

高等学校计算机教材

Visual FoxPro 程序设计

高怡新 编著

人民邮电出版社

图书在版编目(CIP)数据

Visual FoxPro 程序设计 / 高怡新编著. —北京:人民邮电出版社, 2003.4

高等学校计算机教材

ISBN 7-115-10911-7

I. V... II. 高... III. 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. TP311.138

中国版本图书馆 CIP 数据核字(2003)第 017532 号

内 容 提 要

本书是根据最新的计算机教学大纲,在作者总结多年从事“Visual FoxPro 程序设计”教学经验的基础上编著而成的,全书共分 10 章,主要内容包括:数据库和数据库管理系统的基本概念、Visual FoxPro 数据表和数据库的基本操作、传统的面向过程的程序设计方法、先进的面向对象的可视化程序设计方法和一个基于数据库的应用程序开发实例等。

本书的特点是概念清晰、编排合理、实例丰富,且理顺了众多知识点之间的关系,有利于读者的理解与掌握。在每章后均附有一定数量的习题,便于复习参考。

本书适合作为各类院校的“Visual FoxPro 程序设计”课程或“数据库应用”课程教材,也可作为各类培训班的相关课程教材。对于参加计算机二级 Visual FoxPro 考试的应试者,也是一本相当实用的参考书。

高等学校计算机教材

Visual FoxPro 程序设计

编 著 高怡新

责任编辑 赵鹏飞

◆ 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 电子函件 315@ptpress.com.cn

网址 <http://www.ptpress.com.cn>

读者热线 010-67180876

北京汉魂图文设计有限公司制作

北京鸿佳印刷厂印刷

新华书店总店北京发行所经销

◆ 开本:787×1092 1/16

印张:20.5

字数:496 千字

2003 年 4 月第 1 版

印数:1-6 000 册

2003 年 4 月北京第 1 次印刷

ISBN 7-115-10911-X/TP·3057

定价:27.00 元(附光盘)

本书如有印装质量问题,请与本社联系 电话:(010)67129223

前 言

近年来, Visual FoxPro 迅速替代其他同类软件,成为 PC 机上影响最大、应用最广的关系型数据库管理系统。在数据操纵方面,它基本具备了大型 SQL 数据库管理系统的特征,如视图、关联、数据字典、触发器、存储过程等;在数据库设计方面,它提供了各种数据库组件的设计向导,以及图形化、智能化的设计过程。

Visual FoxPro 安装容易,运行环境要求不高,便于学习掌握,易于被广大用户所接受。更为可取的是,它不仅支持先进的面向对象程序设计方法,而且提供了用以开发各种数据库应用系统界面的可视化设计工具。它一改传统程序设计的概念与方法,使得用户只需编写少量简单的代码,就可利用鼠标方便、快捷、直观地创建各种 Windows 风格的窗口、对话框、菜单和报表等,从而极大地提高了应用程序界面的友好性和程序设计的效率,因而在企事业管理和办公自动化等各方面得到了普遍的应用。

本书是根据最新的计算机教学大纲,在作者多年从事“ Visual FoxPro 程序设计”教学的经验基础上编著而成的。全书共分 10 章,大致可归纳为如下 5 部分内容。

第 1、2 章主要介绍数据库和数据库管理系统的概念,以及 Visual FoxPro 数据元素的基础知识。其中有关数据类型、常量、变量、函数、表达式等的概念至为重要,是学好 Visual FoxPro 的基础。

第 3、4、5 章主要讲解 Visual FoxPro 数据表和数据库的基本操作,详细说明在交互方式下用户如何建立、维护、使用数据表和数据库,如何增删和修改表中的数据记录、按要求统计汇总数据、按指定的关键字对数据记录进行物理或逻辑排序,以及如何从单个或多个数据表中查询所需的数据等。掌握了这部分内容,就可利用 Visual FoxPro 做一些简单的数据管理工作,同时这些知识也是进一步学习程序设计的基础。

第 6 章讲述面向过程的程序设计方法,这是进一步学习面向对象程序设计的基础。本章在阐述顺序、分支、循环等基本程序结构及相应流程控制语句的同时,介绍了结构化的程序设计思想。其间通过大量的例子介绍了传统程序设计的常用方法与技巧,力图使读者在学习理解的基础上,能够举一反三,迅速掌握这些方法并学会编写一些简单的应用程序。

第 7、8、9 章是本书的精华部分,在阐述了面向对象程序设计的概念、方法与术语之后,着重介绍了利用各种设计器可视化地设计表单、报表和菜单的方法和步骤。从某种意义上讲,面向对象程序设计的关键就是对应用程序用户界面的设计,而组成用户界面的各种窗口和对话框则均是表单的不同表现形式,因而表单设计是这一部分最重要的内容。

