

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

# Visual FoxPro 程序设计基础

Visual FoxPro Chengxu Sheji Jichu

田俊华 刘爱军 主编

- 结合多年的教学经验和教学成果
- 以全国计算机等级考试大纲为参照
- 融理论与实验为一体，注重可操作性



高校系列

138F0



人民邮电出版社  
POSTS & TELECOM PRESS

21世纪

21st Century

TP311.138FO  
T622

计算机规划教材

Textbooks of Computer Science

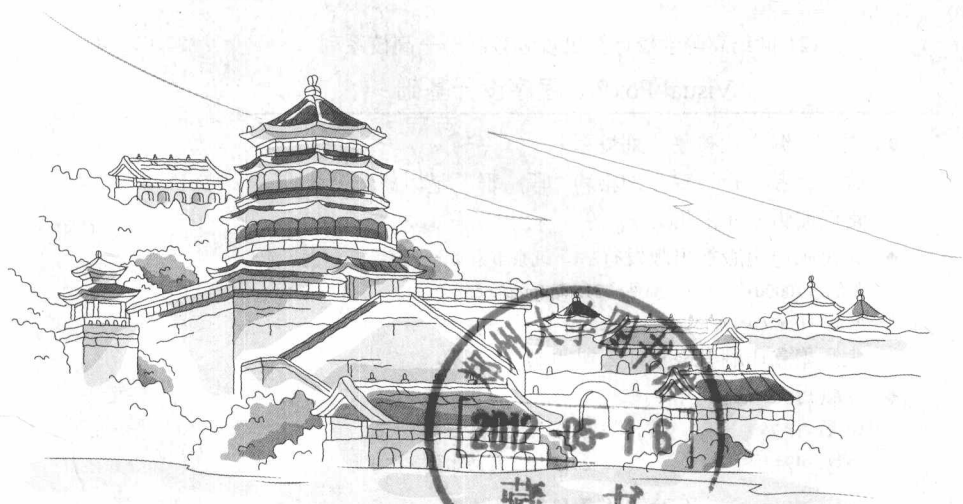


郑州大学 \*04010773473-\*

# Visual FoxPro 程序设计基础

Visual FoxPro Chengxu Sheji Jichu

田俊华 刘爱军 主编

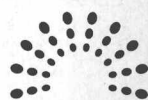


TP311.138FO

T622

人民邮电出版社

北京



高校系列

## 图书在版编目 (CIP) 数据

Visual FoxPro程序设计基础 / 田俊华, 刘爱军主编

— 北京: 人民邮电出版社, 2011. 7 (2012. 2 重印)

21世纪高等学校计算机规划教材. 高校系列

ISBN 978-7-115-25381-1

I. ①V… II. ①田… ②刘… III. ①关系数据库—数据库管理系统, Visual FoxPro—程序设计—高等学校—教材 IV. ①TP311.138

中国版本图书馆CIP数据核字(2011)第082173号

## 内 容 提 要

本书介绍关于数据库的基础知识以及利用 Visual FoxPro 开发数据库应用系统的方法, 给出了大量的应用实例。全书共分 13 章, 内容包括数据库系统基础知识、Visual FoxPro 概述、Visual FoxPro 的数据与数据运算、表的基本操作、数据库操作、SQL 基础、查询和视图、结构化程序设计、面向对象程序设计、表单的设计与应用、菜单工具栏设计、报表设计和综合应用实例等。

本书融理论与实验为一体, 所有操作步骤都按实际操作界面逐步讲解, 读者可一边学习, 一边上机操作。通过本书的学习, 读者对使用 Visual FoxPro 进行数据库软件开发会有一个较完整的认识, 并能掌握开发数据库系统的基本思想和方法, 逐步具备数据库管理系统的设计、应用和开发能力。

本书可作为普通高等学校和高职高专 Visual FoxPro 程序设计课程的教材, 也可作为计算机等级考试(二级)的培训教材以及计算机技术爱好者的自学参考书。

21 世纪高等学校计算机规划教材——高校系列

### Visual FoxPro 程序设计基础

- ◆ 主 编 田俊华 刘爱军  
编 者 卢 琼 刘雅莉 段 群 赵 蕾 樊景博  
责任编辑 王亚娜
- ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街 14 号  
邮编 100061 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
大厂聚鑫印刷有限责任公司印刷
- ◆ 开本: 787×1092 1/16  
印张: 18.75  
字数: 494 千字

2011 年 7 月第 1 版

2012 年 2 月河北第 3 次印刷

ISBN 978-7-115-25381-1

定价: 36.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223

反盗版热线: (010)67171154

# 前言

能够掌握满足社会与专业本身需求的计算机应用能力已成为一名合格大学毕业生所必须具备的素质。文科类专业与信息技术的相互结合、交叉、渗透,是现代科学技术发展的趋势之一,是不可忽视的一个新学科生长点。加强文科类专业的计算机教育是培养信息化社会文科人才的重要举措,是培养跨学科、综合型文科通才的重要环节。因此,使用一定层次、一定内容的计算机科学与技术知识来武装文科类专业(包括哲学、经济学、法学、教育学、文学、外语、历史学等学科和管理学中的一些专业)的学生,开设具有文科专业特色的计算机课程是十分必要的。为了指导文科类专业的计算机教学工作,教育部高等教育司组织制定了《高等学校文科类专业大学计算机教学基本要求(2003年版)》,该要求把文科类计算机教学的知识结构分为两大部分:一是大学计算机公共基础课;二是在开设计算机公共基础课之后,体现专业特色或与专业教学相结合的后续课程。

