

高等教育 21 世纪课程教材

Visual FoxPro

CHENGXU SHEJI JIAOCHENG

Visual FoxPro

程序设计教程

主 编 杨克昌 莫 照

主 审 蒋加伏

 湖南科学技术出版社

高等教育 21 世纪课程教材

Visual FoxPro

CHENGXU SHEJI JIAOCHENG

Visual FoxPro

程序设计教程

主 编 杨克昌 莫 照

主 审 蒋加义

 湖南科学技术出版社

高等教育 21 世纪课程教材

Visual FoxPro 程序设计教程

主 编:杨克昌 莫 照

副 主 编:王岳斌 肖伟平 杨秀平 胡虚怀

主 审:蒋加伏

责任编辑:曹 阳

文字编辑:胡捷晖

出版发行:湖南科学技术出版社

社 址:长沙市湘雅路 276 号

<http://www.hnstp.com>

印 刷:长沙化勘印刷有限公司

(印装质量问题请直接与本厂联系)

厂 址:长沙市青园路 4 号

邮 编:410004

出版日期:2005 年 7 月第 1 版第 3 次

开 本:787mm×1092mm 1/16

印 张:19.75

字 数:483000

书 号:ISBN 7-5357-3575-4/TP·170

定 价:28.00 元

(版权所有·翻印必究)

前 言

以计算机为核心的信息科学的迅猛发展与广泛应用,正在对人类社会的发展进程以至人们的工作方式与思维方式的改变产生深远的影响。计算机技术的应用与发展,是一个国家综合国力的体现与科技发展水平的象征。进入 21 世纪,计算已成为与理论研究、科学实验相并列的第三种研究方法。作为人类智慧的结晶与人脑功能的延伸,计算机已经成为人类进行复杂计算与模拟探索的“通用智能工具”,广泛应用于现代社会的各个领域,并发挥着越来越大的作用。

面向 21 世纪的高层次人才都需要学习和掌握数据库的基本知识和数据处理的基本方法,以开发出适用的数据库应用系统。《Visual FoxPro 程序设计教程》就是为满足这种需求而编写的,它既可以作为高等院校数据库应用课程的教材,又可供社会各类计算机应用人员阅读参考。

Visual FoxPro 是新一代小型数据库管理系统的杰出代表,它以完善的性能、丰富的工具、较高的处理速度、友好的界面以及完备的兼容性等特点,备受广大用户的欢迎。Visual FoxPro 提供了一个集成化的系统开发环境,它使数据的组织与操作变得简单方便。它在语言体系方面做了强大的扩充,不仅支持传统的结构化程序设计,而且支持面向对象程序设计,并拥有功能强大的可视化程序设计工具。利用可视化的设计工具和向导,用户可以快速创建表单、菜单,查询和打印报表。目前,Visual FoxPro 是数据库应用系统较为理想的开

发工具之一,也成为首选的数据库教学软件。

Visual FoxPro 及其中文版是可运行于 Windows 98、Windows 2000和 Windows NT 平台的 32 位数据库开发系统。本书以高等院校非计算机专业学生为对象,以计算机等级考试纲要为依据,以课程教学为主线,系统讲授 Visual FoxPro 数据库技术与程序设计。全书主要内容有:数据库系统基础知识,Visual FoxPro 基础,数据表的基本操作,项目管理器与数据库操作,SQL 语言与查询,Visual FoxPro 程序设计,表单与面向对象程序设计,菜单、报表与标签设计,应用程序开发。本书在编写中力求做到概念清晰,注重基础,突出重点,面向教学,方便实用。