第 10 章提供了一个应用程序的开发实例,较详细地讲述了开发一个基于数据库的应用系统所需的各个步骤,包括用户需求分析与目标设计、系统功能分析与模块设计、数据库需求分析与逻辑结构设计、各功能模块的创建与调试,以及整个系统的编译与运行等。该实例融汇了前面所讲的各种知识,介绍了一些编程思想与技巧,并经实际上机验证通过,因而对于读者提高水平、开阔眼界,进而开发出真正实用的管理系统甚至商品化的应用软件,均具有一定的参考价值。

本书的特点是概念清晰、编排合理、实例丰富，且理顺了众多知识点之间的关系，有利于读者的理解与掌握。在每章后均附有一定数量的习题，便于复习参考。

本书适合作为各类院校的“ Visual FoxPro 程序设计 ”课程或“ 数据库应用 ”课程教材，也可作为各类培训班的相关课程教材。对于参加计算机二级 Visual FoxPro 考试的应试者，也是一本相当实用的参考书。衷心希望每位读者都能从本书中获益，同时欢迎对本书的不足提出宝贵意见。

作者

2002 年 12 月

目 录

| | |
|---------------------------|----|
| 第 1 章 数据库系统概论 | 1 |
| 1.1 数据库系统的基本概念 | 1 |
| 1.1.1 信息、数据与数据库 | 1 |
| 1.1.2 数据管理技术的发展 | 2 |
| 1.1.3 数据模型 | 3 |
| 1.1.4 数据库管理系统 | 5 |
| 1.1.5 数据库系统 | 6 |
| 1.2 关系型数据库 | 7 |
| 1.2.1 关系术语与关系特点 | 8 |
| 1.2.2 关系运算 | 8 |
| 1.3 Visual Foxpro 概述 | 10 |
| 1.3.1 Visual FoxPro 的特点 | 10 |
| 1.3.2 Visual FoxPro 的用户界面 | 12 |
| 1.3.3 Visual FoxPro 的工作方式 | 13 |
| 1.3.4 Visual FoxPro 的设计工具 | 14 |
| 1.3.5 Visual FoxPro 的环境设置 | 16 |
| 习题一 | 19 |
| 第 2 章 数据与数据运算 | 21 |
| 2.1 常量与变量 | 21 |
| 2.1.1 常量 | 21 |
| 2.1.2 字段变量 | 23 |
| 2.1.3 内存变量 | 24 |
| 2.2 运算符与表达式 | 26 |
| 2.2.1 数值表达式 | 27 |
| 2.2.2 字符表达式 | 27 |
| 2.2.3 日期表达式 | 28 |
| 2.2.4 逻辑表达式 | 29 |
| 2.2.5 表达式书写规则 | 32 |
| 2.2.6 表达式输出命令 | 33 |
| 2.3 常用函数 | 33 |
| 2.3.1 数学运算函数 | 34 |
| 2.3.2 字符处理函数 | 35 |
| 2.3.3 日期时间函数 | 37 |
| 2.3.4 转换函数 | 38 |
| 2.3.5 测试函数 | 40 |

| | | |
|--------------|----------------|-----------|
| 2.4 | 命令格式与文件类型 | 42 |
| 2.4.1 | 命令格式 | 42 |
| 2.4.2 | 文件类型 | 43 |
| | 习题二 | 44 |
| 第 3 章 | 数据表基本操作 | 48 |
| 3.1 | 数据表的建立 | 48 |
| 3.1.1 | 表结构的建立与修改 | 48 |
| 3.1.2 | 表记录的输入 | 51 |
| 3.1.3 | 表数据的浏览与维护 | 53 |
| 3.2 | 数据表操作命令 | 55 |
| 3.2.1 | 创建数据表命令 | 55 |
| 3.2.2 | 追加与插入记录命令 | 59 |
| 3.2.3 | 记录指针移动命令 | 60 |
| 3.2.4 | 显示记录命令 | 61 |
| 3.2.5 | 删除记录命令 | 62 |
| 3.2.6 | 修改记录命令 | 63 |
| 3.3 | 数据表的排序与索引 | 66 |
| 3.3.1 | 物理排序 | 66 |
| 3.3.2 | 索引类型 | 68 |
| 3.3.3 | 索引文件的建立 | 69 |
| 3.3.4 | 索引文件的打开与关闭 | 73 |
| 3.4 | 数据表的查询与统计 | 76 |
| 3.4.1 | 数据表的查询 | 76 |
| 3.4.2 | 数据表的统计 | 79 |
| 3.5 | 数据表间的操作 | 82 |
| 3.5.1 | 工作区的选择 | 82 |
| 3.5.2 | 表间的数据更新 | 83 |
| 3.5.3 | 表间的物理连接 | 84 |
| 3.5.4 | 表间的逻辑连接 | 85 |
| | 习题三 | 86 |
| 第 4 章 | 数据库及其操作 | 92 |
| 4.1 | 数据库的创建与维护 | 92 |
| 4.1.1 | 数据库的创建 | 92 |
| 4.1.2 | 数据库的打开与维护 | 93 |
| 4.1.3 | 数据库中表的组织 | 93 |
| 4.1.4 | 数据库中表的关联 | 96 |
| 4.2 | 数据字典 | 99 |
| 4.2.1 | 长表名与长字段名 | 99 |
| 4.2.2 | 设置字段属性 | 99 |

