



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

Visual FoxPro 数据库 及程序设计基础 (第2版)

Visual FoxPro Programing (2nd Edition)

周明红 主编

杨潞霞 孙咏梅 张季良 副主编

张永奎 相万让 审

- 按知识点讲解、实例说明、课后习题的模式组织内容
- 一个数据库管理系统贯穿全书，分解到各章实例中
- 覆盖全国计算机等级考试二级VFP的知识要点



高校系列



人民邮电出版社

POSTS & TELECOM PRESS



工业和信息化普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

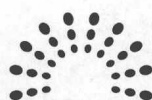
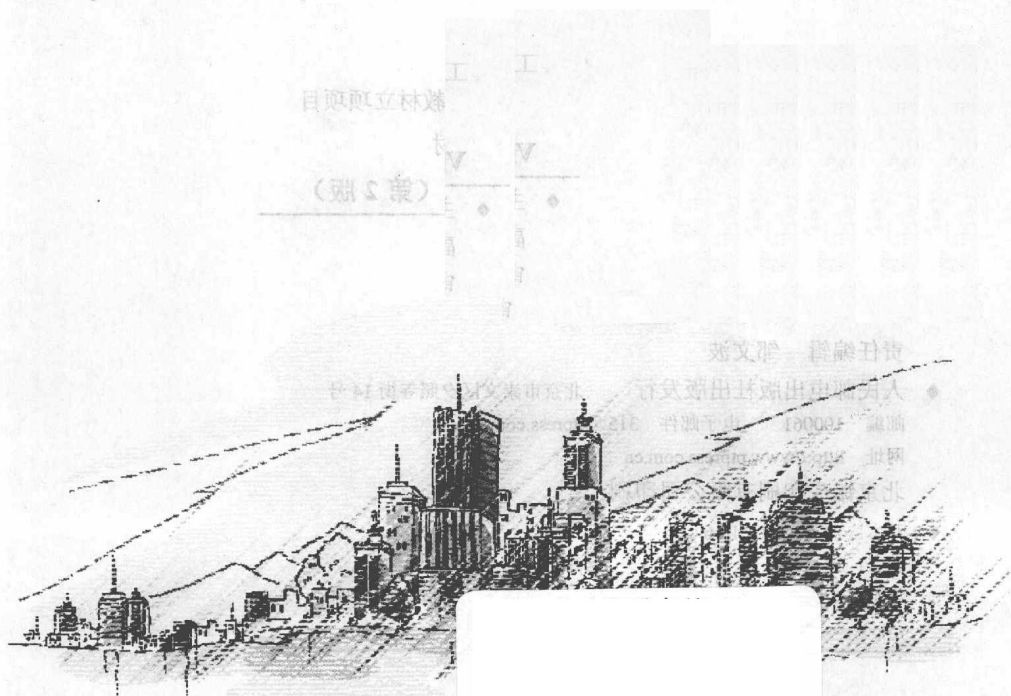
Visual FoxPro 数据库 及程序设计基础 (第2版)

Visual FoxPro Programing (2nd Edition)

周明红 主编

杨潞霞 孙咏梅 张季良 副主编

张永奎 相万让 审



高校系列

人民邮电出版社

北京

图书在版编目(CIP)数据

Visual FoxPro数据库及程序设计基础 / 周明红主编

— 2版. — 北京: 人民邮电出版社, 2011. 9

21世纪高等学校计算机规划教材. 高校系列

ISBN 978-7-115-25722-2

I. ①V… II. ①周… 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141201号

内 容 提 要

本书是针对普通高等院校非计算机专业的“Visual FoxPro 数据库及程序设计基础”课程的教材。全书紧扣全国计算机等级考试大纲, 主要内容包括数据库基础知识、Visual FoxPro 6.0 基础知识、Visual FoxPro 6.0 基本数据元素、表的创建及相关操作、数据库的创建和使用、视图与查询、SQL 标准语言、结构化程序设计、面向对象程序设计、表单的设计、报表和标签、菜单和工具栏, 以及应用程序的创建和发布。每章开始有简短的导读, 指出本章主要内容; 结尾配有相应的理论练习题及参考答案, 供学习者练习。教材最后还给出了几套综合练习题, 内容和方式类似于计算机等级考试题。全书贯穿一个数据库管理系统的设计, 图、文、例相结合, 通俗易懂, 特别是给出了大量的实例, 可以帮助学习者很容易地理解教材内容。

本书可作为普通高等院校非计算机专业相关课程的教材, 也可作为相关计算机等级考试的培训教材。

工业和信息化部普通高等教育“十二五”规划教材立项项目

21 世纪高等学校计算机规划教材

Visual FoxPro 数据库及程序设计基础 (第 2 版)

- ◆ 主 编 周明红
- 副 主 编 杨潞霞 孙咏梅 张季良
- 审 张永奎 相万让
- 责任编辑 邹文波
- ◆ 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>
北京铭成印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 18
字数: 470 千字
- 2011 年 9 月第 2 版
2011 年 9 月北京第 1 次印刷

ISBN 978-7-115-25722-2

定价: 34.00 元

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

反盗版热线: (010) 67171154

广告经营许可证: 京崇工商广字第 0021 号

第 2 版前言

“Visual FoxPro 数据库及程序设计基础”课程是普通高等院校非计算机专业的计算机公共课程之一,也是计算机等级考试的课程之一。本书在第 1 版《Visual FoxPro 程序设计知识与练习》使用将近 6 年的基础上,征求了许多使用教材的教师和学生的意见,结合计算机等级考试新大纲的内容进行了第 2 版的修订工作。但为了与计算机等级考试相一致,本书仍采用 Visual FoxPro 6.0 的软件版本进行编写。

