

# Visual FoxPro

## 实用教程

崔洪芳 聂玉峰 主编



32.3



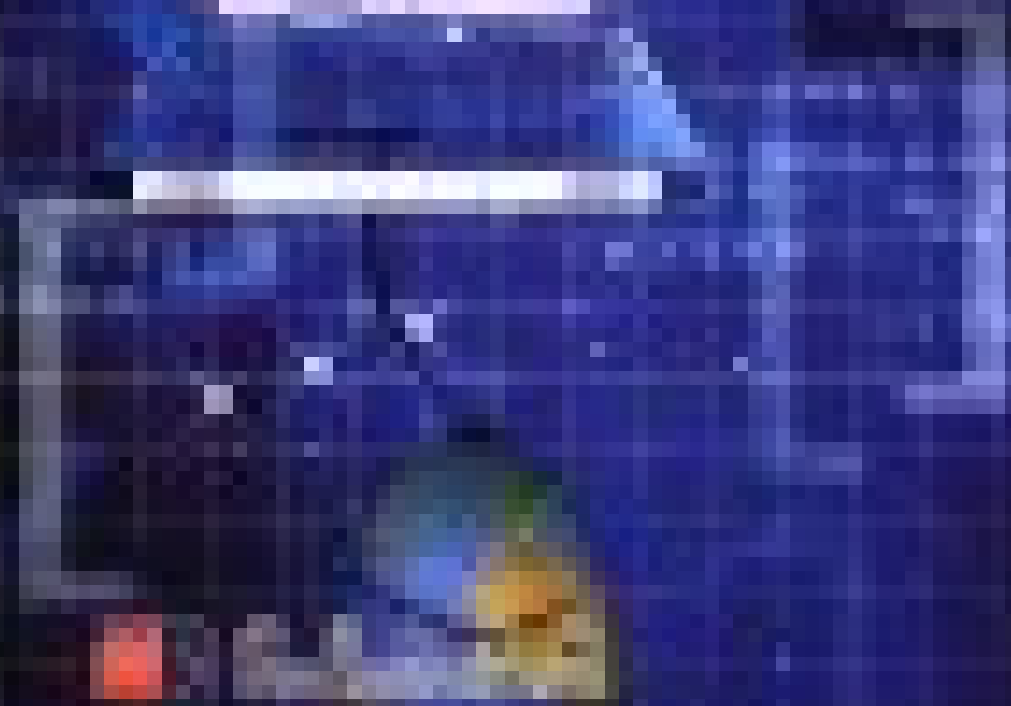
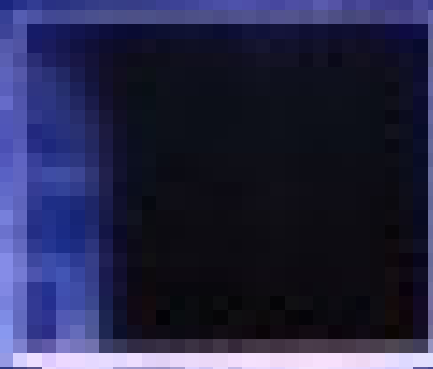
科学出版社

[www.sciencep.com](http://www.sciencep.com)

# Visual FoxPro

## 实用教程

第 1 章 数据库系统概论



# Visual FoxPro 实用教程

崔洪芳 聂玉峰 主 编

田萍芳 邹 琼 副主编

科 学 出 版 社

北 京

## 内 容 简 介

本书为 Visual FoxPro 课程教材,内容包括 Visual FoxPro 基础知识, Visual FoxPro 的基本语法,数据库的基本操作,项目管理器,结构化查询语言 SQL,数据查询与视图,Visual FoxPro 程序设计,面向对象的程序设计,管理信息系统开发简例。

本书是根据教育部考试中心最新推出的《全国计算机等级考试二级考试大纲(Visual FoxPro 程序设计)》的要求编写的。每章后附有习题,内容通俗易懂,图文并茂,有利于实际教学,实用性强。

本书可作为大专院校 Visual FoxPro 课程教材,也可供职业学校、成人教育学生使用,亦可作为计算机等级考试培训教材。

### 图书在版编目(CIP)数据

Visual FoxPro 实用教程/崔洪芳,聂玉峰主编. —北京:科学出版社,2003  
ISBN 7-03-011936-3

I. V… II. ①崔…②聂… III. 关系数据库-数据库管理系统, Visual FoxPro-教材 N. TP311.138

中国版本图书馆 CIP 数据核字 (2003) 第 063588 号

责任编辑:冯贵层/责任校对:王望荣

责任印制:高 嵘/封面设计:李 静

科学出版社出版

北京东黄城根北街16号

邮政编码:100717

<http://www.sciencep.com>

湖北省京山金美印刷有限责任公司印刷

科学出版社出版 各地新华书店经销

\*

2003年8月第 一 版 开本:787×1092 1/16

2003年8月第一次印刷 印张:16 3/4

印数:1—8000 字数:410 000

定价:23.00元

(如有印装质量问题,我社负责调换)