| | | |
|--------------|--------------------------|------------|
| 4.2.3 | 设置记录规则 | 100 |
| 4.2.4 | 参照完整性 | 102 |
| 4.3 | 查询的建立与运行 | 103 |
| 4.3.1 | 使用“查询设计器”进行查询 | 103 |
| 4.3.2 | “查询设计器”的界面 | 106 |
| 4.3.3 | “查询”菜单的使用 | 110 |
| 4.4 | 视图的建立与维护 | 111 |
| 4.4.1 | 视图的概念 | 111 |
| 4.4.2 | 视图的创建 | 112 |
| 4.4.3 | 视图与数据更新 | 114 |
| 4.4.4 | 视图的使用与维护 | 115 |
| | 习题四 | 116 |
| 第 5 章 | 结构化查询语言 SQL | 118 |
| 5.1 | SQL 语言概述 | 118 |
| 5.2 | SQL 的定义功能 | 119 |
| 5.2.1 | 建立表结构 | 119 |
| 5.2.2 | 修改表结构 | 122 |
| 5.2.3 | 建立视图 | 124 |
| 5.2.4 | 删除表 | 125 |
| 5.3 | SQL 的数据修改功能 | 125 |
| 5.3.1 | 插入数据 | 125 |
| 5.3.2 | 更新数据 | 126 |
| 5.3.3 | 删除数据 | 127 |
| 5.4 | SQL 的数据查询功能 | 127 |
| 5.4.1 | SQL 的查询命令 | 127 |
| 5.4.2 | 简单查询 | 129 |
| 5.4.3 | 嵌套查询 | 131 |
| 5.4.4 | 连接查询 | 131 |
| 5.4.5 | 使用库函数查询 | 133 |
| 5.4.6 | 分组查询 | 134 |
| | 习题五 | 135 |
| 第 6 章 | 程序设计基础 | 137 |
| 6.1 | 程序文件的建立与运行 | 137 |
| 6.1.1 | 程序文件的建立 | 137 |
| 6.1.2 | 程序文件的运行 | 138 |
| 6.1.3 | 程序中的辅助命令 | 139 |
| 6.1.4 | 程序中的交互输入命令 | 141 |
| 6.2 | 顺序结构程序设计 | 143 |
| 6.2.1 | 基本程序结构 | 143 |

| | | |
|--------------|-----------------|------------|
| 6.2.2 | 顺序结构程序设计 | 144 |
| 6.3 | 分支结构程序设计 | 144 |
| 6.3.1 | 简单分支结构 | 145 |
| 6.3.2 | 选择分支结构 | 146 |
| 6.3.3 | 分支嵌套结构 | 147 |
| 6.3.4 | 多路分支结构 | 147 |
| 6.4 | 循环结构程序设计 | 149 |
| 6.4.1 | 当型循环结构 | 149 |
| 6.4.2 | 步长型循环结构 | 152 |
| 6.4.3 | 扫描型循环结构 | 153 |
| 6.4.4 | 循环嵌套结构 | 154 |
| 6.5 | 模块结构程序设计 | 155 |
| 6.5.1 | 程序模块的建立与运行 | 156 |
| 6.5.2 | 应用程序的模块化设计 | 158 |
| 6.5.3 | 带参模块的建立与调用 | 159 |
| 6.5.4 | 内存变量作用域 | 161 |
| 6.5.5 | 过程与过程文件 | 164 |
| 6.5.6 | 用户自定义函数 | 165 |
| 6.6 | 数组及其应用 | 167 |
| 6.6.1 | 数组的定义与使用 | 168 |
| 6.6.2 | 数组与表之间的数据传递 | 169 |
| 6.6.3 | 数组应用举例 | 172 |
| | 习题六 | 173 |
| 第 7 章 | 面向对象程序设计 | 177 |
| 7.1 | 面向对象程序设计简介 | 177 |
| 7.2 | 对象与类 | 178 |
| 7.2.1 | 对象与类的概念 | 178 |
| 7.2.2 | 对象与类的特性 | 179 |
| 7.3 | 控件类与容器类 | 180 |
| 7.3.1 | 控件类 | 181 |
| 7.3.2 | 容器类 | 182 |
| 7.4 | 属性、方法与事件 | 182 |
| 7.4.1 | 属性 | 182 |
| 7.4.2 | 方法 | 184 |
| 7.4.3 | 事件 | 186 |
| 7.5 | 类的创建与定义 | 187 |
| 7.5.1 | 用程序方式定义类 | 187 |
| 7.5.2 | 用“类设计器”定义类 | 189 |
| 7.6 | 对象的创建与引用 | 191 |