数据库技术是计算机应用科学中发展最快的技术,也是目前应用最广泛的技术。微机数据库系统以其开发成本低、简单易学、方便用户等优点应用最为普遍,Visual FoxPro 就是目前微机上优秀的数据库管理系统之一,它采用了可视化的、面向对象的程序设计方法,大大简化了应用系统的开发过程,并提高了系统的模块性和紧凑性。Visual FoxPro 也是文科类学生学习计算机程序设计最好的入门语言,由于它与 SQL 一脉相承,所以学习 Visual FoxPro 也可为后续学习其他编程语言以及其他体现专业特色的计算机知识打下坚实的基础。

本书是依据教育部高等教育司组织制定的《高等学校文科类专业大学计算机教学基本要求(2003年版)》编写而成的,书中针对文科学生的特点,充分强调“抓住基础,少讲技巧”,同时注意到“强化应用,淡化语法”。书中先从数据库基本原理、概念出发,介绍数据表以及数据库对象的建立、查看、修改、使用与维护等操作,然后再介绍结构化程序设计的结构与基本方法,并由浅入深地引入了面向对象程序设计的思想。在本书的有关章节中,分别通过大量的实例,介绍解决问题的方法和思路,逐步讲解 Visual FoxPro 中最实用、最常用的技术,最后结合一个综合案例阐述了利用 Visual FoxPro 开发数据库应用系统的方法。

目前,全国计算机等级考试认证是社会上比较权威的认证,也是人事部门录用和考核人员的一种测试手段,因此得到各高等院校师生的高度重视,本书在编写过程中也兼顾了学生参加等级考试的需求,内容涵盖了《全国计算机等级考试二级 Visual FoxPro 考试大纲》的全部内容。

