

21世纪高等教育规划教材

数据库技术

——SQL Server 2000

贺桂英 王杰 李雪婵 编著



西南交通大学出版社
[Http://press.swjtu.edu.cn](http://press.swjtu.edu.cn)

21 世纪高等教育规划教材

数据库技术

—SQL Server 2000

贺桂英 王 杰 李雪婵 编著

西南交通大学出版社

• 成 都 •

图书在版编目(CIP)数据

数据库技术:SQL Server 2000/贺桂英,王杰,李雪婵编著.

—成都:西南交通大学出版社,2007.8

21 世纪高等教育规划教材

ISBN 978-7-81104-713-4

I. 数… II. ①贺…②王…③李… III. 关系数据库—数据库管理系统,SQL Server 2000 IV. TP311.138

中国版本图书馆 CIP 数据核字(2007)第 131902 号

21 Shiji Gaodeng Jiaoyu Guihua Jiaocai

21 世纪高等教育规划教材

Shujuku Jishu——SQL Server 2000

数据库技术——SQL Server 2000

贺桂英 王 杰 李雪婵 编著

*

责任编辑 张华敏

特邀编辑 高青松 陈旭文

封面设计 水木时代

西南交通大学出版社出版发行

(成都市二环路北一段 111 号 邮政编码:610031 发行部电话:028-87600564)

<http://press.swjtu.edu.cn>

北京广达印刷有限公司印刷

*

成品尺寸:185 mm×260 mm 印张:13.75

字数:365 千字

2007 年 8 月第 1 版 2007 年 8 月第 1 次印刷

ISBN 978-7-81104-713-4

定价:23.80 元

版权所有 盗版必究 举报电话:028-87600562

前 言

随着计算机技术的发展,特别是互联网技术的飞速发展,作为互联网技术的一个重要支持技术,数据库已成为信息社会的重要基础技术之一。掌握数据库技术及开发数据库应用系统成为当前最热门的计算机职业技能之一,正因为数据库应用及开发的专业人才的需求量非常大,所以学习和掌握数据库技术的相关技能和知识是非常必要的。

SQL Server 2000 是 Microsoft 公司推出的大型分布式客户/服务器结构的关系型数据库管理系统,它功能强大、操作简便,是目前最为流行的数据库管理系统之一。正因为其优势突出,SQL Server 得到了越来越多的服务器和客户端开发工具的支持,如 ASP.NET、Visual Basic.NET、C#、Visual C++ 和 Delphi 等都为其提供了接口,因此,了解和掌握 SQL Server 对于学习数据库技术的人员来说是非常实用的。

本教材的编写注重实践、兼顾理论,通过讲授和实训两条主线来安排课程内容,旨在使读者能通过讲解的实例和实训操作内容两个方面来掌握 SQL Server 2000 的管理技术及相关数据库理论和开发知识。

本书第 1 章介绍关系数据库系统的基础知识及数据库设计和规范化基本理论。第 2 章介绍 SQL Server 2000 安装的基本过程,安装后的主要组件介绍和服务器注册等内容。第 3 章介绍通过企业管理器建立与管理用户数据库的基本方法,以及 SQL Server 2000 数据库的体系结构和 4 个系统数据的作用。第 4 章介绍 SQL Server 2000 数据库中表的一些基本概念、数据完整性概念及其应用,并介绍了在企业管理器中建立用户表、修改用户表的结构、向表中插入、删除和修改数据以及表的删除操作。第 5 章介绍索引和视图的概念及建立和使用方法。第 6 章介绍 Transact-SQL 的基本语句,如数据库、表的建立语句,数据的插入、删除和修改命令、查询语句的使用等。第 7 章介绍 Transact-SQL 的编程基础,各种流程控制语句及 SQL Server 数据库游标的使用。第 8 章介绍存储过程的创建、修改、执行和删除操作,以及另一种特殊类型的存储过程触发器的创建和使用操作。第 9 章介绍有关数据库的备份和恢复、数据的导入与导出及系统的安全管理等方面的内容。第 10 章介绍数据库应用开发的基本知识,最后通过一个 Visual Basic 实例说明数据库应用项目开发的基本步骤和方法。

本书可作为高等院校计算机及相关专业的数据库技术教材或教学参考书,也可作为从事计算机应用开发的工程技术人员学习数据库技术的参考书。本书的编写参考了本专业的部分资料和文献,在此向所有原作者表示衷心的感谢!