| | | |
|--------------|--------------------------------|------------|
| 7.6.1 | 用程序创建对象 | 191 |
| 7.6.2 | 向容器中添加对象 | 193 |
| 7.6.3 | 对象的引用 | 197 |
| 习题七 | | 198 |
| 第 8 章 | 可视化表单设计 | 201 |
| 8.1 | 表单设计概述 | 201 |
| 8.1.1 | 创建表单的途径 | 201 |
| 8.1.2 | 表单的修改与运行 | 202 |
| 8.1.3 | 常用表单事件与方法 | 203 |
| 8.2 | 使用表单向导创建表单 | 204 |
| 8.2.1 | 使用向导创建单表表单 | 204 |
| 8.2.2 | 使用向导创建一对多表单 | 206 |
| 8.3 | 使用“表单设计器”创建表单 | 209 |
| 8.3.1 | “表单设计器”的启动 | 209 |
| 8.3.2 | 表单设计工具 | 210 |
| 8.3.3 | 表单生成器 | 212 |
| 8.4 | 数据环境与数据绑定 | 214 |
| 8.4.1 | 数据环境 | 214 |
| 8.4.2 | 数据绑定 | 215 |
| 8.4.3 | 向表单添加字段 | 216 |
| 8.5 | 常用表单控件 | 216 |
| 8.5.1 | 标签、线条、形状与图像 | 216 |
| 8.5.2 | 文本框与编辑框 | 217 |
| 8.5.3 | 列表框与组合框 | 219 |
| 8.5.4 | 命令按钮与命令按钮组 | 222 |
| 8.5.5 | 复选框与选项按钮组 | 224 |
| 8.5.6 | 微调控件 | 226 |
| 8.5.7 | 计时器 | 227 |
| 8.5.8 | 表格 | 229 |
| 8.5.9 | 页框与容器 | 230 |
| 8.5.10 | ActiveX 控件与 ActiveX 绑定控件 | 232 |
| 8.6 | 表单设计综合举例 | 235 |
| 8.6.1 | 通用类表单 | 235 |
| 8.6.2 | 计算类表单 | 236 |
| 8.6.3 | 表维护类表单 | 238 |
| 习题八 | | 241 |
| 第 9 章 | 报表与菜单设计 | 245 |
| 9.1 | 报表设计基础 | 245 |
| 9.1.1 | 报表设计概述 | 245 |

| | | |
|---------------|-----------------------|------------|
| 9.1.2 | 使用向导创建报表 | 247 |
| 9.1.3 | 创建快速报表 | 251 |
| 9.2 | 报表设计器 | 252 |
| 9.2.1 | “报表设计器”窗口 | 252 |
| 9.2.2 | 报表设计工具 | 254 |
| 9.2.3 | 在报表设计中使用控件 | 255 |
| 9.3 | 报表设计举例 | 259 |
| 9.3.1 | 设计带表格线的报表 | 259 |
| 9.3.2 | 设计档案卡片 | 261 |
| 9.3.3 | 设计分组报表 | 261 |
| 9.3.4 | 设计标签 | 264 |
| 9.4 | 菜单设计 | 266 |
| 9.4.1 | 菜单设计基本步骤 | 266 |
| 9.4.2 | 快速建立一个下拉菜单 | 268 |
| 9.4.3 | “菜单设计器”概述 | 268 |
| 9.4.4 | 创建一个应用程序的菜单 | 271 |
| 9.4.5 | 创建快捷菜单 | 274 |
| | 习题九 | 275 |
| 第 10 章 | 应用程序开发实例 | 278 |
| 10.1 | 应用程序开发步骤 | 278 |
| 10.1.1 | 需求分析 | 278 |
| 10.1.2 | 数据库设计 | 278 |
| 10.1.3 | 系统总体构架设计 | 279 |
| 10.1.4 | 创建各程序模块 | 280 |
| 10.1.5 | 主文件设计 | 281 |
| 10.1.6 | 测试、连编与发行 | 281 |
| 10.2 | 项目与“项目管理器” | 283 |
| 10.2.1 | “项目管理器”的启动 | 283 |
| 10.2.2 | “项目管理器”的使用 | 285 |
| 10.3 | 数据库设计基础 | 286 |
| 10.3.1 | 数据库设计原则 | 286 |
| 10.3.2 | 数据库设计过程 | 287 |
| 10.4 | 应用程序开发实例 | 290 |
| 10.4.1 | 系统设计 | 290 |
| 10.4.2 | 数据库设计 | 291 |
| 10.4.3 | 创建新类 | 293 |
| 10.4.4 | 主界面设计 | 295 |
| 10.4.5 | 查询表单设计 | 297 |
| 10.4.6 | 维护模块设计 | 299 |

