

高校非计算机专业计算机基础教育改革型教材

Visual FoxPro

教程(2版)

GAOXIAOFEIJISUANJIZHUANYE
JISUANJICHUJIAOYUGAIGEXINGJIAOCAI

©卢雪松 主编



东南大学出版社
SOUTHEAST UNIVERSITY PRESS

Visual FoxPro 教程

(第 2 版)

卢雪松 主编



CQU1291602

东南大学出版社

· 南京 ·

内容提要

本书将数据库基本理论、数据库管理系统和数据库应用程序开发为一体,既注重基础理论,又强调实践。在提出面向对象概念的同时又兼顾了传统的面向过程的程序设计方法。本书图文并茂、示例翔实,便于读者阅读和上机操作。

本书是在总结作者多年的 VFP 教学经验的基础上,根据计算机等级考试大纲的要求组织编写,可供各高等院校本、专科学生使用,也适用于计算机等级考试二、三级的考生。

图书在版编目(CIP)数据

Visual FoxPro 教程 / 卢雪松主编. —2 版. —南京:

东南大学出版社, 2005. 8

ISBN 7-5641-0088-5

I. V... II. 卢... III. 关系数据库—数据库管理系统, Visual FoxPro—教材 IV. TP311.138

中国版本图书馆 CIP 数据核字 (2005) 第 078785 号

Visual FoxPro 教程(第 2 版)

主 编	卢雪松
责任编辑	张 煦
装帧设计	毕 真
出版发行	东南大学出版社
社 址	南京市四牌楼 2 号(210096)
经 销	江苏省新华书店
印 刷	江苏省通州市印刷总厂有限公司
版 次	2005 年 8 月第 2 版 2005 年 8 月第 8 次印刷
开 本	787×1092 mm 1/16
印 张	20
字 数	512 千字
印 数	24001~30000 册
定 价	29.80 元

凡因印装质量问题,请直接向出版社读者服务部调换。电话:025-83792328

再 版 前 言

随着计算机及其应用的推广普及,各行各业纷纷建立了以数据库为核心的信息系统。数据库技术在当今信息社会的地位和作用显得尤为重要。目前微型计算机上普遍使用的 Visual FoxPro 是一种可视化的(Visual)、面向对象的关系型数据库管理系统。

本书融数据库基本理论、数据库管理系统和数据库应用程序开发为一体,既注重基础理论,又强调实际应用。与一般教材不同的是,本书既突出了面向对象的概念,同时又兼顾到了传统的面向过程的程序设计方法。这是因为面向过程是面向对象的基础,离开这个基础,要进行面向对象的程序设计将举步维艰。本书图文并茂、示例典型、内容翔实,便于读者阅读和上机操作。

本书作者在总结多年的 Visual FoxPro 教学经验的基础上,根据计算机等级考试大纲的要求组织编写,可供各类高等院校本、专科学生使用。自本书第 1 版于 2001 年出版至今,广受读者欢迎,作者结合近 4 年的教学心得和等级考试考前培训的经验,对各章节内容进行了优化重组,并在每章后附上练习题,以求内容更实用、文字更精炼、示例更典型、习题更丰富。

本书由卢雪松主编。参加编写工作的有:杨晓秋(第 1 章)、徐晶(第 2 章)、方益明(第 3~4 章)、张平(第 5~6 章)、聂士澄(第 7 章)、卢雪松(第 8~9 章)、沈启坤(第 10~11 章)、楚红(第 12 章)。

在本书的编写过程中,参考了许多同类书籍,同时得到了扬州大学教务处有关同志的关心和支持,在此一并表示衷心的感谢。

由于编者水平有限,不妥之处敬请广大读者批评指正。

编 者

2005 年 7 月于扬州大学

目 录

第 1 章 数据库系统及 Visual FoxPro 概述	(1)
1.1 数据库系统的基本概念	(1)
1.2 数据模型	(5)
1.3 关系数据库理论基础	(8)
1.4 关系数据库系统简介	(13)
1.5 项目管理器	(17)
第 2 章 Visual FoxPro 的数据类型	(23)
2.1 数据类型	(23)
2.2 数据存储	(25)
2.3 函 数	(29)
2.4 表达式	(45)
第 3 章 表的使用	(54)
3.1 表结构的创建和使用	(54)
3.2 表记录的编辑修改	(57)
3.3 表的使用	(65)
3.4 表的索引	(69)
3.5 建立表之间的临时关系	(74)
第 4 章 数据库的创建和使用	(78)
4.1 VFP 数据库	(78)
4.2 数据库的设计过程	(78)
4.3 建立数据库文件	(81)
4.4 数据字典	(83)
4.5 数据库表字段的扩展属性	(84)
4.6 数据库表的表属性	(86)
4.7 表之间的关系	(89)
4.8 参照完整性	(90)
4.9 使用多个数据库	(92)
4.10 查看和设置数据库的属性	(93)
第 5 章 查询和视图	(97)
5.1 SQL 语言和查询技术	(97)
5.2 查询的创建和使用	(103)
5.3 视图的创建和使用	(115)
第 6 章 报表和标签	(125)
6.1 报表类型	(125)