由于作者水平有限,书中不足之处在所难免,敬请专家和读者批评指正。

作 者

2007 年 7 月

目 录

第 1 章 关系数据库基础	(1)
1.1 基本概念	(1)
1.2 数据库系统的体系结构	(3)
1.3 关系数据库系统的基本概念	(5)
1.4 关系代数	(8)
1.5 数据库设计	(12)
1.6 关系规范化的基本思想	(16)
本章小结	(19)
习 题	(20)
第 2 章 SQL Server 2000 的安装与配置	(22)
2.1 安装前的准备工作	(22)
2.2 安装 SQL Server 2000	(23)
2.3 SQL Server 2000 的服务器管理	(28)
2.4 客户机上的主要管理工具	(30)
2.5 注册 SQL Server 服务器	(33)
本章小结	(35)
习 题	(35)
第 3 章 SQL Server 2000 数据库的建立与管理	(36)
3.1 SQL Server 2000 数据库的体系结构	(36)
3.2 SQL Server 2000 的系统数据库	(38)
3.3 创建用户数据库	(39)
3.4 配置用户数据库	(42)
3.5 分离和附加用户数据库	(44)
3.6 删除用户数据库	(45)
本章小结	(46)
习 题	(47)
第 4 章 SQL Server 2000 表的建立与管理	(48)
4.1 SQL Server 2000 的数据类型	(48)
4.2 用户表与系统表	(52)
4.3 创建用户表	(54)
4.4 数据完整性基本概念	(57)
4.5 修改用户表结构	(65)
4.6 用户表中插入、修改与删除数据	(68)
4.7 删除用户表	(70)
本章小结	(71)

习 题	(71)
第 5 章 索引与视图的建立与管理	(73)
5.1 概 述	(73)
5.2 创建索引	(75)
5.3 修改索引	(78)
5.4 删除索引	(79)
5.5 视 图	(79)
本章小结	(85)
习 题	(85)
第 6 章 Transact-SQL 使用基础	(87)
6.1 SQL 语言	(87)
6.2 Transact-SQL 语言	(88)
6.3 使用 Transact-SQL 定义数据	(94)
6.4 使用 Transact-SQL 操纵数据	(100)
6.5 使用 Transact-SQL 查询数据	(102)
本章小结	(109)
习 题	(109)
第 7 章 Transact-SQL 编程	(111)
7.1 Transact-SQL 编程基础	(111)
7.2 Transact-SQL 批处理	(113)
7.3 流程控制语句	(115)
7.4 功能性语句	(122)
7.5 游标(CURSOR)	(124)
本章小结	(129)
习 题	(129)
第 8 章 Transact-SQL 事务编程	(130)
8.1 事 务	(130)
8.2 锁	(132)
8.3 存储过程	(135)
8.4 触发器	(139)
本章小结	(144)
习 题	(145)
第 9 章 SQL Server 2000 数据库的管理与维护	(146)
9.1 数据库的备份与恢复	(146)
9.2 数据的导入与导出	(153)
9.3 系统安全管理	(157)
9.4 数据库安全管理相关命令	(171)
本章小结	(174)
习 题	(175)

第 10 章 SQL Server 2000 数据库的应用开发	(176)
10.1 C/S 与 B/S 应用程序模式	(176)
10.2 数据库应用项目开发流程	(177)
10.3 SQL Server 2000 应用项目开发相关技术	(181)
10.4 VB + SQL Server 2000 应用开发实例	(187)
本章小结	(207)
习 题	(208)
参考文献	(209)

第 1 章 关系数据库基础

关系数据库是当前数据库产品的主流,如 Microsoft 公司的 ACCESS、SQL Server 2000 和 ORACLE 公司的 Oracle、IBM 公司的 DB2、Sybase 公司的 Sybase 等。在这一章里首先说明什么是数据库,再介绍关系数据库的一些基本概念和关系代数理论,以及数据库设计的过程和关系规范化基本知识。

1.1 基本概念

数据库离不开数据,数据按一定的方式组织成为数据库。数据库需要有相应的软件来进行管理,这就是数据库管理系统,数据管理技术经过了较长的发展过程,才有了今天的管理方式。而数据库系统是一个与数据库有关的计算机软件、硬件和人员的总称。

1.1.1 数 据

数据(Data)是人们用来反映客观世界而记录下来的可以鉴别的数字、字母、符号、图形、声音、图像、视频信号等的总称。人们通过数据来认识世界,交流信息。这里所说的数据是经编码后可存入计算机中进行相关处理的符号集合,而信息是人们消化理解了的数据。

1.1.2 数据库

数据库(DataBase,DB)是长期存储在计算机存储器中、按照一定的方式组织起来的、可由多个用户共享的数据集合。也就是说,数据库是计算机中存放数据的仓库。数据库中的数据不是随意堆积在一起的内容,而是有组织有管理的数据聚集。

1.1.3 数据库管理系统

数据库管理系统(DataBase Management System,DBMS)能够为数据库提供数据的定义、建立、查询、统计、维护等功能,并完成对数据完整性、安全性、多用户同时使用等控制功能,以及发生故障后进行系统恢复的功能。