## 前 言

Visual FoxPro 是 Microsoft 公司推出的目前流行的微机关系型数据库管理系统。该系统最大的特点是简单易学、功能强大,特别适用于中、小型数据库的管理,并为数据库结构和应用程序开发提供功能强大的面向对象的环境。无论是组织信息、运行查询、创建集成的关系型数据库系统,还是为最终用户编写功能全面的数据管理应用程序,Visual FoxPro 都可以提供管理数据所需的有用工具。

本书为 Visual FoxPro 课程教材,是根据教育部考试中心最新推出的《全国计算机等级考试二级考试大纲(Visual FoxPro 程序设计)》的要求编写的。在编写过程中,作者力图用深入浅出、清晰简洁的叙述,全面介绍 Visual FoxPro 的操作使用和程序设计方法。

本书分为五个部分。第一部分是基础知识,由第一章和第二章组成,第一章介绍了 Visual FoxPro 基础概念和操作环境,系统的启动和退出;第二章介绍了 Visual FoxPro 的基本语法和常用函数。

第二部分介绍数据库的基本操作,由第三章和第四章组成,采用命令方式和菜单方式介绍了数据表 and 数据库的建立、维护、操作,多个数据表的操作,项目管理器的使用。

第三部分由第五章和第六章组成,介绍了结构化查询语言 SQL,数据查询的命令,查询与视图的建立和使用。

第四部分介绍 Visual FoxPro 程序设计,由第七章和第八章组成。第七章介绍过程化程序设计方法,包括三种基本控制结构、过程与自定义函数的设计及程序的调试技术;第八章介绍面向对象的程序设计,包括表单的生成与设计,菜单设计器和报表设计器的使用。

第五部分由第九章组成。通过一个管理信息系统开发简例对 Visual FoxPro 的内容进行了总结,也使读者对应用程序开发有一个全面的认识。

本书每章后附有习题,内容通俗易懂,图文并茂,有利于实际教学,实用性强。

本书由崔洪芳、聂玉峰任主编,田萍芳、邹琼任副主编。其中:第一章、第二章和第七章由聂玉峰编写,第三章、第四章、第五章和第六章由崔洪芳编写,第八章由田萍芳编写,第九章由邹琼编写。

Visual FoxPro 的命令和函数共有 700 多条,限于篇幅和学时,必然要有所取舍,有所侧重。至于这种取舍是否恰当,详略是否合适,恳请读者不吝赐教。

编者

2003 年 4 月

# 目 录

<b>第一章 Visual FoxPro 基础知识</b> .....	(1)
1.1 数据库基本概念 .....	(1)
1.1.1 数据、数据库、数据库管理系统和数据库系统 .....	(1)
1.1.2 数据模型 .....	(3)
1.2 关系数据库 .....	(5)
1.2.1 关系数据库的基本概念 .....	(5)
1.2.2 关系运算 .....	(6)
1.2.3 关系的完整性 .....	(7)
1.3 面向对象程序设计的概念 .....	(7)
1.3.1 对象、属性、事件和方法 .....	(7)
1.3.2 类、子类和封装 .....	(8)
1.4 Visual FoxPro 系统特点与性能指标 .....	(9)
1.4.1 Visual FoxPro 的特点 .....	(9)
1.4.2 Visual FoxPro 的性能指标 .....	(10)
1.5 Visual FoxPro 的启动与退出 .....	(11)
1.6 Visual FoxPro 的界面 .....	(12)
1.6.1 Visual FoxPro 的工作环境 .....	(12)
1.6.2 Visual FoxPro 的工作方式 .....	(13)
1.6.3 Visual FoxPro 的辅助设计工具 .....	(14)
习题一 .....	(15)
<b>第二章 Visual FoxPro 的基本语法</b> .....	(17)
2.1 Visual FoxPro 的数据类型 .....	(17)
2.2 常量、变量与表达式 .....	(19)
2.2.1 常量 .....	(19)
2.2.2 变量 .....	(19)
2.2.3 表达式 .....	(23)
2.2.4 简单输出命令 .....	(26)
2.3 Visual FoxPro 的命令结构与书写规则 .....	(26)
2.3.1 Visual FoxPro 的命令结构 .....	(26)
2.3.2 Visual FoxPro 命令和短语的书写规则 .....	(27)
2.3.3 Visual FoxPro 的文件类型 .....	(28)
2.4 Visual FoxPro 的常用函数 .....	(29)
2.4.1 字符处理函数 .....	(29)
2.4.2 数值处理函数 .....	(31)
2.4.3 日期和时间函数 .....	(33)
2.4.4 数据类型转换函数 .....	(34)
2.4.5 状态测试函数 .....	(35)