6.2	创建报表	(126)
6.3	报表的预览与打印	(132)
6.4	设计标签	(134)
第 7 章	VFP 程序设计基础	(136)
7.1	创建、修改和运行应用程序	(136)
7.2	Visual FoxPro 基本语法结构	(137)
7.3	数组	(149)
7.4	过程和用户自定义函数	(153)
7.5	综合应用	(159)
第 8 章	表单	(169)
8.1	创建表单	(169)
8.2	对象的属性、事件和方法	(179)
8.3	常用的事件和方法	(182)
8.4	添加属性和方法程序	(188)
8.5	多文档界面与表单类型	(190)
第 9 章	控件	(193)
9.1	标签	(193)
9.2	命令按钮	(194)
9.3	文本框	(196)
9.4	编辑框	(200)
9.5	微调框	(201)
9.6	列表框	(202)
9.7	组合框	(204)
9.8	复选框	(205)
9.9	选项按钮组	(206)
9.10	命令按钮组	(207)
9.11	表格	(209)
9.12	页框	(211)
9.13	计时器	(213)
9.14	图形控件	(214)
9.15	图像控件	(216)
9.16	OLE 控件	(217)
第 10 章	类	(220)
10.1	VFP 的类层次	(220)
10.2	类的创建与修改	(223)
10.3	扩展基类	(230)
10.4	为类指定外观	(231)
10.5	类的应用	(233)
第 11 章	菜单和工具栏设计	(239)
11.1	设计菜单	(239)

11.2 创建自定义工具栏.....	(251)
第 12 章 数据库应用系统开发	(258)
12.1 开发 VFP 数据库应用系统的一般步骤	(258)
12.2 教学管理系统的开发.....	(263)
12.3 应用程序的管理和发布.....	(278)
附录 1 ASCII 码表	(289)
附录 2 VFP 常用函数	(290)
附录 3 VFP 常用命令	(295)
附录 4 控件的主要属性、事件和方法	(302)
参考文献.....	(310)

第1章 数据库系统及 Visual FoxPro 概述

数据库是数据处理的重要工具,也是管理信息系统(MIS)、办公自动化(OA)系统和决策支持系统(DSS)等应用系统的核心部分,能够有效、合理地存储各种数据,为有关应用准确、快速地提供有用的信息。数据库技术是计算机领域中最重要技术之一,其应用已渗透到人类社会各个领域,并正在改变着人们的生活方式和工作方式。因此,我们有必要学习和掌握数据库系统的原理和技术,用以解决各种计算机应用中的实际问题。

1.1 数据库系统的基本概念

1.1.1 数据管理技术的发展

数据管理技术是指对数据进行分类、组织、编码、存储、检索和维护的技术。在计算机环境下,数据管理技术经历了从低级到高级的3个发展阶段:

1. 人工管理阶段(20世纪50年代中期以前)

该阶段尚无统一的数据管理软件,对数据的管理完全由各个程序员在其程序中进行管理。程序员在编制其课题程序时,必须考虑数据的逻辑定义和组织、数据存放的存储设备、物理存储方式和地址分配,并通过物理地址来存取数据,表示处理流程的程序与其处理对象——数据相互结合成一个整体。两者紧密地相互依赖,数据的管理仍然是分散的,计算机在数据管理中还没有发挥应有的作用。因此,严重地影响了计算机的使用效率。

2. 文件系统管理阶段(20世纪50年代后期至60年代中期)

在这一阶段,由于计算机技术的发展,出现了磁带、磁鼓和磁盘等容量较大的存储设备。软件方面有了操作系统,计算机的应用范围也由科学计算领域扩展到数据处理领域。

这一阶段的特点是:

① 数据可以以操作系统的文件形式长期保存在计算机中,文件的组织方式由顺序文件逐步发展到随机文件。

② 操作系统的文件管理系统提供了对数据的输入和输出操作接口,进而提供数据存取方法。

③ 一个应用程序可以使用多个文件,一个文件可为多个应用程序使用,数据可以共享。

④ 数据仍然是面向应用的,文件之间彼此孤立,不能反映数据之间的联系,因而仍存在数据的大量冗余和不一致性。

3. 数据库系统管理阶段(20世纪60年代后期开始)