《Visual FoxPro 数据库及程序设计基础》(第 2 版)做了如下修改。

1. 改变了教材名称。第 1 版教材的名称中没有提及数据库,而本课程实际是由两大部分组成的,一部分是数据库,另一部分是程序设计,所以第 2 版教材将名称改为《Visual FoxPro 数据库及程序设计基础》更能反映出课程的组成结构。

2. 统一了数据库。在《Visual FoxPro 数据库及程序设计基础(第 2 版)》整个教材中,将数据库和表文件及字段名等进行了统一,构成一个完整的数据库管理系统体系结构。

3. 增加了实例。《Visual FoxPro 数据库及程序设计基础(第 2 版)》突出的一个特点就是每个知识点都配有相应的实例加以强化,使学习者可以从中对知识进一步理解和应用。这样当学习完本书的内容后,就可利用每章的实例连编成一个完整的数据库管理系统,使知识得到连贯。

4. 辅以后练习。《Visual FoxPro 数据库及程序设计基础(第 2 版)》教材每章后都附有相应的练习题及参考答案,可以使学习者对所学知识进行检验。

教学建议:

1. 由于本课程的实践性和应用性很强,本书可以与配套的《Visual FoxPro 数据库及程序设计基础实验指导(第 2 版)》教材同时使用。另外,教材涉及的许多资料以附录的形式编写在《Visual FoxPro 数据库及程序设计基础实验指导(第 2 版)》教材中。

2. 课程学时可以为 72 学时(18 个教学周,每周 2 学时理论课,2 学时实训课),建议理论讲解与上机实训的课时比例各占一半;上课形式可以在多媒体环境教室,也可以在机房边讲解边实训,这样效果会更好。

3. 本教材难度适中,各校可以根据学生的具体情况,对教材中的知识适当进行相应的增删,对知识的难易度进行把握。

4. 根据课程特点,课程的考核形式可以用机试的方式进行,也可以与计算机等级考试相结合,以等级考核成绩的一定分值确定该课程的成绩。

本书由周明红主编,并负责全书的统稿和定稿工作。杨潞霞、孙咏梅、张季良任副主编。第 1 章、第 3 章由杨潞霞编写,第 2 章由闫俊伢编写,第 4 章、第 5 章由孙咏梅编写,第 6 章由周明红、闫俊伢编写,第 7 章、第 11 章、第 12 章由周明

红编写,第8章、第9章由张季良编写,第10章由高瑾编写,第13章由于华编写,本书最后的综合练习题及答案由闫俊伢编写。张永奎、相万让担任本书的主审。由于编者水平有限,书中难免有不足和疏漏之处,恳请广大读者批评指正。

编者

2011年7月

“Visual FoxPro 数据库及程序设计基础”是计算机专业的一门必修课程,也是计算机专业的一门必修课程。本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

本书在编写过程中,参考了国内外许多优秀的教材,力求做到概念清晰、重点突出、循序渐进、由浅入深。本书可作为高等院校计算机专业及相关专业的教材,也可作为从事计算机工作的工程技术人员参考。