本书由田俊华、刘爱军任主编,第1、2、3、5章由田俊华编写,第4章由赵蕾编写,第6、7、8章由段群编写,第9、13章由刘爱军编写,第10章由卢琼编

写,第11章由樊景博编写,第12章由刘雅莉编写。

由于编者水平有限,书中难免存在错误和不妥之处,欢迎广大读者批评指正。

编者

2011年3月

# 目 录

## 第 1 章 数据库系统基础知识.....1

### 1.1 数据库基础知识.....1

#### 1.1.1 信息与数据.....1

#### 1.1.2 数据处理.....2

#### 1.1.3 数据库管理技术的发展.....3

### 1.2 数据模型.....5

#### 1.2.1 基本概念.....5

#### 1.2.2 实体及其之间的关系.....6

#### 1.2.3 数据模型.....7

### 1.3 数据库系统.....9

#### 1.3.1 数据库系统的组成.....9

#### 1.3.2 数据库系统的分类.....10

#### 1.3.3 数据库系统的结构.....11

### 1.4 关系数据库.....13

#### 1.4.1 关系模型.....13

#### 1.4.2 关系数据库.....15

#### 1.4.3 关系运算.....15

#### 1.4.4 关系完整性.....16

### 1.5 数据库系统的体系结构与开发工具.....17

#### 1.5.1 数据库系统的体系结构.....18

#### 1.5.2 常见的数据库管理系统.....19

#### 1.5.3 常见的数据库开发工具.....20

## 第 2 章 Visual FoxPro 概述.....21

### 2.1 Visual FoxPro 的发展过程.....21

### 2.2 Visual FoxPro 的安装、启动与退出.....22

#### 2.2.1 Visual FoxPro 的运行环境.....22

#### 2.2.2 Visual FoxPro 的安装.....22

#### 2.2.3 Visual FoxPro 的启动.....23

#### 2.2.4 Visual FoxPro 的退出.....23

### 2.3 Visual FoxPro 的用户界面.....23

#### 2.3.1 Visual FoxPro 的窗口介绍.....23

#### 2.3.2 Visual FoxPro 的系统菜单.....24

#### 2.3.3 Visual FoxPro 的工具栏.....25

#### 2.3.4 Visual FoxPro 的状态栏.....26

#### 2.3.5 Visual FoxPro 的命令窗口.....26

### 2.4 Visual FoxPro 的操作概述.....26

#### 2.4.1 Visual FoxPro 的操作方式.....26

#### 2.4.2 Visual FoxPro 的可视化设计工具.....27

#### 2.4.3 Visual FoxPro 的系统环境设置.....29

#### 2.4.4 Visual FoxPro 的帮助系统.....30

#### 2.4.5 Visual FoxPro 的文件类型.....31

### 2.5 Visual FoxPro 的命令概述.....31

#### 2.5.1 Visual FoxPro 命令结构.....31

#### 2.5.2 Visual FoxPro 命令常用短语.....31

#### 2.5.3 Visual FoxPro 命令的书写规则.....32

#### 2.5.4 Visual FoxPro 命令的语法约定.....33

## 第 3 章 Visual FoxPro 的数据与 数据运算.....34

### 3.1 Visual FoxPro 的数据类型.....34

### 3.2 Visual FoxPro 的常量与变量.....36

#### 3.2.1 Visual FoxPro 的常量.....36

#### 3.2.2 Visual FoxPro 的变量.....37

### 3.3 Visual FoxPro 表达式.....38

#### 3.3.1 算术运算符与算术表达式.....38

#### 3.3.2 字符运算符与字符表达式.....39

#### 3.3.3 关系运算符与关系表达式.....39

#### 3.3.4 逻辑运算符与逻辑表达式.....39

#### 3.3.5 日期和时间表达式.....40

### 3.4 Visual FoxPro 常用的内部函数.....40

#### 3.4.1 数值函数.....41

#### 3.4.2 字符函数.....42

#### 3.4.3 日期和时间函数.....43

#### 3.4.4 数据类型转换函数.....44

|                   |           |                   |            |
|-------------------|-----------|-------------------|------------|
| 3.4.5 测试函数        | 45        | 5.2.2 数据库的设计过程    | 77         |
| 3.4.6 显示位置函数      | 46        | 5.3 数据库的基本操作      | 81         |
| 3.4.7 文件管理函数      | 46        | 5.3.1 建立数据库文件     | 82         |
| 3.4.8 数组函数        | 46        | 5.3.2 编辑数据库       | 83         |
| <b>第4章 表的基本操作</b> | <b>48</b> | 5.3.3 向数据库添加自由表   | 85         |
| 4.1 表的创建          | 48        | 5.3.4 创建数据库表      | 85         |
| 4.1.1 设计表结构       | 48        | 5.3.5 编辑数据库表      | 86         |
| 4.1.2 建立表结构       | 49        | 5.3.6 设置数据库表字段属性  | 87         |
| 4.1.3 显示表结构       | 50        | 5.3.7 为数据库表添加索引   | 92         |
| 4.1.4 修改表结构       | 51        | 5.4 建立数据库表之间的联系   | 92         |
| 4.2 表的基本操作        | 51        | 5.4.1 工作区的基本概念    | 92         |
| 4.2.1 打开和关闭表      | 51        | 5.4.2 工作区的选择和使用   | 93         |
| 4.2.2 向表中输入记录     | 52        | 5.4.3 数据库表间的关联    | 95         |
| 4.2.3 浏览表记录       | 53        | 5.4.4 设置参照完整性     | 98         |
| 4.2.4 显示表记录       | 53        | <b>第6章 SQL 基础</b> | <b>100</b> |
| 4.2.5 表记录的修改      | 54        | 6.1 SQL 概述        | 100        |
| 4.2.6 表记录的删除      | 55        | 6.1.1 SQL 简介      | 100        |
| 4.2.7 表记录指针的定位    | 56        | 6.1.2 SQL 的主要特点   | 101        |
| 4.3 表的查询、统计与计算    | 57        | 6.2 数据定义          | 102        |
| 4.3.1 表的查询        | 57        | 6.2.1 定义表结构       | 102        |
| 4.3.2 统计表中记录个数    | 58        | 6.2.2 删除表         | 104        |
| 4.3.3 求和与平均值      | 59        | 6.2.3 修改表结构       | 104        |
| 4.3.4 统计函数        | 59        | 6.3 数据查询          | 106        |
| 4.3.5 分类统计        | 60        | 6.3.1 基本查询        | 107        |
| 4.4 表的排序与索引       | 60        | 6.3.2 条件查询        | 108        |
| 4.4.1 表的排序        | 60        | 6.3.3 嵌套查询        | 110        |
| 4.4.2 索引文件概述      | 61        | 6.3.4 多表查询        | 111        |
| 4.4.3 索引文件的建立     | 62        | 6.3.5 连接查询        | 112        |
| 4.4.4 索引文件的使用     | 64        | 6.3.6 查询结果处理      | 112        |
| <b>第5章 数据库操作</b>  | <b>67</b> | 6.4 数据操纵          | 115        |
| 5.1 项目管理器         | 67        | 6.4.1 插入记录        | 115        |
| 5.1.1 项目的创建       | 67        | 6.4.2 删除记录        | 116        |
| 5.1.2 项目管理器概述     | 69        | 6.4.3 更新记录        | 116        |
| 5.2 数据库的设计        | 76        | <b>第7章 查询和视图</b>  | <b>118</b> |
| 5.2.1 数据库的设计原则    | 76        | 7.1 查询设计          | 118        |

|                             |            |                                  |            |
|-----------------------------|------------|----------------------------------|------------|
| 7.1.1 查询设计器 .....           | 118        | 9.1.5 事件过程 .....                 | 175        |
| 7.1.2 使用查询向导建立查询 .....      | 120        | 9.1.6 方法 .....                   | 175        |
| 7.1.3 使用查询设计器建立查询 .....     | 122        | 9.2 Visual FoxPro 中的类和对象 .....   | 175        |
| 7.1.4 查询文件的操作 .....         | 127        | 9.2.1 Visual FoxPro 中的常用基类 ..... | 175        |
| 7.2 视图设计 .....              | 130        | 9.2.2 类的创建 .....                 | 177        |
| 7.2.1 视图的概念与视图设计器 .....     | 130        | 9.3 对象的操作 .....                  | 182        |
| 7.2.2 建立视图 .....            | 132        | 9.3.1 对象的创建 .....                | 182        |
| 7.2.3 使用视图更新数据 .....        | 134        | 9.3.2 对象的引用 .....                | 183        |
| 7.2.4 操作和使用视图 .....         | 137        | 9.3.3 设置对象属性 .....               | 184        |
| <b>第 8 章 结构化程序设计 .....</b>  | <b>138</b> | 9.3.4 方法的调用 .....                | 184        |
| 8.1 程序设计概述 .....            | 138        | <b>第 10 章 表单的设计与应用 .....</b>     | <b>186</b> |
| 8.1.1 程序设计相关概念 .....        | 138        | 10.1 创建表单 .....                  | 186        |
| 8.1.2 程序文件的建立和编辑 .....      | 140        | 10.1.1 使用表单向导创建表单 .....          | 186        |
| 8.1.3 交互式输入输出命令 .....       | 142        | 10.1.2 使用“表单设计器”创建表单 .....       | 192        |
| 8.2 程序的基本结构 .....           | 145        | 10.1.3 使用“表单生成器”创建表单 .....       | 196        |
| 8.2.1 顺序结构 .....            | 145        | 10.2 表单的操作 .....                 | 197        |
| 8.2.2 选择结构 .....            | 147        | 10.2.1 表单的保存、运行和修改 .....         | 197        |
| 8.2.3 循环结构 .....            | 151        | 10.2.2 设置表单的属性 .....             | 198        |
| 8.3 过程与过程文件 .....           | 155        | 10.2.3 表单的事件和方法 .....            | 200        |
| 8.3.1 子程序和自定义函数 .....       | 156        | 10.2.4 为表单添加数据环境 .....           | 201        |
| 8.3.2 过程的建立和调用 .....        | 158        | 10.3 表单控件的操作 .....               | 205        |
| 8.3.3 变量的作用域 .....          | 160        | 10.3.1 控件的基本操作 .....             | 205        |
| 8.4 数组的应用 .....             | 164        | 10.3.2 控件的布局 .....               | 205        |
| 8.4.1 数组中常用的语句 .....        | 164        | 10.3.3 设置控件的 Tab 次序 .....        | 206        |
| 8.4.2 数组中常用的函数 .....        | 166        | 10.4 常用表单控件 .....                | 207        |
| 8.5 程序的调试 .....             | 167        | 10.4.1 标签控件和文本框控件 .....          | 208        |
| 8.5.1 程序调试概述 .....          | 167        | 10.4.2 编辑框控件和列表框控件 .....         | 212        |
| 8.5.2 调试器 .....             | 167        | 10.4.3 命令按钮控件和命令组控件 .....        | 218        |
| <b>第 9 章 面向对象程序设计 .....</b> | <b>171</b> | 10.4.4 复选框控件和选项按钮组控件 .....       | 222        |
| 9.1 面向对象基本概念的引入 .....       | 171        | 10.4.5 组合框控件与表格控件 .....          | 225        |
| 9.1.1 对象 .....              | 172        | 10.4.6 计时器控件和微调控件 .....          | 229        |
| 9.1.2 类 .....               | 172        | 10.4.7 页框控件和图像控件 .....           | 230        |
| 9.1.3 属性 .....              | 174        | 10.5 表单应用实例 .....                | 231        |
| 9.1.4 事件 .....              | 174        | 10.5.1 设计表单的一般过程 .....           | 232        |
|                             |            | 10.5.2 设计系统说明表单 .....            | 232        |

