



普通高等学校“十二五”规划教材 ■■■

# 全国计算机等级考试教程 ——Visual FoxPro

邵洪成 邵 俊 主编

QUANGUO JISUANJI DENGJI KAOSHI JIAOCHENG  
——VISUAL FOXPRO

中国铁道出版社  
CHINA RAILWAY PUBLISHING HOUSE

普通高等学校“十二五”规划教材

基于计算思维的理工类“大学计算机系列课程”改革及教材建设成果

# 全国计算机等级考试教程

## ——Visual FoxPro

邵洪成 邵俊 主编

朱锦新 张宏 陈瑾 季鹏宇 副主编

**中国铁道出版社**  
CHINA RAILWAY PUBLISHING HOUSE

---

## 内 容 简 介

本书根据教育部高等学校计算机基础课程教学指导委员会提出的教学基本要求,以全国计算机等级考试二级 Visual FoxPro 数据库程序设计考试大纲(2013 年版)为依据编写。全书共分为 10 章,主要内容包括关系数据库系统概述、VFP 程序设计基础、表的建立与操作、数据库的创建与操作、关系数据库结构化查询语言 SQL、VFP 程序设计、表单设计、菜单设计、报表和标签设计、开发应用程序等,各章末均提供了练习题,并附有参考答案供读者自我检查。

本书各章节的内容安排合理、理论联系实际、阐述由浅入深,并配有《Visual FoxPro 程序设计实验与上机考试教程》一书,适合作为全国计算机等级考试二级 Visual FoxPro 数据库程序设计的教材,也可作为其他人员学习 Visual FoxPro 的教学和参考用书。

### 图书在版编目(CIP)数据

全国计算机等级考试教程. Visual FoxPro/邵洪成,  
邵俊主编. — 北京:中国铁道出版社,2014.2

普通高等学校“十二五”规划教材 基于计算思维的  
理工类“大学计算机系列课程”改革及教材建设成果

ISBN 978-7-113-17930-4

I. ①全… II. ①邵…②邵… III. ①电子计算机—  
水平考试—教材②关系数据库系统—程序设计—水平考试  
—教材 IV. ①TP3

中国版本图书馆 CIP 数据核字(2014)第 008397 号

书 名:全国计算机等级考试教程——Visual FoxPro

作 者:邵洪成 邵 俊 主编

策 划:张围伟

读者热线:400-668-0820

责任编辑:何红艳 何 佳

封面设计:刘 颖

封面制作:白 雪

责任校对:汤淑梅

责任印制:李 佳

出版发行:中国铁道出版社(100054,北京市西城区右安门西街8号)

网 址: <http://www.51eds.com>

印 刷:北京鑫正大印刷有限公司

版 次:2014 年 2 月第 1 版 2014 年 2 月第 1 次印刷

开 本:787mm×1092mm 1/16 印张:14.5 字数:350 千

书 号:ISBN 978-7-113-17930-4

定 价:31.00 元

版权所有 侵权必究

凡购买铁道版图书,如有印制质量问题,请与本社教材图书营销部联系调换。电话:(010) 63550836

打击盗版举报电话:(010) 51873659

密集型的数据处理技术是目前计算机应用中最为广泛的领域，它依赖于数据库技术组成数据处理系统，对数据资源进行统一管理，使数据能为各类用户与应用程序共享。在现代计算机系统中，已将数据库管理系统作为主要的系统软件之一，微软公司推出的 Visual FoxPro 关系型数据库管理系统是目前比较流行的数据库管理系统之一，它采用面向对象的程序设计思想，可视化的操作方法，易学易用。

本书涵盖了《全国计算机等级考试二级 Visual FoxPro》教学与考试大纲规定的全部内容，注重基础、注重应用、实例丰富、图文并茂、循序渐进、通俗易懂、符合教学规律，同时也方便读者自学。

本书共 10 章，按照由浅入深、循序渐进的方式，全面而详细地介绍了 Visual FoxPro 6.0 中文版的功能，包括关系数据库系统概述、VFP 程序设计基础、表的建立与操作、数据库的创建与操作、关系数据库结构化查询语言 SQL、VFP 程序设计、表单设计、菜单设计、报表和标签设计、开发应用程序等，本书建议教学课时为 72，上机课时为 36。

读者可以到盐城工学院计算中心网站 (<http://jszx.ycit.cn>) 上下载相关素材，也可以直接与编者联系（编者 E-mail: shc@ycit.cn）。

本书由邵洪成、邵俊任主编，朱锦新、张宏、陈瑾和季鹏宇任副主编。本书的编者长期从事“Visual FoxPro 程序设计”等课程的教学研究工作和指导学生进行计算机等级考试强化训练等教学工作，有着丰富的教学经验。本书在编写过程中得到了我校教务处、信息工程学院的领导及同行教师的关心和支持，在此一并表示衷心的感谢！

