

高等学校计算机应用基础通用教材

数据库及其应用

宋 玉 梁 飞 丁玉荣 编著

郑州大学出版社

图书在版编目(CIP)数据

数据库及其应用/宋玉,梁飞,丁玉荣编著. —郑州:
郑州大学出版社,2002. 2
ISBN 7-81048-575-X

I. 数… II. ①宋… ②梁… ③丁… III. 关系数
据库-数据库管理系统, FoxPro IV. TP311.138

中国版本图书馆 CIP 数据核字(2002)第 008086 号

出版社:郑州大学出版社

(地址:郑州市大学路 40 号

邮政编码:450052)

发行单位:郑州大学出版社

承印单位:河南东方制图印刷有限公司

开本:787 mm × 1 092 mm

1/16

印张:19.125

字数:453 千字

印数:1~5 200 册

版次:2002 年 2 月第 1 版

印次:2002 年 2 月第 1 次印刷

书号:ISBN 7-81048-575-X/O·4

定价:25.80 元

内 容 提 要

本书较全面地介绍了 FoxPro 2.5/2.6 关系型数据库管理系统,包括数据库的基本概念、FoxPro 的基础知识、数据库文件的基本操作、多重数据库文件操作、程序设计、程序设计实例,以及简单介绍了最新可视化数据库管理系统 Visual FoxPro 6.0 的使用、数据库和数据表、面向对象的程序设计思想。

本书内容实用、概念清晰。各章都安排有大量的例题,章末附有习题,这些例题和习题都具有代表性,便于教学和自我检测。

本书可以作为大中专院校计算机和非计算机专业数据库应用课程的教材,也可以作为各类培训班的培训教材,同时适合各类数据库应用开发人员自学或作为参考用书。

前 言

关系数据库管理系统近年来得到了广泛的应用,由 Microsoft 公司推出的关系型数据库管理系统 FoxPro 2.5/2.6 受到了普遍欢迎,它以其快速的查询、灵活的操作迅速地占据了广大的市场。而 Visual FoxPro 6.0(简称 VFP)是 Microsoft 公司近期推出的正在逐渐流行起来的微机数据库管理系统,它以其可视化的编程方法和面向对象的程序设计思想以及卓越的数据库管理性能和良好的开发环境赢得了广大用户的信赖和喜爱。

作者根据多年的教学经验,认为 FoxPro 2.5/2.6 提供的数据库管理方法容易被读者所接受,学习其结构化程序设计的方法,有利于提高读者的编程能力。而 VFP 引入了面向对象的编程思想,采用可视化的编程方式,大大减少了编程的工作量;同时,VFP 还继承了 FoxPro 2.5/2.6 的优点并兼容了其功能。基于以上考虑,编者在重点介绍 FoxPro 2.5/2.6 的基础上也简单地介绍了 VFP 的一些基本知识,使读者可以系统、全面地掌握数据库理论知识和应用技巧。

本书是参照国家教育部计算机基础课程教育指导委员会制定的有关大纲和作者长期从事数据库教学实践中的总结来编写的。全书共分为 9 章,主要包括数据库的基本概念、FoxPro 的基础知识、数据库文件的基本操作、多重数据库文件操作、程序设计、程序设计实例及 Visual FoxPro 6.0 的简介等。

本书力求语言精练、概念准确、结构清晰,避免空泛的论述,目的是使读者能在有限的时间内,通过阅读较少的篇幅来掌握大纲中所规定的内容。本书的参编作者都是有多年教学和科研经验的专业教师,在内容组织、结构编排、实例选用等方面有一定的特色,目的是突出 FoxPro 的特点和培养读者程序设计的能力,相信本书对读者会有所帮助。

本书是由宋玉、梁飞、丁玉荣编著,宋玉统稿,李正民参加了编写工作。其中:梁飞编写第 1 章、第 2 章、附录 1 和附录 2;丁玉荣编写了第 3 章和第 4 章;宋玉编写第 5 章、第 6 章、第 7 章;李正民编写了第 8 章和第 9 章。