目 录

第 1 章 数据库基础知识.....1

1.1 数据库基础知识.....1

1.1.1 数据和数据库的概念.....1

1.1.2 数据库系统的组成.....2

1.2 数据模型.....3

1.2.1 数据模型的基本概念.....4

1.2.2 E-R 模型.....6

1.3 关系数据模型.....9

1.3.1 关系模型的数据结构.....9

1.3.2 关系模型的数据操作.....11

1.3.3 关系的完整性.....12

1.3.4 关系代数.....13

习题 1.....15

第 2 章 Visual FoxPro 6.0 基础知识.....17

2.1 Visual FoxPro 6.0 的基本操作.....17

2.1.1 Visual FoxPro 6.0 的启动和退出.....17

2.1.2 Visual FoxPro 6.0 的窗口.....17

2.1.3 Visual FoxPro 6.0 的工作方式.....20

2.1.4 Visual FoxPro 6.0 中命令的格式要求.....21

2.1.5 简单的输入输出命令.....21

2.1.6 Visual FoxPro 6.0 的环境配置.....23

2.2 Visual FoxPro 6.0 的辅助设计工具.....24

2.2.1 Visual FoxPro 6.0 的设计器.....24

2.2.2 Visual FoxPro 6.0 的向导.....24

2.2.3 Visual FoxPro 6.0 的生成器.....25

2.3 Visual FoxPro 6.0 中的文件操作.....26

2.3.1 项目管理器简介.....26

2.3.2 文件操作.....28

习题 2.....29

第 3 章 Visual FoxPro 6.0 基本数据元素.....31

3.1 数据类型.....31

3.2 常量与变量.....33

3.2.1 常量.....33

3.2.2 变量.....35

3.2.3 内存变量的常用命令.....36

3.2.4 数组.....39

3.3 运算符和表达式.....41

3.3.1 算术运算符与数值表达式.....41

3.3.2 字符串运算符与字符表达式.....42

3.3.3 关系运算符与关系表达式.....42

3.3.4 日期时间运算符与日期时间表达式.....44

3.3.5 逻辑运算符与逻辑表达式.....45

3.3.6 名表达式.....46

3.3.7 类与对象操作符.....46

3.4 常用函数.....46

3.4.1 数值型函数.....47

3.4.2 字符型函数.....49

3.4.3 日期与时间函数.....53

3.4.4 数据类型转换函数.....54

3.4.5 测试函数.....55

3.4.6 表操作函数.....57

3.4.7 其他常用函数.....58

习题 3.....58

第 4 章 表的创建及相关操作.....60

4.1 创建和修改表.....60

4.1.1 定义表结构.....60

4.1.2 在表设计器中创建表.....63

4.1.3 表中数据的输入.....64

4.1.4 修改表结构	67	6.1 视图	111
4.2 表操作	67	6.1.1 视图的概念	111
4.2.1 更改表文件名	67	6.1.2 视图的创建	112
4.2.2 打开表与关闭表	68	6.1.3 视图的操作	119
4.2.3 表和表结构的复制	70	6.1.4 利用视图更新数据	119
4.3 表记录操作	70	6.2 查询	121
4.3.1 浏览记录	71	6.2.1 查询的含义	121
4.3.2 修改记录	72	6.2.2 查询的创建	121
4.3.3 插入记录	73	6.2.3 查询的操作	123
4.3.4 记录的删除和恢复	73	6.2.4 视图与查询的区别	125
4.3.5 记录指针的定位	76	习题 6	126
4.4 表的筛选	79	第 7 章 SQL 标准语言	128
4.4.1 记录的筛选	79	7.1 SQL 概述	128
4.4.2 字段的筛选	80	7.2 SQL 的数据查询功能	129
习题 4	81	7.2.1 Select 查询命令	129
第 5 章 创建和使用数据库	85	7.2.2 单表查询	130
5.1 有关数据库的操作	85	7.2.3 连接查询	138
5.1.1 创建数据库	85	7.2.4 嵌套查询	139
5.1.2 打开、关闭、修改数据库	87	7.2.5 集合查询	141
5.1.3 删除数据库	89	7.3 SQL 的数据定义功能	142
5.2 表与数据库的关系	90	7.3.1 表和视图的定义	142
5.2.1 向数据库中添加表	90	7.3.2 表和视图的删除	143
5.2.2 从数据库中移去表	91	7.3.3 表结构的修改	144
5.2.3 数据库表的属性	92	7.4 SQL 的数据操作功能	145
5.3 建立和使用索引	95	7.4.1 插入数据	145
5.3.1 索引的概念与作用	95	7.4.2 更新数据	146
5.3.2 索引的类型	96	7.4.3 删除数据	147
5.3.3 索引的建立与使用	97	习题 7	147
5.4 表间的关联	101	第 8 章 结构化程序设计	152
5.4.1 表间关系的类型	101	8.1 程序的建立与维护	152
5.4.2 建立表之间的关系	103	8.1.1 菜单方式建立与维护程序文件	152
5.4.3 数据完整性	104	8.1.2 命令方式建立和维护程序文件	153
习题 5	106	8.1.3 程序的编译	154
第 6 章 视图与查询	111	8.1.4 程序中常用命令	154

8.2 程序的流程控制.....157	10.1.1 表单控件的使用场景.....192
8.2.1 顺序结构.....157	10.1.2 Visual FoxPro 6.0 中的常用控件.....194
8.2.2 选择结构.....157	10.2 表单的设计实例.....210
8.2.3 循环结构.....159	10.2.1 界面表单的设计.....210
8.2.4 编程实例.....162	10.2.2 登录及注册表单的设计.....211
8.3 模块结构程序设计.....164	10.2.3 主操作界面表单.....211
8.3.1 子程序的调用.....164	10.2.4 表记录管理表单的设计.....212
8.3.2 过程及过程调用.....165	10.2.5 查询及统计表单的设计.....213
8.3.3 过程调用中的参数传递.....167	10.2.6 索引表单的设计.....215
8.3.4 变量的作用域.....168	习题 10.....216
8.3.5 自定义函数.....169	
8.4 调试程序.....170	第 11 章 报表和标签.....219
8.4.1 程序调试.....170	11.1 创建报表.....219
8.4.2 调用调试器.....170	11.1.1 利用报表向导创建报表.....219
8.4.3 设置断点.....172	11.1.2 快速创建报表.....224
习题 8.....173	11.1.3 利用报表设计器创建报表.....226
第 9 章 面向对象程序设计.....176	11.2 报表设计器及其使用.....226
9.1 面向对象编程的基本概念.....176	11.2.1 报表设计器的结构.....226
9.1.1 类与对象.....176	11.2.2 报表中各种对象的添加.....228
9.1.2 子类与继承.....178	11.2.3 报表的排序.....231
9.2 类和对象的使用.....179	11.2.4 报表的分组与总结.....231
9.2.1 Visual FoxPro 6.0 中的基类.....179	11.3 报表的输出.....232
9.2.2 常用的属性、方法和事件.....179	11.3.1 报表的页面设置.....232
9.2.3 对象的属性设置和方法的引用.....180	11.3.2 报表的预览和打印.....233
9.2.4 类的创建.....182	11.4 标签.....234
9.3 表单设计器.....185	11.4.1 利用标签向导创建标签.....234
9.3.1 表单设计器环境.....185	11.4.2 利用设计器创建和修改标签.....235
9.3.2 控件的操作与布局.....188	习题 11.....235
9.3.3 设置【Tab】键次序.....189	
9.3.4 数据环境.....189	第 12 章 菜单和工具栏.....237
习题 9.....190	12.1 菜单概述.....237
第 10 章 表单的设计.....192	12.1.1 菜单系统的规划.....237
10.1 表单控件介绍.....192	12.1.2 菜单的组成及结构.....238
	12.2 菜单的设计.....238