本书虽经过多次讨论和反复修改，但由于编者的水平有限，书中难免有不当之处，敬请读者批评指正。

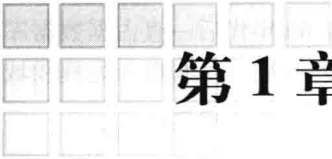
编 者

2013 年 12 月

|                                 |    |                               |    |
|---------------------------------|----|-------------------------------|----|
| 第 1 章 关系数据库系统概述 .....           | 1  | 2.3.5 逻辑运算符与逻辑<br>表达式 .....   | 36 |
| 1.1 数据库系统的发展与特点 .....           | 1  | 2.3.6 运算符优先级 .....            | 37 |
| 1.1.1 数据库系统的发展 .....            | 1  | 2.4 常用函数 .....                | 37 |
| 1.1.2 数据库系统中的概念 .....           | 2  | 2.4.1 数学函数 .....              | 37 |
| 1.1.3 数据库管理系统中的<br>数据模型 .....   | 5  | 2.4.2 字符处理函数 .....            | 39 |
| 1.2 关系数据库系统 .....               | 9  | 2.4.3 日期和时间函数 .....           | 42 |
| 1.2.1 关系模型的基本概念 .....           | 9  | 2.4.4 数据类型转换函数 .....          | 43 |
| 1.2.2 关系的完整性 .....              | 10 | 2.4.5 测试函数 .....              | 44 |
| 1.2.3 关系运算 .....                | 11 | 2.4.6 表操作函数 .....             | 46 |
| 1.3 关系型数据库的范式理论 .....           | 15 | 2.4.7 其他函数 .....              | 47 |
| 1.4 Visual FoxPro 系统概述 .....    | 16 | 小结 .....                      | 48 |
| 1.4.1 VFP 6.0 的发展及特点 .....      | 16 | 思考与练习 .....                   | 48 |
| 1.4.2 VFP 6.0 启动与主界面 .....      | 16 | 第 3 章 表的建立与操作 .....           | 50 |
| 小结 .....                        | 18 | 3.1 创建表 .....                 | 50 |
| 思考与练习 .....                     | 18 | 3.1.1 表结构的定义 .....            | 51 |
| 第 2 章 VFP 程序设计基础 .....          | 25 | 3.1.2 表结构的修改 .....            | 56 |
| 2.1 数据类型 .....                  | 25 | 3.1.3 表的打开、关闭 .....           | 56 |
| 2.1.1 基本数据类型 .....              | 25 | 3.1.4 表的复制 .....              | 57 |
| 2.1.2 字段数据类型 .....              | 26 | 3.1.5 文件操作的其他命令 .....         | 58 |
| 2.2 常量和变量 .....                 | 27 | 3.2 表的基本操作 .....              | 59 |
| 2.2.1 常量 .....                  | 27 | 3.2.1 记录的显示 .....             | 59 |
| 2.2.2 变量 .....                  | 29 | 3.2.2 记录的定位 .....             | 61 |
| 2.2.3 内存变量常用命令 .....            | 30 | 3.2.3 记录的添加 .....             | 62 |
| 2.3 运算符和表达式 .....               | 33 | 3.2.4 记录的修改 .....             | 63 |
| 2.3.1 算术运算符与算术<br>表达式 .....     | 33 | 3.2.5 记录的删除 .....             | 63 |
| 2.3.2 字符运算符与字符<br>表达式 .....     | 34 | 3.2.6 内存变量与表之间的数据<br>传递 ..... | 65 |
| 2.3.3 日期时间运算符与日期<br>时间表达式 ..... | 34 | 3.3 表的高级操作 .....              | 67 |
| 2.3.4 关系运算符与关系<br>表达式 .....     | 35 | 3.3.1 表的排序 .....              | 67 |
|                                 |    | 3.3.2 表的索引 .....              | 68 |
|                                 |    | 3.3.3 查询与统计 .....             | 71 |
|                                 |    | 3.4 多表的同时使用 .....             | 74 |

|                                          |           |                             |            |
|------------------------------------------|-----------|-----------------------------|------------|
| 3.4.1 表和工作区 .....                        | 74        | 5.4 SQL 数据操作 .....          | 107        |
| 3.4.2 表之间的临时关联 .....                     | 75        | 5.5 SQL 表定义 .....           | 107        |
| 小结 .....                                 | 76        | 小结 .....                    | 108        |
| 思考与练习 .....                              | 76        | 思考与练习 .....                 | 108        |
| <b>第 4 章 数据库的创建与操作 .....</b>             | <b>78</b> | <b>第 6 章 VFP 程序设计 .....</b> | <b>112</b> |
| 4.1 数据库的建立与操作 .....                      | 78        | 6.1 程序与程序文件 .....           | 112        |
| 4.1.1 数据库的建立、打开与<br>关闭 .....             | 78        | 6.1.1 程序的基本概念 .....         | 112        |
| 4.1.2 数据库的修改与删除 .....                    | 80        | 6.1.2 程序的建立、修改与<br>执行 ..... | 113        |
| 4.2 数据库表 .....                           | 80        | 6.1.3 常用的交互命令 .....         | 114        |
| 4.2.1 数据库中建表 .....                       | 80        | 6.2 程序的基本结构 .....           | 115        |
| 4.2.2 数据库中添加自由表 .....                    | 81        | 6.2.1 顺序结构 .....            | 115        |
| 4.3 数据库表的设置 .....                        | 83        | 6.2.2 选择结构 .....            | 116        |
| 4.3.1 “字段”选项卡的<br>设置 .....               | 83        | 6.2.3 循环结构 .....            | 118        |
| 4.3.2 “表”选项卡的设置 .....                    | 84        | 6.3 多模块程序 .....             | 121        |
| 4.4 数据库表之间的永久关系 .....                    | 85        | 6.3.1 模块的定义和调用 .....        | 121        |
| 4.4.1 建立表间关系 .....                       | 85        | 6.3.2 变量的作用域 .....          | 124        |
| 4.4.2 设置参照完整性 .....                      | 87        | 6.3.3 参数传递 .....            | 125        |
| 小结 .....                                 | 88        | 6.4 程序调试 .....              | 127        |
| 思考与练习 .....                              | 88        | 6.4.1 调试器 .....             | 127        |
| <b>第 5 章 关系数据库结构化<br/>查询语言 SQL .....</b> | <b>91</b> | 6.4.2 设置断点 .....            | 128        |
| 5.1 SQL 概述 .....                         | 91        | 小结 .....                    | 129        |
| 5.2 SQL 查询 .....                         | 92        | 思考与练习 .....                 | 129        |
| 5.2.1 简单查询 .....                         | 93        | <b>第 7 章 表单设计 .....</b>     | <b>133</b> |
| 5.2.2 排序查询 .....                         | 94        | 7.1 面向对象的程序设计 .....         | 133        |
| 5.2.3 计算查询 .....                         | 94        | 7.1.1 对象与属性 .....           | 133        |
| 5.2.4 嵌套查询 .....                         | 94        | 7.1.2 事件与方法 .....           | 134        |
| 5.2.5 分组查询 .....                         | 95        | 7.2 用表单向导设计表单 .....         | 135        |
| 5.2.6 连接查询 .....                         | 95        | 7.2.1 建立单表表单 .....          | 135        |
| 5.2.7 量词和谓词查询 .....                      | 96        | 7.2.2 建立多表表单 .....          | 137        |
| 5.2.8 其他查询 .....                         | 96        | 7.3 表单设计器 .....             | 139        |
| 5.2.9 查询去向 .....                         | 97        | 7.3.1 表单设计器环境 .....         | 139        |
| 5.3 查询与视图 .....                          | 98        | 7.3.2 控件的操作与布局 .....        | 142        |
| 5.3.1 查询和查询设计器 .....                     | 98        | 7.3.3 数据环境 .....            | 143        |
| 5.3.2 视图和视图设计器 .....                     | 103       | 7.3.4 设置属性与编辑代码 .....       | 145        |
| 5.3.3 建立本地视图 .....                       | 104       | 7.3.5 表单的修改与运行 .....        | 146        |
| 5.3.4 建立远程视图 .....                       | 106       | 7.4 常用表单控件 .....            | 146        |
|                                          |           | 7.4.1 标签控件 .....            | 146        |
|                                          |           | 7.4.2 命令按钮控件 .....          | 147        |