为了方便教学和上机操作练习,编者配套编写了《Visual FoxPro上机实验指导与习题》一书,配套制作了教学课件(附习题解答),供教师教学参考。

本书可以作为高等院校数据库应用课程的教材,也可供社会各类计算机应用人员与参加各类计算机等级考试的读者阅读参考。

本书由杨克昌、莫照任主编,王岳斌、肖伟平、杨秀平、胡虚怀任副主编,蒋加伏任主审。参加编写的教师还有许静、王鑫、陈忠文、梁小芝等。

由于编者学识水平有限,书中的疏漏或错误之处在所难免,恳请广大读者批评指正。

编 者

2004 年 7 月

目 录

第1章 数据库系统基础知识	(1)
1.1 信息、数据与数据处理	(1)
1.1.1 数据与数据处理	(1)
1.1.2 计算机数据管理技术的发展	(2)
1.2 数据模型	(4)
1.2.1 实体及其联系	(4)
1.2.2 数据模型	(5)
1.3 数据库系统	(8)
1.3.1 数据库系统的组成	(8)
1.3.2 数据库系统的特点	(9)
1.3.3 关系数据库	(10)
1.3.4 Visual FoxPro 系统的发展	(11)
第2章 Visual FoxPro 基础	(12)
2.1 Visual FoxPro 的特点、安装与启动	(12)
2.1.1 Visual FoxPro 的特点	(12)
2.1.2 安装与启动	(13)
2.2 Visual FoxPro 的用户界面	(15)
2.2.1 系统菜单	(15)
2.2.2 命令窗口	(17)
2.2.3 工具栏、状态栏与帮助系统	(17)
2.3 Visual FoxPro 命令概述	(18)
2.3.1 命令与短语	(18)
2.3.2 命令的书写规则	(20)
2.4 Visual FoxPro 的数据类型、常量与变量	(20)
2.4.1 数据类型	(20)
2.4.2 常量	(22)
2.4.3 变量	(24)
2.5 Visual FoxPro 的常用函数	(28)
2.5.1 数值函数	(28)

2.5.2	字符函数	(29)
2.5.3	日期和时间函数	(31)
2.5.4	数据类型转换函数	(32)
2.5.5	测试函数	(34)
2.6	Visual FoxPro 的表达式	(35)
2.6.1	算术表达式	(36)
2.6.2	字符表达式	(36)
2.6.3	日期和时间表达式	(37)
2.6.4	关系表达式	(37)
2.6.5	逻辑表达式	(38)
第3章	数据表的基本操作	(40)
3.1	VFP 数据表的概念	(40)
3.2	数据表的建立、打开与关闭	(40)
3.2.1	设计数据表文件结构	(40)
3.2.2	创建数据表	(41)
3.2.3	数据的输入	(43)
3.2.4	数据表的打开与关闭	(45)
3.3	数据表结构的显示与修改	(46)
3.3.1	数据表文件结构的显示	(46)
3.3.2	修改数据表文件结构	(47)
3.4	记录的显示、追加、定位与插入	(48)
3.4.1	记录显示命令	(48)
3.4.2	记录的追加	(51)
3.4.3	记录指针的定位	(54)
3.4.4	记录的插入	(56)
3.5	记录的删除与修改	(56)
3.5.1	记录的删除	(57)
3.5.2	记录的修改	(58)
3.6	复制数据表文件	(61)
3.6.1	表文件结构与数据的复制	(61)
3.6.2	表文件的结构文件	(62)
3.7	表的排序与索引	(64)
3.7.1	排序	(65)
3.7.2	索引	(66)
3.7.3	建立索引文件	(67)

3.7.4 索引文件的使用	(68)
3.7.5 索引的更新	(70)
3.8 数据查询	(71)
3.8.1 直接查询	(71)
3.8.2 索引查询	(72)
3.9 数据表的统计与汇总	(73)
3.9.1 数据统计	(74)
3.9.2 分类求和	(76)
3.10 多个表的操作	(77)
3.10.1 多工作区	(77)
3.10.2 表的关联	(78)
3.10.3 表的联接	(79)
第4章 项目管理器与数据库操作	(80)
4.1 项目管理器	(80)
4.1.1 项目管理器的功能	(80)
4.1.2 项目管理器的界面操作	(81)
4.1.3 项目管理器的使用	(84)
4.2 数据库操作	(86)
4.2.1 建立数据库文件	(86)
4.2.2 向数据库添加自由表	(87)
4.2.3 数据库的打开与关闭	(88)
4.2.4 数据库的修改	(89)
4.2.5 数据库的删除	(89)
4.3 数据库中表的使用	(90)
4.3.1 为数据库表建立索引	(90)
4.3.2 建立表之间的永久联系	(90)
4.3.3 设置参照完整性	(91)
4.3.4 在数据库中直接建立表	(93)
4.3.5 修改数据库中的表	(96)
第5章 SQL 语言与查询	(97)
5.1 结构化查询语言 SQL	(97)
5.1.1 SQL 概述	(97)
5.1.2 SQL 语句	(98)
5.1.3 SQL 功能简介	(99)
5.1.4 Visual Foxpro 中 SQL 的数据定义	(100)