在本书编写过程中,郑州大学教务处和信息工程学院的各位领导给予了大力的支持和帮助,并提出了许多指导性的建议,谨此一并表示感谢。

限于编者的水平,本书难免有不妥之处,敬请各位同行和读者给予指教。

编者

2001 年 12 月

目 录

第1章 数据库的基本概念	(1)
1 数据库概述	(1)
1.1 数据库简史	(1)
1.2 信息、数据和数据处理	(1)
1.3 数据库和数据库系统	(2)
1.4 数据库管理系统	(2)
2 数据模型	(3)
3 关系的定义和性质	(5)
4 关系操作	(5)
小 结	(7)
习 题	(7)
第2章 数据库的基本知识	(8)
1 FoxPro 简介	(8)
1.1 FoxPro 的发展和特点	(8)
1.2 FoxPro 的启动	(10)
2 FoxPro 的菜单操作系统	(13)
2.1 FoxPro 的菜单系统	(13)
2.2 菜单操作	(14)
2.3 窗口及对话框	(15)
3 FoxPro 基础	(18)
3.1 FoxPro 的数据类型和文件类型	(19)
3.2 FoxPro 的命令结构及语法规则	(20)
3.3 关系的投影和选择运算	(23)
4 常量、变量和表达式	(24)
4.1 常量与变量	(24)
4.2 运算符及表达式	(30)
4.3 FoxPro 函数	(35)
小 结	(48)
习 题	(48)
第3章 数据库文件的基本操作	(51)
1 数据库文件结构的建立	(51)

1.1 定义数据库文件的结构	(51)
1.2 数据库文件结构的建立	(52)
2 数据库文件的打开和关闭	(56)
2.1 数据库文件的打开	(56)
2.2 数据库文件的关闭	(57)
3 数据库文件的数据输入	(58)
3.1 直接输入数据	(58)
3.2 追加数据	(60)
4 数据库文件的显示	(60)
4.1 数据库文件的结构显示	(61)
4.2 数据库文件的记录显示	(62)
5 记录的定位、插入和删除	(64)
5.1 记录指针及其定位	(64)
5.2 记录的插入	(67)
5.3 记录的删除	(68)
6 数据库文件的修改	(73)
6.1 数据库文件结构的修改	(73)
6.2 数据库文件记录数据的修改	(75)
7 数据库文件的复制	(88)
7.1 库文件结构的复制	(88)
7.2 结构描述文件的建立与应用	(89)
7.3 库结构与记录数据的复制	(92)
7.4 从其他库文件中读取数据	(94)
7.5 磁盘文件操作	(96)
8 数组及其应用	(98)
8.1 数组的定义	(98)
8.2 数组元素的赋值	(99)
8.3 记录数据传送到数组	(100)
8.4 数组数据传送到记录	(102)
8.5 多记录数据的传送	(103)
小 结	(105)
习 题	(105)
第4章 数据库文件的排序、索引与统计	(107)
1 记录的筛选与字段的屏蔽	(107)
1.1 记录的筛选	(107)
1.2 字段的屏蔽	(109)
2 数据库文件的排序	(111)
2.1 排序与索引	(111)

2.2 排序文件的建立	(111)
3 数据库文件的索引	(114)
3.1 索引文件及索引类型	(114)
3.2 单索引文件的建立	(115)
3.3 复合索引文件的建立	(117)
3.4 索引文件的打开	(119)
3.5 重新指定主索引文件	(122)
3.6 索引文件的关闭	(123)
3.7 重新索引	(123)
3.8 索引的删除	(124)
4 数据库记录的查询	(125)
4.1 直接查询	(125)
4.2 索引查询	(128)
4.3 三个查找命令的比较	(131)
5 数据库记录的统计与汇总	(131)
5.1 数据库记录的统计	(131)
5.2 数值字段纵向求和	(133)
5.3 数值字段纵向求平均值	(134)
5.4 纵向综合统计	(135)
5.5 纵向分类汇总	(136)
小 结	(138)
习 题	(138)
第5章 多重数据库的操作	(141)
1 工作区与多个数据库	(141)
1.1 工作区的选择	(141)
1.2 工作区的联访	(145)
2 数据库文件的关联	(146)
2.1 建立关联关系	(146)
2.2 取消关联关系	(148)
3 数据库文件间的更新	(148)
4 数据库文件的连接	(150)
小 结	(151)
习 题	(151)
第6章 FoxPro 程序设计	(153)
1 程序设计基础	(153)
1.1 程序的建立、运行、调试和修改	(153)
1.2 程序注释及程序流程图	(158)
2 人机对话命令	(160)

