外码与主码之间的引用规则。

若属性(或属性组)F 是基本关系 R 的外码,它与基本关系 S 的主码 K_s 相对应(基本关系 R 和 S 不一定是不同的关系),则对于 R 中每个元组在 F 上的值必须取空值(F 的每个属性值均为空值)或者等于 S 中某个元组的主码值。

例如,在关系数据库 student 中有两个关系模式:

S(学号,姓名,性别,年龄,班级号,系别),PK(学号)

SC(学号、课程号、成绩),PK(学号,课程号),FK1(学号),FK2(课程号)

根据规则要求,关系 SC 中的“学号”值应该在关系 S 中出现。如果关系 SC 中有一个元组(S07, C02, 86)。而学号 S07 却在关系 S 中找不到,那么就认为在关系 SC 中引用了一个不存在的学生实体,这就违反了参照完整性规则。另外,在关系 SC 中“学号”不仅是外键,也是主键的一个部分,因此这里的“学号”值不允许空。

(3) 用户定义的完整性。

实体完整性和参照完整性适用于任何关系数据库系统,除此之外,不同的关系数据库系统根据其应用环境的不同,往往还需要一些特殊的约束条件。

用户定义的完整性就是针对某一具体关系数据库的约束条件,它反映某一具体应

用所涉及的数据必须满足的语义要求。

例如,表 1.1 中学生的年龄定义为两位整数,但是范围仍然太大,为此用户可以写出如下规则,把年龄限制在 16~28 岁。

CHECK (AGE BETWEEN 16 AND 28)

2.2 关系数据库语言 SQL

关系数据库是采用关系模型作为数据组织方式的数据库,它建立在严格的数学理论基础。1970 年 6 月,IBM 公司研究员 E. F. Codd 发表了题为《大型共享数据库的关系模型》论文,首先概述了关系数据模型及其原理,并把它用于数据库系统中。1981 年,IBM 公司开发出数据库产品 SQL/DS,同时,美国加州大学与 INGRES 公司研制了 INGRES 数据库产品,使关系方法从实验走向市场。

目前关系数据库系统的研究取得了辉煌的成就,涌现出许多良好的商品化关系数据库管理系统,著名的有 DB2、Oracle、Sybase、Informix、SQL Server 等。关系数据库被广泛应用于各个领域,成为主流数据库。

关系数据库语言 SQL(Structured Query Language),又称为结构化查询语言,是关系数据库管理系统中最流行的数据查询和操作语言,用户可以使用 SQL 语言对数据库执行各种操作,包括数据定义、数据操作和数据控制等与数据库有关的全部功能。

SQL 语言是在 1974 年由美国 IBM 公司的 San Jose 研究所中的科研人员 Boyce 和 Chamberlin 提出的,并于 1975 年至 1979 年在关系数据库管理系统原型 System R 上实现了这种语言。1986 年 10 月,美国国家标准局(American National Standards Institute, ANSI)的数据库委员会批准了 SQL 作为关系数据库语言的美国标准,同年公布了 SQL 标准文本 SQL-86。1987 年国际标准化组织(International Standards Organization, ISO)将其采纳为国际标准。

1. 数据定义功能

SQL 的数据定义功能通过数据定义语言(Date Definition Language, DDL)实现。它用来定义数据库的逻辑结构,包括基本表,视图和索引。基本的 DDL 包括 3 类,即定义、修改和删除。

(1)SQL 定义基本表的语句。

CREATE TABLE 创建表

DROP TABLE 删除表

ALTER TABLE 修改表

(2)SQL 定义视图的语句。

CREATE VIEW 创建视图

DROP VIEW 删除视图

(3)SQL 定义索引的语句。

CREATE INDEX 创建索引

DROP INDEX 删除索引

2. 数据操纵功能

SQL 的数据操纵功能通过数据操纵语言(Data Manipulation Language, DML)实现。它包括数据查询和数据更新两大类操作。SQL 的数据查询主要通过 SELECT 语句



来实现。SQL 的数据更新包 INSERT、DELETE 和 UPDATE 三条语句。

3. 数据控制功能

数据库的控制是指数据的安全和完整性控制。SQL 的数据控制功能通过数据控制语言(Data Control Language, DCL)实现。SQL 通过授权语句 GRANT 和回收语句 REVOKE 来实现数据控制功能。

SQL 通过对数据库用户的授权和取消授权命令来实现相关数据的存取控制,以保证数据库的安全性。另外还提供了数据完整性约束条件的定义和检查机制,来保证数据库的完整性。

关于 SQL 语言的具体操作将在后续内容中陆续介绍。

2.3 关系数据库规范化理论

1. 数据依赖

现实世界随着时间在不断地变化,从现实世界中经过抽象而得到的关系模式的关系也会有所变化。但是,现实世界的许多事实限定关系模式的关系必须满足一定的完整性约束条件。这些约束条件通过对属性取值范围的限定反映出来,称为依赖于值域元素语义的限制。例如,学生出生于 1988 年而入学时间也为 1988 年,这显然是不合理的。这些约束条件通过对属性值之间的相互关联反映出来,这类限制统称为数据依赖,其中最重要的是函数依赖和多值依赖。

(1) 函数依赖。

函数依赖普遍存在于现实生活中。例如,描述一名学生的关系(学号、姓名、系名),由于一个学号只对应一名学生,一个学生只在一个系里学习,因而,当学号值确定之后,姓名和该学生所在的系名的值也就唯一地确定了,我们称“学号”函数决定“姓名”和“系名”函数,或者说“姓名”和“系名”函数依赖于“学号”函数。记为:学号 $\rightarrow$ 姓名,学号 $\rightarrow$ 系名。