5.1.5 Visual Foxpro 中 SQL 数据操纵	(102)
5.2 数据查询	(103)
5.2.1 单表查询	(104)
5.2.2 联接查询	(106)
5.2.3 嵌套查询	(108)
5.2.4 查询结果处理	(109)
5.3 创建和使用视图	(111)
5.3.1 视图设计器	(112)
5.3.2 使用视图更新数据	(121)
5.3.3 视图的 SQL 语句	(122)
第 6 章 Visual Foxpro 程序设计	(125)
6.1 程序设计基础	(125)
6.1.1 程序设计的概念	(125)
6.1.2 程序文件的建立与执行	(126)
6.2 顺序结构程序	(128)
6.2.1 程序文件中的常用命令	(128)
6.2.2 顺序结构程序设计	(132)
6.3 选择结构程序	(135)
6.3.1 双分支选择语句	(135)
6.3.2 多分支选择语句	(137)
6.4 循环结构程序	(140)
6.4.1 DO WHILE 循环	(140)
6.4.2 FOR 循环	(144)
6.4.3 SCAN 循环	(147)
6.4.4 循环嵌套	(149)
6.5 过程与自定义函数	(155)
6.5.1 过程	(155)
6.5.2 内部过程与过程文件	(158)
6.5.3 内存变量的作用域	(161)
6.5.4 自定义函数	(166)
6.6 程序调试	(168)
6.6.1 调试器窗口	(168)
6.6.2 设置断点	(169)
6.6.3 调试菜单项	(171)
第 7 章 表单与面向对象程序设计	(172)

7.1 创建表单	(172)
7.1.1 表单向导	(172)
7.1.2 表单设计器	(178)
7.1.3 表单的运行	(180)
7.2 面向对象可视化编程基础	(180)
7.2.1 面向对象的基本概念	(181)
7.2.2 Visual FoxPro 的类层次结构	(183)
7.2.3 Visual FoxPro 类的设计	(184)
7.2.4 对象的设计	(190)
7.3 表单中的常用控件	(195)
7.3.1 输出类控件	(196)
7.3.2 输入类控件	(202)
7.3.3 控制类控件	(204)
7.3.4 容器类控件	(207)
7.3.5 其他类控件	(211)
7.4 表单综合设计举例	(213)
第8章 菜单、报表与标签设计	(229)
8.1 菜单设计	(229)
8.1.1 菜单设计基础知识	(229)
8.1.2 Visual FoxPro 的菜单设计器	(231)
8.1.3 下拉式菜单设计	(238)
8.1.4 快速菜单设计	(240)
8.1.5 为顶层表单添加菜单	(240)
8.1.6 快捷菜单设计	(242)
8.2 报表与标签设计	(243)
8.2.1 报表向导	(243)
8.2.2 快速报表设计	(247)
8.2.3 报表设计器	(249)
8.2.4 报表输出	(259)
8.2.5 报表设计实例	(261)
8.2.6 标签设计	(263)
第9章 应用程序开发	(264)
9.1 应用程序的开发步骤	(264)
9.1.1 需求分析	(264)
9.1.2 数据库设计	(265)

9.1.3 应用程序的创建	(266)
9.2 应用程序的测试、调试及编译	(269)
9.2.1 测试与调试的任务	(269)
9.2.2 Visual FoxPro 的调试器	(269)
9.2.3 应用程序的编译生成	(270)
9.3 发布应用程序	(272)
9.3.1 应用程序的发布过程	(272)
9.3.2 安装向导	(273)
附录1 Visual FoxPro 常用文件类型一览表	(275)
附录2 Visual FoxPro 常用命令一览表	(276)
附录3 Visual FoxPro 常用函数一览表	(289)
参考文献	(302)