针对数据的文件管理方式存在的上述缺点,计算机软件工作者经过长期不懈的努力,提出了数据库的概念。数据库技术为数据管理提供了一种较完善的高级管理方式,它克服了文件系统方式下分散管理的弱点,对所有的数据实行统一、集中的管理,使数据的存储独立于使用它的程序,从而实现数据共享。这样也就克服了文件管理方式的缺点。如图1-1所示。

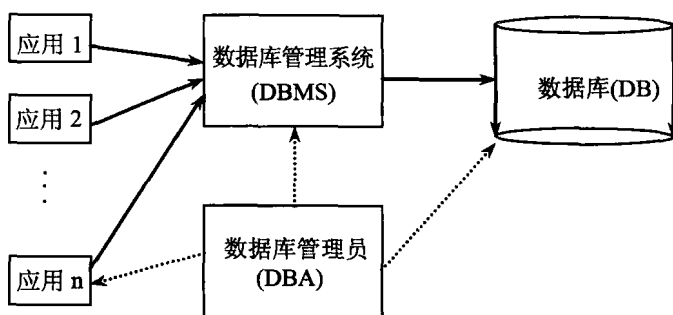


图 1-1 数据库系统

1.1.2 数据库的特点

现在数据库已经成为各种计算机应用系统的核心部分。其所以如此,是因为数据库有着许多独特的优点,它们主要是:

一、数据被集成化

所谓集成化就是按照一定的数据模型来组织和存放数据,故而又称数据的结构化。数据的集成化或结构化是数据库系统管理和文件系统管理之间的一个本质区别,是实现数据的集中控制和减少数据冗余的前提和保证。

由于数据库是从一个组织的全部应用来全盘考虑并集成数据的结构,所以数据库中的数据不再是面向个别应用,而是面向系统。各个不同的应用系统所需的数据只是整体模型的一个子集。

二、具有数据独立性

所谓数据独立性就是数据与应用程序之间不存在相互依赖关系,也就是数据的逻辑结构、存储结构和存取方法等,不因应用程序的修改而修改,反之亦然。这是数据库系统与文件系统之间的另一个重要区别,也是使得数据库系统结构复杂的一个重要原因。数据独立性分为物理独立性和逻辑独立性两级。

1. 物理独立性

即数据的物理结构(或存储结构)的改变,如物理存储设备的更换、物理存储位置的变更、存取方法的改变等等,不影响数据库的逻辑结构,从而不致引起应用程序的修改。

2. 逻辑独立性

数据库总体逻辑结构的改变,如修改数据的定义、增加新的数据类型、改变数据间的联系等等,无需修改原来的应用程序,这就是数据的逻辑独立性。遗憾的是,到目前为止,数据的逻辑独立性还没有完全彻底地实现。

三、实现数据共享

数据共享是促成发展数据库技术的重要原因之一,也是数据库技术先进性的一个重要体现。数据库中数据的共享性体现在两个方面:

(1) 数据库中的数据可供多个应用程序用于不同的目的,每个应用程序各有其自己的局部数据逻辑结构。数据库中的数据不但可供现有的各个应用程序共享,还可开发新的应用程序而无需附加新的数据,实现新、老应用程序共享数据库中的数据。

(2) 数据库可以为批处理用户和终端的多个用户同时共享。

四、数据的冗余度小

数据冗余是指相同的数据重复出现。在非数据库系统中,每个应用有它自己的文件,从而造成了存储数据的大量重复。大量的数据冗余带来的弊病是:

(1) 浪费了存储空间。

(2) 为了更新某些冗余副本,保持数据的一致性,必须执行多次更新操作,从而增加不必要的更新阶段和机器时间。

(3) 由于数据的不同副本可能处于不同的更新阶段,从而可能给出不一致的信息。

数据库则是从整体观点来组织和存储数据的,数据是集成化、结构化的。它是为多种应用所共享的,从而大大减少了数据冗余。

五、具有统一的数据控制功能

由于数据库是多用户的共享资源,而计算机的共享一般是并发的,即多用户同时使用数据库,因此,数据的安全性和可靠性是一个数据库系统能否实用的关键问题,所以必须采用有效措施保护数据。

1. 安全性控制

这里的安全性,主要是指数据的保密性控制,例如用户的身份号,验明身份后才能进入系统,对某些特定的数据限定使用权限;对不同的操作采用不同的保护级别(如有的用户只允许查询,有的用户有修改权);采用密码方式存放数据等。

2. 完整性控制

指利用某些条件判断来排除不正确的数据,它包括正确性(如数值型数据中不应含有字母)、有效性(如表示月份必须为1至12的正整数)、相容性(如一人的工资在两个地方不应出现不同的数值,数据不相容主要是由于数据冗余引起的)的控制。