DBMS 是用户与计算机之间的数据管理软件,是计算机操作系统支持的计算机系统软件,类似于 Visual C++、Delphi 等高级程序设计语言。

DBMS 有很多,目前流行的有 Oracle、Sybase、SQL Server、Access、DB2 等。

1.1.4 数据库系统

数据库系统(DataBase System,DBS)包括和数据库有关的整个系统: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、用户和维护人员等。有时也将人以外与数据库有关的硬件和软件系统称为数据库系统。一个数据库系统应该具有的功能有:使用数据定义语言建立数据库;使用数据操作或查询语言对数据库中的数据进行查询和更新;支持存储大量的数据,保证对数据的正确安全使用;支持多用户并发访问且不相互影响和损坏数据。

1.1.5 数据管理技术的发展

数据在计算机中的管理方式伴随着计算机硬件、软件技术的发展以及计算机应用的不断扩充,计算机进行数据处理也经历了从低级到高级 4 个发展阶段。

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。当时的计算机硬件状况是:外存只有磁带、卡片、纸带,没有磁盘等直接存取的存储设备;从软件状况看,没有操作系统,没有管理数据的软件,数据处理方式是批处理。

在这个阶段,数据是程序的组成部分,数据的输入、输出和使用都是由程序来控制的,使用时随程序一起进入内存,用完后完全撤出计算机,没有文件的概念,程序之间不能共享数据,没有管理数据的软件,数据完全由程序员人工进行管理,如图 1-1 所示。

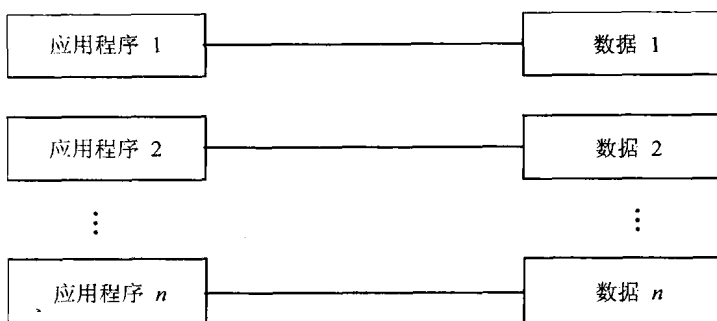


图 1-1 数据和程序关系图

2. 文件系统阶段

从 20 世纪 50 年代后期到 60 年代中期,计算机软硬件都得到了发展。计算机不仅用于科学计算,还大量用于管理工作。这时硬件方面已经有了磁盘、磁鼓等直接存取的存储设备;软件方面,出现了操作系统,可利用操作系统中的文件管理功能来进行数据处理。

在这一阶段,数据不再是程序的组成部分,而是按照一定的规则把成批数据组织在数据文件中,存放于外存储器上,由操作系统统一存取。程序通过文件名把文件从磁盘等外存调入内存而使用其中的数据,如图 1-2 所示。

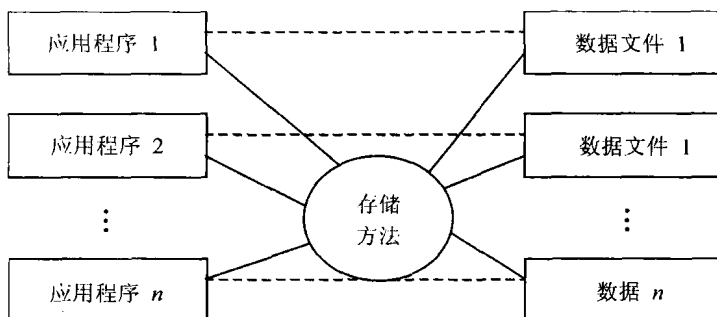


图 1-2 文件系统阶段程序和数据关系图

3. 数据库系统阶段

20 世纪 60 年代后期,数据处理的规模急剧增长。同时,计算机系统中采用了大容量的磁盘

(数百 MB 以上)系统,使联机存储大量数据成为可能。为了解决数据的独立性问题,实现数据的统一管理,达到数据共享的目的,数据库技术得到了极大的发展。为数据库的建立、使用和维护而配置的软件称为数据库管理系统,它是在操作系统支持下运行的。数据库系统是一种可以有组织地、动态地存储大量关联数据,方便用户访问的计算机应用软件。