(2) 多值依赖。

例如,学校中的某一门课程由多个教师讲授,他们使用同一本参考书,每个教师可以讲授多门课程。当某一门课程要去掉一本参考书时,必须删除多个元组。这种情况称为多值依赖。

2. 关系模式的规范化

针对一个具体的数据库应用问题,应该构造几个关系模式,每个关系由哪些属性组成,即如何构造适合于它的数据模式,这是数据库逻辑设计的问题。为了使数据库设计的方法趋于规范,数据库专家研究了关系规范化理论。从 1971 年起, E. F. Codd 相继提出了第一范式、第二范式、第三范式, Codd 与 Boyce 合作提出了 Boyce-Codd 范式。在 1976—1978 年间, Fagin、Delobe 以及 Zaniolo 又定义了第四范式。到目前为止,已经提出了第五范式。

所谓范式(Normal Form)是指规范化的关系模式。由于规范化的程度不同,产生了不同的范式。满足最基本规范化的关系模式称为第一范式(1NF),第一范式的关系模式再满足另外一些约束条件就产生了第二范式(2NF)、第三范式(3NF)、BC 范式等。每种范式都规定了一些限制约束条件。

一个低一级的关系范式模式通过模式分解转换为若干高一级范式的关系模式集合

的过程,称为关系模式的规范化(Normalization)。

(1)第一范式(1NF)。

若一个关系模式 R 的所有属性都是不可分的基本数据项,则该关系属于第一范式。

在任何一个关系数据库系统中,第一范式是对关系模式的一个必须的要求,不满足第一范式的数据库模式不能称为关系数据库。

满足第一范式的关系模式并不一定是好的关系模式。

例如,学生(学号,姓名,所在系,系主任姓名,课程名,成绩)

它显然满足第一范式,但是本身存在插入异常、删除异常、更新异常和数据冗余问题,所以它不是一个好的关系模式。

(2)第二范式(2NF)。

若关系模式 R 属于第一范式,且每个非主属性都完全函数依赖于主关键字,则 R 属于第二范式。

第二范式不允许关系模式中的非主属性部分函数依赖于主关键字。

例如,学生(学号,姓名,所在系,系主任姓名,课程名,成绩)

学生关系模式存在部分依赖:

(学号,课程名)→姓名

(学号,课程名)→所在系

(学号,课程名)→系主任姓名

不满足“每个非主属性都完全函数依赖于主关键字”的条件,故不属于第二范式。

对学生关系模式进行分解,使其满足第二范式的条件,即要消除非主属性对主关键字的部分依赖。

学生关系模式分解成:

学生一系(学号,姓名,所在系,系主任姓名)

考试(学号,课程名,成绩)

学生一系、考试属于第二范式。

学生关系模式:学生一系(学号,姓名,所在系,系主任姓名)

中存在:

学号→所在系

所在系→系主任姓名

学号→系主任姓名

仍有数据冗余,从学生一系关系模式的实例可以看出,当一个系有很多学生时,“系主任姓名”会大量重复出现和更新异常。所以应进一步对其进行规范化。

(3)第三范式(3NF)。

若关系模式 R 属于第一范式,且每个非主属性都不传递依赖于主关键字,则 R 属于第三范式。

将学生一系(学号,姓名,所在系,系主任姓名)

关系模式分解为:

学生(学号,姓名,所在系)

系(所在系,系主任姓名)



关系模式学生与系均已满足 3NF。

第三范式是一个可用的关系模式应满足的最低范式。一个关系模式如果不满足 3NF，实际上是不能使用的。

把关系模式分解到 3NF，可以减轻原关系中的异常和信息冗余，但也不能保证完全消除关系模式中的各种异常和信息冗余。

► 任务 3 设计关系数据库

情境描述(任务背景、主要内容等)

数据库设计是指在给定的环境下，创建一个性能良好，能满足不同用户使用要求，又能被选定的 DBMS 所接受的数据模式。本任务了解数据库设计的内容、方法和步骤，进行需求分析、概念结构设计和逻辑结构设计的方法。

相关知识与技能点

1. 数据库设计的步骤。
2. 需求分析的任务和阶段性成果。
3. 概念结构设计的方法和步骤。
4. 逻辑结构设计的步骤，E-R 模型向关系数据库的转换规则。
5. 数据库的物理设计、数据库的实施、数据库的运行与维护。

3.1 数据库设计的内容

数据库系统设计的内容主要有：数据库的结构特性设计，数据库的行为特性设计，数据库的物理模式设计。

(1) 数据库的结构特性设计。

先将现实世界中的事物、事物间的联系用 E-R 图表示，再将各个分 E-R 图汇总，得出数据库的概念结构模型，最后将概念结构模型转化为数据库的逻辑结构模型表示。

(2) 数据库的行为特性设计。

首先要将现实世界中的数据及应用情况用数据流程图和数据字典表示，并详细描述其中的数据操作要求(即操作对象、方法、频度和实时性要求)，进而得出系统的功能模块结构和数据库的子模式。

(3) 数据库的物理模式设计。

根据数据库结构的动态特性(即数据库应用处理要求)，在选定的 DBMS 环境下，把数据库的逻辑结构模型加以物理实现，从而得出数据库的存储模式和存取方法。

3.2 数据库设计的步骤

人们不断地研究探索，提出了各种数据库的规范设计方法，其中比较著名的是新奥尔良法(New Orleans)。按照规范设计方法，数据库系统设计应分 6 个阶段进行。这 6 个阶段是：需求分析阶段、概念结构设计阶段、逻辑结构设计阶段、物理结构设计阶段、数据库实施阶段、数据库运行和维护阶段，如图 1.2 所示。