习题二 .....	(37)
<b>第三章 数据库的基本操作</b> .....	(39)
3.1 数据表的建立 .....	(39)
3.1.1 建立数据表结构 .....	(39)
3.1.2 修改数据表结构 .....	(42)
3.1.3 输入数据记录 .....	(43)
3.1.4 打开与关闭数据表文件 .....	(44)
3.1.5 显示数据表结构与记录 .....	(46)
3.2 数据表的维护 .....	(48)
3.2.1 记录指针 .....	(48)
3.2.2 修改数据记录 .....	(49)
3.2.3 插入与删除数据记录 .....	(53)
3.2.4 间接建立数据表文件 .....	(55)
3.2.5 文件的显示、复制、删除与改名 .....	(58)
3.2.6 记录的筛选与字段的屏蔽 .....	(59)
3.3 数据库的建立 .....	(61)
3.3.1 建立数据库 .....	(61)
3.3.2 数据库的打开、关闭与删除 .....	(61)
3.3.3 向数据库中添加或移去数据表 .....	(63)
3.4 字段级规则和记录规则的设定 .....	(65)
3.4.1 字段级规则的设定 .....	(65)
3.4.2 长表名和长字段名 .....	(66)
3.4.3 记录级规则的设定 .....	(67)
3.5 数据表的排序、索引与查询 .....	(67)
3.5.1 排序 .....	(68)
3.5.2 索引 .....	(69)
3.6 数据表的统计与汇总 .....	(74)
3.6.1 数据表的统计 .....	(74)
3.6.2 数据表的汇总 .....	(75)
3.7 多个表的同时使用 .....	(76)
3.7.1 选择工作区 .....	(76)
3.7.2 数据表的连接 .....	(78)
3.7.3 数据表文件的更新 .....	(79)
3.7.4 建立表之间的关联 .....	(80)
3.7.5 设置参照完整性 .....	(82)
习题三 .....	(85)
<b>第四章 项目管理器</b> .....	(89)
4.1 启动项目管理器 .....	(89)
4.1.1 创建新项目 .....	(89)
4.1.2 打开和关闭项目 .....	(90)
4.2 项目管理器的操作 .....	(90)
4.2.1 项目管理器的组成 .....	(90)
4.2.2 使用项目管理器 .....	(91)

4.3	项目管理器的定制 .....	(92)
	习题四 .....	(93)
<b>第五章</b>	<b>结构化查询语言 SQL .....</b>	<b>(94)</b>
5.1	SQL 的数据定义功能 .....	(94)
5.1.1	数据表的定义 .....	(94)
5.1.2	表结构的修改 .....	(96)
5.2	SQL 的数据修改功能 .....	(97)
5.2.1	插入 .....	(97)
5.2.2	更新 .....	(98)
5.2.3	删除 .....	(98)
5.3	SQL 的数据查询功能 .....	(98)
5.3.1	SELECT 命令 .....	(98)
5.3.2	简单查询 .....	(100)
5.3.3	复杂查询 .....	(101)
	习题五 .....	(103)
<b>第六章</b>	<b>数据查询与视图 .....</b>	<b>(104)</b>
6.1	常用查询命令 .....	(104)
6.1.1	直接查找命令 .....	(104)
6.1.2	索引查询 .....	(105)
6.2	Rushmore 技术 .....	(107)
6.3	查询 .....	(108)
6.3.1	查询设计器 .....	(108)
6.3.2	查询向导 .....	(115)
6.4	视图 .....	(116)
6.4.1	建立视图 .....	(116)
6.4.2	修改视图 .....	(118)
6.4.3	使用视图 .....	(119)
6.4.4	远程视图 .....	(119)
	习题六 .....	(121)
<b>第七章</b>	<b>Visual FoxPro 程序设计 .....</b>	<b>(122)</b>
7.1	Visual FoxPro 程序的建立、执行与显示 .....	(122)
7.1.1	Visual FoxPro 程序的概念 .....	(122)
7.1.2	Visual FoxPro 程序的建立与修改 .....	(123)
7.1.3	Visual FoxPro 程序的执行 .....	(124)
7.1.4	Visual FoxPro 程序的显示和打印 .....	(124)
7.1.5	求解的基本逻辑与程序通用性 .....	(125)
7.2	FoxPro 程序设计中常用的命令 .....	(125)
7.2.1	交互式输入输出命令 .....	(125)
7.2.2	其他辅助命令 .....	(129)
7.2.3	常用环境参数设置命令 .....	(130)
7.3	结构化程序设计 .....	(131)
7.3.1	程序的结构 .....	(131)