|                       |     |                              |     |
|-----------------------|-----|------------------------------|-----|
| 10.5.3 设计系统登录表单       | 233 | 12.1.2 使用报表向导创建报表            | 254 |
| 10.5.4 设计数据查询表单       | 234 | 12.1.3 使用快速报表创建简单规范<br>报表    | 257 |
| <b>第 11 章 菜单工具栏设计</b> | 236 | 12.1.4 使用报表设计器创建自定义<br>报表    | 259 |
| 11.1 菜单系统概述           | 236 | 12.2 报表的设计                   | 261 |
| 11.1.1 菜单结构与组成        | 236 | 12.2.1 报表的控件设计               | 261 |
| 11.1.2 系统菜单的结构与定制     | 237 | 12.2.2 美化报表                  | 267 |
| 11.1.3 建立菜单系统的步骤      | 238 | 12.2.3 报表输出                  | 269 |
| 11.2 创建下拉式菜单          | 238 | <b>第 13 章 综合应用实例</b>         | 272 |
| 11.2.1 菜单设计器窗口        | 239 | 13.1 数据库应用系统的开发步骤            | 272 |
| 11.2.2 设置菜单的属性        | 240 | 13.1.1 应用系统开发的步骤             | 272 |
| 11.2.3 创建下拉菜单实例       | 242 | 13.1.2 Visual FoxPro 应用系统的组成 | 275 |
| 11.2.4 为顶层表单添加菜单      | 246 | 13.2 应用系统的分析与设计              | 276 |
| 11.3 创建快捷菜单           | 247 | 13.2.1 系统需求分析                | 276 |
| 11.4 工具栏的设计           | 249 | 13.2.2 系统设计                  | 276 |
| 11.4.1 定制工具栏          | 249 | 13.3 应用系统的发布过程               | 288 |
| 11.4.2 自定义工具栏         | 250 | 13.3.1 编译应用系统                | 288 |
| <b>第 12 章 报表设计</b>    | 253 | 13.3.2 系统发布过程                | 291 |
| 12.1 创建报表             | 253 | 13.3.3 制作应用系统发布盘             | 292 |
| 12.1.1 报表布局及布局文件      | 254 |                              |     |

# 第 1 章

## 数据库系统基础知识

### 内容导读:

数据库系统 (DataBase System, DBS) 是指引进数据库技术的计算机系统。数据库技术是从 20 世纪 60 年代末逐步发展起来的计算机软件技术, 本章介绍有关数据库的一些基本概念和知识, 主要内容有: 数据库的基础知识、数据库系统的组成、关系模型、关系完整性约束等。

### 教学目标:

随着计算机技术的蓬勃发展, 计算机应用已经涉足人们的日常生活、工作的各个领域。学习 Visual FoxPro 就是希望能够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数据, 作为进行各项决策活动的依据。所以, 首先应该了解和掌握有关数据库的一些基本概念和知识。通过学习这些知识, 读者可以了解数据库技术的相关概念、知识和技能, 从而为进一步学习数据库技术及其应用奠定基础。

### 重点难点:

- 数据与信息
- 数据库系统的组成
- 数据模型
- 关系数据库

## 1.1 数据库基础知识

客观世界是由信息、能源和材料这三大要素构成。在当今世界, 信息是一种非常重要、有价值的资源。信息是企业进行生产活动、经济活动和社会活动必不可少的基本资源, 是其赖以生存和发展的根本。信息也是决定一个国家的建设和发展的重要因素。因此, 人们想要获取对其决策有价值的信息, 就必须对信息和用于表示信息的数据进行处理和管理。通常, 把用计算机对数据进行处理的应用系统称为计算机信息系统, 其核心是数据库。本节将介绍数据库中的一些基本知识。

### 1.1.1 信息与数据

信息和数据是两个不同的概念。信息是客观事物属性的反映, 是事物之间相互联系、相互作用的状态描述。为了认识客观世界和进行彼此交流, 人们需要各种信息。如一位教师的信息可以用如下格式的数据来描述: “李明, 男, 1975, 陕西, 副教授, 信息工程学院”。该数据包含的信

息为：一位叫李明的男教师，1975 年出生，陕西人，职称是副教授，是该校信息工程学院的教师。

信息具有如下重要特征。

- (1) 信息具有表征性。它能够表达事物的属性、运动特征及状态。
- (2) 信息具有可传播性。信息可以进行获取、存储、传递和共享。
- (3) 信息具有可处理性。信息可以进行压缩、加工以及再生。
- (4) 信息具有价值性和可转换性。