12.2.1	菜单设计器	238
12.2.2	创建菜单	240
12.3	使用菜单	244
12.3.1	为应用程序添加主菜单	245
12.3.2	快捷菜单的使用	246
12.3.3	启用和禁用菜单项	247
12.4	创建自定义工具栏	248
12.4.1	创建类	248
12.4.2	在表单集中使用工具栏	249
12.4.3	协调菜单和自定义工具栏	249
习题 12		250

第 13 章 应用程序的创建和发布

13.1	应用程序的开发过程	252
13.1.1	系统开发的一般步骤	252
13.1.2	主程序及主菜单的设计	253
13.1.3	连编应用程序	255
13.2	应用程序生成器	255
13.2.1	应用程序向导的使用	255
13.2.2	应用程序生成器简介	256
13.3	应用程序的发布	261
习题 13		265
附录 A	模拟试题及其答案	268
参考文献		278

第 1 章

数据库基础知识

1.1 数据库基础知识

在社会飞速发展的今天，人们接触的事物越来越多，信息量也急剧增大，这样就需要借助计算机来保存和管理大量复杂的数据，从而充分利用这些宝贵的资源。数据库技术正是由于这一需求驱动而发展起来的。目前，随着数据库技术的发展，其应用已深入到工农业生产、商业、金融、行政管理、科学研究、工程技术、国防和军事等各个领域，成为信息管理、办公自动化、计算机辅助决策、计算机辅助设计、智能信息处理等计算机应用系统的核心部分。

本章将从作为数据库基本元素的数据概念出发，详细讲解信息、数据、数据处理等数据库系统的知识和概念，这些都是学习和掌握 Visual FoxPro 的基础和前提。

1.1.1 数据和数据库的概念

1. 数据

谈起数据 (Data)，人们首先想到的是数字，其实数字只是最简单的一种数据。数据的种类很多，它可以是体重、年龄、价格的多少，可以是姓名、单位、地址的标志，还可以是声音、视频和图像等。无论什么样的数据，都是对客观事物反映的一种物理符号。

从计算机角度来看，数据就是存储在某一媒体上，能反映客观现实世界中各种信息的符号记录。这个概念包含两方面内容，一方面是数据的存储方式，另一方面是数据的具体内容。在数据库系统中，数据是数据库存储的基本对象。

2. 信息

信息 (Information) 是指经过加工处理之后的数据，是从原始数据中提炼出来的对人类客观行为产生影响的具有使用价值的信息。例如，可以通过记录全班每个同学的每门功课成绩，计算出每个同学的总成绩以及全班排名，根据这两个数据就可以得到每个同学在班里的学习情况这样一条重要信息。因此，可以通过数据了解所需的信息。

总之，信息是通过数据符号来传播的，是有用的数据；而数据则是信息的表现形式，若其不具有知识性和有用性，则不能称为信息。

3. 数据处理

数据处理 (Data Processing) 是指对各种形式的数据进行收集、整理、存储、分类、排序、检索、加工、统计和传输等一系列活动的总和。其目的是从大量的、原始的数据中抽取、导出有价

值的信息, 以此作为行为和决策的依据。

4. 数据库

数据库 (Data Base, DB) 是指长期存储在计算机设备上, 结构化的、可共享的相关数据的集合。它不仅包括描述事物的数据本身, 而且还包括相关事物之间的联系。

数据库中的数据是面向多种应用的, 具有集中性和共享性。所谓集中性, 是指把数据库看成性质不同的数据文件的集合, 其中的数据冗余很小。所谓共享性, 是指多个不同用户使用不同语言, 为了不同目的可同时存取数据库中的数据。数据库中的数据由数据库管理系统进行统一管理和控制, 用户对数据库进行各种数据操作都是通过数据库管理系统来实现的。

1.1.2 数据库系统的组成

数据库系统是指带有数据库并利用数据库技术进行数据管理的计算机系统。数据库系统是在计算机系统中引入了数据库技术之后的系统, 它一般由数据库、数据库管理系统、应用系统、数据库管理员和用户 5 部分组成。

在一般不引起混淆的情况下, 常常把数据库系统简称为数据库, 如图 1-1 所示。

1. 数据库

数据库 (DB) 就是长期存储在计算机设备上, 结构化的、可共享的相关数据的集合。它以文件的形式存储在计算机外存中, 用户通过数据库管理系统来统一管理和控制数据。

2. 数据库管理系统

收集并抽取一个应用所需的大量数据后, 如何科学地组织和存储数据, 又如何高效地处理和维持这些数据呢? 完成这个任务的是一个软件——数据库管理系统 (DataBase Management System, DBMS)。它是一个介于用户和操作系统之间的系统软件, 是数据库系统中的一个重要的组成部分, 可以帮助用户建立、使用和维护数据库。目前, 广泛使用的大型数据库管理系统有 Oracle、Sybase 等, 小型数据库管理系统有 SQL Sever、Visual FoxPro、Access 等。