|                             |            |                            |            |
|-----------------------------|------------|----------------------------|------------|
| 7.4.3 命令按钮组控件 .....         | 148        | 8.4.1 建立一个工具栏类 .....       | 194        |
| 7.4.4 文本框控件 .....           | 150        | 8.4.2 为表单添加工具栏 .....       | 194        |
| 7.4.5 编辑框控件 .....           | 151        | 小结 .....                   | 194        |
| 7.4.6 复选框控件 .....           | 153        | 思考与练习 .....                | 194        |
| 7.4.7 选项按钮组控件 .....         | 155        | <b>第 9 章 报表和标签设计</b> ..... | <b>196</b> |
| 7.4.8 列表框控件 .....           | 157        | 9.1 报表的创建 .....            | 196        |
| 7.4.9 组合框控件 .....           | 159        | 9.1.1 报表的总体规划和<br>布局 ..... | 196        |
| 7.4.10 表格控件 .....           | 160        | 9.1.2 用报表向导创建报表 .....      | 197        |
| 7.4.11 页框控件 .....           | 162        | 9.1.3 用快速报表创建报表 .....      | 198        |
| 7.4.12 计时器控件与<br>微调控件 ..... | 164        | 9.1.4 用报表设计器创建<br>报表 ..... | 199        |
| 7.4.13 图像、形状、<br>线条控件 ..... | 166        | 9.1.5 用命令打印或预览<br>报表 ..... | 205        |
| 7.4.14 容器控件 .....           | 169        | 9.1.6 编程打印输出报表 .....       | 206        |
| 7.5 自定义属性与方法 .....          | 170        | 9.2 标签的创建 .....            | 206        |
| 7.5.1 自定义属性 .....           | 170        | 9.2.1 用标签向导创建标签 .....      | 207        |
| 7.5.2 自定义方法 .....           | 172        | 9.2.2 用标签设计器创建<br>标签 ..... | 207        |
| 7.6 表单集与多重表单 .....          | 175        | 9.3 报表和标签布局 .....          | 208        |
| 7.6.1 表单集 .....             | 175        | 小结 .....                   | 208        |
| 7.6.2 多重表单 .....            | 178        | 思考与练习 .....                | 208        |
| 7.7 对象数组 .....              | 181        | <b>第 10 章 开发应用程序</b> ..... | <b>209</b> |
| 7.7.1 VFP 基类 .....          | 181        | 10.1 VFP 数据库设计过程 .....     | 209        |
| 7.7.2 运行时创建对象 .....         | 182        | 10.2 VFP 应用程序设计过程 .....    | 209        |
| 小结 .....                    | 183        | 10.3 开发教学管理系统 .....        | 210        |
| 思考与练习 .....                 | 184        | 10.3.1 系统总体规划 .....        | 210        |
| <b>第 8 章 菜单设计</b> .....     | <b>185</b> | 10.3.2 数据库设计 .....         | 210        |
| 8.1 VFP 菜单概述 .....          | 185        | 10.3.3 功能模块设计 .....        | 211        |
| 8.2 VFP 菜单规划 .....          | 185        | 10.3.4 连编应用程序 .....        | 215        |
| 8.3 建立菜单 .....              | 186        | <b>附录 A 表结构与记录</b> .....   | <b>217</b> |
| 8.3.1 菜单设计器窗口 .....         | 186        | <b>附录 B 部分习题参考答案</b> ..... | <b>222</b> |
| 8.3.2 建立下拉式菜单 .....         | 190        | <b>参考文献</b> .....          | <b>224</b> |
| 8.3.3 建立快捷菜单 .....          | 192        |                            |            |
| 8.3.4 为顶层表单添加菜单 .....       | 193        |                            |            |
| 8.4 工具栏的使用 .....            | 194        |                            |            |



# 第1章 关系数据库系统概述

21 世纪是一个高度信息化的时代, 计算机科学的发展使得它的应用领域逐步扩大, 已进入社会的各个方面。信息化包括三项技术: 计算机技术、通信技术和控制技术, 而计算机技术是信息化的主要处理工具。信息化社会用计算机进行数据处理已成为日常工作的内容。数据库系统技术是用计算机进行数据处理的最简单、易学、易掌握的技术。Visual FoxPro (以下简称 VFP) 数据库继承了以往传统数据库的优点, 是采用面向对象的程序设计思想和可视化的编程工具, 它操作简单、易学, 极大地方便了软件的开发。人们在使用它作为开发工具时, 可以高效地开发出全新的可视化图形界面应用程序, 所开发出的软件易于推广和应用, 因此 VFP 应用较为广泛。本章主要介绍 VFP 的基本知识, 为后续章节的学习打下基础。