7.3.2	分支结构程序设计	(131)
7.3.3	循环结构程序设计	(137)
7.4	数组及其应用	(146)
7.4.1	数组的定义与赋值	(147)
7.4.2	数组与表之间数据的传递	(148)
7.5	多模块程序	(151)
7.5.1	子程序	(151)
7.5.2	自定义函数	(152)
7.5.3	过程及过程文件	(153)
7.5.4	内存变量的作用域	(155)
7.5.5	过程调用中数据的传递	(157)
7.6	程序的调试	(160)
7.6.1	程序中常见错误	(160)
7.6.2	常用的程序调试方法	(160)
7.6.3	Visual FoxPro 调试工具应用举例	(162)
	习题七	(164)
<b>第八章</b>	<b>面向对象的程序设计</b>	(170)
8.1	用表单设计应用程序界面	(170)
8.1.1	表单设计器的基本操作	(170)
8.1.2	表单设计器常用工具	(172)
8.1.3	面向对象编程的步骤	(177)
8.1.4	控件的操作方法	(180)
8.1.5	常用表单控件	(182)
8.1.6	自定义属性与方法	(201)
8.1.7	多表的表单设计	(203)
8.1.8	表单集与多重表单	(205)
8.2	菜单设计	(207)
8.2.1	创建菜单系统的步骤	(207)
8.2.2	菜单设计器的使用	(207)
8.2.3	菜单设计	(211)
8.3	报表设计	(214)
8.3.1	报表设计的步骤与方法	(214)
8.3.2	报表设计器的使用	(215)
8.3.3	报表设计	(216)
8.4	应用程序向导	(221)
	习题八	(226)
<b>第九章</b>	<b>管理信息系统开发简例</b>	(229)
9.1	系统分析	(229)
9.1.1	功能分析	(229)
9.1.2	模块设计	(229)
9.1.3	教学管理系统组成	(229)
9.2	数据库设计	(230)
9.2.1	数据表设计	(230)

9.2.2	创建项目.....	(231)
9.2.3	创建数据库 .....	(231)
9.3	各功能模块的实现 .....	(231)
9.3.1	欢迎界面.....	(231)
9.3.2	系统主菜单.....	(233)
9.3.3	“学生基本信息查询”表单.....	(236)
9.3.4	“学生成绩查询”表单.....	(239)
9.3.5	“教师任课查询”表单.....	(241)
9.3.6	“学生基本信息维护”表单.....	(243)
9.3.7	“成绩录入和修改”表单.....	(248)
9.3.8	“教师任课信息维护”表单.....	(251)
9.3.9	“成绩打印”报表.....	(252)
9.4	系统的编译和发行 .....	(253)
9.4.1	设置主文件.....	(253)
9.4.2	设置包含和排除.....	(254)
9.4.3	连编应用程序.....	(254)
9.4.4	创建发布磁盘.....	(254)
附录	全国计算机等级考试二级 Visual FoxPro 考试大纲 .....	(256)

# 第一章 Visual FoxPro 基础知识

## 1.1 数据库基本概念

信息在现代社会中起着越来越重要的作用,信息资源的开发和利用水平已成为衡量一个国家综合国力的重要标志。随着计算机技术的发展,计算机的主要应用已从科学计算逐渐转变为事务处理。据统计,目前全世界 80% 以上的计算机主要从事事务处理。在进行事务处理时,并不需要进行复杂的科学计算,而主要从事大量数据的存储、查找、统计等工作。为了有效地使用保存在计算机系统的大量数据,必须采用一整套严密合理的数据处理方法,即数据管理。数据管理是指对数据的收集、整理、组织、存储、查询、维护、传送和使用等。数据库技术就是作为数据管理中的一项技术而发展起来的。

数据库技术所研究的问题就是如何科学地组织和存储数据,如何高效地获取和处理数据。近年来,数据库技术和计算机网络技术的相互渗透、相互促进,已成为当今计算机领域发展迅速、应用广泛的两大领域。