在这个阶段,所有程序中的数据由 DBMS 统一管理,应用程序和数据完全独立,数据得到高度共享,如图 1-3 所示。

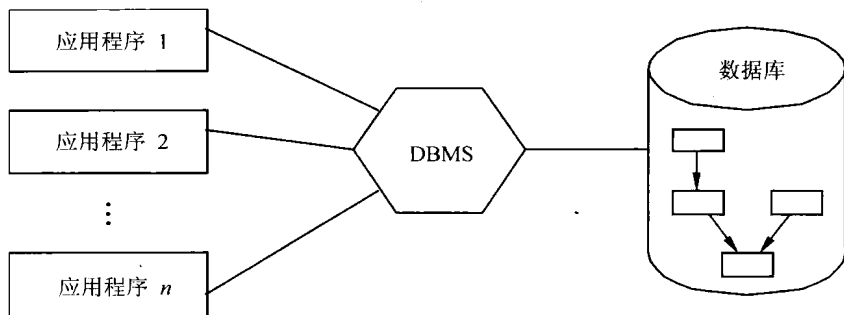


图 1-3 数据库系统阶段中程序与数据之间的关系

4. 分布式数据库管理阶段

分布式数据库系统是数据库技术与计算机网络技术相结合的产物,在 20 世纪 80 年代中期已有商品化产品问世。分布式数据库系统是一个逻辑上统一、地域上分布的数据集合,是计算机网络环境中各个局部数据库的逻辑集合,同时受分布式数据库管理系统的控制和管理。

分布式数据库系统适合于那些各部门在地理上分散的组织机构的事务处理,如银行业务、飞机订票等。

当今,计算机信息系统已从管理信息系统发展到帮助企业领导分析和做出决策的决策支持系统(Decision Support System, DSS)和以办公自动化(Office Automation, OA)技术为支撑的办公自动化信息系统。

决策支持系统和办公信息系统的目标在于,借助计算机技术及其他高技术手段,综合经营、管理与决策为一体,追求信息系统的高效益,使其在企业管理中发挥更大的作用。

1.2 数据库系统的体系结构

站在不同的角度或不同层次上看数据库系统的体系结构会有所不同。站在最终用户的角度看,数据库系统的体系结构分为集中式、分布式、客户/服务器(C/S)和并行结构。站在数据库管理系统的角度看,数据库系统体系结构一般采用三级模式结构:外模式、模式和内模式。

1.2.1 从用户的角度分类数据库系统的体系结构

1. 集中式数据库体系结构

将 DBMS 软件、所有用户数据和应用程序放在一台计算机(作为服务器)上,其余计算机作为终端通过通信线路向服务器发出数据库应用请求,这种网络数据库应用系统称为集中式数据库体系结构。

2. 客户/服务器(C/S)体系结构

这是在客户/服务器计算机网络上运行的数据库系统,这个计算机网络中,有一些计算机称为客户,另一些计算机称为服务器(即客户机/服务器)。客户/服务体系结构的关键在于功能的分布,一些功能放在客户机(前端机)上运行,另一些功能则放在服务器(后端机)上执行。

3. 分布式数据库体系结构

分布式数据库体系结构将分散存储在计算机网络中的多个节点上的数据库在逻辑上统一管理,它是建立在数据库技术与网络技术发展的基础之上的。最初的数据库一般是集中管理的,随着网络的扩大,增加了网络的负荷,对数据库的管理也困难了,分布式数据库则可克服这些缺点,分布式数据库可供地理位置分散的用户共享彼此的数据资源。

4. 并行结构数据库体系结构

这是多个物理上连在一起的CPU(而分布式系统是多个地理上分开的CPU),根据承担数据库服务责任划分它们自身的数据,通过划分的任务以及通过每秒兆位级的高速网络通信共同完成事务查询。

1.2.2 三层模式结构

数据库在计算机系统中是由数据库管理系统进行管理,为保证数据库系统各项功能得以实现,它把数据库建立成为三级模式结构和二级存储映像的体系结构。

(1)外模式,亦称子模式或用户模式。是数据库用户与数据库系统的接口,是数据库用户的数据视图。它属于模式的一部分,描述用户需要使用的数据的结构、类型、长度等。

所有的应用程序都是根据外模式中对数据的描述而不是根据模式中对数据的描述而编写的。在一个外模式中可以编写多个应用程序,但一个应用程序只能对应一个外模式。根据应用的不同,一个模式可以对应多个外模式,外模式可以互相覆盖。外模式由外模式描述语言SDDL进行具体描述。

(2)模式,又可细分为概念模式和逻辑模式,是所有数据库用户的公共数据视图,是全部用户的所有数据的逻辑结构和特征的描述,是数据库的总框架。描述数据库中关于目标存储的逻辑结构和特性,基本操作和目标的关系和依赖性,以及对数据的安全性、完整性等方面的定义,所有数据都按这一模式进行装配。模式由模式描述语言DDL来进行描述。

(3)内模式亦称存储模式。是对数据库在物理存储器上具体实现的描述。它规定数据在存储介质上的物理组织方式、记录寻址技术,定义物理存储块的大小,溢出处理方法等,与概念模式相对应。内模式由数据存储描述语言DSDL进行描述。

1.2.3 两层映像功能

映像用来指定映像双方进行数据转换的规则,说明双方如何进行转换,其实际的转换工作是由数据库管理系统来完成的。

1. 外模式/模式映像

通过外模式与模式之间的映像把描述局部逻辑结构的外模式与描述全局逻辑结构的模式联系起来。因为一个模式与多个外模式对应,所以对于每个外模式都有一个外模式/模式映像用于描述外模式与模式之间的对应关系。外模式与模式之间的映像通常放在内模式中进行描述。