## 1.1 数据库系统的发展与特点

### 1.1.1 数据库系统的发展

数据处理技术形成和发展大致经历了的人工管理、文件系统、数据库系统、分布式数据库管理四个阶段。人工管理是最原始的数据管理方式, 由于数据的不共享性, 而被文件系统所代替。文件系统中的数据虽然具有共享性, 但共享性差、独立性差、冗余度高而被数据库系统所代替。由于数据库系统具有共享性高、冗余度低、数据的独立性好的优点使它成为当今数据处理的主要工具。分布式数据库系统更适合分布式的管理与控制, 具有灵活的体系结构、系统经济、可靠性高、可用性好、在一定条件下响应速度快、可扩展性好、易于集成现有系统、也易于扩充。在 20 世纪 60 年代末美国 IBM 公司成功研制出第一个商品化的数据库系统 IMS (Information Management System), 以后又相继出现了 DBASE、FoxBASE、FoxPro、VFP 等数据库系统。社会科技的进步促进了数据库技术的发展, 与此同时数据库技术的发展又推动了计算机应用领域的扩大和深入。

#### 1. 文件系统阶段

文件系统是数据库系统发展的初级阶段; 它提供了简单的数据共享与数据管理能力, 但是它无法提供完整的、统一的、管理和数据共享的能力。由于它的功能简单, 因此它附属于操作系统而不成为独立的软件, 目前一般仅将其看成是数据库系统的雏形, 而不是真正的数据库系统。

#### 2. 层次数据库与网状数据库系统阶段

从 20 世纪 60 年代末期起, 真正的数据库系统——层次数据库与网状数据库开始发展, 它们为统一管理与共享数据提供了有力支撑, 这个时期数据库系统蓬勃发展形成了有名的“数据库时代”。但是这两种系统也存在不足, 主要是它们脱胎于文件系统, 受文件的物理影响较大, 对

数据库使用带来诸多不便，同时，此类系统的数据模式构造烦琐，不易于推广使用。

### 3. 关系数据库系统阶段

关系数据库系统出现于 20 世纪 70 年代，在 80 年代得到蓬勃发展，并逐渐取代前两种系统。关系数据库系统结构简单，使用方便，逻辑性强，物理性少，因此在 80 年代后一直占据数据库领域的主导地位。但是由于此系统来源于商业应用，适合于事务处理领域而对非事务处理领域应用受到限制，因此在 80 年代末期兴起与应用技术相结合的各种专用数据库系统。

### 4. 分布式数据库管理阶段

在数据库管理阶段之后，随着网络技术的产生和发展，出现了分布式数据库系统 (Distributed DataBase System, DDBS)。分布式数据库系统是地理上分布在计算机网络的不同结点，逻辑上属于同一系统的数据库系统，它不同于将数据存储在服务器上供用户共享存取的网络数据库系统，分布式数据库系统不仅能支持局部应用，存取本地结点或另一结点的数据，而且能支持全局应用，同时存取两个或两个以上结点的数据。分布式数据库系统的主要特点是：

① 数据是分布的。数据库中的数据分布在计算机网络的不同结点上，而不是集中在一个结点，区别于数据存放在服务器上由各用户共享的网络数据库系统。

② 数据是逻辑相关的。分布在不同结点的数据逻辑上属于同一数据库系统，数据间存在相互关联，区别于由计算机网络连接的多个独立数据库系统。

③ 结点的自治性。每个结点都有自己的计算机软硬件资源、数据库、数据库管理系统（即局部数据库管理系统，Local DataBase Management System, LDBMS），因而能够独立地管理局部数据库。局部数据库中的数据可以仅供本结点用户存取使用，也可供其他结点上的用户存取使用，提供全局应用。

## 1.1.2 数据库系统中的概念

### 1. 数据

数据 (Data) 实际上就是描述事物的符号记录。计算机中的数据一般分为两部分，其中一部分与程序仅有短时间的交互关系，随着程序的结束而消亡，它们称为临时性数据，这类数据一般存放于计算机内存中；而另一部分数据则对系统起着长期持久的作用，它们称为持久性数据。数据库系统中处理的就是这种持久性数据。

### 2. 数据库

数据库 (DataBase, DB) 是数据的集合，它具有统一的结构形式，并存放于统一的存储介质内，是多种应用数据的集成，并可被各个应用程序所共享。

数据库中的数据具有“集成”“共享”的特点，亦即是数据库集中了各种应用的数据，进行统一的构造与存储，而使它们可被不同应用程序所使用。

### 3. 数据库管理系统

数据库管理系统 (DataBase Management System, DBMS) 是数据库的机构，它是一种系统软件，负责数据库中的数据组织、数据操纵、数据维护、控制及保护和数据服务等。数据库中的数据是具有海量级的数据，并且其结构复杂，因此需要提供管理工具。

#### (1) 数据库管理系统的主要功能

数据库管理系统是数据库系统的核心，它主要有如下几方面的具体功能：

① 数据模式定义。数据库管理系统负责为数据库构建模式，也就是为数据库构建其数据框架。

② 数据存取的物理构建。数据库管理系统负责为数据模式的物理存取及构建提供有效的存取方法与手段。

③ 数据操纵。数据库管理系统为用户使用数据库中的数据提供方便,它一般提供查询、插入、修改及删除数据的功能。此外,它自身还具有做简单算术运算及统计的能力,而且还可以与某些过程性语言结合,使其具有强大的过程性操作能力。

④ 数据的完整性、安全性定义与检查。数据库中的数据具有内在语义上的关联性与一致性,它们构成了数据的完整性,数据的完整性是保证数据库中数据正确的必要条件,因此必须经常检查以维护数据正确。