数据库管理系统在计算机系统中的地位如图 1-2 所示, 其功能主要有以下几个方面。

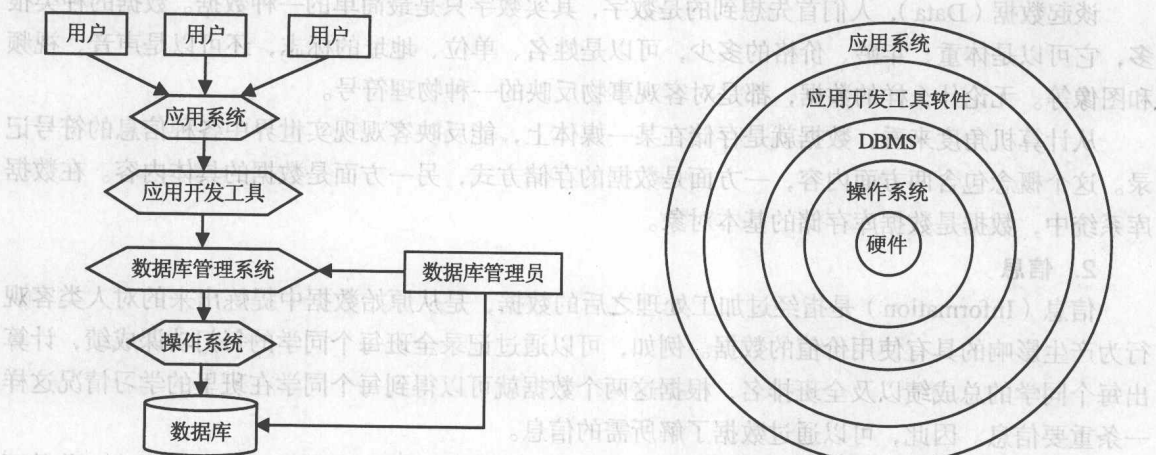


图 1-1 数据库系统

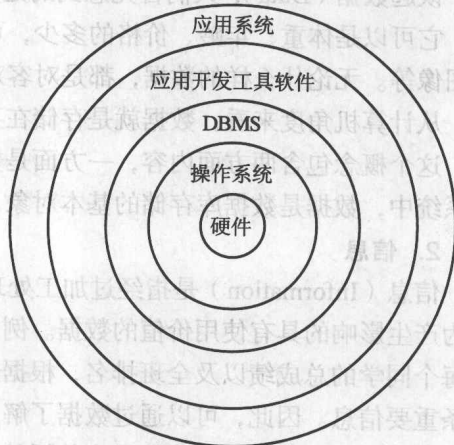


图 1-2 DBMS 在计算机系统中的地位

(1) 数据定义功能

DBMS 提供了数据定义语言 (Data Definition Language, DDL)。用户通过使用 DDL, 可以很方便地定义数据库中的数据对象。

(2) 数据操作功能

DBMS 提供了数据操纵语言 (Data Manipulation Language, DML)。用户通过使用 DML, 可以对数据库中的数据进行操纵, 如查询、插入、修改和删除数据等。

(3) 数据库的运行管理功能

数据库在建立、运行和维护时, 由 DBMS 统一管理、统一控制, 以保证数据的安全性、完整性和多用户对数据操纵的并发控制。

(4) 数据库的建立和维护功能

该功能包含数据库初始数据输入、数据转换、数据库的转存和恢复以及数据库的重构、性能监测和分析等。这些功能主要由 DBMS 提供的一些实用程序来实现。

3. 软件系统

在数据库系统中, 除了数据库管理系统外, 还必须有相关软件的支持, 包括操作系统、应用程序和开发工具。

应用程序主要是指数据库应用系统 (DataBase Application System, DBAS)。它是系统开发人员利用数据库系统资源开发出来的、面向某一类实际应用的应用软件系统。例如, 以数据库为基础的人事管理系统、财务管理系统、火车票售票系统等, 它们都是以数据库为基础和核心的计算机应用系统。

开发工具是指开发软件的具体工具, 如 Delphi、VB、ASP、JSP 等软件, 而 Visual FoxPro 本身也可作为开发工具。

4. 数据库管理员

数据库系统中还包括设计、建立、管理、使用数据库的各类人员。

(1) 数据库管理员: 数据库建立好后, 还要有专门的人员来完成, 这些人称为数据库管理员 (DataBase Administrator, DBA)。数据库管理员除了参加数据库的设计过程外, 还参与数据库系统的全面维护和管理。

(2) 系统开发人员: 负责系统的整个开发过程, 包括负责系统需求分析、数据库设计、详细设计、程序设计等的人员。

5. 用户

用户是指不需要了解数据库的设计、维护、管理等问题, 只需直接使用数据库应用系统来处理数据的一类人员。

数据库、数据库管理系统和数据库系统是 3 个不同的概念。数据库强调的是相互关联的数据; 数据库管理系统强调的是管理数据库的系统软件而数据库系统强调的是基于数据库技术的计算机系统, 它包含数据库、数据库管理系统、操作系统、应用软件、硬件系统、数据库管理员、最终用户和开发人员等。



1.2 数据模型

数据库是某个企业、组织或部门所涉及的数据的一个综合, 它不仅要反映数据本身的内容, 而且还要反映数据之间的联系。由于计算机不可能直接处理现实世界中的具体事物, 所以人们必须借助数据模型, 把具体事物转换成计算机能够处理的数据。

数据模型必须满足3方面的要求：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。一种数据模型要很好地满足这些要求，在目前尚很困难。在数据库系统中，针对不同的使用对象和应用目的，采用不同的数据模型。

根据模型应用的不同目的，可以将模型划分为两类，它们分属于两个不同的层次。第一类模型是概念模型，也称信息模型，它是按用户的观点对数据和信息建模；另一类模型是数据模型，它是按计算机系统的观点对数据建模。