### 1.1.1 数据、数据库、数据库管理系统和数据库系统

#### 1. 数据

数据(data)是指存储在某种媒介上能够识别的物理符号。数据的概念包括两个方面:其一是描述事物特征的数据内容;其二是存储在某种媒介上的数据形式。数据形式可以是多种多样的:可以是数字,如成绩;可以是文字,如姓名;也可以是特定的一串符号;还可以是图形、图像、动画、影像、声音等多媒体数据。

#### 2. 数据库

形象地说,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,比如一个学校可以将全部学生的情况存入数据库进行管理。在数据库系统尚未产生以前,人们往往采用表格、卡片或档案来进行人事管理、图书管理以及各种档案资料的管理。数据库的作用就在于把这些数据有组织地存储到计算机中去,减少数据的冗余,使人们能快速方便地对数据进行查询、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如下的定义:数据库是以一定的数据模型组织和存储的、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。它有如下的几个特点:

- (1) 数据的共享性。数据库中的数据能为多个用户服务。
- (2) 数据的独立性。用户的应用程序与数据的逻辑组织和物理存储方式无关。
- (3) 数据的完整性。数据库中的数据在操作和维护过程中可以保证正确无误。
- (4) 数据库中的冗余数据少,尽可能避免数据的重复。

#### 3. 数据库管理系统

数据库的建立、使用和维护,都是通过特定的数据库语言进行的。正如使用高级语言需要解释/编译程序的支持一样,使用数据库语言也需要一个特定的支持软件,这就是数据库管理

系统(database management system,简称 DBMS)。数据库管理系统是位于用户与操作系统之间的一层数据管理软件,它建立在操作系统的基础上对数据库进行统一的管理。用户利用数据库管理系统提供的一整套命令,可以对数据进行各种操作,从而实现用户的数据处理要求。一般地说,数据库管理系统应该具有下列功能:

(1) 数据定义功能。DBMS 向用户提供“数据定义语言”(data definition language,简称 DDL),用户通过它可以方便地对数据库中的数据对象进行定义,如建立或删除数据库、基本表和视图等。

(2) 数据操作功能。对数据进行检索和查询,是数据库的主要应用。为此,DBMS 向用户提供“数据操作语言”(data manipulation language,简称 DML),支持用户对数据库中的数据进行查询、更新(包括增加、删除、修改)等操作。

(3) 控制和管理功能。除 DDL 和 DML 两类功能外,DBMS 还具有必要的控制和管理功能。其中包括:在多用户使用时对数据进行的“并发控制”;对用户权限实施监督的“安全性检查”,数据的备份、恢复和转储功能;对数据库运行情况的监控和报告等。通常数据库系统的规模越大,这类功能也越强,所以大型机 DBMS 的管理功能一般比 PC 机 DBMS 更强。

#### 4. 数据库系统

数据库系统(database system,简称 DBS)是指在计算机系统中引入数据库技术后的系统,一般由 4 个部分构成:

(1) 计算机硬件系统。数据库系统对计算机系统硬件资源提出了较高的要求,这些要求主要包括:有足够大的内存来存放操作系统,DBMS 的核心模块、数据缓冲区和应用程序;有足够大的直接存取设备存放数据(如磁盘);有足够的其他存储设备来进行数据备份;有较高的数据传输能力,以提高数据传输率。

(2) 计算机软件系统。包括操作系统、数据库管理系统、与数据库接口的高级语言及其编译系统和以 DBMS 为核心的应用开发工具。

(3) 数据库应用系统。对数据库中的数据进行处理和加工的软件,它面向特定应用。如基于数据库的各种管理软件:管理信息系统、决策支持系统、办公自动化等都属于数据库应用系统。

(4) 各类人员。包括数据库管理员和用户。

#### 5. 数据库系统的特点

数据库系统的主要特点表现如下:

(1) 数据结构化。数据库系统实现整体数据的结构化,是数据库的主要特征之一,也是数据库系统与文件系统的本质区别。

(2) 数据共享。在数据库系统中,所有的程序都存取同一个数据库。一个库中的数据不仅可为同一企业或机构之内的各个部门所共享,也可为不同单位、地域甚至不同国家的用户所共享。

(3) 数据独立性。用户的应用程序与存储在磁盘上的数据库中的数据是相互独立的。用户不需要了解数据实际的存取方式,只通过数据库系统的存取命令就可得到需要的数据。

(4) 可控冗余度。数据冗余度是指在数据库中的数据重复程度。实现共享后,不必要的重复将全部消除,这样可以节省存储空间,减少存取时间,避免数据之间的不相容性和不一致性。但为了提高查询效率,有时也保留少量重复数据,其冗余度可由设计人员控制。