| | | |
|---------|----------------------------------|-----|
| 10.4.7 | 统计与报表模块设计 | 303 |
| 10.4.8 | 系统主菜单设计 | 303 |
| 10.4.9 | 创建主程序 | 304 |
| 10.4.10 | 连编与运行 | 304 |
| 附录 1 | Visual Foxpro 常用命令表 | 307 |
| 附录 2 | Visual Foxpro 常用函数表 | 313 |

第 1 章 数据库系统概论

现代社会是一个信息化的社会，而信息的载体则是各种各样的数据。计算机是现代化的信息处理工具，基于计算机的数据库技术能有效地存储和组织大量的数据，而基于数据库技术的计算机系统则被称为数据库系统。目前，各种数据库系统不仅已成为办公自动化系统（OAS）、管理信息系统（MIS）和决策支持系统（DSS）的核心，并且已经和正在与计算机网络技术紧密地结合起来，成为电子商务、电子政务及其他各种现代信息处理系统的核心，得到了越来越广泛的应用。

本章将介绍信息与数据的概念、计算机数据管理的发展、数据库的基本概念及其主要数据结构模型、数据库管理系统以及数据库系统的概念等。在最后一节将介绍数据库管理系统软件 Visual FoxPro 的一些基本知识。

1.1 数据库系统的基本概念

1.1.1 信息、数据与数据库

信息是客观世界在人们头脑中的反映，是客观事物的表征，是可以传播和加以利用的一种知识。而数据是信息的载体，是对客观存在实体的一种记载和描述。也就是说，对信息的记载和描述产生了数据；反之，对众多相关的数据加以分析和处理又将产生新的信息。目前，各国学者对“信息”一词的概念仍众说纷纭，然而对于信息重要性的认识是完全一致的，即：信息如同材料和能源一样，是人类社会最重要的资源之一。

计算机能够处理的信息必须是数字化的，从这个意义上讲，数据又可以看作是数字化的信息。在这里，数据的概念已在通常意义的基础上大大地拓展了，数据不但包括数字、文字，还包括图形、图像、声音和视频等各种数字化的信息。

事实上，信息与数据的概念是密切相关的，在某些场合甚至是可以不加区分的。信息处理常常又被称为数据处理，包括数据的收集、存储、传输、加工、排序、检索和维护等一系列的活动。此外，信息和数据是有价值的，其价值取决于它的准确性、可靠性、及时性与完整性。为了提高信息或数据的价值，就必须用科学的方法对其进行管理，这种科学的方法就是数据库技术。

数据库（Database）是指存储在计算机外部存储器上的、结构化的相关数据集合。为了便于数据的管理和检索，数据库中的大量数据必须按一定的逻辑结构加以存储，这就是数据“结构化”的概念。数据库中的数据具有较高的数据共享性、独立性和安全性，以及较低的数据冗余度，能有效地支持对数据进行的各种处理，并能保证数据的一致性和完整性。

1.1.2 数据管理技术的发展

自从计算机应用于数据处理领域以来，就面临着如何管理大量复杂数据的问题。时至今日，随着计算机软、硬件技术与数据管理手段的不断发展，计算机数据管理技术已经大致经历了 3 个发展阶段。