与信息相关的是数据。数据是信息的具体表现形式，是信息的载体。一般认为，数据是人们用于记录事物情况的物理符号。为了描述客观事物而用到的数字、字符，以及所有能输入到计算机中并能被计算机处理的符号都被认为是数据。在实际应用中，数据的表示形式有两种：一种是可以参与数值运算的数值型数据，如表示工资、成绩的数据；另一种是由不能参与数值运算的数据，如字符（文字和符号）、图表（图形、图像和表格）、动画、影像、声音等多媒体数据。但其中使用最多、最基本的仍然是文字数据。数据以格式化的形式来表示事实和概念，这种形式有助于通信、解释和处理。

数据有两方面的特征：一是客体属性的反映，这是数据的内容；二是记录信息的符号，这是数据的形式。形式是内容的表现方式，内容是形式的实质。

信息和数据既有区别，又有联系。一方面，信息是有价值的，它以数据为载体，依靠数据来完成信息的传播，而数据却不一定具有价值。信息是向人们提供关于现实事物的知识，数据则是载荷信息的物理符号。另一方面，信息可以用不同的数据来表示，它不会因数据的形式不同而不同；数据具有任意性，在实际应用中可以用不同的数据来表示同一信息。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。在一些非严格的场合中，信息和数据之间又没有做严格的区分，甚至可以混用，如信息处理就是数据处理，信息采集就是数据采集等。

## 1.1.2 数据处理

数据处理是指将数据转换成信息的过程，数据处理也可称为信息处理。它包括对数据的采集、整理、存储、分类、检索、排序、统计、维护、传输等一系列活动。对数据进行处理的主要目的就是大量杂乱无章的、难以理解的、原始的数据中整理出对人们有价值、有意义的信息，作为日常工作和决策的依据。例如，公务员考试中，全体考生的考试成绩记录了考生的考试情况，属于原始数据，对考试成绩进行分析和处理，如对成绩从高到低排序，可以根据招考人数确定分数线。所以日常工作中，财务、人事、审计、办公自动化等方面离不开数据处理。反过来，要想使所获得的信息能够充分地发挥作用，就必须对所拥有的数据进行处理。

通常将数据处理分为两个操作层次：一是数据采集、分类、组织、编码、存储、检索、传输、维护等基本操作，这些基本操作称为数据管理；二是加工、计算、输出等操作，管理对象不同，操作的要求也不同。由于这些操作主要由应用程序来实现，因此这些操作称为应用操作。严格地说，信息处理中包含数据处理，而数据处理仅仅是信息处理中最主要的内容。

根据处理设备的结构和工作方式以及数据分布方式的不同，数据处理的方式主要有以下 4 种。

- (1) 按处理设备结构方式分，可分为联机处理方式和脱机处理方式。
- (2) 按数据处理时间的分配方式分，可分为批处理方式、分时处理方式和实时处理方式。
- (3) 按数据处理空间的分布方式分，可分为集中式处理方式和分布处理方式。
- (4) 按计算机中央处理器的工作方式分，可分为单道作业处理方式、多道作业处理方式和交

交互式处理方式。

数据处理有不同的方式,不同的处理方式要求不同的硬件和软件支持。因此,每种处理方式都有自己的特点,应当根据应用问题的实际环境选择合适的处理方式。

数据处理是系统工程和自动控制的基本环节,贯穿于社会生产和社会生活的各个领域。目前,随着数据处理技术的发展及其应用的广泛和深入,数据处理已经极大地影响了人类社会发展的进程。当然在数据处理中离不开软件的支持,常见的数据处理软件包括:各种程序设计语言及其编译程序,管理数据的文件系统和数据库系统,各种数据处理方法的应用软件包。同时,为了保证数据的安全,还应具有一整套数据安全的保密技术。

### 1.1.3 数据库管理技术的发展

#### 1. 数据库的概念

数据库(DataBase)是依照某种数据模型组织起来的、相互关联的、并存放于二级存储器中的数据集合。在数据库中集中了一个部门或单位的完整的数据资源,这些数据能够为多个用户同时共享,且具有冗余度小、独立性强和安全性高的特点。这种数据集合以最优方式为某个特定组织提供多种应用服务。换句话说,在日常工作中,需要处理的数据量往往都很大,为便于计算机对这些数据进行有效的处理,我们可以将采集的数据存放在磁盘、光盘等外存媒介的“仓库”中,这个“仓库”就是数据库。由于数据库本身的数据结构特征,使得数据库独立于使用它的应用程序,对数据的增加、删除、修改和检索则由相关的软件进行管理和控制,从而实现了数据和操作的分离。

将所有数据集中存放在数据库中,一方面便于人们对其进行统一管理,另一方面也便于人们提炼出对决策有用的数据和信息。例如,一个学校把采购的大量书籍存放在资料室(书库)中,以供教师和学生借阅;一个汽车企业生产制造出的汽车要先存放在仓库中,这样既便于统计和管理,又便于以后分批地把汽车销售给客户。因此可以说,数据库就是在计算机存储器中用于存储数据的仓库。从历史的发展角度看,数据库是数据管理的高级阶段,它是由文件管理系统发展起来的。

#### 2. 数据库的产生

实际上,数据库系统并不是和计算机同时出现的,而是随着计算机硬件技术和软件技术的发展,以及社会对数据处理需求的不断发展而产生的。计算机数据管理的方式也在不断改进,经历了从人工管理到文件系统再到数据库系统3个阶段。