(5) 安全性保护。数据安全性是指保护数据以防止不合法的使用所造成的数据破坏和泄

密,例如设置访问权限、对数据加密等。

(6) 数据完整性控制。数据完整性是指数据的正确性、有效性和相容性。数据库系统提供了必要的功能,保证了数据在输入、修改过程中始终符合原来的数据定义和规定。

(7) 并发控制。并发操作是指多个用户进程在同一时刻存取同一数据时发生的事件。为了避免并发进程间相互干扰进而导致错误的结果或破坏数据完整性,必须对多用户的并发操作加以控制和协调。

(8) 故障发现和恢复控制。在数据库系统运行中,由于用户操作失误和硬件及软件的故障,可能使数据库遭到局部性或全局性损坏,但系统能进行应急性处理,把数据库恢复到正确状态。

## 1.1.2 数据模型

模型(model)是现实世界特征的模拟和抽象。

计算机不可能直接处理现实世界中的具体事物,人们必须把具体事物转换成计算机能够处理的数据。在数据库用数据模型(data model)这个工具来抽象、表示和处理现实世界中的数据和信息。

数据模型应满足三方面要求:一是能比较真实地模拟现实世界;二是容易为人所理解;三是便于在计算机上实现。数据模型主要有层次模型、网状模型、关系模型三种,它是按计算机系统的观点对数据建模,用于 DBMS 的实现。

在了解数据模型之前,先介绍两个术语:实体与属性。

实体:现实世界任何可相互区别的事物,不论是实际存在的东西(如人、计算机),还是概念性的东西(如教学质量、管理水平),或是事物与事物之间的联系(如一场球赛),一律统称为实体。

属性:实体所具有的性质统称为属性。不同实体是由其属性的不同而被区分的,换句话说,实体靠属性来描述。

### 1. 层次模型(hierarchical model)

若用图来表示,层次模型是一棵倒立的树,具有父子关系,如图 1.1 所示。每个父结点可以有許多子结点,但每个子结点仅有一个父结点。

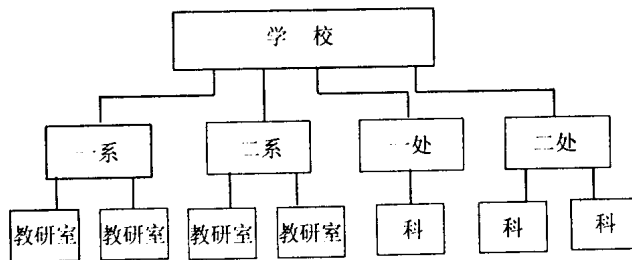


图 1.1 层次模型示意图

层次模型数据库是最基本的、最容易实现的数据库。它易于操作,可利用树状数据结构来完成。每一个结点有其具体的功能,如果需要寻找较远的结点,则必须先往上通过很多父结点,然后再往下寻找另一个结点。显然,对于一个较大的数据库将会消耗很多搜索时间。层次模型在不同结点之间只允许存在单线联系,因此只适合于管理具有家族型系统结构的数据库。

## 2. 网状模型(network model)

网状模型的结构如图 1.2 所示。网状模型允许子结点有多个父结点,并且子结点之间的关系较接近,联系很容易。以学生选课为例,看一看网状模型如何来组织数据。

一个学生可以选修若干门课程,某一课程可以被多个学生选修,因此学生与课程之间是多对多的联系。这样的实体联系图不能直接用网状模型来表示,因为网状模型不能直接表示记录间多对多的联系。为此,我们可以把联系作为一个结点来表示,并引入学生选课记录。它由三个数据项组成,即学号、课程号、成绩,表示某个学生选修某一门课程及其成绩。利用学生选课的一对多联系和课程到选课的一对多联系表示学生和课程之间的多对多的联系。

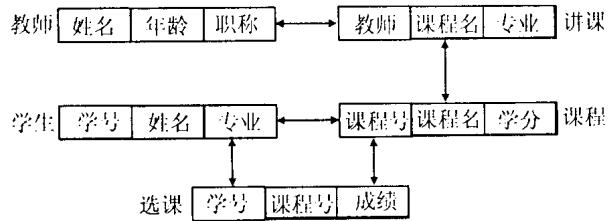


图 1.2 网络模型示意图

网状模型比层次模型更具有灵活性,更适用于管理在数据之间具有复杂联系的数据库。明显的缺点是路径太多,当加入或删除数据时,涉及相关数据过多,不易于维护与重建。

网状模型表达能力强,它能反映实体间的多对多的联系,但网状模型在概念上、结构上和使用上都比较复杂,而且对计算机的硬件环境要求较高。

网状模型和层次模型都是用指针来实现两个实体之间的联系的。它们都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数据模型。1970年,美国 E. F. Codd 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。关系模型中的数据逻辑结构是一张二维表,它由行和列组成。一个关系对应通常说的一张表,表中的一列表示实体的一项属性,称为一个字段;表中的一行包含了一个实体的全部属性值,称为一个记录。