第 1 章 数据库系统基础知识

计算机作为高速准确地进行各种算术运算与逻辑运算的现代化计算工具,已广泛应用于数据处理的各个领域。在信息时代,各行各业都需要对大量的信息进行加工处理,都需要对各种数据进行科学管理。学习 Visual FoxPro,就是要利用计算机完成对大量数据的组织、存储、维护和处理,从而方便、准确和迅速地获取有价值的信息,作为各种决策活动的依据。

学习和使用 Visual FoxPro,有必要了解和掌握有关数据库系统的一些基本概念。作为学习的理论先导,本章介绍数据库系统的基本概念与基础知识。

1.1 信息、数据与数据处理

应用计算机进行信息处理,我们有必要对信息、数据与数据处理等基本概念有一个初步的认识与了解。

1.1.1 数据与数据处理

1. 数据和信息

数据和信息是数据处理中的两个基本概念。信息是客观世界的抽象描述,数据是人们用于记录客观事物属性的物理符号。

数据与信息既有联系,又有区别。信息是数据所包含的意义,数据是信息的表现形式。通俗地讲,信息是经过加工处理并对人类社会实践和生产活动产生决策影响的数据。不经过加工处理的数据只是一种原始材料,对人类活动产生不了决策作用,它的价值只是在于记录了客观世界的事实。只有经过提炼和加工,原始数据才发生了质的变化,给人们以新的知识和智慧。数据是表示信息的,但并非任何数据都能表示信息,信息只是加工处理后的数据,是数据所表达的内容。另一方面,信息不随表示它的数据形式而改变,它是反映客观现实世界的知识,而数据则具有任意性,用不同的数据形式可以表示同样的信息。例如一个城市的天气预报情况是一条信息,而描述该信息的数据形式可以是文字、图像或声音等。

数据通常分为文本形式数据与多媒体数据。

文本形式的数据是数据的最基本的形式,由数字、字母、文字和其他特殊字符组成。例如,张小明的职称为工程师,月工资为 1356 元,这里的“张小明”、“工程师”与“1356”都是数据。在实际应用中,文本形式的数据有两种基本形式,一种是可以参与数值运算的数值型数据,如表示成绩、工资的数据;另一种是由字符组成,不能参与数值运算的字符型数据,如表

示姓名、职称的数据。

多媒体数据是通过各种输入设备输入到计算机中以二进制形式存储在磁介质、光介质和半导体存储器里的图形、图像、动画、影像、声音等数据,如人的照片、商品的商标以及语音、音乐等。

2. 数据处理

数据处理是指对数据的收集、存储、分类、计算、加工、检索和传输等一系列活动。数据处理的基本目的是从大量的、杂乱无章的、难以理解的数据中整理出对人们有价值、有意义的信息(即数据),作为决策的依据。例如,全体考生各门课程的考试成绩记录了考生的考试情况,属于原始数据,对考试成绩进行分类、汇总、分析和处理,如按成绩从高分到低分顺序排列,统计各分数段的人数,等等,进而可以根据招生人数确定录取分数线。

1.1.2 计算机数据管理技术的发展

数据处理的中心问题是数据管理。计算机对数据的管理是指对数据的组织、分类、编码、存储、检索和维护提供操作手段。计算机数据管理技术的发展与任何其他技术发展一样,也是一个由低级到高级的发展过程,大致经历了人工管理、文件系统和数据库系统 3 个发展阶段。

1. 人工管理阶段

20 世纪 50 年代中期以前,计算机主要应用于科学计算,数据量较少,一般不需要长期保存数据。硬件方面,没有磁盘等直接存取的外存储器。软件方面,没有对数据进行管理的系统软件。在此阶段,对数据的管理是由程序员个人考虑和设计的,他们既要确定算法,又要考虑数据的逻辑结构、物理结构以及输入输出方法等问题。程序与数据是一个整体,一个程序中的数据无法被其他程序使用,因此程序与程序之间存在大量的重复数据。数据存储结构一旦有所改变,则必须修改相应程序。应用程序的设计与维护负担繁重。

2. 文件系统阶段

20 世纪 50 年代后期至 60 年代,计算机开始大量用于数据管理。硬件上出现了直接存取的大容量外存储器,如磁盘、磁鼓等,这为计算机管理系统数据提供了物质基础。软件方面,出现了操作系统,其中包含文件系统,为数据管理提供了技术支持。