3. 并发控制

避免一个用户读出另一个用户正在修改的错误数据。

4. 故障的发现与恢复

防止硬件和软件故障以及用户操作上的错误所造成的对数据的破坏。

1.1.3 数据库系统

数据库系统(Data Base System, 简称 DBS)是指在计算机环境下引进数据库技术后构成的整个计算机系统。除计算机软、硬件环境外,数据库系统包括以下几个组成部分:

一、数据库(Data Base, 简称 DB)

数据库被定义为是在计算机存储设备上合理存放的、互相关联的数据集合。

二、数据库管理系统(Data Base Management System, 简称 DBMS)

数据库管理系统是数据库系统中专门用于数据管理的软件,是用户与数据库的接口。提供有:

1. 数据库定义功能

提供数据描述语言(Data Defined Language, 简称 DDL)及其翻译程序,用于定义数据库结构(模式及模式间映射)、数据完整性和保密性约束等。

2. 数据库操作功能

提供数据操纵语言(Data Manipulation Language, 简称 DML)及其翻译程序,用于实现对数据库数据的查询、插入、更新和删除等操作。

数据操纵语言亦称数据库语言。一般有两种使用方式:一种是交互式命令语言,它语法简明,可以独立使用,称为自含型语言;另一种是嵌入到某种程序设计语言中,如 C 语言、FORTRAN 语言、COBOL 语言、PASCAL 语言等,称为宿主型语言。

3. 数据库运行和控制功能

包括数据安全性控制、数据完整性控制、多用户环境的并发控制等。

4. 数据库维护功能

包括数据库数据的载入、转储和恢复,数据库的维护和数据库的功能及性能分析和监测等。

三、应用程序

应用程序是数据库中特定用户的数据处理业务,利用 DBMS 支持的程序设计语言编写的程序。

四、人员

参与分析、设计、管理、维护和使用数据库中数据的人员都是数据库系统的组成部分,分析、设计、管理和使用数据库系统的人员主要是:

1. 数据库管理员(DataBase Administrator, 简称 DBA)

数据库管理员是管理和维护数据库系统正常运转的专职人员。具体的职责包括:①决定数据库中的数据内容和结构;②决定数据库的存储结构和存取策略;③定义数据的安全性和完整性约束条件;④监控数据库的使用和运行;⑤数据库的重构和重组。

2. 系统分析员

系统分析员是数据库系统建设期主要的参与人员,负责应用系统的需求分析和规范说明,确定系统的基本功能、数据库结构和应用程序的设计以及软、硬件配置,并组织整个系统的开发。

3. 应用程序员

应用程序员根据应用系统的功能需求负责设计和编写应用系统的程序模块,并参与程序模块的测试。

4. 用户

这里用户是指最终用户(End User)。

目前,微型机到大型计算机都有数据库系统的应用。一般而言,大型机上的系统是多用户系统,系统提供的设施较完整,功能也较强。微型机上的系统大多是单用户系统,这种系统提供的设施往往是大型多用户系统的一个子集,即使设施相同,也没有大型多用户系统那么强的功能。

1.1.4 数据库系统体系结构

数据库的体系结构是数据库系统的一个总框架。尽管实际数据库软件产品种类繁多,使用的数据库语言各异,基础操作系统不同,采用的数据结构模型相差甚大,但是绝大多数数据库系统在总体结构上都具有三级模式的结构特征。数据库的三级模式结构由外模式、模式和内模式组成,如图 1-2 所示。

(1) 外模式:又称子模式或用户模式,是模式的子集,是数据的局部逻辑结构,也是数据库用户看到的数据视图。

(2) 模式:又称逻辑模式或概念模式,是数据库中全体数据的全局逻辑结构和特性的描

述,也是所有用户的公共数据视图。

(3) 内模式:又称存储模式,是数据在数据库系统中的内部表示,即数据的物理结构和存储方式的描述。

数据库系统的三级模式是对数据的三级抽象。为了实现 3 个抽象层次的转换,数据库系统在三级模式中提供了两次映像:外模式/模式映像和模式/内模式映像。所谓映像就是存在某种对应关系。

外模式到模式的映像,定义了外模式与模式之间的对应关系。模式到内模式的映像,定义了数据的逻辑结构和物理结构之间的对应关系。正是由于这两级映像,使数据库管理的数据具有两个层次的独立性:物理独立性和逻辑独立性。