因为有了外模式与模式之间的映像,当数据库的模式有改变时,比如修改某个属性的数据类

型、增加或删除某个表中的属性,这时只要对外模式与模式之间的映像进行修改,而保持与应用程序相关的外模式不变,从而应用程序不变,从而保证了数据与程序之间的逻辑独立性,这也称之为数据的逻辑独立性。

2. 模式/内模式映像

通过模式与内模式之间的映像把描述全局逻辑结构的模式与描述物理结构的内模式联系起来。一个数据库中只有一个模式与一个内模式,因此,模式/内模式的映像也只有一个,通常放在内模式中进行描述。

因为有了模式与内模式之间的映像,当数据库的内模式改变时,比如存储设备或存储方式有所改变,这时只要对模式与内模式之间的映像进行修改,从而使模式保持不变,而与应用程序相关的外模式也就不会变,从而应用程序不变,保证了数据与程序之间的物理独立性,也就是数据的物理独立性。

数据库管理系统把数据库分为三级模式结构和二级映像,这样当存储数据库的内模式发生变化或者当数据库模式发生变化时,都可以使外模式在最大限度内保持不变。由于应用程序是在外模式所描述的数据结构的基础上编写的,外模式的稳定性就保证了应用程序的稳定性。而这正是数据库系统采用三层模式、两层映像为系统提供了高度的数据独立性所得到的结果。

数据库系统的三层模式结构如图 1-4 所示。

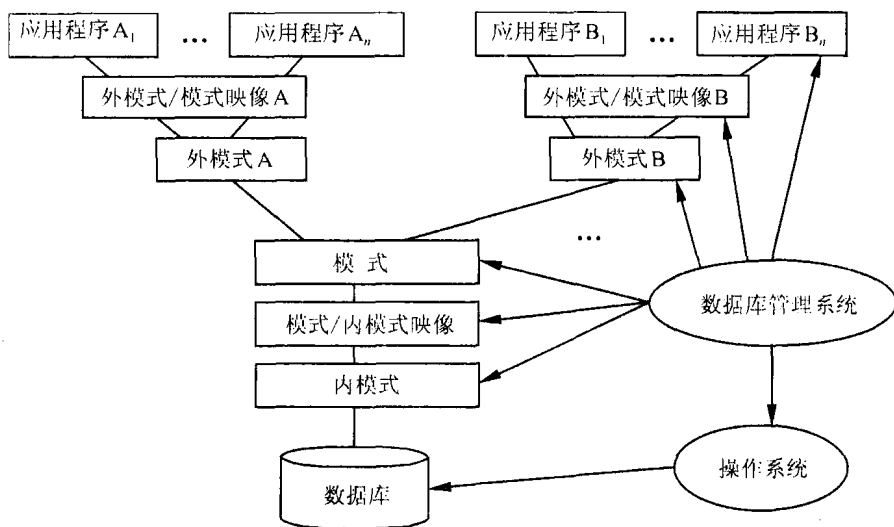


图 1-4 数据库系统的三层模式结构

1.3 关系数据库系统的基本概念

关系数据库是当前数据库产品的主流,它不仅简单易于理解并且具有坚实的理论基础。它和早期的网状和层次数据库系统相比,关系数据库系统以其简易性简化了大量的编程工作。

关系数据库的理论基础是关系代数。这一节主要讲述关系的基本概念、关系模型、关系运算等方面的内容。

1.3.1 关 系

关系是一种简单的二维表结构,即一个关系可以表示为一张二维表。例如,表 1-1 就是一个

“学生”关系，记录了学生的学号、姓名、性别和年龄信息。表的每一列标题栏中的名字称为关系的属性，属性描述了该列数据的意义；表中除了标题行之外的每一行称之为关系的“元组”或“记录”，它表示了具体的数据信息。例如，表 1-1 中有 4 个“元组”，记录了 4 个学生的学号、姓名、性别和年龄信息。

关系中不同属性列的数目称之为关系的度或度数。例如表 1-1 所表示的“学生”关系共有 4 个属性列，所以此关系的度为 4；表 1-2 所表示的“课程”关系中，共有三个属性列，所以其度数为 3。

表 1-1 “学生”关系

学号	姓名	性别	年龄	属性
20060003001	张小光	男	19	元组
20060003002	刘和平	男	20	
20060003003	陈 新	女	19	
20060003004	蔡忠明	男	21	