文件系统提供了在外存储器上长期保存数据并对数据进行存取的手段。文件的逻辑结构与存储结构有一定区别,即程序与数据有一定的独立性。数据的存储结构变化,不一定影响到程序,因此程序员可集中精力进行算法设计,大大减少了维护程序的工作量。

文件系统使计算机在数据管理方面有了长足的进步。时至今日,文件系统仍是一般高级语言普遍采用的数据管理方式。然而当数据量增加,使用数据的用户越来越多时,文件系统便不能适应更有效的管理,其症结表现在 3 个方面:

(1) 数据的冗余度大。由于数据文件是根据应用程序的需要而建立的,当不同的应用程序所需要使用的数据有许多部分相同时也必须建立各自的文件,即数据不能共享,造成大量重复。这样不仅浪费存储空间,而且使数据修改变得非常困难,容易产生数据不一致,即同样的数据在不同的文件中所存储的数值不同,造成矛盾。

(2) 数据独立性差。在文件系统中,数据和应用程序是互相依赖的,即程序的编写与数

据组织方式有关,如果改变数据的组织方式,就必须修改有关应用程序。这无疑将增加用户的负担。此外,数据独立性差也不利于系统扩充、系统移植等开发推广工作。

(3) 缺乏对数据的统一控制管理。在同一个应用项目中的各个数据文件没有统一的管理机构,数据完整性和安全性很难得到保证。数据的保护等均交给应用程序去解决,使得应用程序的编制相当烦琐。

较好地解决文件系统存在的问题,正是数据库系统的目标。

3. 数据库系统阶段

20 世纪 60 年代后期,计算机在数据管理中应用规模更加庞大,数据量急剧增加,数据共享性更强。硬件价格下降,软件价格上升,编制和维护软件所需成本相对增加,其中维护成本更高。这些成为数据管理在文件系统的基础上发展到数据库系统的原动力。

在数据库系统中,由一种叫做数据库管理系统(DBMS: Database Management Systems)的软件来对数据进行统一的控制和管理,从而有效地减少了数据冗余,实现了数据共享,解决了数据独立性问题,并提供统一的安全性、完整性和控制功能。

数据库是在数据库管理系统的集中控制之下,按一定的组织方式存储起来的、相互关联的数据集合。在数据库中集中了一个部门或单位完整的数据资源,这些数据能够为多个用户同时共享,且具有冗余度小、独立性和安全性高等特点。

数据库技术的发展先后经历了层次数据库、网状数据库和关系数据库。层次数据库和网状数据库可以看做是第一代数据库系统,关系数据库可以看做是第二代数据库系统。自 20 世纪 70 年代提出关系数据模型和关系数据库后,数据库技术得到了蓬勃发展,应用也越来越广泛。但随着应用的不断深入,占主导地位的关系数据库系统已不能满足新的应用领域的需求。例如,在实际应用中,除了需要处理数字、字符数据的简单应用之外,还需要存储并检索复杂的嵌套数据、复合数据(如集合、数组、结构)、多媒体数据、计算机辅助设计绘制的工程图纸和 GIS(地理信息系统)提供的有关数据等。对于这些复杂数据,关系数据库无法实现对它们的管理。正是实际中涌现出的许多问题,促使数据库技术不断向前发展,涌现出许多不同类型的新型数据库系统。下面概要性地作一些介绍。

(1) 分布式数据库系统。分布式数据库系统(Distributed Database System)是在集中式数据库基础上发展起来的,是数据库技术与计算机网络技术、分布处理技术相结合的产物。分布式数据库系统是地理上分布在计算机网络不同结点,逻辑上属于同一系统的数据库系统,能支持全局应用,同时存取两个或两个以上结点的数据。

(2) 面向对象数据库系统。面向对象数据库系统(Object-Oriented Database System)是将面向对象的模型、方法和机制,与先进的数据库技术有机地结合而形成的新型数据库系统。它从关系模型中脱离出来,强调在数据库框架中发展类型、数据抽象、继承和持久性。它的基本设计思想是,一方面把面向对象语言向数据库方向扩展,使应用程序能够存取并处理对象,另一方面扩展数据库系统,使其具有面向对象的特征,提供一种综合的语义数据建模概念集,以便对现实世界中复杂应用的实体和联系建模。因此,面向对象数据库系统首先是一个数据库系统,具备数据库系统的基本功能,其次是一个面向对象的系统,针对面向对象的程序设计语言的永久性对象存储管理而设计的,充分支持完整的面向对象概念和机制。