数据库中的数据具有共享性,而数据共享可能会引发数据的非法使用,为此应作出必要的规定,并在使用时做检查,这就是数据的安全性。

数据完整性与安全性的维护是数据库管理系统的基本功能。

⑤ 数据库的并发控制与故障恢复。数据库是一个集成、共享的数据集合体,它能为多个应用程序服务,所以就存在着多个应用程序对数据库的并发操作。在并发操作中如果不加控制和管理,多个应用程序间就会相互干扰,从而对数据库中的数据造成破坏。因此,数据库管理系统必须对多个应用程序的并发操作做必要的控制以保证数据不被破坏,这就是数据库的并发控制。

数据库中的数据一旦遭受破坏,数据库管理系统必须有能力及及时进行恢复,这就是数据库的故障恢复。

⑥ 数据的服务。数据库管理系统提供对数据库中数据的多种服务功能,如数据复制、转存、重组、性能监测、分析等。

## (2) 数据库管理系统的语言

为实现以上六个功能,数据库管理系统一般提供相应的数据语言(Data Language),它们是:

① 数据定义语言(Data Definition Language, DDL)。该语言负责数据的模式定义与数据的物理存取构建。

② 数据操纵语言(Data Manipulation Language, DML)。该语言负责数据的操纵,包括查询及增加、删除、修改等操作。

③ 数据控制语言(Data Control Language, DCL)。该语言负责数据完整性、安全性的定义与检查,以及并发控制、故障恢复等功能,包括系统初启程序、文件读/写与维护程序、存取路径管理程序、缓冲区管理程序、安全性控制程序、完整性检查程序、并发控制程序、事务管理程序、运行日志管理程序、数据库恢复程序等。

## (3) 数据库管理系统的语言结构形式

上述数据语言按其使用方式具有两种结构形式:

① 交互式命令语言。它的语言简单,能在终端上即时操作,它又称为自含型或自主型语言。

② 宿主型语言。它一般可嵌入某些宿主语言(Host Language)中,如C、C++和COBOL等高级过程性语言中。

目前流行的DBMS均为关系数据库系统,比如Oracle、Sybase的PowerBuilder及IBM的DB2、微软的SQL Server等,它们均为严格意义上的DBMS。另外有一些小型的数据库,如微软的Visual FoxPro和Access等,它们只具备数据库管理系统的一些简单功能。

## 4. 数据库管理员

由于数据库的共享性,因此对数据库的规划、设计、维护、监视等需要有专人管理,称他

们为数据库管理员 ( DataBase Administrator, DBA )。其主要工作如下:

① 数据库设计 ( DataBase Design )。DBA 的主要任务之一是进行数据库设计,具体地说是进行数据模式的设计。由于数据库的集成与共享性,因此需要有专门人员 (即 DBA) 对多个应用的数据需求进行全面的规划、设计与集成。

② 数据库维护。DBA 必须对数据库中数据的安全性、完整性、并发控制及系统恢复、数据定期转存等进行实施与维护。

③ 改善系统性能,提高系统效率。DBA 必须随时监视数据库运行状态,不断调整内部结构,使系统保持最佳状态与最高效率。当效率下降时,DBA 需采取适当的措施,如进行数据库的重组、重构等。

## 5. 数据库系统

数据库系统 ( DataBase System, DBS ) 是由硬件、软件、数据库和用户四部分构成的整体,如图 1.1 所示。

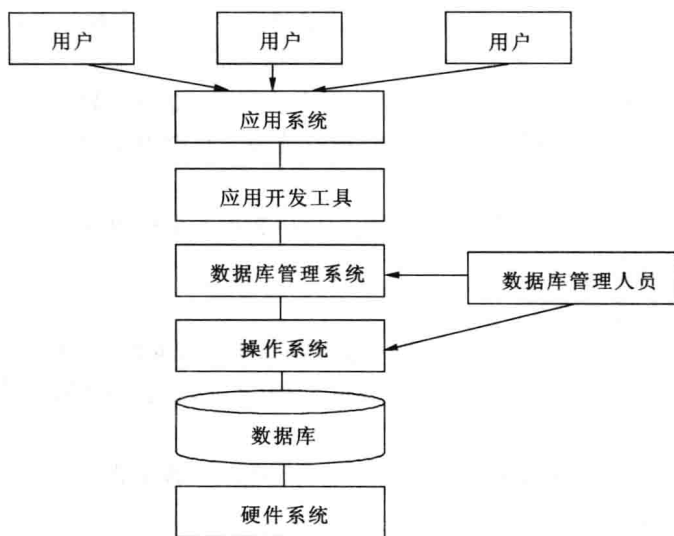


图 1.1 数据库系统

**数据库:** 是数据库系统的核心和管理对象,数据库是存储在一起的相互有联系的数据集合。数据库中的数据是集成的、共享的、最小冗余的、能为多种应用服务,数据是按照数据模型所提供的形式框架存放在数据库中。

**硬件:** 数据库系统建立在计算机系统上,运行数据库系统的计算机需要有足够大的内存以存放系统软件、需要足够大容量的磁盘等联机存取设备存储数据库庞大的数据。需要足够的脱机存储介质 (磁盘、光盘、磁带等) 以存放数据库备份。需要较高的通道能力,以提高数据传输速率。要求系统联网,以实现数据共享。

**软件:** 数据库软件主要指数据库管理系统 (DBMS)。

DBMS 是为数据库存取、维护和管理而配置的软件,它是数据库系统的核心组成部分,DBMS 在操作系统支持下工作。DBMS 主要包括数据库定义功能、数据操纵功能、数据库运行和控制功能、数据库建立和维护功能、数据通信功能。