2.1 非格式化命令	(160)
2.2 简单的格式化命令	(161)
3 FoxPro 结构化程序设计的结构形式	(165)
3.1 三种程序控制结构	(165)
3.2 顺序结构	(165)
4 选择结构	(166)
4.1 IF ~ ELSE ~ ENDIF 语句	(166)
4.2 DO CASE ~ ENDCASE 语句	(169)
5 循环结构	(172)
5.1 DO WHILE ~ ENDDO 语句	(173)
5.2 FOR ~ ENDFOR 语句	(178)
5.3 SCAN ~ ENDSCAN 语句	(179)
5.4 多重循环	(181)
6 过程与自定义函数	(184)
6.1 子程序	(184)
6.2 过程	(189)
6.3 自定义函数	(194)
6.4 变量的作用域	(197)
6.5 程序设计中常见的错误和要注意的问题	(199)
小 结	(200)
习 题	(200)
第7章 输入/输出设计及工具	(203)
1 格式化输入/输出语句	(203)
1.1 格式语句及其功能	(203)
1.2 格式化输入/输出语句的应用	(205)
2 窗口设计	(208)
2.1 定义窗口	(208)
2.2 激活或显示窗口	(209)
2.3 挂起或隐去窗口	(210)
2.4 从内存中删除窗口	(211)
3 菜单设计	(212)
3.1 光棒菜单设计	(212)
3.2 数组菜单设计	(214)
3.3 弹出式菜单设计	(217)
3.4 下拉菜单设计	(219)
3.5 多级下拉菜单设计	(225)
3.6 组合式下拉菜单设计	(226)
4 报表输出设计	(231)

4.1 常用的报表设计命令和函数	(231)
4.2 特殊图形符号及打印字形设置	(232)
4.3 两种设计方式	(233)
小 结	(235)
习 题	(235)
第8章 FoxPro 程序设计示例	(237)
1 系统分析	(237)
2 系统设计	(238)
2.1 系统结构设计	(238)
2.2 数据库设计	(239)
2.3 程序设计	(241)
3 系统评价	(258)
小 结	(258)
习 题	(258)
第9章 Visual FoxPro 6.0 简介	(259)
1 从 FoxPro 到 Visual FoxPro	(259)
1.1 VFP 的发展	(259)
1.2 VFP 的主要改进	(259)
1.3 VFP 的启动和退出	(260)
1.4 VFP 的基本菜单简介	(261)
1.5 学习 VFP 时要注意的问题	(264)
2 数据库和数据表	(265)
2.1 全新概念的库和表	(265)
2.2 字段数据类型	(267)
2.3 创建数据库	(269)
2.4 数据库表的好处	(272)
2.5 库和表的命令操作	(274)
3 面向对象的编程思想	(275)
3.1 类和对象	(275)
3.2 封装、继承与多态性	(276)
3.3 面向对象编程的优点	(277)
小 结	(278)
习 题	(278)
附 录	(279)
附录1 FoxPro 命令简表	(279)
附录2 FoxPro 函数集(字母序)	(290)

第 1 章 数据库的基本概念

1 数据库概述

1.1 数据库简史

随着计算机技术的发展,计算机的主要功能已经从科学计算转变为事物管理。在进行事物管理时,并不要求复杂的科学计算,而是要求从大量有关数据中提取有关信息。因此,在进行事物处理时,必须在计算机中存放大量的数据。为了有效地使用存放在计算机系统的大量有关的数据,必须采用一套严密合理的存取数据、使用数据的方法,这就是数据管理。

数据管理是指对数据的组织、存储、维护和使用等。随着计算机技术的发展,数据管理的技术也在发展。大体上可分为三个阶段:人工管理阶段、文件管理阶段和数据库系统阶段。

人工管理阶段是最早的数据管理阶段,大致在 20 世纪 50 年代中期之前。那时,数据包含在程序中,用户必须考虑存储数据和使用数据的所有问题。如在存取数据时,必须使用物理地址,即直接与计算机的硬件设备打交道。这就使得用户必须要首先了解计算机的硬件设备和工作过程。况且,当数据稍有变动时,整个程序都要修改。因此,该阶段的数据管理是最低级的数据管理。