1.2.1 数据模型的基本概念

1. 数据表示

为了把现实世界中的具体事物抽象、组织为某一 DBMS 支持的数据模型，人们常常首先将现实世界抽象为信息世界，然后将信息世界转换为机器世界。也就是说，首先把现实世界中的客观对象抽象为某一种信息结构，这种信息结构并不依赖于具体的计算机系统，不是某一个 DBMS 支持的数据模型，而是概念级的模型；然后再把概念模型转换为计算机上某一 DBMS 支持的数据模型，这一过程如图 1-3 所示。

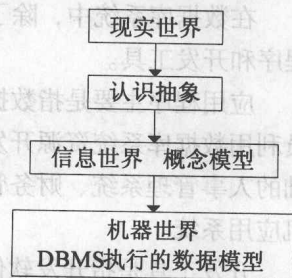


图 1-3 现实世界中客观对象的抽象过程

从图中不难看出，数据处理过程中涉及不同的数据描述领域。它从事物的客观存在到计算机的具体表示，实际上经历了3个数据领域：现实世界、信息世界、机器世界。

(1) 现实世界

现实世界就是现实存在的客观世界，包括事物及其相互联系。现实世界反映到人的大脑中，对这些事物有个认识过程，经过选择和加工，将有意义成分进行命名、分类之后就进入了信息世界。

(2) 信息世界

信息世界是人脑对现实世界的反映，人们用文字或符号记载事物及事物与事物之间的联系。在信息世界用到以下几个基本概念。

① 实体 (Entity)。客观存在并可相互区别的事物称为实体。实体可以是具体的人、物、事（如一个职工、一本书），也可以是抽象的概念或联系（如学生的一次选课、一个班级）等。

② 属性 (Attribute)。实体所具有的某一特性称为属性。一个实体可以用若干属性来刻画。例如，学生实体可以由学号、姓名、性别、出生年份、系和入学时间等属性组成，这些属性组合起来表征了一个学生。但要注意，尽管实体有许多属性，在研究时，只选择其中对管理和处理有用或有意义的属性。

③ 码 (Key)。唯一标识实体的属性集称为码。例如，学号是学生实体的码。

④ 域 (Domain)。属性的取值范围称为该属性的域。例如，性别的域为（男，女）。

⑤ 实体集 (Entity Set) 和实体型 (Entity Type)。具有相同属性的一类实体的集合称为实体集，如学生实体集和图书实体集等。

属性值的集合表示一个实体，而属性的集合表示一种实体的类型，称为实体型。例如，在学生实体集中，（92211，张三，男，1980-02-12，1998-09-01）表示某个具体学生，即一个实体，而学生（学号，姓名，性别，出生日期，入学日期）为实体型。

在 Visual FoxPro 中，用“表”来存放相同属性的一类实体，即实体集，如职工表、部门表等。表的结构称为实体型，每个表都包含若干字段，这些字段就是实体的属性。字段值的集合组成表

中的一条记录,代表一个实体,即每条记录代表一个实体。

(3) 机器世界

信息世界的信息在机器世界中以数据形式存储,机器世界对数据描述的术语如下。

① 字段(Field)。标识实体属性的符号集称为字段或数据项,它是可以命名的最小数据单位。

② 记录(Record)。记录是字段的有序集合,用来描述一个实体,即能完整描述一个实体的符号集。因此,一个记录可以由若干字段构成。

③ 文件(File)。同一记录的集合称为文件,用来描述一个实体集。

④ 关键字。关键字是能唯一标识一个记录的字段或字段的组合,用以标识不同的实体的标识符。

由以上可以看出,在数据的表示过程中涉及两类模型,它们分别属于两个不同的层次。第一类模型是概念模型,也称信息模型,它是按用户的观点对数据和信息建模,主要用于数据库设计;另一类模型是数据模型,它是按计算机系统的观点对数据建模,主要用于DBMS的实现。概念模型实际上是现实世界到机器世界的一个中间层次。

例如,对客观世界存在的学生“张三”建立数据模型,具体过程为:首先,从对象“张三”中抽象出他的一些特性,诸如学号、姓名、年龄、长相、身高、体重、血型、口音等,通过这些特性可以清楚地描述出现实中的“张三”,这些描述就是建立的概念模型。然后应用数字、字母、文字、图像、图形、声音等相应的数据形式,把抽象出来的特性用有组织的数据存储到计算机内,进行各种处理,这就是可用的数据模型。

2. 数据模型的要素

一般地讲,任何一种数据模型都是严格定义的概念的集合。这些概念必须能够精确地描述系统的静态特性、动态特性和完整性约束条件。因此,数据模型通常都是由数据结构、数据操作和完整性约束3个要素组成的。

(1) 数据结构

数据结构用于描述系统的静态特性,它是刻画一个数据模型性质最重要的方面。因此在数据库系统中,人们通常按照其数据结构的类型来命名数据模型。例如,层次结构、网状结构和关系结构的数据模型分别命名为层次模型、网状模型和关系模型。

(2) 数据操作

数据操作用于描述系统的动态特征。它是指对数据库中各种对象的实例允许执行的操作的集合,包括操作及其有关的操作规则。数据库主要有检索和更新(包括插入、删除和修改)两大类操作。数据模型必须定义这些操作的确切含义、操作符号、操作规则(如优先级)以及实现操作的语言。

(3) 数据的完整性约束

数据的完整性约束条件是一组完整性规则的集合。完整性规则是给定的数据模型中数据及其联系所具有的制约和存储规则,用以限定符合数据模型的数据库状态以及状态的变化,以保证数据的正确、有效和相容。