(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,数据量较少,一般不需要长期保存。硬件方面,没有磁盘等直接存取的外存储器,数据只能存放于卡片、纸带或磁带上。软件方面只有汇编语言,没有专门的数据管理软件,数据由计算或处理它的程序自行携带,因此数据的管理是靠人工进行的,当计算机运行结束后,便将结果输出,由人工保存,计算机并不存储数据。在此阶段,对数据的管理是由程序员个人考虑和安排的,他们既要设计算法,又要考虑数据的逻辑结构、物理结构、输入/输出方法等,应用程序的设计和维持的负担繁重。

该阶段主要存在以下几方面的问题。

① 数据不能长期保存。数据被包含在程序中,程序运行结束后数据和程序一起从内存中释放。

② 数据不独立。编写的程序是针对程序中的数据。所以,当数据修改时程序也得随之修改,而程序修改后,数据的格式、类型也得做相应的变化。

③ 数据不能重复使用。由于没有对数据管理的软件应用,程序和数据是一个整体,一个程序

的数据不能被其他程序使用，导致程序与程序之间存在大量的重复数据。

(2) 文件系统阶段。20 世纪 50 年代后期到 60 年代中期，计算机开始大量用于数据管理。计算机硬件和软件技术得到了飞速的发展。其中，硬件方面有了磁盘、磁鼓等大容量且能长期保存数据的存储设备，这为计算机管理系统数据提供了物质基础；软件方面有了操作系统，其中包含文件系统，后来还出现了高级语言，这又为计算机管理系统数据提供了操作基础。操作系统中设有专门的文件系统用于管理外部存储器上的数据文件，不同数据用不同的文件名表示，程序只需文件名访问数据，不必关心记录在存储器上的地址和内、外存交换数据的过程。

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的区别，即程序与数据具有一定的独立性。数据的存储结构变化，不一定影响程序，因此程序员可集中精力进行算法的设计，大大减少了维护程序的工作量。可以说，该阶段做到了数据与程序分开。该阶段是按照数据文件的形式来存放数据的。在一个文件中包含了若干个“记录”，一个记录又包含若干个“数据项”，用户通过对文件的访问实现对记录的存取。这种数据管理方式称为文件管理系统。文件系统使计算机在数据管理方面有了长足进步。时至今日，文件系统依然是一般高级语言普遍采用的数据管理方式。

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地使用数据的需求了。文件管理系统的一个严重不足就是数据的管理没有实现结构化组织，数据与数据之间没有联系，文件与文件之间也没有形成有机的联系，数据不能脱离建立其数据文件的程序，也致使文件管理系统中数据的独立性和一致性较差，冗余度较大，从而限制了大量数据的共享和有效应用。文件系统阶段存在着以下几个问题。

① 数据冗余度大。数据没有合理和规范的结构，使得数据的共享性极差，即使不同程序使用部分相同数据，也得要创建各自的数据文件，造成数据的重复存储。

② 数据独立性差。在文件系统中，虽然数据和程序分开，但是数据文件是为了满足特定业务领域或某部门的需要而专门设计的，该数据只服务于某一特定应用程序。这样，所设计的数据是针对某一特定程序，所以无论是修改数据文件还是修改程序文件都会相互影响。

③ 缺乏对数据的集中管理。在同一个项目中的各个数据文件没有统一的管理机制，数据的安全性和完整性都很难得到保证。各数据之间、数据文件之间缺乏联系，数据处理很不方便。数据的保护等都交给应用程序去解决，使得应用程序的编写非常烦琐。

(3) 数据库系统阶段。20 世纪 60 年代后期，随着计算机技术的发展，计算机在管理中应用越来越广泛，规模越来越庞大，数据量急剧增加，数据共享性更强。为了克服文件管理系统的缺点，人们对文件系统进行了扩充，研制了一种结构化的数据组织和处理方式，即数据库系统。在数据库系统中，有一种叫做数据库管理系统 (DataBase Management System, DBMS) 的系统软件将所有的数据集中到一个数据库中，形成一个数据中心，对数据进行统一的控制。它使数据的存取独立于使用数据的程序，从而有效地减少了数据冗余，解决了多用户、多应用程序共享数据的需求，实现了数据的独立和集中管理。用户的应用程序与数据的逻辑结构及数据的物理存储方式无关，用户只需通过数据库管理系统来使用数据库中的数据。

在数据库系统阶段，应用程序和数据完全独立，应用程序对数据管理和访问更加灵活，一个数据库可以为多个应用程序共享，使得程序的编制和效率大大提高。同时还减少了数据的冗余度，实现数据资源共享，提高了数据的完整性、一致性以及数据的管理效率。

### 3. 数据库的特点

数据库就是在数据库管理软件的集中控制下，按照以一定的组织方式存储在一起的、相互关

联的数据的集合。它不仅存储数据本身，而且还存储数据之间的联系。数据库技术在 20 世纪 60 年代后期发展起来后，在计算机应用中得到了迅速的发展。数据库技术先后经历了层次数据库、网状数据库和关系数据库几个阶段，其中占主导地位的是关系数据库。随着应用的不断深入，数据库技术已经成为信息管理最新、最重要的技术。

数据库具有数据共享、数据独立、最小冗余、数据安全、数据完整、灵活性、可恢复性等特点。

(1) 数据共享。因为数据库中的数据是按某种数据模型组织为一个结构化的数据，因此能够实现多个应用程序、多种语言及多个用户能够共享一个库中的数据，甚至可以在一个单位或更大范围内共享。数据共享不仅可以提高数据的利用率，同时也可以提高工作效率。数据共享是数据库技术先进性的重要体现。

(2) 数据独立性强。在数据库技术中，数据与程序相互独立，互不依赖。数据和程序之间不因一方的改变而改变，减少了应用程序设计与维护的工作量，而且数据也不会因程序的结束而消失，数据可长期保留在计算机系统中。

(3) 数据冗余度小。在数据库技术之前，许多应用系统都需要建立各自的数据文件，即使相同的数据都需要在各自的系统中保留，从而造成大量的数据重复存储，这一现象称为数据的冗余。数据库实现了数据共享后，便减少了存储数据的重复，节省了存储空间，减少了数据冗余。