文件管理阶段从 20 世纪 50 年代中期开始,直至 20 世纪 60 年代末。它是通过操作系统中的文件管理系统来管理和使用数据的。用户只需把数据组织成单独的数据文件,使用时通过文件名调用。用户不必考虑数据在计算机系统实际存储方法即物理结构,不直接与计算机的存储设备打交道,使用比较方便。但是,由于数据文件的内容是由用户自行组织,会产生随意性。一般情况下,数据文件是专用的,即各程序分别组织自己的数据文件。这样,不能实现数据共享,反而会出现数据重复即数据的冗余,并且还会出现重复数据的不一致性。为了克服文件管理阶段存在的缺点,就产生了数据库管理技术。

数据库管理阶段是从 20 世纪 70 年代初开始的。该技术的出现,标志着数据管理达到了一个全新的高级阶段。不同用户的数据可以按照一定的组织结构存放在一起,形成一个大的数据集,称为数据库。不同用户可以共同使用数据库中的数据。实现了数据共享,最大限度地降低了数据的冗余,并且避免了数据的不一致性。

1.2 信息、数据和数据处理

(1) 信息

信息泛指通过各种方式传播的声音、文字、图像、符号等所表征的某一特定事物的知

识、情况和消息。它直接地、本质地反映了客观世界,是人们从事某项决策的依据。

当今,社会已经进入信息化的时代,人们在各种活动中将产生大量的信息。计算机技术的发展,使得人们可以对这些信息进行保存和加工处理。

(2) 数据

数据是用一定方式记录下来的客观事物的特征,是信息的符号化表示。我们每个人在生活和工作中都离不开数据。比如:身份证、工资单、通讯录以及录像带等等。这些记录可以是数字,如工资;可以是文字,如姓名;也可以是一串特定的符号,如真假、日期;还可以是声音、图片、图像等等。

(3) 数据处理

数据处理是指对各种形式的数据进行收集、存储、计算、加工、检索和传输的过程。数据处理的目的是为了从大量的、原始的数据中提取有价值的信息,以此作为行为和决策的依据。对数据的组织、编码、分类、存储、检索和维护是数据处理的核心。

1.3 数据库和数据库系统

(1) 数据库

数据库(DataBase,简称DB)就是存放数据的仓库。它保存的是属于单位、团体和个人的有关数据。比如,我们可以把一个单位所有职工的姓名、性别、出生年月、籍贯、职称、工资等许多数据有序地组织起来,存储在计算机磁盘上,构成一个人事数据库。此后,用户可方便地对该数据库中的数据进行查询、修改,并按照一定的格式输出。从而达到管理和使用这些数据的目的。因此,我们对数据库可以作以下的定义:数据库就是按照一定的数据模型来组织和存储的、能为多个用户共享的、独立于应用程序的、相互关联的数据集合。

(2) 数据库系统

数据库系统(DataBase System,简称DBS)是一个计算机的应用系统,由相关的计算机硬件、软件、数据和人员组成。

数据库系统的软件是指负责数据库的存取、维护 and 管理的软件系统,即数据库管理系统(DataBase Management System,简称DBMS)。它对数据库中的数据资源进行统一管理和控制,并将应用程序和数据库数据联系起来。数据库管理系统是数据库系统的核心,其功能强弱体现了数据库系统性能的优劣。数据库管理系统一般由计算机软件公司提供。

数据库系统的硬件是指存储数据库及运行DBMS的硬件资源。

这里的数据是指数据库系统中存储的数据,它是数据库系统操作的对象。

数据库系统的人员是指管理、开发和使用数据库系统的人员。

1.4 数据库管理系统

数据库管理系统(DBMS)是管理数据库的系统软件。它的作用是为用户提供一个建立数据库、使用数据库和维护数据库的环境。数据库管理系统处于操作系统和应用软件之间,实现了对数据库的统一管理和控制。

数据库管理系统通常由三部分组成:数据库描述语言(Data Description Language,简

称 DDL) 及其编译程序、数据操纵语言 (Data Manipulation Language, 简称 DML) 或查询语言及其编译 (或解释) 程序。数据描述语言用于定义数据库。数据操纵语言提供对数据库数据的检索、插入、修改及删除等基本操作。数据库管理例行程序是数据库管理的核心部分, 它包括并发控制、存取控制、完整性条件检查与执行、数据库内部维护等, 所有对数据库的操作都在上述控制程序的统一管理下进行, 以确保任务的正确运行和数据库数据的正确有效。