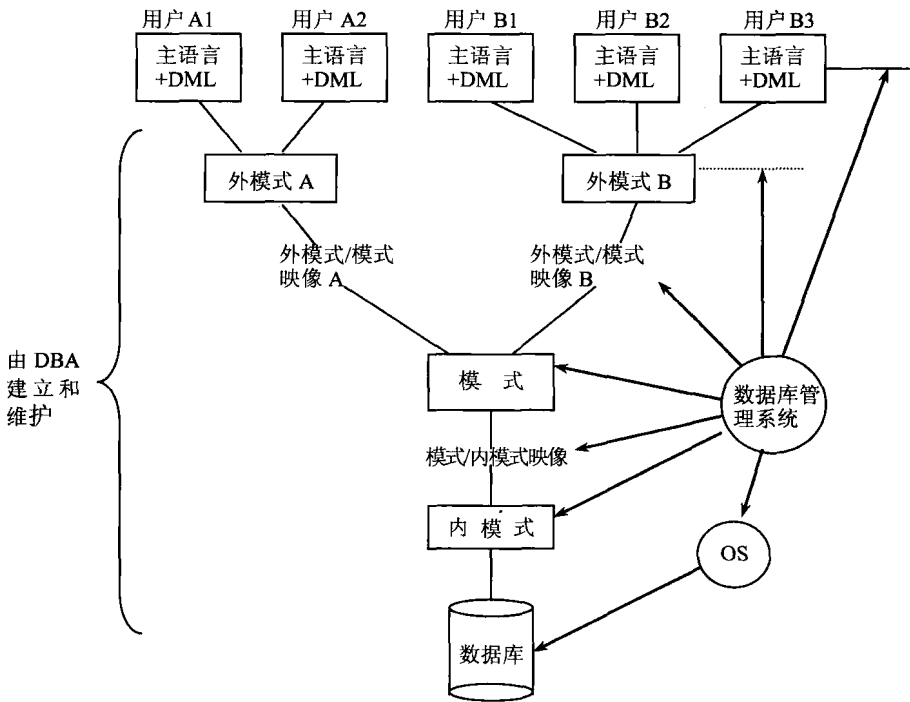


图 1-2 数据库系统的体系结构

1.2 数据模型

模型一词本身含意是现实世界事物的一种抽象,例如汽车模型、船舶模型等。

数据模型是数据库结构和语意的一种抽象,通俗地说,数据模型是现实世界事物及事物之间联系的数据表示,从数据组织形式而言,数据库的数据结构形式就是数据模型。

1.2.1 概念模型

从现实世界到数据世界,实际上经历了“现实世界”→“信息世界”→“数据世界”3 个世界的演变过程,信息世界是现实世界事物间的内在联系在人们头脑中的反映。其认识、理解、抽象和表达的结果,用概念模型来表示,又称“信息模型”。

一、几个术语

信息世界涉及的主要概念有:

1. 实体(Entity)

客观存在并可相互区分的事物。实体可以指实际的对象,也可以指某些概念。例如一个职工、一个学生、一个部门、一门课、一次定货等。

2. 属性(Attribute)

实体所具有的某一特性。一个实体可以由若干个属性来刻画。例如,某学生实体可以由学号、姓名、年龄、性别、班级等属性组成。(970103331,王平,21,男,计 9701)这些属性值组合起来表征了一个学生。

3. 关键字(Key)

惟一标识实体的属性集称为关键字,又称码、键。例如,学号是学生实体的关键字。

4. 域(Domain)

某个(些)属性的取值范围。例如,学号的域为 9 位数字,姓名的域为字符串集合,年龄的域为小于 35 的整数,性别的域为(男,女)。

5. 实体型(Entity type)与实体集(Entity Set)

具有相同属性的实体具有共同的特征和性质。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、年龄、性别、班级)是一个实体型。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。

6. 联系(Relationship)

现实世界的事物之间是有联系的,这种联系必然要在信息世界中加以反映,一般存在两类联系:一是实体内部的联系,如组成实体的属性之间的联系;一是实体之间的联系。

二、联系的类型

这里我们讨论的是实体型之间的联系。两个实体型之间的联系可分为 3 类。

1. 一对一联系(1:1)

若对于实体集 A 中的每一个实体,实体集 B 中至多有一个实体与之联系,反之亦然,则称实体集 A 与实体集 B 具有一对一联系,记为 1:1。

2. 一对多联系(1:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n \geq 0)$ 个实体与之联系,反之,对于实体集 B 中的每一个实体,实体集 A 中至多只有一个实体与之联系,则称实体集 A 与实体集 B 具有一对多的联系。记为 1:n。

3. 多对多联系(m:n)