1. 人工管理阶段

人工管理阶段约在 20 世纪 50 年代中期以前，那时计算机刚诞生不久，主要用于科学与工程计算。当时的硬件系统尚没有大容量的存储设备，而就其软件而言，操作系统和数据管理软件尚未开发。因此，这一阶段人们在进行数据处理时，就需要在编制的程序中，对所处理的数据作专门的定义，并需要对数据的存取及输入、输出方式作具体的安排。程序与数据不具有独立性，同一组数据在不同的程序中不能被共享。因此，各应用程序之间存在大量的重复数据，被称为数据冗余。

2. 文件管理阶段

文件管理阶段约为 50 年代后期至 60 年代后期，由于计算机软、硬件技术的发展，大容量的存储设备逐渐地被投入使用，操作系统也已诞生，从而为数据管理技术的发展提供了物质条件和工具手段，计算机开始大量地运用于管理中的数据处理工作。在当时的操作系统中，通常包含着一种专门进行文件管理的软件，或称文件管理程序，它可将数据的集合按照一定的形式存放到计算机的外部存储器中形成数据文件，而不再需要人们去考虑这些数据的存储结构、存储位置以及输入输出方式等。用户只需运用简单的操作命令，即可通过文件管理程序实现对数据的存取、查询及修改等多项操作，操作系统则提供了应用程序与相应数据文件之间的接口。这样一来，同一个应用程序可以调用多个数据文件，而同一个数据文件即同一组数据也可以被多个应用程序所调用。从而提高了数据的应用效率，并使数据和程序之间有了一定的独立性。

然而，随着社会生活中数据量的急剧增加，文件管理程序的功能仍不能适应新的需要。因为在文件管理方式下，数据文件本身仍仅仅应用于一个或几个应用程序，数据的独立性较差、共享性较弱、冗余度较大。因而在一定程度上浪费了存储空间，并带来了数据修改工作的麻烦，也容易造成数据的不一致性。

3. 数据库管理阶段

60 年代中期之后，为了克服文件管理方式的不足，有关数据库的理论研究和具体应用得到了迅速的发展，进而出现了各种数据库管理系统。致使数据管理技术推进到了一个新的阶段。数据库管理方式是将大量的相关数据按照一定的逻辑结构组织起来，构成一个数据库 (Database)，然后借助于专门的数据库管理系统软件 (DataBase Management System) 对这些数据资源进行统一的、集中的管理。如此一来，不仅减少了数据的冗余度、节约了存储空间，而且充分地实现了数据的共享，并具有相当好的易维护性和易扩充性，极大地提高了程序运行和数据利用的效率。

从 60 年代后期开始到 80 年代初期是数据库管理技术的发展时期，此后便进入了成熟期。多年来，数据库技术已成为计算机科学最重要的分支之一，并得到了惊人的发展，成

为了各种管理信息系统和决策支持系统的核心。

1.1.3 数据模型

在考察数据模型之前，先来考察一下实体及实体之间的联系。

1. 实体及实体间的联系

如前所述，数据是对客观存在事物的一种记载和描述，一般将客观存在的并且可以相互区分的事物称为实体。实体可以是实际的事物，例如一个学生、一本书等；也可以是象的事件，例如一个创意、一场比赛等。

实体的特性称之为属性。例如，关于学生实体可用学号、姓名、性别、出生日期等属性来描述；关于比赛实体可用比赛名、时间、地点、参赛者、举办方等属性来描述。

实体之间的关联称为联系，它反映了客观事物之间相互依存的状态。实体之间的联系可以归结为以下 3 种类型。

- 一对一联系：如果 A 公司只与 B 公司做生意，而 B 公司也只与 A 公司做生意，那么这两家公司实体之间就存在着一对一的联系。

- 一对多联系：如果一家总公司有多家子公司，那么总公司与子公司两个实体之间就存在着一对多的联系。一对多的联系是最普遍的联系，也可以将一对一的联系看作是一对多联系的特殊情况。

- 多对多联系：如果一家公司经营多种业务，而一种业务可以被多家公司所经营，那么这家公司与这种业务两个实体之间就存在着多对多的联系。多对多联系比较复杂，在实际应用中，可以将多对多联系分解为几个一对多联系来处理。

2. 数据模型

数据库中的数据是按一定的逻辑结构存放的，这种结构是用数据模型来表示的。任何一个数据库管理系统都是基于某种数据模型的。目前比较流行的数据模型主要有 3 种，即按图论理论建立起来的层次模型与网络模型，以及按关系理论建立起来的关系模型。

(1) 层次模型

在层次模型的数据集合中，各数据对象之间是一种依次的一对一的或一对多的联系。在这种模型中，层次清楚，可沿层次路径存取和访问各个数据。层次结构犹如一棵倒置的树，因而也称其为树型结构。如图 1-1 所示即为层次模型数据集合的一个例子。