2 数据模型

数据模型 (Data Model) 是指数据库的组织形式, 它决定了数据库中数据之间联系的表达方式, 常用的数据模型有三种: 层次模型、网络模型和关系模型。

(1) 层次模型 (Hierarchical Model)

层次模型是数据库系统最早使用的一种模型, 它的数据结构像一棵倒着的树。树结点是实体, 树的枝是联系。层次模型的特征如下。

- ① 有且仅有一个结点, 没有父结点, 这个结点即为父结点 (或简称根)。
- ② 其他子结点有且仅有一个父结点。

层次模型结构清晰, 各结点之间联系简单, 适用于描述现实世界中具有层次联系的事物, 如图 1-1 所示。

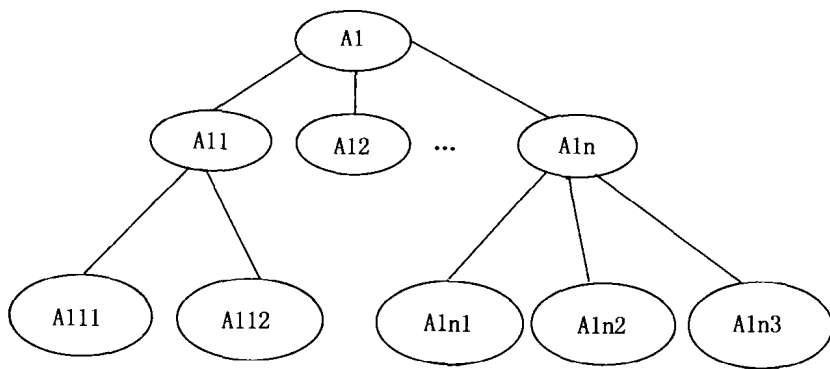


图 1-1 层次模型

层次模型只能反映实体间的一对多的联系, 而对于实体间的多对多的联系则无能为力。

(2) 网络模型 (Network Model)

网络模型其结点之间的联系像一张网, 网上的连接点都是结点。结点之间是平等的, 不分层次。网络模型的特征如下。

- ① 有一个以上的结点无父结点。
- ② 允许结点有多于一个的父结点。
- ③ 允许两个结点之间有两种或两种以上的联系。

网络模型表达能力较强, 它能反映实体间的多对多的联系。如图 1-2 表示了任课教

师、授课内容、听课学生及上课地点四者的联系。但网络模型在概念上、结构上和使用上都比较复杂。

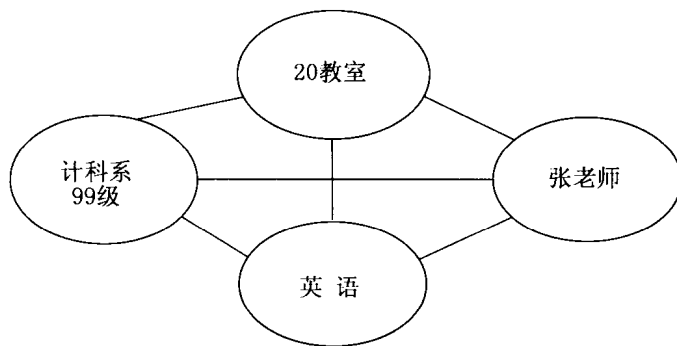


图 1-2 网络模型

层次模型和网络模型都是用指针来实现两个实体之间的联系的。它们都是建立在图论的基础上,通常被称为格式化数据模型。

(3) 关系模型(Relational Model)

早期的数据库系统都采用格式化模型。1970年,美国 E. F. Good 提出了关系模型的概念,首次运用数学方法来研究数据库的结构和数据操作,将数据库的设计从以经验为主提高到以理论为指导。关系模型是用表格形式表示实体间联系的模型。它将数据的逻辑结构归结为满足一定条件的二维表格。日常生活和工作中的各种报表,如工资表、学生成绩表、人事档案表、生产报表等都具有二维表格的结构,都可以很方便地用关系数据模型描述,如表 1-1 和表 1-2。

表 1-1 工资表

职工号	姓名	工资	津贴	应发	扣款	实发
01001	张三元	550	100	650	10	640
01002	李四新	595	100	695	10	685
03021	王五君	600	120	720	10	710
03022	赵六梅	470	95	565	10	555
03101	李自力	810	130	940	20	920