**用户:** 数据库系统中存在一组管理 (数据库管理员 DBA)、开发 (应用程序员)、使用数

据库（终端用户）的用户。

## 6. 数据库应用系统

利用数据库系统进行应用开发可构成一个数据库应用系统（DataBase Application System, DBAS），数据库应用系统由数据库系统、应用软件及应用界面组成，具体包括：数据库、数据库管理系统、数据库管理员，硬件平台、软件平台、应用软件、应用界面。其中应用软件是由数据库系统所提供的数据库管理系统（软件）及数据库系统开发工具所书写而成，而应用界面大多由相关的可视化工具开发而成。

数据库应用系统的七个部分以一定的逻辑层次结构方式组成一个有机的整体。

### 1.1.3 数据库管理系统中的数据模型

数据是现实世界符号的抽象，而数据模型（data model）则是数据特征的抽象，它从抽象层次上描述了系统的静态特征、动态行为和约束条件，为数据库系统的信息表示与操作提供一个抽象的框架。

#### 1. 数据模型的内容

数据模型所描述的内容有三部分，分别是数据结构、数据操作与数据约束。

① 数据结构。数据模型中的数据结构主要描述数据的类型、内容、性质以及数据间的联系等。数据结构是数据模型的基础，数据操作与约束均建立在数据结构上。不同数据结构有不同的操作与约束，因此，一般数据模型均以数据结构的不同而分类。

② 数据操作。数据模型中的数据操作主要描述在相应数据结构上的操作类型与操作方式。

③ 数据约束。数据模型中的数据约束主要描述数据结构内数据间的语法、语义联系，它们之间的制约与依存关系，以及数据动态变化的规则，可保证数据的正确、有效与相容。

#### 2. 数据模型的类型

数据模型按不同的应用层次分成三种类型，它们是概念数据模型、逻辑数据模型、物理数据模型。

① 概念数据模型简称概念模型，它是一种面向客观世界、面向用户的模型，它与具体的数据库管理系统无关，与具体的计算机平台无关。概念模型着重于对客观世界复杂事物的结构描述及它们之间的内在联系的刻画。概念模型是整个数据模型的基础。目前，较为有名的概念模型有 E-R 模型、扩充的 E-R 模型、面向对象模型及谓词模型等。

② 逻辑数据模型又称数据模型，它是一种面向数据库系统的模型，该模型着重于在数据库系统一级的实现。概念模型只有在转换成数据模型后才能在数据库中得以表示。目前，逻辑数据模型也有很多种，较为成熟并先后被人们大量使用过的有：层次模型、网状模型、关系模型、面向对象模型等。

③ 物理数据模型又称物理模型，它是一种面向计算机物理表示的模型，此模型给出了数据模型在计算机上物理结构的表示。

#### 3. E-R 模型的基本概念

##### （1）实体

现实世界中的事物可以抽象成为实体，实体是概念世界中的基本单位，它们是客观存在的且又能相互区别的事物。凡是有共性的实体可组成一个集合，称为实体集。

##### （2）属性

现实世界中事物均有一些特性，这些特性可以用属性来表示。属性刻画了实体的特征。一个实

体往往可以有若干个属性。每个属性可以有值，一个属性的取值范围称为该属性的值域或值集。

### (3) 联系

现实世界中事物间的关联称为联系。在概念世界中联系反映了实体集间的操作关系，上、下级间的领导关系，生产者与消费者间的供求关系。

一对一 (one to one) 的联系，简记为  $1:1$ 。这种函数关系是常见的函数关系之一，如学校与校长间的联系，一个学校与一个校长间相互一一对应，如图 1.2 所示。

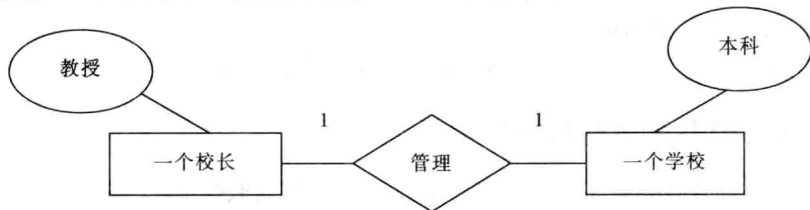


图 1.2 一对一联系

一对多 (one to many) 或多对一 (many to one) 联系，简记为  $1:M$  ( $1:m$ ) 或  $M:1$  ( $m:1$ )。这两种函数关系实际上是一种函数关系，如学生与其宿舍房间的联系是多对一的联系 (反之，则为一对多联系)，即多个学生对应一个房间，如图 1.3 所示。

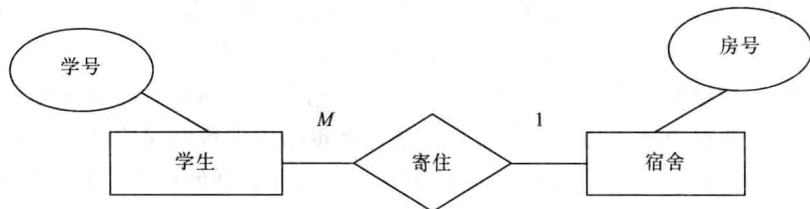


图 1.3 一对多或多对一联系

多对多 (many to many) 联系，简记为  $M:N$  或  $m:n$ 。这是一种较为复杂的函数关系，如教师与学生这两个实体集间的教与学的联系是多对多的，因为一个教师可以教授多个学生，而一个学生又可以受教于多个教师，如图 1.4 所示。

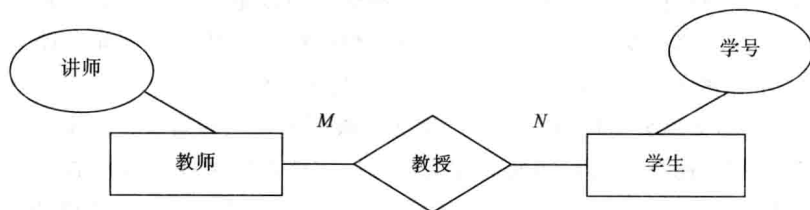


图 1.4 多对多联系

### (4) E-R 模型的图示法

E-R 模型可以用一种非常直观的图的形式表示，这种图称为 E-R 图 (Entity Relationship Diagram)。在 E-R 图中分别用下面不同的几何图形表示 E-R 模型中的三个概念与两个连接关系，如图 1.5 所示。