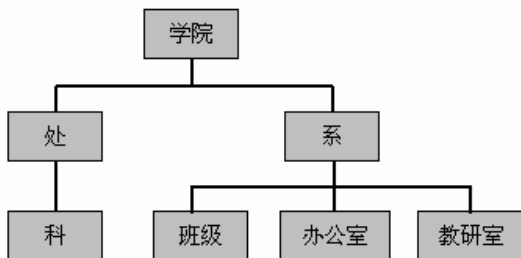


图 1-1 层次模型举例

层次模型的特点如下。

- 有且仅有一个根结点，其层次最高。
- 一个父结点向下可以有若干个子结点，而一个子结点向上只有一个父结点。
- 同层次的结点之间没有联系。

层次模型的优点是结构简单、层次清晰，并且易于实现。适宜描述类似于行政编制、家族关系及书目章节等信息载体的数据结构。但用层次模型不能直接表示多对多的联系，因而难以实现对复杂数据关系的描述。

(2) 网状模型

在网状模型中，各数据实体之间建立的往往是一种层次不清楚的一对一、一对多、或多对多的联系，此种结构可用来表示数据间复杂的逻辑关系。如图 1-2 所示的是一个网状数据模型的例子。

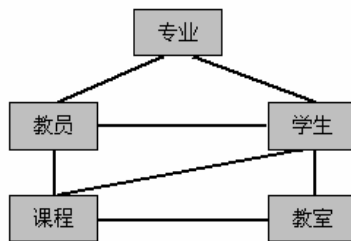


图 1-2 网状模型举例

网状模型的特点如下。

- 一个结点可以有多个父结点。
- 可以有一个以上的结点无父结点。
- 两个结点之间可以有多个联系。

网状模型的主要优点是在表示数据之间的多对多联系时具有很大的灵活性，但这种灵活性是以数据结构的复杂化为代价的。

事实上，网状模型和层次模型在本质上是类似的，它们都是用结点表示实体，用连线表示实体之间的联系。在计算机中具体实现时，每一个结点都是一个存储的数据或记录，而用链接指针来实现数据或记录之间的联系。这种用指针将数据或记录联系在一起的方法，很难对整个数据集合进行修改和扩充。

(3) 关系模型

关系模型是一种易于理解并具有较强的数据描述能力的数据库模型。关系模型用二维表格表示实体及实体之间的联系，即用若干行与若干列构成的表格来描述数据集合以及它们之间的联系，每一个这样的表格被称为一个关系。如图 1-3 所示的职工档案表格就是一个典型的关系模型数据集合的例子。

对于一个符合关系模型的二维表格，通常将其中的每一列称为一个字段 (Field)；而将其中的每一行称为一个记录 (Record)。一张二维表格若能构成一个关系模型的数据集合，必须满足以下条件。

- 表中不允许有重复的字段名。
- 表中每一列中数据的类型必须相同。

- 表中不应有内容完全相同的数据行。
- 表中行的顺序或列的顺序的任意排列，应不影响表中各数据项间的关系。

| 编号 | 姓名 | 性别 | 出生日期 | 年龄 | 职称 | 基本工资 | 婚否 |
|---------|------|----|----------|----|-----|---------|----|
| 2000101 | 刘红梅 | 女 | 09/10/72 | 30 | 工程师 | 880.00 | F |
| 2000102 | 张立功 | 男 | 08/16/67 | 35 | 工程师 | 960.00 | T |
| 2000103 | 薛小妹 | 女 | 09/20/68 | 34 | 工程师 | 960.00 | T |
| 2000104 | 王刚 | 男 | 03/25/48 | 54 | 高工 | 1280.00 | T |
| 2000105 | 蒋大伟 | 男 | 10/14/56 | 46 | 工程师 | 960.00 | T |
| 2000106 | 李永远 | 男 | 09/15/52 | 50 | 高工 | 1280.00 | T |
| 2000107 | 马丽 | 女 | 11/25/78 | 24 | 助工 | 760.00 | F |
| 2000108 | 张小龙 | 男 | 06/12/66 | 36 | 工程师 | 880.00 | F |
| 2000109 | 欧阳敏慧 | 女 | 11/23/57 | 45 | 高工 | 1088.00 | T |
| 2000110 | 王水清 | 男 | 08/09/57 | 45 | 高工 | 1088.00 | T |