关系模型的特点是:

- (1) 描述一致性。无论是实体还是实体之间的联系都用关系来表示。
- (2) 可以直接表示多对多联系。比如“教师任课表”可表示一个教师担任几个班的教学,也可表示一个班有几个教师任教。
- (3) 关系规范化。二维表格中每一栏目都是不可分的数据项,即不允许表中有表。
- (4) 数学基础严密。
- (5) 概念简单,操作方便。用户对数据的检索是从原来的表中得到一张新表,具体操作毋须用户关心,数据的独立性高。

基于关系模型的数据库管理系统因其严格的数学理论、使用简单灵活、数据独立性强等特点,被公认为最有前途的一种数据库管理系统。它的发展十分迅速,目前已占据主导地位。自 20 世纪 80 年代以来,作为商品推出的数据库管理系统几乎都是关系型的。例如,Oracle, Sybase, Informix, Visual FoxPro 等。

## 1.2 关系数据库

### 1.2.1 关系数据库的基本概念

为了进一步了解关系数据库,首先给出关系模型中的一些基本概念。

#### 1. 关系

一个关系就是一张二维表,每个关系有一个关系名。在 Visual FoxPro 中,一个关系存储为一个文件,文件扩展名为 dbf,称为“表”。下面所示的两个表文件中,学生表记载了学生的基本情况,成绩表记载了学生各门功课的成绩信息。

表 1-1 学生表

学号	姓名	性别	出生日期	团员	专业	平均成绩	简历
020101	刘晓刚	男	03/15/83	.T.	自动化	69.5	
020102	韩爱芳	女	10/05/84	.F.	自动化	63.5	
030205	周子康	男	06/16/84	.T.	机械	85.0	
030206	胡冬琴	女	01/12/85	.T.	机械	57.8	
030310	王世洪	男	08/21/83	.F.	管理	67.3	
030315	李梦茹	女	04/03/84	.T.	管理	90.3	

表 1-2 成绩表

学号	数学	英语	计算机	物理	总分
020101	75	80	67	56	278
020102	60	70	50	74	254
030205	80	88	83	89	340
030206	60	56	45	70	231
030310	55	76	65	73	269
030315	90	91	86	94	361

关系可以有三种形式:基本表、查询表和视图。基本表是实际存在的表,它是实际存储数据的逻辑表示,如学生表、成绩表。查询表是查询结果对应的表。视图是由基本表或其他视图导出的表,是虚表,不对应实际存储的数据。

#### 2. 元组

二维表中水平方向的行称为元组(tuple),每一行是一个元组。元组对应存储文件中的一个具体记录。

#### 3. 属性

二维表中垂直方向的列称为属性(attribute),每一列有一个属性名,在 Visual FoxPro 中表示为字段名。每个字段的数据类型、宽度等在创建表的结构时设定。

#### 4. 域

一个属性的取值范围称为域(domain),即不同元组对同一个属性的取值所限定的范围。

#### 5. 分量

元组中的一个属性值叫做元组的一个分量,在 Visual FoxPro 中又称为字段。

#### 6. 关键字

二维表中的一个属性或几个属性的组合,其值能够惟一地标识一个元组,这种属性或其组

合称为关键字,也叫主键(或主关键字)。例如学生表的学号。

在关系数据库中,表与表之间的联系是通过公共属性实现的。如学生表和成绩表中都含有学号属性,尽管学生数据与成绩数据分别存储在不同的表中,但是通过学生与成绩之间的公共属性(学号)就可以建立两个表之间的关联。

## 7. 外部关键字

如果表中的一个属性不是本表的主关键字,而是另外一个表的主关键字,这个属性就称为外部关键字(也叫外键)。

例如,教工表中的课程号就是外部关键字,是用于实现与课程表之间联系的外键。由此可见,尽管关系数据库中表是独立存储的,但是表与表之间可通过外键相互联系,从而构成一个整体的逻辑结构。