若对于实体集 A 中的每一个实体,实体集 B 中有 $n(n \geq 0)$ 个实体与之联系。反过来,对于实体集 B 中的每一个实体,实体集 A 中也有 $m(m \geq 0)$ 个实体与之联系,则称实体集 A 与实体集 B 具有多对多联系。记为 m:n。

例如:一个部门有一个经理,而每个经理只在一个部门任职,则部门和经理之间具有一对一联系;一个部门有若干职工,而每个职工只在一个部门工作,则部门与职工之间是一对多的联系;一个项目有多个职工参加,而一个职工可以参加若干项目的工作,则项目和职工之间具有多对多的联系。

三、实体联系模型(Entity Relationship Model)

概念模型的表示方法最常用的是实体联系方法。这个方法是用 E-R 图来描述某一组织的概念模型,这种模型也称为实体联系模型(E-R 模型)。

在 E-R 图中,实体用“方框”表示,属性用“椭圆框”表示,联系用“菱形框”表示,再用线

段将各个成分连接起来并注明联系的类型,若实体之间联系也具有属性,则把属性和菱形也用线段连接上。我们可以用 E-R 图表示学生—课程的概念模型,如图 1-3。

E-R 模型是独立于计算机系统的模型,简单、清晰,易于用户理解,但现有数据库系统没有一个能直接接受 E-R 模型。主要是因为 E-R 模型只能说明实体以及实体间语义的联系,还不能进一步说明详细的数据结构。一般遇到一个实际问题总是先设计一个 E-R 模型,然后再把 E-R 模型转换成计算机能实现的结构数据模型。

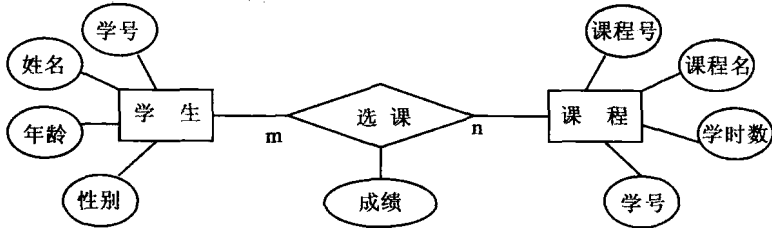


图 1-3 学生选课实体联系模型

1.2.2 结构数据模型

结构数据模型是直接面向数据库的逻辑结构,它是现实世界的第二层抽象。这类模型有严格的形式化定义,以便于在计算机系统中实现,又简称“数据模型”。

一、常用的数据模型

常用的数据模型主要有层次、网状和关系 3 种类型。

1. 层次模型 (Hierarchical Model)

层次模型采用树状结构表示实体及其联系,适合于表示实体之间 1:n 联系。其结构自顶向下层次分明,如一个学校行政机构可以抽象成为一个层次模型。对层次模型的数据搜索仅允许自上向下的单向搜索,故使得层次模型中数据操纵受到很大限制。1969 年,IBM 公司推出 IMS 系统,它是最典型的层次模型系统。

2. 网状模型 (Network Model)

网状模型采用结点间的连通图(网状结构)表示实体及其联系,能表示实体之间各种复杂联系情况。网状模型是美国 DASYL 委员会数据库任务组(DBTG)于 1969 年提出的一种模型。在网状模型中,对数据的搜索可以用两种方式:

- (1) 可以从网络中任一点开始搜索。
- (2) 可沿着网中的路径按任意方向搜索,但在计算机中实现较为困难,使用不太方便。

3. 关系模型 (Relational Model)

关系模型采用“二维表”表示实体及其联系,能直接表示实体之间各种复杂联系情况。关系数据模型具有简单明了、理论严谨等优点,是 3 种数据模型中最重要的模型。目前常见的数据库管理系统都是依据关系模型的。

二、面向对象的数据模型

现实世界中存在着许多含有复杂数据结构的应用领域,如 CAD 数据、图形数据等,人们需要更高级的数据模型来表达这类信息。面向对象的数据模型就是面向对象概念与数据库技术相结合的产物。

1.3 关系数据库理论基础

关系数据库是当今数据处理型信息系统中最常用也是最有效的一种数据库。关系数据库采用关系数据模型。在本节中将对关系数据模型作进一步的阐述。关系数据模型简称关系模型,它包含数据结构、数据操作和数据完整性约束 3 个部分。

1.3.1 关系模型的数据结构和基本术语

一、关系模型的数据结构

在关系模型中,无论是实体还是实体之间的联系均由单一的结构类型即关系来表示。数据在用户观点下的逻辑结构是一张张二维表。图 1-4 是一张学生档案的二维表,它与学生实体集相对应。下面我们以它为例,介绍关系模型中的主要术语。