图 1-3 关系模型举例

按关系模型建立的数据库称为关系型数据库 (Relational Database)，关系型数据库与层次型、网络型数据库的主要区别在于它描述数据的一致性。它把每个数据子集都分别按同一方法描述为一个关系，并且不像后两者那样事先规定子集之间的先后顺序或从属、层次等关系，而是让子集之间彼此独立，然后在使用时，通过筛选、投影、关联等方法，使数据之间或子集之间按某种关系进行操作。因此，关系数据库的数据表示能力较强、易于理解、使用上也比较方便，而且容易实现通用的数据管理功能。

目前，关系型数据库以其完备的理论基础、简单的模型和使用的方便性等优点得到了最广泛的应用，实际使用中的数据库基本上都是关系型的。

1.1.4 数据库管理系统

数据库管理系统 DBMS (DataBase Management System) 是管理数据库的软件工具，是帮助用户创建、维护和使用数据库的软件系统。它建立在操作系统的基础之上，实现对数据库的统一管理和操纵，满足用户对数据库进行访问的各种需要。

一般说来，数据库管理系统应具有以下一些功能。

(1) 数据定义功能

数据库管理系统软件都具有专门的数据定义语言，用于描述数据库的结构。例如关系型数据库管理系统的标准语言 SQL，有 CREATE、ALTER、DROP 等命令分别用来创建、修改和删除关系数据库的二维表结构。此外，SQL 语言还具有定义数据库中数据之间的约束条件、定义二维数据表之间的关联等功能。

(2) 数据操作功能

数据库管理系统提供的数据库操作功能，可支持用户对数据库中的数据进行查询、追加、插入、删除、修改、更新、统计、排序等操作。不同的数据库管理系统实现数据库操作的方法和命令格式不尽相同，然而大多数的数据库管理系统都支持 SQL 语言，因而可通过相应的 SQL 命令来实现各种数据库操作功能。

(3) 控制和管理功能

数据库管理系统必须具有控制和管理功能，以保障数据资源的安全。通常的安全措施包括对数据的备份、恢复和转储等功能，对用户的身份检查和用户权限控制，以及在

多个用户同时操作数据库时进行并发控制等。数据库系统的规模越大,要求的这类功能也就越强。

(4) 数据字典

数据库管理系统通常提供数据字典功能,以便对数据库中数据的各种描述进行集中管理。数据字典中存放了系统中所有数据的定义和设置信息,例如字段的属性、字段间的规则和记录间的规则、数据表之间的联系等。用户可以利用数据字典功能,为数据表的字段设置默认值、创建表之间的永久关系等。

总之,数据库管理系统是用户和数据库之间的交互界面。用户只需通过它就能实现对数据库的各种操作与管理。在其控制之下,用户在对数据库进行操作时,可以不必关心数据的具体存储位置、存放方式以及命令代码执行的细节等问题,就能完成对各种相关数据的处理任务,而且可以保证这些数据的安全性、可靠性与一致性。

目前,在各种计算机软件中,数据库管理系统软件占有极为重要的位置。广泛运用的大型数据库管理系统软件有 ORACLE、SYBASE、DB2 等,而在 PC 机上广泛应用的则有 Access、Visual FoxPro 和 SQL-Server 等。

1.1.5 数据库系统

1. 数据库系统的组成

人们通常将引入了数据库技术的计算机系统称为数据库系统。利用它可以实现有组织地、动态地存储大量的相关数据,并提供数据处理和共享的便利手段,为用户提供数据访问和所需的数据查询服务。一个数据库系统通常由 5 个部分组成,包括相关的计算机硬件、数据库集合、数据库管理系统、相关软件和人员。

(1) 计算机硬件系统

任何一个计算机系统都需要有存储器、处理器和输入/输出设备等硬件平台,一个数据库系统更需要有足够容量的内存与外存来存储大量的数据,同时需要有足够快的处理器来处理这些数据,以便快速响应用户的数据处理和数据检索请求。对于网络数据库系统,还需要有网络通信设备的支持。

(2) 数据库集合

数据库(Database)是指存储在计算机外部存储器上的结构化的相关数据集合。数据库不仅包含数据本身,而且还包括数据之间的联系。数据库中的数据通常可以被多个用户和(或)多个应用程序所共享。在一个数据库系统中,常常可以根据实际应用的需要创建多个数据库。

(3) 数据库管理系统

数据库管理系统 DBMS 用来对数据库进行集中统一的管理,是帮助用户创建、维护和使用数据库的软件系统。数据库管理系统是整个数据库系统的核心。

(4) 相关软件

除了数据库管理系统软件之外,一个数据库系统还必须要有其他相关软件的支持。这些软件包括:操作系统、编译系统、应用软件开发工具等。对于大型的多用户数据库系统和网络数据库系统,则还需要多用户系统软件和网络系统软件的支持。