表 1-2 职工表

职工号	姓名	性别	职称	婚否
01001	张三元	男	教授	已
01002	李四新	女	副教授	已
03021	王五君	女	讲师	未

3 关系的定义和性质

一个二维表格称作一个关系,如前面的工资表和职工表均可称作关系,可分别命名为工资表和职工表。在二维表中,表的每一行称为一个元组或记录;表的每一列称为一个属性或字段,相当于记录中的一个数据项。属性的取值范围称为域。为了相互区分和识别,字段要分别命名。由所有字段名组成的二维表的表头称为关系框架。一个或若干个属性的集合称为关键字,它惟一表示一个元组。每一行中对应于每一列的属性值叫做一个分量。表格有几列,则称该关系为几元关系。关系应满足如下条件。

- ①二维表中的每列都是同类型的数据。
- ②关系的每列都必须有不同的名字。
- ③一个关系中的任意两个元组不能相同。
- ④关系的行列次序不影响其信息内容,可以任意交换。
- ⑤关系的每个量都必须是不可再分的基本数据项。
- ⑥每个关系都有一个关键字惟一标识它的元组。

因此,关系中的每个记录都是惟一的,所有记录都具有相同数据类型的字段,即所有记录都具有同样的固定长度和格式。关系模型具有如下性质。

①描述一致性。在关系模型中,无论是实体还是实体之间的联系,均用关系(二维表)或关系间的运算来描述。

②可以直接表示多对多的联系。

③关系最小化。每一个分量都是不可再分的数据项,即不允许表中有表。

④概念简单,操作方便,数据的独立性强。

⑤支持关系运算,特别是关系的投影、选择和连接运算。使用这些运算,以对二维表进行任意的分割和组装。

⑥关系可以为空关系,即不包括任何元组。

4 关系操作

在关系代数中,把关系看做是元组的集合。因此,集合中的定义与运算均适用于关系。

(1) 传统的集合运算

关系代数中传统的集合运算包括并、交、差等三种二元运算(即在两个关系之间进行)。设关系 R 与 S 具有相同的框架(即结构),它们的并(结果为包括在两个关系中的元组)、交(结果为两个关系中相同的元组)、差(结果为从一个关系中减去与另一个关系中相同的元组)将产生同一框架下的新关系。分别记作 $R \cup S$ 、 $R \cap S$ 和 $R - S$ 。

(2) 专门的关系操作

对关系的操作,常用的有三种:选择、投影与连接。

1) 选择操作

选择操作是一元操作(只对一个关系进行),用于从一个指定的关系中按给定条件在水平方向选择符合条件的元组。选择操作可用下面两种方式描述。

SELECT <关系名> WHERE <条件>

或者 $\sigma_{\langle \text{条件} \rangle}(\langle \text{关系名} \rangle)$

其中选择的条件通常是逻辑表达式描述。例如,要在关系工资表(见表1-1)中选择工资大于500的元组信息,便可描述为:

SELECT 工资表 WHERE 工资 > 500

或者 $\sigma_{\text{工资} > 500}(\text{工资表})$

2) 投影操作

投影操作也是一元操作(只对一个关系进行),用于从一个指定的关系中选择含有给定属性全部值的子集。投影可描述为:

PROJECT <关系名> ON <属性表>

或者 $\pi_{\langle \text{属性表} \rangle}(\langle \text{关系名} \rangle)$

例如,从关系工资表中仅取“姓名”和“扣款”构成新关系,可以描述为:

PROJECT 工资表 ON 姓名,扣款

或者 $\pi_{\text{姓名,扣款}}(\text{工资表})$

需要指出的是,经投影操作所得的新关系中不仅会取消关系中的若干属性,而且也有可能取消其中的某些元组。例如,新关系中如果有两个元组的属性值完全相同,按关系的定义应取消其中的一个元组。

综上所述,在经过选择或投影操作之后所得的新关系中,元组个数总是小于或等于原关系中的元组个数。

3) 连接操作

连接操作是二元操作(对两个关系进行)。用于按规定条件把两个关系中的所有元组以一切可能的组合拼接成为一个新的关系。对两个关系R与S进行连接操作,就是按给定条件同时在两个关系上进行选择,生成一个新关系。连接用下面的方法描述:

JOIN <关系1> AND <关系2> WHERE <条件>

或者 $\langle \text{关系1} \rangle \bowtie \langle \text{关系2} \rangle \langle \text{条件} \rangle$