3. 常用的数据模型

不同的数据模型具有不同的数据结构形式。目前最常用的数据模型有层次模型、网状模型和关系模型。其中层次模型和网状模型统称为非关系模型。非关系模型的数据库系统在20世纪70年代与80年代初非常流行,现在已逐渐被关系模型的数据库系统取代。

(1) 层次模型

层次模型是数据库系统中最早出现的数据模型, 它用树型结构表示各类实体以及实体间的联系。现实世界中许多实体之间的联系本来就呈现出一种很自然的层次关系, 如行政机构、家族关系等。

① 层次数据模型的基本结构如下。

- 只有一个结点没有双亲, 该结点称为根结点;

- 根结点以外的其他结点有且只有一个双亲结点。

这就使得层次数据库系统只能直接处理一对多的实体联系。

② 层次数据模型的另外一个基本的特点是: 任何一个给定记录值, 只有按其路径查找时, 才能显出它的全部意义, 没有一个子女记录值能够脱离双亲记录值而单独存在。

(2) 网状模型

网状模型是层次模型的扩展, 它表示多个从属关系的层次结构, 它是以记录为结点的网络结构。网状模型去掉了层次模型的两个限制, 允许多个结点没有双亲结点, 允许结点有多个双亲结点, 此外它还允许两个结点之间有多种联系。可见, 网状模型可以更直接地描述现实世界。但是, 网状模型结构复杂, 实现的算法难以规范化。

(3) 关系模型

关系模型是目前最重要的一种数据模型。关系模型与层次模型、网状模型的本质区别体现在表示实体间的联系是不一样的。对于层次模型和网状模型来说, 它们是用链接指针来存储和体现联系; 而在关系模型中, 实体和实体之间的联系都是用关系 (二维表) 来表示。在用户看来, 一个关系模型的逻辑结构是一张二维表, 它由行和列组成, 共同构成该关系的全部内容。

1.2.2 E-R 模型

1. 实体间的联系

在现实世界中, 事务内部以及事务之间是有联系的, 如学生和课程之间有“选课”的联系, 读者和图书之间有“借阅”的联系。这些联系在信息世界中反映为实体内部的联系和实体之间的联系。实体内部的联系通常是指组成实体的各属性之间的联系。两个实体型之间的联系可以分为以下 3 类。

(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 中至多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, 记为 1:1。例如, 在学校中, 一个班级只有一个班长, 而一个班长只在一个班中任职, 则班级与班长之间具有一对一联系。

(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 n 个实体 ($n \geq 0$) 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 有一对多联系, 记为 1:n。例如, 一个班级中有若干学生, 而每个学生只在一个班级中学习, 则班级与学生之间具有一对多联系。

(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 n 个实体 ($n \geq 0$) 与之联系, 反之, 对于实体集 B 中的每一个实体, 实体集 A 中也有 m 个实体 ($m \geq 0$) 与之联系, 则称实体集 A 与实体集 B 具有多对多联系, 记为 $m:n$ 。例如, 一门课程同时有若干学生选修, 而一个学生可以同时选修多门课程, 则课程与学生之间具有多对多联系。

实际上,一对一联系是一对多联系的特例,而一对多联系又是多对多联系的特例。

实体型之间的这种一对一、一对多和多对多联系不仅存在于两个实体型之间,也存在于两个以上的实体型之间。例如,对于课程、教师与参考书3个实体型,如果一门课程可以由若干教师讲授,使用若干参考书,而每一个教师只讲授一门课程,每一本参考书只供一门课程使用,则课程与教师、参考书之间的联系是一对多的。

另外,同一个实体集内的各实体之间也可以存在一对一、一对多和多对多的联系。例如,学生实体集内部具有领导与被领导的联系,即某一学生(班干部)“领导”若干学生,而一个学生仅被另外一个学生直接领导,因此这是一对多的联系。

2. 概念模型的表示方法

由于概念模型是现实世界到信息世界的第一层抽象,是用户与数据库设计人员之间进行交流的语言,因此,概念模型一方面应该具有较强的语义表达能力,能够方便、直接地表达应用中的各种语义知识,另一方面它还应该简单、清晰、易于用户理解。

概念模型的表示方法很多,其中最为常用的是 P.P.S.Chen 于 1976 年提出的实体-联系方法 (Entity-Relationship Approach)。该方法用 E-R 图来描述现实世界的概念模型。