实体集表示法：在 E-R 图中用矩形表示实体集。

属性表示法：在 E-R 图中用椭圆形表示属性。

联系表示法：在 E-R 图中用菱形表示联系。

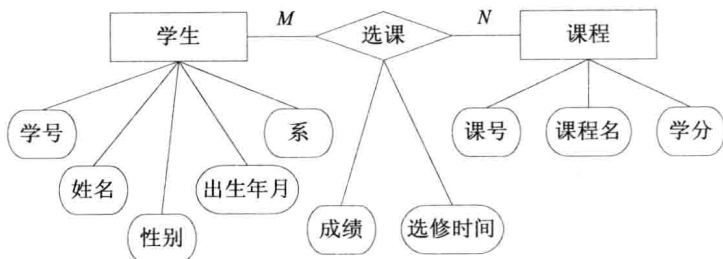


图 1.5 E-R 模型的图示法

#### 4. 层次模型

层次模型是最早发展起来的数据库模型。层次模型（Hierarchical Model）的基本结构是树形结构，这种结构方式在现实世界中很普遍，如家族结构、行政组织机构，它们自顶向下、层次分明。

层次模型的特点：

- ① 每棵树有且仅有一个无双亲结点，称为根。
- ② 树中除根外所有结点有且仅有一个双亲。

层次模型支持的操作主要有查询、插入、删除和更新。在对层次模型进行插入、删除、更新操作时，要满足层次模型的完整性约束条件：进行插入操作时，如果没有相应的双亲结点值就不能插入子女结点值；进行删除操作时，如果删除双亲结点值，则相应的子女结点值也被同时删除；进行更新操作时，应更新所有相应记录，以保证数据的一致性。

层次模型的数据结构比较简单，操作简单；对于实体间联系是固定的、且预先定义好的应用系统，层次模型有较高的性能；同时，层次模型还可以提供良好的完整性支持。但由于层次模型形成早，受文件系统影响大，模型受限制多，物理成分复杂，操作与使用均不甚理想，它不适合于表示非层次性的联系；对于插入和删除操作的限制比较多；此外，查询子女结点必须通过双亲结点，如图 1.6 所示。

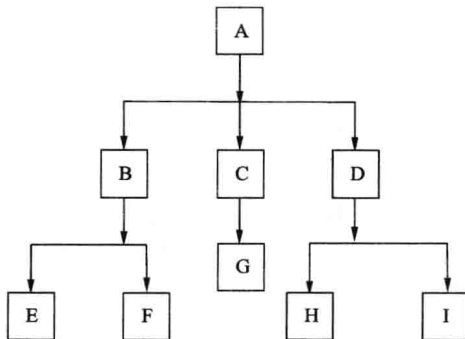


图 1.6 层次模型

#### 5. 网状模型

网状模型的出现略晚于层次模型。从图论观点看，网状模型是一个不加任何条件限制的无向图。网状模型在结构上较层次模型好，不像层次模型那样要满足严格的条件。

在实现中，网状模型将通用的网络拓扑结构分成一些基本结构。一般采用的分解方法是将一个网络分成若干个二级树，即只有两个层次的树。换句话说，这种树是由一个根及若干个叶所组成。为实现方便，一般规定根结点与任一叶结点间的联系均是一对多的联系（包含一对一

联系)。一般地,现实世界的一个实体结构往往可以由若干个系组成,如图 1.7 所示。

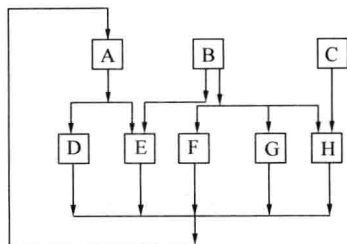


图 1.7 网状模型

## 6. 关系模型

关系模型采用二维表来表示,简称表。二维表由表框架及表的元组组成。表框架由  $n$  个命名的属性组成, $n$  称为属性元数。每个属性有一个取值范围,称为值域。表框架对应了关系的模式,即类型的概念。

在表框架中按行可以存放数据,每行数据称为元组,实际上,一个元组是由  $n$  个元组分量所组成,每个元组分量是表框架中每个属性的投影值。一个表框架可以存放  $m$  个元组, $n$  元表框架及框架内  $m$  个元组构成了一个完整的二维表。

- ① 二维表中元组个数是有限的——元组个数有限性。
- ② 二维表中元组均不相同——元组的唯一性。
- ③ 二维表中元组的次序可以任意交换——元组的次序无关性。
- ④ 二维表中元组的分量是不可分割的基本数据项——元组分量的原子性。
- ⑤ 二维表中属性名各不相同——属性名唯一性。
- ⑥ 二维表中属性与次序无关,可任意交换——属性的次序无关性。
- ⑦ 二维表属性的分量具有与该属性相同的值域——分量值域的同—性。

满足以上七个性质的二维表称为关系,以二维表为基本结构所建立的模型称为关系模型,关系模型中的一个重要概念是键(码),键具有标识元组、建立元组间联系等重要作用。

在二维表中凡能唯一标识元组的最小属性集称为该表的键(码)。

二维表中可能有若干个键,它们称为该表的候选码(候选键)。

从二维表的所有候选键中选取一个作为用户使用的键,称为主键或主码,简称键(码)。

在关系元组的分量中允许出现空值以表示信息的空缺。空值用于表示未知的值或不可能出现的值,一般用 NULL 表示。一般关系数据库系统都支持空值,但是有两个限制,即关系的主键中不允许出现空值,因为如主键为空值则失去了其元组标识的作用;需要定义有关空值的运算。