(3) 多媒体数据库系统。多媒体数据库系统(Multi-media Database System)是数据库技术与多媒体技术相结合的产物。在许多数据库应用领域中,都涉及大量的多媒体数据,这些与传统的数字、字符等格式化数据有很大的不同,都是一些结构复杂的对象。

(4) 数据仓库。信息技术的高速发展,数据库应用的规模、范围和深度不断扩大,一般的事务处理已不能满足应用的需要,企业界需要在大量信息数据基础上的决策支持,数据仓库(Data Warehouse)技术的兴起满足了这一需求。数据仓库作为决策支持系统(Decision Support System)的有效解决方案,涉及 3 个方面的技术内容:数据仓库技术、联机分析处理(On-Line Analysis Processing, OLAP)技术和数据挖掘(Data Mining)技术。

1.2 数据模型

计算机信息处理的对象是现实生活中的客观事物,客观事物是信息之源。人们把客观存在的事物以数据的形式存储到计算机中,经历了对现实生活中事物特征的认识、概念化到计算机数据库里的具体表示的过程。

1.2.1 实体及其联系

1. 实体

从数据处理的角度看,现实世界中的客观事物称为实体。实体可以指人,如一个教师、一个学生,等等;也可以指物,如一本书、一张桌子等;也可以指抽象的事件,如一次借书、一次奖励,等等。它还可以指事物与事物之间的联系,如学生选课、客户订货等。

一个实体可有不同的属性,属性描述了实体某一方面的特性。例如,教师实体可以用教师编号、姓名、性别、出生日期、职称、基本工资、研究方向等属性来描述。每个属性可以取不同的值,对于具体的某一教师,其编号为 30121,姓名为杨昆,性别为男,出生日期为 1974 年 10 月 10 日,职称为教授,基本工资为 1960 元,研究方向为数据库系统,分别为上述教师实体属性的取值。属性值的变化范围称作属性值的域。如性别这个属性的域为(男,女),职称的域为(助教,讲师,副教授,教授)等,由此可见,属性是个变量,属性值是变量所取的值,而域是变量的变化范围。

由上可见,属性值所组成的集合表征一个实体,相应的这些属性的集合表征了一种实体的类型,称为实体型,例如上面的教师编号、姓名、性别、出生日期、职称、基本工资、研究方向等表征“教师”这样一种实体的实体型。同类型的实体的集合称为实体集。

在 Visual FoxPro 中,用“表”来表示同一类实体,即实体集,用“记录”来表示一个具体的实体,用“字段”来表示实体的属性。显然,字段的集合组成一个记录,记录的集合组成一个表。

2. 实体间的联系

实体之间的对应关系称为联系,它反映了现实世界事物之间的相互关联。实体间的联系是指一个实体集中可能出现的每一个实体与另一实体集中多少个具体实体存在联系。实体之间有各种各样的联系,归纳起来有 3 种类型:

(1) 一对一联系(1:1)。如果对于实体集 A 中的每一个实体,实体集 B 中有且只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。例如,一所学校只有一个校长,一个校长只在一所学校任职,校长与学校之间的联系是一对一的联系。

(2) 一对多联系(1:n)。如果对于实体集 A 中的每一个实体,实体集 B 中有多个实体与之联系,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 有一对多的联系。例如,一所学校有许多学生,但一个学生只能就读于一所学校,所以学校和学生之间的联系是一对多的联系。

(3) 多对多联系(m:n)。如果对于实体集 A 中的每一个实体,实体集 B 中有多个实体与之联系,而对于实体集 B 中的每一个实体,实体集 A 中也有多个实体与之联系,则称实体集 A 与实体集 B 之间有多对多的联系。例如,一个读者可以借阅多种图书,任何一种图书可以为多个读者借阅,所以读者和图书之间的联系是多对多的联系。

1.2.2 数据模型

数据库需要根据应用系统中数据的性质、内在联系,按照管理的要求来设计和组织。计算机信息处理的对象是现实生活中的客观事物,人们把客观存在的事物以数据的形式存储到计算机中,经历了对现实生活中事物特征的认识、概念化到计算机数据库里的具体表示的逐级抽象过程。