表 1-3 课程表

课程号	课程名	学时	学分
101	英语	160	8
102	数学	120	6
103	计算机	80	4
104	物理	54	3
105	哲学	40	2

表 1-4 教工表

职工号	姓名	课程号	任课班级
01001	江明	101	0201
03012	杨柳	102	0201
02103	刘茜	103	0201
02061	代淳	104	0201
01045	黄河	101	0202

## 8. 关系模式

对关系的描述称为关系模式,一个关系模式对应一个关系的结构。其格式为:

关系名(属性名 1,属性名 2,⋯,属性名 n)

在 Visual FoxPro 中表示为表结构:

表名(字段名 1,字段名 2,⋯,字段名 n)

## 9. 关系数据库

采用关系模型的数据库叫做关系数据库。作为关系数据库中的表应满足以下要求:

- (1) 表中不允许有重复的字段名。
- (2) 表中每一字段中的数据类型必须相同。
- (3) 表中的记录的次序及字段的次序可任意排列。
- (4) 一般表中的字段之间不应相互关联,如有“出生日期”字段,就不应有“年龄”字段。
- (5) 主关键字一般应为惟一的。

学生表中包含了 6 条记录,每条记录有 8 个字段。为了区别每一条记录,作为主关键字的字段不能重复。例如,学号字段是主关键字,不可重复,但姓名字段和性别字段的数据则可以重复。

### 1.2.2 关系运算

对关系数据库进行查询时,若要找到用户关心的数据,就需要对关系进行一定的关系运算(选择、投影、联接)。关系运算的操作对象是关系,运算的结果仍为关系。

(1) 选择(select)。选择是指从指定的关系中选择某些满足条件的元组构成一个新的关系。也就是说,选择是在二维表中选择满足指定条件的行,它是一种水平方向上的选择。例如,从学生表中找出所有平均成绩在 80 分以上的记录,就是通过选择操作来完成的。



(2) 投影(project)。投影是指从指定的关系中选择某些属性的所有值组成的新关系,它是一种垂直方向上的选择。例如,在学生表中,若要仅显示所有学生的学号、姓名、性别、专业和平均成绩,那么可以使用投影运算来实现。

(3) 联接(join)。联接是将两个关系联接在一起,形成一个新的关系。每一个联接操作都包括一个联接条件和一个联接类型。联接条件决定了两个关系中哪些元组相互匹配,以及联接结果中有哪些属性出现。联接类型有内联接、自然联接、左联接、右联接、全联接等。联接类型将决定如何处理与联接条件不相匹配的元组。在后面的章节将详细介绍这些内容。

以上这些关系运算,在关系数据库管理系统中都有相应的操作命令。对关系数据库的实际操作,往往是以上几种操作的综合应用。

### 1.2.3 关系的完整性

关系模型的完整性(integrity)规则是对关系的某种约束条件。关系模型中可以有三类完整约束:实体完整性、参照完整性和用户自定义的完整性。其中,实体完整性和参照完整性是关系模型必须满足的完整性约束条件,统称为关系完整性规则。

#### 1. 实体完整性

实体完整性(entity integrity)是指关键字的值在关系中必须非空并且是惟一的。

表 1-1 中学生表的主关键字是学号,这个主键的值必须是惟一的和确定的,只有这样才能有效地标识每一个学生。主关键字不能取空值(NULL),空值不是 0,也不是空字符串,而是没有值,是不确定的值,所以空值无法标识表中的任何一行。为了保证每一个实体有惟一的标识符,主关键字不能取空值。

#### 2. 参照完整性

参照完整性(referential integrity)是指一个关系中外键的值必须是另一个关系中主键的有效值,或者是空值。

例如,在表 1-3 和表 1-4 中,课程号是课程表的主键、教工表的外键。教工表中任何一个课程号必须是课程表中一个存在的值,或者是空值(该教师没有分配教学任务)。换句话说,外键可以没有值,但不允许是一个无效值。外键的这种性质称为参照完整性。

#### 3. 用户自定义的完整性

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,而用户自定义的完整性就是针对某一具体关系数据库的约束条件。

例如,成绩的取值必须在 0~100 之间。

## 1.3 面向对象程序设计的概念

面向对象程序设计(object oriented programming,简称 OOP)是一种全新的程序设计方法,本节介绍它的基本概念。OOP 技术将在第八章通过设计表单、菜单和报表来逐步介绍。

### 1.3.1 对象、属性、事件和方法

#### 1. 对象

现实世界的事物均可抽象视为对象(object)。例如,现实生活中的计算机、电话机、电视等,Windows 中的窗口、命令按钮、标签等都可以视为对象。对象包括数据和过程。每个对象