二、关系模型的基本术语

- (1) **关系**: 一张二维表对应一个关系,代表一个实体集。
- (2) **元组**: 表中的一行称为一个元组,代表一个实体。一个关系可看成是元组的集合。
- (3) **属性**: 表中的一列称为属性,给每一列起一个名称即属性名。
- (4) **域**: 属性的取值范围。
- (5) **关系模式**: 在二维表中,行定义(记录的型)称为关系模式,关系模式是对关系的描述,它包括关系名、组成该关系的诸属性名等。
- (6) **关键字**: 关系中的某一属性组,若它的值惟一地标识了一个元组,则称该属性组为关键字或码(或键)。对于图 1-4 中的学生关系和班级关系,其关键字分别是学号和班号。

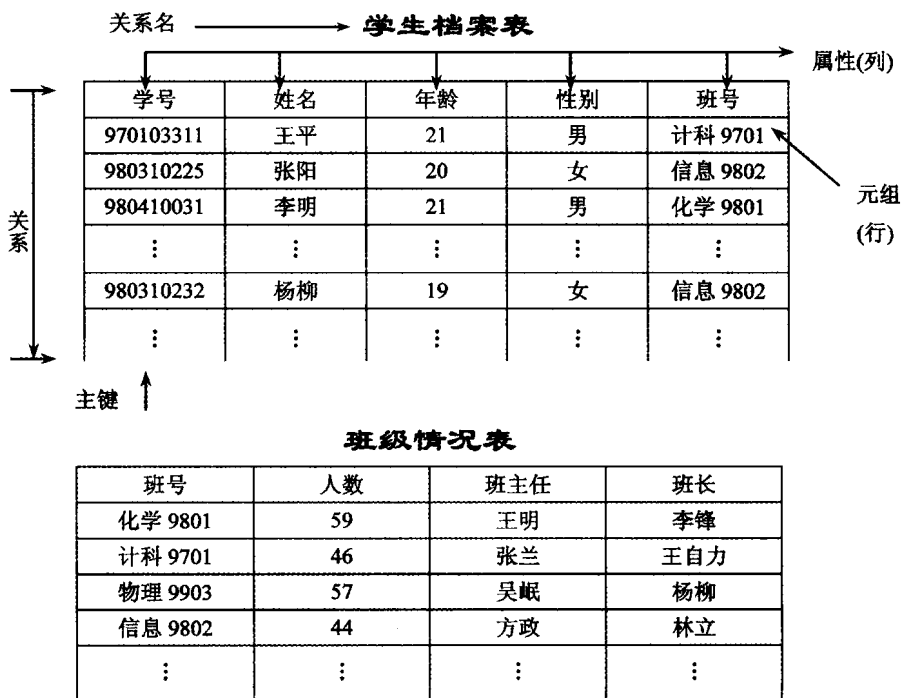


图 1-4 关系数据结构示例

学生关系中一般姓名不能作为关键字,尽管当前该关系中没有两个元组有相同的姓名,但不能保证在将来的任何时候不会有同名的学生进入学生关系。有些关系的关键字是由单个属性组成的,还有一些关系的关键字常常是由若干个属性组合而构成的,即这种关系中的元组不能由任何一个属性惟一标识,必须由多个属性的组合才能惟一标识。如考试成绩关系:考试成绩(学号、考试日期、考试科目、姓名、性别、成绩),它的关键字由(学号、考试日期、考试科目)属性的组合构成。

● **超关键字** 在一个关系中,能用来惟一标识该关系元组的单个属性或属性组(由多个属性构成),称为该关系的超关键字。根据关系的性质,一个关系中必定存在超关键字。当然,就惟一标识关系的元组来讲,超关键字中可能含有多余的属性。

● **候选关键字** 如果一个超关键字中去掉任一属性后不再能惟一标识该关系的元组,则称该关键字为候选关键字。因此,候选关键字是最精简的超关键字。一个关系可以有一个或多个候选关键字。

● **主关键字** 在一个关系的若干个候选关键字中,选定其中一个作为主关键字,亦称主键。

● **外部关键字** 若关系中的某属性虽非该关系的主关键字,但却是另一关系的主关键字,则称其为该关系的外部关键字。例如,对于图 1-4 中的学生关系,班号不是主关键字,但班号是班级关系的主关键字,所以班号是学生关系的外部关键字或外键。

在外键相关联的两个表中,外键所在的表称为从表;以外键作为主键的表称为主表。如图 1-4 中学生关系对外键“班号”而言,它是从表,而班级关系是主表。

(7) 关系数据库:使用关系模型所设计的数据库称为关系数据库。关系数据库有型和值之分,关系数据库的型由若干个关系模式构成;关系数据库的值由在某一时刻各关系模式对应的关系构成。在一个关系数据库中,关系模式是稳定的,而关系是随时间不断变化的,因为数据库中的数据在不断更新。