数据模型是对客观事物及其联系的数据描述,反映实体内部和实体之间的联系。由于采用的数据模型不同,相应的数据库管理系统也就不同。

在数据库系统中,常用的数据模型有层次模型、网状模型和关系模型三种。

1. 层次模型

层次模型用树形结构来表示实体及其之间的联系。在这种模型中,数据被组织成由“根”开始的“树”,每个实体由根开始沿着不同的分支放在不同的层次上。树中的每一个结点代表实体型,连线则表示它们之间的关系。根据树形结构的特点,建立数据的层次模型需要满足两个条件:

- ① 有一个结点没有父亲,称之为根结点。
- ② 其他结点有且仅有一个父亲。

事实上,许多实体间的联系本身就是自然的层次关系。如一个学校的行政机构、一个家庭的世代关系等。图 1-1 是层次模型示意图。

层次模型具有层次清晰、构造简单、易于实现一对一和一对多的实体联系等优点。但不能直接表示出多对多的实体,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,实现起来较为麻烦,这是层次模型的局限性。

采用层次模型来设计的数据库称为层次数据库。层次模型的数据库管理系统是最早出现的,它的典型代表是 IBM 公司的 IMS 系统,这是世界上最早出现的大型数据库系统。

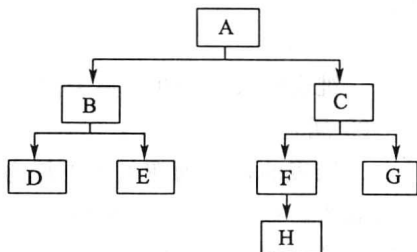


图 1-1 层次模型示意图

2. 网状模型

网状数据模型用以实体型为结点的有向图来表示各实体及其之间的联系。其特点是:

- ① 可以有一个以上的结点无父亲。

② 至少有一个结点有多于一个的父亲。

网状模型可以用图 1-2 来表示。

网状模型要比层次模型复杂,但它可以直接用来表示“多对多”联系。然而由于技术上的困难,一些已实现的网状数据库管理系统(如 DBTG)中仍然只允许处理“一对多”联系。

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(这对数据库系统来说是理所当然的),众多的指针使得管理工作相当复杂,对用户来说使用也比较麻烦。

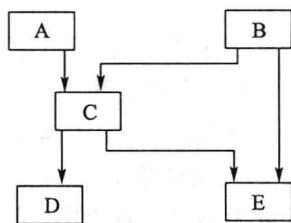


图 1-2 网状模型示意图

3. 关系模型

关系模型是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,把实体集看成一个二维表,每一个二维表称为一个关系。每个关系均有一个名字,称为关系名。例如表 1.1 就是一个学生成绩关系。

表 1.1 学生成绩关系

学 号	姓 名	性 别	入学成绩
03083101	王国文	男	510
02083106	李 义	男	520
02083103	邓 斌	男	524
02093108	杨雪晴	女	542

虽然关系模型比层次模型和网状模型发展得晚,但是因为它建立在严格的数学理论基础之上,所以是目前比较流行的一种数据模型。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。

(1) 关系模型的基本概念。

① 关系。

一个关系就是一张二维表,通常将一个没有重复行、重复列的二维表看成一个关系,每个关系都有一个关系名。在 Visual FoxPro 中,一个关系对应于一个表文件,其扩展名为 .dbf。

② 元组。

二维表的每一行在关系中称为元组。在 Visual FoxPro 中,一个元组对应表中一个记录。

③ 属性。

二维表的每一列在关系中称为属性,每个属性都有一个属性名,属性值则是各个元组属性的取值。在 Visual FoxPro 中,一个属性对应表中一个字段,属性名对应字段名,属性值对应于各个记录的字段值。

④ 域。

属性的取值范围称为域。域作为属性值的集合,其类型与范围具体由属性的性质及其所表示的意义确定。同一属性只能在相同域中取值。

⑤ 关键字。

关系中能惟一区分、确定不同元组的属性或属性组合,称为该关系的一个关键字。单个属性组成的关键字称为单关键字,多个属性组合的关键字称为组合关键字。需要强调的是,