E-R 图提供了表示实体型、属性和联系的方法。

① 实体型:用矩形表示,矩形框内写明实体名。

② 属性:用椭圆形表示,并用无向边将其与相应的实体连接起来。

③ 联系:用菱形表示,菱形框内写明联系名,并用无向边分别与有关实体连接起来,同时在无向边旁标上联系的类型(1:1, 1:n 或 m:n)。

需要注意的是,联系本身也是一种实体型,也可以有属性。如果一个联系具有属性,则这些属性也要用无向边与该联系连接起来。

图 1-4 用 E-R 图描述了上面有关两个实体型之间的 3 类联系、3 个实体型之间的一对多联系和一个实体型内部的一对多联系。

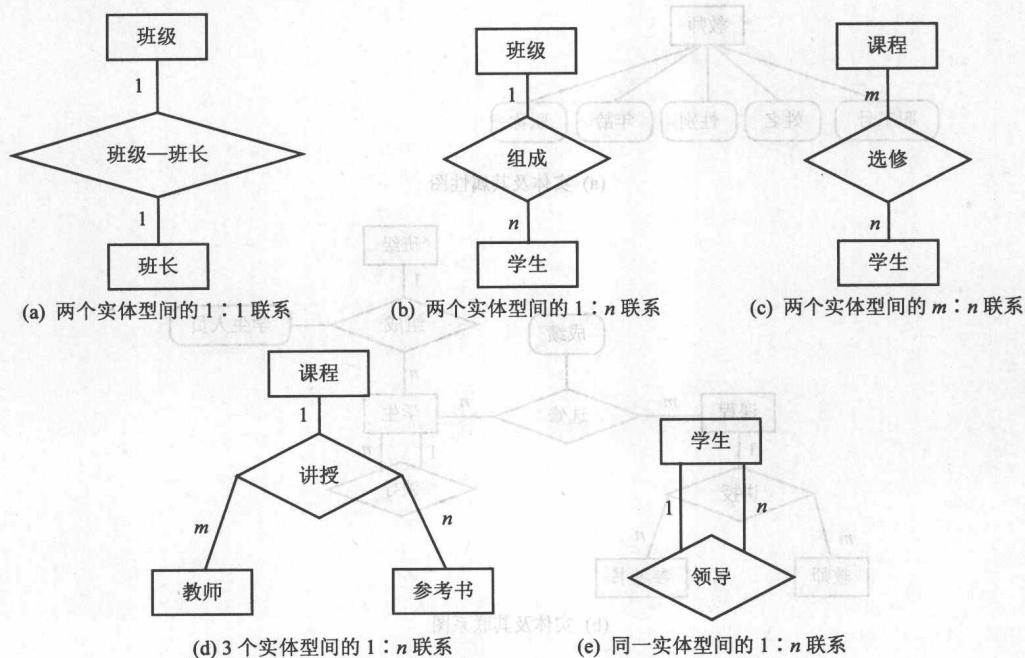


图 1-4 实体型之间及实体型的联系

假设上面的 5 个实体型, 即学生、班级、课程、教师和参考书分别具有下列属性。

学生: 学号、姓名、性别、年龄

班级: 班级编号、所属专业系

课程: 课程号、课程名、学分

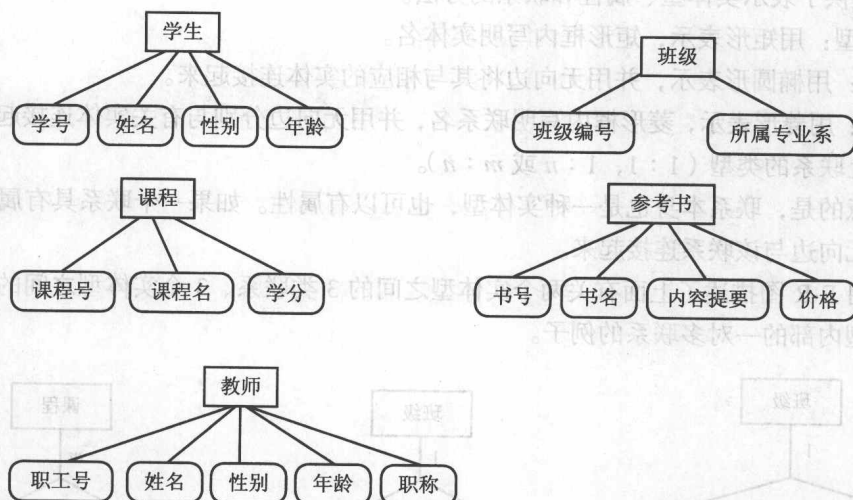
教师: 职工号、姓名、性别、年龄、职称

参考书: 书号、书名、内容提要、价格

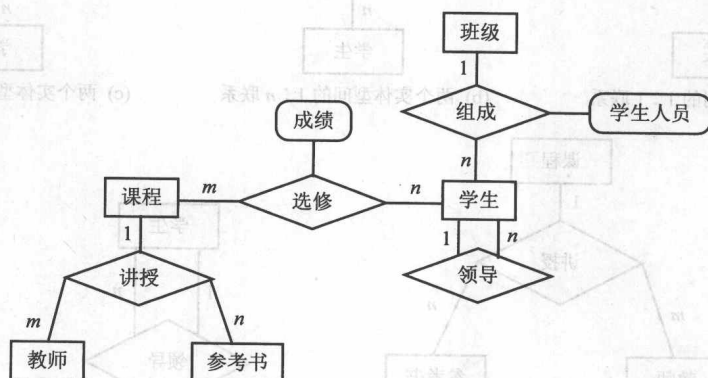
这 5 个实体的属性用 E-R 图表示, 如图 1-5 (a) 所示。这 5 个实体之间的联系可以用 E-R 图表示, 如图 1-5 (b) 所示。注意, 选修和班级两个联系又都分别具有各自的属性。

将图 1-5 (a) 与图 1-5 (b) 合并在一起 (见图 1-6) 就是一个完整的关于学校学生管理的概念模型了。但在实际当中, 在一个概念模型中涉及的实体和实体的属性较多时, 为了清晰起见, 往往采用图 1-6 的方法, 将实体及其属性与实体及其联系分别用两张 E-R 图表示。

实体-联系方法 (E-R 方法) 是抽象和描述现实世界的有力工具。用 E-R 图表示的概念模型独立于具体的 DBMS 所支持的数据模型, 它是各种数据模型的共同基础, 因而比数据模型更一般和抽象, 更接近于现实世界。



(a) 实体及其属性图



(b) 实体及其联系图

图 1-5 E-R 图实例