关系框架与关系元组构成了一个关系。一个语义相关的关系集合构成一个关系数据库。关系的框架称为关系模式,而语义相关的关系模式集合构成了关系数据库模式。

### (1) 关系操纵

关系模型的数据操纵即是建立在关系上的数据操纵,一般有查询、增加、删除及修改四种操作。

① 数据查询。用户可以查询关系数据库中的数据,它包括一个关系内的查询以及多个关系间的查询。

② 数据删除。

③ 数据插入。

④ 数据修改。

以上四种操作的对象都是关系，而操作结果也是关系，因此都是建立在关系上的操作。

## (2) 关系中的数据约束

关系模型允许定义三类数据约束，它们是实体完整性约束、参照完整性约束以及用户定义的完整性约束，其中前两种完整性约束由关系数据库系统自动支持。对于用户定义的完整性约束，则由关系数据库系统提供完整性约束语言，用户利用该语言写出约束条件，运行时由系统自动检查。

① 实体完整性约束。该约束要求关系的主键中属性值不能为空值，这是数据库完整性的基本要求，如为空值则其唯一性就不可能实现。

② 参照完整性约束。该约束是关系之间相关联的基本约束，它不允许关系引用不存在的元组：即在关系中的外键要么是所关联关系中实际存在的元组，要么就为空值。

③ 用户定义的完整性约束。这是针对具体数据环境与应用环境由用户具体设置的约束，它反映了具体应用中数据的语义要求。

## 1.2 关系数据库系统

关系数据库 (Relational Database)，是建立在关系数据库模型基础上的数据库，借助于集合代数等概念和方法来处理数据库中的数据。关系数据库是一个被组织成一组拥有正式描述性的表格，该形式的表格作用的实质是装载着数据项的特殊集体，这些表格中的数据能以许多不同的方式被存取或重新召集而不需要重新组织数据库表格。关系数据库是在 1970 年被 IBM 公司的 E.F.Codd 发明的。

标准用户和应用程序到一个关系数据库的接口是结构化查询语言 (SQL)。SQL 声明被用来交互式查询来自一个关系数据库的信息和为报告聚集数据。

除了相对地容易创建和存取之外，关系数据库还具有容易扩充的重要优势。在最初的数据库创造之后，一个新的数据种类能被添加而不需要修改所有的现有应用软件。

### 1.2.1 关系模型的基本概念

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（表）来表示。下面讲述关系模型的一些基本术语：

#### (1) 实体

客观存在并可以相互区别的事物称为实体。如：一个学生、一门课等。

#### (2) 属性

实体所具有的某一特性称为属性。如：一个学生实体可由学号、姓名、成绩等属性组成。

#### (3) 域

属性的取值范围称为域。如：学生的一门课成绩取值为 0~100 之间。

#### (4) 实体型

若干个属性名所组成的集合表示一个实体的类型称为实体型。

#### (5) 实体集

同型实体的集合称为实体集。如：全班学生实体就是一个实体集。

#### (6) 联系

实体联系有内部联系与外部联系。实体的内部联系是指实体内部各属性之间的联系；实体

的外部联系也称实体之间的联系，通常是指不同实体集之间的联系。实体集之间的联系可分为三类，即一对一联系、一对多（多对一）联系、多对多联系。

### 1.2.2 关系的完整性

关系的完整性包括实体完整性、参照完整性和用户定义完整性。

#### 1. 概念

##### (1) 关键字

若在关系中能唯一标识元组的一个或一组属性称为关键字。如学生成绩表中的学号。

##### (2) 候选关键字

候选关键字也是关键字，它是具有关键字特性的一个或多个属性的统称。若学生成绩表中学号、姓名都无重复值，那么学号与姓名就是候选关键字。若姓名有重复值就不能成为候选关键字。

##### (3) 主关键字

主关键字是在多个候选关键字中选出一个，一个关系中只能有一个主关键字。如在学生成绩表中学号为主关键字。

##### (4) 外部关键字

若在一个关系  $R$  中一个属性不是本关系的主关键字或候选关键字，而是另外一个关系  $S$  的主关键字或候选关键字，则称此属性为本关系的外部关键字， $R$  为参照关系， $S$  为被参照关系。

#### 2. 关系的完整性

##### (1) 实体完整性

实体完整性是指关系中的主关键字不能取空值，空值就是（NULL）不确定的值。由于主关键字为空值，那这个元组是无意义的。如学生成绩表中，学号为空值，而其他属性却有值，则该条记录无意义。

##### (2) 参照完整性

参照完整性是用来约束关系与关系之间的关系，即数据库约束数据库中表与表之间的关系。

参照完整性是指一个关系  $R$  的外部关键字  $F$  与另一个关系  $S$  的主关键字  $K$  相对应（即  $F$  是  $S$  的主关键字）则对  $R$  中的每一个元组在  $F$  上的值必须为空值或等于  $S$  中某一个元素的主关键字的值，如有以下两个关系（表）：学生（学号、姓名、性别、专业代号）和专业（专业代号、专业号），学生关系中的专业代号是外部关键字，但在专业关系中它是主关键字。这时，在学生关系中，专业代号的值或者取空值或者取专业关系中某个元组的专业代号的值；当在学生关系中的专业代号取空值时，表示该学生还没有分专业。在 VFP 数据库中的表实现参照关系时，应注意以下几点：

① 当对含有外部关键字的参照表进行创建、插入、修改时，必须检查外部关键字的值在被参照表中是否存在，若不存在，则不能进行该操作。

② 当对被参照的表进行删除、修改时必须检查被删除的行或修改的行的主关键字值是否正在被参照表的外部关键字参照，若是，则不能进行该操作。

##### (3) 用户定义完整性

用户定义完整性是指根据用户的实际需要，确保属性域完整性的规定。在 VFP 数据库表中是指字段的数据类型、宽度、精度、取值范围、是否允许空值。如学生成绩表中的数学成绩，其数据类型可为整型，取 0~100 之间等，取值范围又可分为静态与动态，静态是指字段的取值