(4) 数据结构化。由于数据库中的数据不再像文件系统中的数据那样从属于某个特定的应用，而是按照某种数据模型组织成为一个结构化的数据整体。它不仅描述了数据本身的特性，而且描述了数据与数据之间的种种联系，这使数据库具备了复杂的内部组织结构。

## 1.2 数据模型

数据模型是描述数据及数据之间联系的结构形式，它主要研究如何组织数据库中的数据，这是数据库的核心内容。通常，数据模型可以用图解的方法来表示数据库中的数据结构形式。为了建立数据模型，就必须首先对需要描述的事物进行抽象。

### 1.2.1 基本概念

现实世界是存在于人们头脑之外的客观世界，现实世界的现实事物经过人们头脑的认识、整理、分类之后进入信息世界，以实体模型的形式表现出来。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形式的逐级抽象过程。实体模型也称概念模型或信息模型，它是按用户的观点对现实世界中的事物所建立的一种模型。这类模型概念简单、清晰、与计算机无关，且用户易于理解，是用户与数据库设计人员之间交流的语言。实现实体模型的过程就是实现现实世界到计算机世界的两级抽象中的第一级抽象——信息抽象过程。

(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，实体可定义为客观存在的并相互区分的“事物”。它可以指人，如一个教师、一个演员等，也可以指物，如一本小说、一把椅子等。它可以指实际的物体，也可以指概念性的东西，如借书、奖励等。性质相同的同类实体的集合称为“实体集”，也称为实体整体，如所有的教师、所有的椅子、所有的工厂、所有的小说等。

(2) 属性。一个实体具有不同的属性，属性描述了实体某一方面的特性。例如，教师实体的

姓名、性别、年龄、职称、学位等都是教师的属性。图书实体用总编号、分类号、书名、作者、单价等多个属性来描述。

(3) 实体集。属性有“型”和“值”的区分,如教师属性的姓名、性别、年龄、职称、学位等是属性的型,而属性的值是其型的具体内容。如“李明、男、35、副教授、博士”分别是姓名、性别、年龄、职称、学位的值。

由上述可见,若干个属性值所组成的集合表征一个实体,相应的属性型的集合表征一种实体的类型,称为实体型。例如,上面的教师姓名、性别、年龄、职称、学位等表征教师实体的实体型。同类型实体的集合称为实体集。

在 Visual FoxPro 中,用“表”来存放同一类实体,如教师表存放教师实体,成绩表存放成绩实体。每一个“表”包含若干个字段,“表”中所包含的“字段”就是实体的属性,字段值的集合组成表中一条记录,代表一个具体的实体,即表中的一条记录表示一个实体。数据库中的数据是结构化的,即建立数据库就需要考虑如何去组织数据,如何表示数据及数据之间的联系,并将其合理地存放在计算机中,以便于对其进行有效的处理。

## 1.2.2 实体及其之间的关系

实体之间的对应关系称为实体间的联系,它反映现实世界事物之间的相互关联。例如,教师和课程之间的关联关系为:一个教师可以讲授多门课程,一门课程可以由多位教师讲授。

实体间联系是指一个实体型中可能出现的每一个实体与另一个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,归纳起来有以下 3 种类型。

(1) 一对一的联系(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 的联系为一对一的联系。例如,一所学校只有一个校长,一个校长只能在一所学校任职,因此校长和学校之间便存在一对一的联系。

在 Visual FoxPro 中,一对一的联系表现为主表中的一条记录只与相关表中的一条记录关联。即表 A 的一条记录在表 B 中只有一条对应的记录,表 B 中的记录最多只能有一条与表 A 的记录相对应。

(2) 一对多的联系(1:n)。如果对于实体集 A 中的一个实体,实体集 B 中有多个实体与之联系,反之,对于实体集 B 中的一个实体,实体集 A 中至多有一个实体与之联系,则称实体集 A 与实体集 B 的联系为一对多的联系。例如,一所学校有多个学生,而一个学生只能就读于一所学校,学校与学生的联系就是一对多的联系,还有公司与员工、母亲与子女都是一对多的联系。

在 Visual FoxPro 中,一对多的联系主要表现为表中的每一条记录,在相关表中有多条记录相关联,即表 A 中的一条记录在表 B 中可以有多条记录与之对应,但表 B 中的一条记录在表 A 中最多只能有一条记录与之对应。

(3) 多对多的联系(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 的联系为多对多的联系。例如,一个教师可以讲授多门课程,而一门课程可以由多位教师讲授,教师与课程的联系即为多对多的联系,还有教师与学生、学生与课程、工厂与产品等都是多对多的联系。

在 Visual FoxPro 中,多对多的联系表现为一个表中的多个记录在相关表中同样有多个记录与其匹配。即表 A 中的一条记录在表 B 中可以对应多条记录,而表 B 的一条记录在表 A 中也可以对应多条记录。

### 1.2.3 数据模型

数据是描述客观事物的载体，而现实世界中事物总是彼此联系的，因此数据与数据之间必然存在一定的联系，我们可以用数据模型来描述这种联系。

数据模型是反映事物间联系的数据组织的结构和形式。一个具体的数据模型应当正确地反映出数据之间存在的整体逻辑关系。数据模型包含两个方面的内容：一为数据的静态特性，即数据的基本结构、数据间的联系和数据的约束；二为数据的动态特性，即定义在数据上的操作，如教师记录中包含姓名、职工编号、性别、出生年月、职称等字段，每个字段都有数据和长度约束，对教师记录也可以进行添加、修改、删除、查询、统计等操作。

由于采用的数据模型不同，相应的数据库管理系统也就完全不同。在数据库管理系统中，常用的数据模型有层次模型、网状模型、关系模型 3 种。因此，使用支持某种特定数据模型的数据库管理系统开发出来的应用系统，相应地称为层次数据库系统、网状数据库系统、关系数据库系统。其中，关系模型对数据库的理论和实践产生很大的影响，它已成为当今最流行的数据库模型。