表 1-2 “课程”关系

课程号	课程名	学分
001	大学英语	3
002	高等数学	4
003	程序设计语言	5
004	数据库技术	4

关系中能唯一标识每个元组的最少属性或属性组称之为关键字或主码。例如“学生”关系中的属性“学号”就是关键字，只要学号确定了，就能知道这个学号对应的姓名、性别和年龄等信息，但学生关系中的“性别”和“年龄”不能作为关键字，因为即使年龄或性别确定了，还是不能确定学生的姓名和学号等信息，同性别或者同年龄的学生太多了。当然如果这个关系中没有同姓名的学生，则姓名也可以作为关键字看待，这要根据具体的语义来决定。当有多个可选的关键字（称之为候选关键字）时，可由关系的设计者或使用用户指定其中之一为主关键字。

在表 1-2 所示的“课程”关系中，“课程号”和“课程名”均可以作为关系的关键字，但为了避免重复的课程名称，一般使用“课程号”作为主关键字。而在表 1-3 所示的“成绩”关系中，单独的学号和课程号都不能作为关键字，因为一个学号代表一个学生，每个学生可以选修多门课，从而对应每个学号会有多个记录，同样地，每个课程号对应一门课程，每门课程会有多个学生选修，也就会对应表中多条记录。只有学号和课程号的组合（学号，课程号）才能确定相应的成绩，所以（学号，课程号）为“成绩”关系的关键字。关键字可能是属性的组合，但必须是最少的，即少一个属性不能成为关键字，但多一个就有了冗余。

表 1-3 “成绩”关系

学号	课程号	成绩	学号	课程号	成绩
20060003001	001	89	20060003002	002	90
20060003001	003	70	20060003002	003	52
20060003001	004	68	20060003003	001	72
20060003004	001	60	20060003004	002	80

根据上面的说明，总结出关系的特点有：

(1) 关系中的每一个属性值都必须是不能再分的元素。例如学生的“姓名”不能再细分为“姓”和“名”两个属性值，必须把其作为一个整体来看待。

(2) 每一列中的数值是同类型的数据。例如学生的年龄列为整数值，性别列从{“男”，“女”}中取值等。

(3) 不同的列应该给予不同的属性名。同一个关系中的两个列即使其取值范围相同也必须有不同的属性名，以便区分其不同意义。

(4) 同一关系中不允许有相同的元组。如果有相同的元组也只保留一个。

(5) 关系是行或列的集合，所以行、列的次序可以任意交换，不影响关系的实际意义。

1.3.2 关系模式

关系名称和关系的属性名集称为该关系的模式,其记法为:

$\langle \text{关系名} \rangle (\langle \text{属性名} 1 \rangle, \langle \text{属性名} 2 \rangle, \dots, \langle \text{属性名} n \rangle)$

所以,表 1-1 所示的学生关系对应的关系模式为:

学生(学号,姓名,性别,年龄)

关系模式简称为模式,模式中的属性是一个集合而非有序列表,也就是说属性名的排列顺序不影响关系模式,但为了便于说明和讨论,一般会为这些属性根据其重要程度而规定一个“标准”顺序。

一个数据库中往往包含多个关系,一个数据库中这些关系的集合称之为“数据库模式”,数据库设计的主要任务是确定其中需要多少个关系,每个关系有多少个属性,属性的名称和数据类型等内容,也就是设计好每个关系的模式。

在同一个数据库中某个关系 R_1 中的属性或属性组若在另一个关系 R_2 中作为主码使用,则该属性或属性组为 R_1 的外关键字或外码。在同一个数据库中的关系往往是通过外关键字而相互关联的。假设设计的数据库包含表 1-1~表 1-3 所示的三个关系:

学生(学号,姓名,性别,年龄)

课程(课程号,课程名,学分)

成绩(学号,课程号,成绩)

“成绩”关系中的“学号”是“学生”关系中的主关键字,所以“学号”是“成绩”关系中的外关键字;同样地,“成绩”关系中的“课程号”是“课程”关系中的主关键字,所以“课程号”也是“成绩”关系中的外关键字。

1.3.3 关系模型

为了用计算机处理现实世界中的具体问题,往往要对复杂的事物进行高度概括和抽象,以便提取事物的主要特征,形成一个清晰且好理解的模型,这就是所说的“建模”。数据模型就是对客观事物抽象化的表现形式,具有三大特点:①它必须真实地反映现实世界中的具体应用,否则就失去了意义;②要便于理解,使用者与设计者要取得一致的看法;③应该便于使用计算机来实现和处理。

数据模型通常由数据结构、数据操作和完整性约束三要素组成。分别表示系统的静态特性、动态特性和数据库的正确性、相容性和有效性。

传统的数据模型有层次、网状和关系。但层次和网状数据模型已很少使用,关系数据库占据了主导地位,近年来,对象模型也得到了一些应用。

数据以“关系”的形式表示,也就是二维表的形式表示,其数据模型就是所说的关系模型。在关系模型中,数据及数据之间的联系均用关系来表达,并且对关系进行各种处理之后得到的还是关系。

采用关系模型建立数据库系统具有以下特点:

(1)组织数据的结构单一。在关系模型中,无论是数据还是数据之间的联系都是以熟悉的二维表(关系)形式来表示的,这种表示方法不仅让人容易理解且便于计算机操作和实现。

(2)采用集合运算。在关系模型中,运算的对象是关系,运算的结果还是关系,而关系可以看做是行(元组或记录)的集合,所以对关系的运算可以转化为对集合的运算。

(3)数据完全独立。因为关系数据库系统中的数据是由 DBMS 进行管理的,对于程序员来说,

不需要知道数据存放的具体位置和组织形式等方面的内容,只需要告诉系统要进行什么样的操作,由系统自动完成相关的任务,即程序和数据高度独立。

(4)数学理论支持。在关系模型中,每个关系都是集合,对关系的运算有集合论、数理逻辑做基础,关系结构可以用关系规范化理论进行优化。总之,关系模型具有严格的数学定义,具有成熟的数学理论为依据,它是目前为止最简单有效、最受欢迎、最广泛应用的数据模型。

1.3.4 关系数据库系统

关系数据库系统就是以关系模型为基础的数据库系统。由于关系模型有严格的数学基础,而且便于人们理解和使用,因此关系数据库系统自 20 世纪 70 年代末推出以来,得到了广泛的应用,并迅速占据了数据库系统的主流位置。近年来各大公司推出的关系数据库产品主要有 Microsoft 公司的 ACCESS,SQL Server 2000,ORACLE 公司的 Oracle,IBM 公司的 DB2,Sybase 公司的 Sybase 等。

1.4 关系代数

从用户的角度来看,关系数据库中保存着他们需要的各种数据。那么保存数据的目的是什么呢?主要是为了今后查询和处理数据方便快捷,也可能是分析问题或决策时有供参考的内容。具体的关系数据库管理系统有针对数据库的数据查询、数据定义和数据控制等语句,这里首先了解关系的一些基本运算,它是所有数据查询语言的理论基础。

关系代数是过程化的查询语言,所谓过程化语言就是需要用户指导系统对数据库执行一系列操作从而计算得到所需要的结果。关系代数包括运算的集合,这些运算都是以一个或两个关系为输入,产生一个新的关系作为结果。

关系代数的运算分为两大类:第一类是传统的集合运算(并、交、差),另一类是专门的关系运算(选择、投影、连接等)。

1.4.1 关系的集合运算

设 P_1 和 P_2 为参加运算的两个关系,它们具有相同的属性集,则可定义以下三种传统的集合运算。

(1)并运算。 $P_1 \cup P_2$,表示关系 P_1 与关系 P_2 的并,结果中的元组或者属于 P_1 或者属于 P_2 。

(2)差运算。 $P_1 - P_2$,表示关系 P_1 与关系 P_2 的差,结果中的元组属于 P_1 但不属于 P_2 。

(3)交运算。 $P_1 \cap P_2$,表示关系 P_1 与关系 P_2 的交,结果中的元组既属于 P_1 又属于 P_2 。

例如关系“课程表 1”与关系“课程表 2”分别如表 1-4 和表 1-5 所示。

表 1-4 “课程表 1”关系

课程号	课程名	学分
001	大学英语	3
002	高等数学	4
003	程序设计语言	5

表 1-5 “课程表 2”关系

课程号	课程名	学分
003	程序设计语言	5
004	数据库技术	4

由表 1-4 和表 1-5 可知,“课程表 1” $\cup$ “课程表 2”应该如表 1-6 所示。要注意的是,关系“课程表 1”与关系“课程表 2”中含有一个相同的元组(003,程序设计语言,5),在并运算的结果集中只能出现一次。正因为如此,“课程表 1” $\cap$ “课程表 2”的结果关系中只包含此一个元组,如表 1-7 所示。

由于元组(003, 程序设计语言, 5)既出现在关系“课程表 1”中又出现在关系“课程表 2”中, 所以“课程表 1”-“课程表 2”的结果应该只包含出现在关系“课程表 1”中不出现在关系“课程表 2”中的两个元组, 结果如表 1-8 所示。

表 1-6 “课程表 1” $\cup$ “课程表 2”结果关系

课程号	课程名	学分
001	大学英语	3
002	高等数学	4
003	程序设计语言	5
004	数据库技术	4

表 1-7 “课程表 1”-“课程表 2”结果关系

课程号	课程名	学分
001	大学英语	3
002	高等数学	4

表 1-8 “课程表 1” $\cap$ “课程表 2”结果关系

课程号	课程名	学分
003	程序设计语言	5

1.4.2 专门的关系运算

专门的关系运算主要介绍三类: 选择、投影和连接(笛卡儿积、等值连接和自然连接)。

1. 选 择

选择是从给定关系中找出满足一定条件的元组的运算, 其运算符号记为“ σ ”。在做选择运算时需要说明是从哪个关系(二维表)中进行选择, 给出关系名; 另外选择时需要说明选择的条件, 一般用“年龄 >19 ”或“成绩 <60 ”等形式来表示, 在比较符号前面的一般是关系中表示某个意义的属性名, 后面为给定的条件值; 当需要满足多个条件时使用逻辑运算符 AND, OR, NOT 等连接条件。选择运算的一般表示形式为:

$$\sigma_{\text{条件}}(\text{关系名})$$

选择运算的结果仍然是一个关系, 这个关系的属性集与原来关系的属性集相同, 但元组一般会减少一些, 它是从行的角度进行的选择, 如图 1-5 所示。

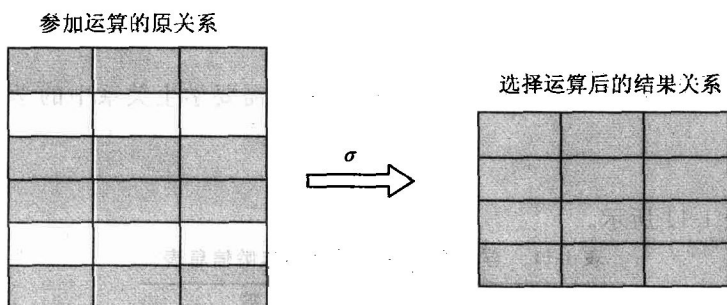


图 1-5 “选择”运算图示

例如, 要查找年龄在 20 岁以下的学生信息, 其关系运算式为:

$$\sigma_{\text{年龄} < 20}(\text{学生})$$

得到的结果关系如表 1-9 所示。

表 1-9 年龄在 20 岁以下的学生关系

学号	姓名	性别	年龄
20060003001	张小光	男	19
20060003003	陈一新	女	19

如果要求查找年龄在 20 岁以下的女学生信息,则条件有两个,一个是“年龄<20”另一个是“性别=‘女’”,要求同时满足条件时使用逻辑运算与(AND)进行连接。其关系代数运算式为:

$$\sigma_{\text{年龄}<20 \text{ AND 性别}='女'}(\text{学生})$$

得到的结果关系如表 1-10 所示。

表 1-10 年龄在 20 岁以下的女学生关系

学号	姓名	性别	年龄
20060003003	陈一新	女	19

2. 投影

投影是从给定的关系中选取若干属性的运算,其运算符号记为“ π ”。进行投影运算时,需要说明是从哪个关系(二维表)中进行投影,即给出关系名;另外投影时还需要说明要投影的列(属性)名,如果选择多个列时属性列之间使用逗号分隔。

投影运算的一般表示如下:

$$\pi_{\text{列名集}}(\text{关系名})$$

投影操作主要是从列的角度进行的运算,如图 1-6 所示。但需要注意的是,投影运算之后不仅取消了原来关系中的某些列,而且还可能取消某些元组(重复行)。

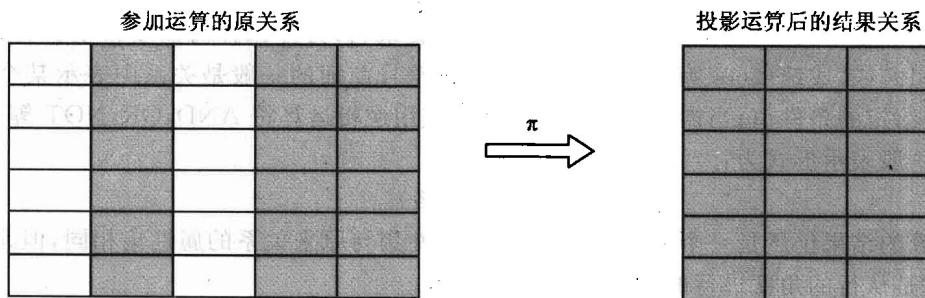


图 1-6 “投影”运算图示

例如,要查找所有学生的姓名和年龄信息,也就是说不需要学生关系中的学号和性别信息,其关系运算式为:

$$\pi_{\text{姓名, 年龄}}(\text{学生})$$

得到的结果关系如表 1-11 所示。

表 1-11 学生关系中的姓名和年龄信息表

姓名	年龄
张小光	19
刘和平	20
陈一新	19
蔡忠明	21

3. 笛卡儿积

前面介绍的选择和投影运算是一元运算,而笛卡儿积是一种二元运算,它把两个关系的元组以所有可能的方式组合起来,其运算符号为“ $\times$ ”。两个关系 A 和 B 的笛卡儿积记为“ $A \times B$ ”,其结果为一个新的关系,其属性集是 A 的所有属性与 B 的所有属性集的合并,如果关系 A 和关系 B 有同名的属性 t,则使用 A.t 和 B.t 来区分是来自 A 的属性 t 还是来自 B 的属性 t,所以 $A \times B$ 的属