三、关系的性质

可以从二维表去理解关系的性质:

- (1) 关系中每个元组(记录)的分量必须是原始的,即表的每一列都是不可再分的。
- (2) 每一列的值只能取自同一个域。
- (3) 列的次序可以任意交换。
- (4) 行的次序可以任意交换。
- (5) 不能出现完全相同的两行。

1.3.2 关系模型的完整性约束条件

关系模式是通过关系数据描述语言描述关系后所生成的关系框架。

作为关系 DBMS,为了维护数据库的完整性,一般对关系模式提供了以下 3 类完整性约束机制:

一、实体完整性约束规则

指关系中的“主键”不允许取“空值”(NULL)。因为关系中的每一记录都代表一个实体,而任何实体都是可标识的,如果主键值为空,就意味着存在不可标识的实体。

二、引用完整性约束规则

亦称参照完整性约束规则。对一个关系上的外键,其值只允许两种可能:一是空值;二

是等于外键对应的关系的主键值。这是由于不同关系之间的联系是通过“外键”实现的,当一个关系通过外键引用另一关系中的记录时,它必须能在引用的关系中找到这个记录,否则无法实现联系。

例如图 1-4 的学生关系和班级关系,其中班级关系中的班号是主键。学生关系中,对每个学生也有班号一项,表明这个学生是属于哪个班的。学生关系中的班号是外键,它要么取空值,表示这个学生还未分配到任何一个班级;要么取值必须与班级关系中的某个元组的班号相同,表示这个学生属于某个班级。若学生关系中某个学生的班号取值不能与班级关系中任何一个元组的班号值相同,表示这个学生是不存在的一个班的学生,这与实际应用环境是不相符的,显然是错误的。这就是为什么关系模型中定义了参照完整性约束规则。

三、用户定义的完整性约束

这是针对某一应用环境的完整性约束条件,它反映了某一具体应用所涉及的数据应满足的要求,往往是对关系模式中的数据类型、长度、取值范围的约束,如学生的成绩不能是负数。DBMS 提供定义和检验这类完整性规则的机制,其目的是用统一的方式由系统来处理它们,不再由应用程序来完成这项工作。

1.3.3 关系数据操纵与关系代数

关系型数据库管理系统提供关系操纵语言来实现数据操纵。关系操纵语言分为关系代数与关系演算两大类。两类语言可表达的数据操纵能力是相当的。关系代数通过对关系的运算来表达查询。关系代数的运算可包括传统的集合操作和专门的关系操作两类。

一、传统的集合操作

这类操作将关系看成元组的集合。其操作是从关系的水平方向,即是对关系的行来进行的。设关系 R 和关系 S 具有相同数目的属性列(n 列属性),并且相应的属性取自同一个域,则可定义以下 4 种集合运算:

(1) 并(Union):关系 R 与关系 S 的并,记为 $R \cup S$ 。它是属于 R 或属于 S 的元组组成的集合,结果为 n 列属性的关系。

(2) 交(Intersection):关系 R 与关系 S 的交,记为 $R \cap S$ 。它是既属于 R 又属于 S 的元组组成的集合,结果为 n 列属性的关系。

(3) 差(Difference):关系 R 与关系 S 的差,记为 $R - S$ 。它是属于 R 而不属于 S 的元组组成的集合,结果为 n 列属性的关系。

(4) 广义笛卡尔积(Extended Cartesian product):关系 R (假设为 n 列)和关系 S (假设为 m 列)的广义笛卡尔积,记为 $R \times S$,是一个 $(m + n)$ 列元组的集合,每一个元组的前 n 列是来自关系 R 的一个元组,后 m 列是来自关系 S 的一个元组。若 R 有 k_1 个元组, S 有 k_2 个元组,则关系 R 和关系 S 的广义笛卡尔积有 $k_1 \times k_2$ 个元组。

二、专门的关系操作

这类操作不仅涉及行,而且也涉及列。

(1) 选择(Selection):选择操作是指在关系中选择满足某些条件的元组,记为 $\sigma_F(R)$ 。例如,要在学生基本档案表中找出年龄为 24 岁的所有学生数据,就可以对学生关系做选择操作: $\sigma_{\text{年龄}='24'}(\text{学生})$ 。

(2) 投影(Projection):投影操作是在关系中选择某些属性列。记为 $\Pi_A(R)$ 。例如,要找出所有班级的班主任和班长,则可以对班级关系做投影操作: $\Pi_{\text{班主任,班长}}(\text{班级})$ 。