(1) 层次模型。层次模型用树型结构来描述实体及它们之间的关系。在这种模型中，数据间的联系是一种从属关系，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的类型，连线表示它们之间的关系。除根之外，所有的子结点都应有唯一的父结点。

实际生活中，许多实体间的联系本身就是自然的层次关系，比如一个单位的行政机构、一个家庭的世代关系等。图 1.1 所示为一个层次模型的例子。

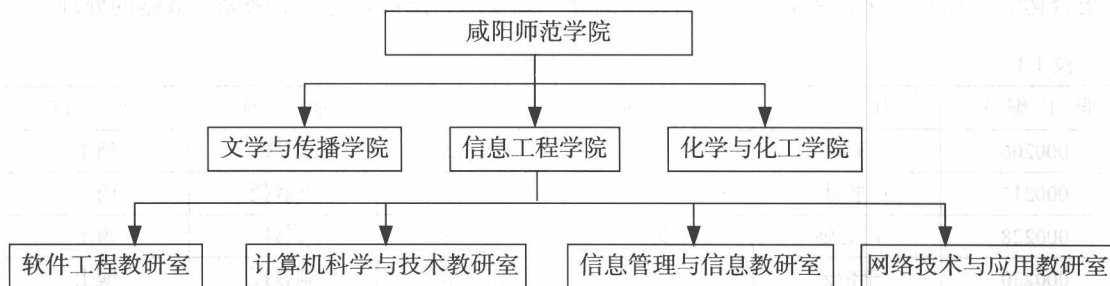


图 1.1 层次模型

层次模型层次清晰、构造简单、易于实现。它可以比较方便地表示实体间一对一和一对多的联系，而不能直接表示出多对多的实体联系。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。因此，对于复杂的数据关系，层次模型实现起来比较麻烦。

采用层次数据库模型的 DBMS 称为层次数据库管理系统，在这种系统中建立的数据库是层次数据库。层次模型的数据库管理系统是世界上最早出现的大型数据库系统，其典型代表是 IBM 的 IMS (Information Management System)。

(2) 网状模型。网状模型用网状结构表示实体及其之间联系。网中的每一个结点代表一个实体类型。网状模型的特点是：一是可以有一个以上的结点没有父结点；二是至少有一个结点有多个父结点。因此，网状模型可以方便地表示各种类型的联系。简单的网状模型示例如图 1.2 所示。

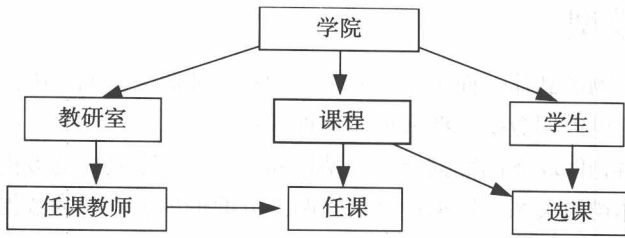


图 1.2 网状模型

网状模型和层次模型在本质上是一样的，树形结构可以看成是网状模型的特例，所以网状模型要比层次模型复杂。网状模型的主要优点是表示多对多的联系具有很大的灵活性，但这种灵活性是以数据结构复杂为代价的。网状模型中，每一个联系都代表实体之间一对多的联系，系统用单项或多项链接指针来具体实现这种联系。支持网状模型的 DBMS 称为网状数据库管理系统，在这种系统中建立的数据库是网状数据库。

从逻辑上看，层次模型和网状模型都是用结点表示实体，用“—►”表示实体间的联系，实体和联系用不同的方法来表示；从物理上看，每一个结点都是一个存储记录，用链接指针来实现记录之间的联系。这种用指针将所有数据记录都“捆绑”在一起的特点，使得层次模型和网状模型存在难以实现系统的修改及扩充等缺陷。

(3) 关系模型。关系模型与以上的两种模型相比，有着本质的区别，它是用二维表格来表示实体以及实体之间联系。在关系模型中，操作的对象和结果都是二维表，这种二维表就是关系。

如表 1.1 所示，表中每一列是一个属性，每一行称为一个元组，即一条记录，可以采用传统的集合运算（如并、交、差）和专门的关系运算（如投影、选择和连接）来完成数据的处理。

表 1.1 关系模型

| 职工编号   | 姓名  | 性别 | 出生年月    | 职称  | 学位 |
|--------|-----|----|---------|-----|----|
| 000206 | 王丽  | 女  | 1982.5  | 讲师  | 硕士 |
| 000217 | 李明  | 男  | 1975.12 | 副教授 | 硕士 |
| 000228 | 冯志强 | 男  | 1979.8  | 讲师  | 硕士 |
| 000230 | 李倩云 | 女  | 1973.10 | 副教授 | 博士 |
| 000232 | 张辉  | 男  | 1964.6  | 教授  | 硕士 |
| 000239 | 吴江波 | 男  | 1985.3  | 助教  | 学士 |

关系模型与层次模型、网状模型的本质区别在于数据描述的一致性，关系模型概念单一。描述实体的数据本身能够自然地反映它们之间的联系。而传统的层次和网状模型数据库是使用链接指针来存储和体现联系。

在关系模型数据库中，无论实体本身还是实体间的联系均用“关系”二维表来表示，每一个实体都是一个二维表，每一个二维表都是一个关系。每个关系均有一个名字，称为关系名。

虽然关系数据模型出现比层次模型和网状模型都晚，但由于关系模型是建立在严格的关系数学理论基础上的，所以是目前十分流行的一种数据模型。关系数据库以其完备的理论基础、简单的模型、说明性的查询语言、使用方便等优点得到了最广泛的应用。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，本书讨论的 Visual FoxPro 就是一种关系数据库管理系统。