例如,有关职工的人事情况和工资情况分别由关系职工表和关系工资表描述,如果要掌握人事和工资的所有资料,可以按职工的编号把两个关系连接起来。即:

JOIN 职工表 AND 工资表 WHERE 职工表[职工号] = 工资表[职工号]

或者 $\text{职工表} \bowtie \text{工资表} \text{ 职工表[职工号] = 工资表[职工号]}$

连接条件虽然可以多种多样,但如同上例那样以关键字值相等作为条件的连接是最常用的,称为自然连接。一般地说,两个关系进行自然连接的前提是两个关系中必须有N个($N \geq 1$)共同属性,以这N个对应属性值相等作为连接的条件就是自然连接,其表示形式可简化为:

$R \bowtie S$

在自然连接所得的新关系中,这N个属性只能出现一次,而两关系原来的其他属性仍被保持。因此,新关系属性个数等于原来两关系属性个数之和再减去N。

小 结

本章着重介绍了数据库的概念、数据模型、关系的定义和性质以及关系操作,要求读者初步了解数据库、数据库管理系统、数据库系统的基本概念,掌握数据的三种模型和三种关系操作,为后面学习关系型数据库管理系统打下基础。

习 题

1. 信息与数据两者在概念上有何区别? 相互间是一种什么关系?
2. 数据库主要任务是数据管理,数据管理的主要内容是什么?
3. 什么是数据库、数据库系统、数据库管理系统? 三者的关系是什么?
4. 数据模型有哪几种? 各自的特征是什么?
5. 一个关系应满足什么样的条件?
6. 关系操作有几种? 怎么来描述它们?

第 2 章 数据库的基本知识

1 FoxPro 简介

1.1 FoxPro 的发展和特点

20 世纪 80 年代中期,美国的 Fox 公司推出了关系型数据库管理系统 FoxBASE,它兼容于早年推出的 dBASE 系统。在同一时期,有许多公司都推出了自己独特的数据库产品,它们和 FoxBASE 具有类似的接口和语言。所有的这些产品形成了被称之为 xBASE 这种事实上的工业标准,它们占据了整个数据库市场的半壁江山。

到 20 世纪 90 年代,Fox 公司又推出了 FoxPro,它以快速的查询、灵活的操作等特点迅速占领了市场。

1.1.1 FoxPro 的发展

FoxPro 2.0 是 1991 年美国 Fox Software 公司自 FoxBASE 2.1 之后推出的新一代 PC 平台的关系数据库管理系统,它可以帮助用户收集、恢复和描述数据。FoxPro 集 FoxBASE 和 dBASE 的优点,以全新的面目出现。它与 FoxBASE 百分之百兼容,FoxBASE 的命令及函数在 FoxPro 中均可使用。FoxPro 的性能强大,运行速度高,有完整丰富的工具,友好的用户界面,数据存取方式简单,并具有良好的兼容性。FoxPro 是微机数据库管理系统的首选产品,它在 FoxBASE 的基础上引入集成环境、报表自动生成和关系举例检索等功能,并且提高了运行速度。

FoxPro 与 FoxBASE 一样,分为单用户版、多用户版。对于两种版本各增加一个 32 位增强型版本。

1992 年著名的 Microsoft 公司收购了 Fox 公司之后,于 1993 年 1 月推出了 FoxPro 2.5 for DOS 和 FoxPro 2.5 for Windows 两种版本。同年又推出了 FoxPro 2.5 b 及中文版,其中 DOS 版的 FoxPro 是在 DOS 系统下最好的微机数据库。1994 年推出的 FoxPro 2.6 较 FoxPro 2.5 增加了多种“向导”工具,但在程序开发方面没有明显的改进。1995 年 6 月,微软公司又推出了 Visual FoxPro 3.0 版,接着又很快推出了 Visual FoxPro 5.0 及其中文版。Visual FoxPro 的诞生使得 FoxPro 在程序开发方面有了很大的改进,它引入了当今流行的面向对象的编程思想。1998 年微软公司又发布了可视化编程语言集成包 Visual Studio 6.0, Visual FoxPro 6.0 (中文版)就是其中的成员之一。它是可运行于 Windows 9X、Windows NT 平台的 32 位数据库开发系统,能充分发挥 32 位微处理器的强大功能,是直观易用的编程工具。