



21世纪高等学校文科计算机课程系列规划教材

Visual FoxPro

程序设计与应用开发教程

周山芙 编著

- 将软件的功能和操作融合到小型应用系统的开发过程中
- 强调可视化操作，使初学者容易接受和掌握
- 提供大量的例题、习题和上机练习题，涵盖等级考试的内容
- 提供习题答案和PPT课件

计算机



人民邮电出版社
POSTS & TELECOM PRESS



21世纪高等学校文科计算机课程系列规划教材

Visual FoxPro

程序设计与应用开发教程

周山芙 © 编著

计算机



人民邮电出版社
北京

图书在版编目 (CIP) 数据

Visual FoxPro程序设计与应用开发教程 / 周山芙编
著. — 北京: 人民邮电出版社, 2011.1
21世纪高等学校文科计算机课程系列规划教材
ISBN 978-7-115-23878-8

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 (2010) 第181450号

内 容 提 要

本书以在微机中广泛使用的小型数据库管理系统软件 Visual FoxPro 6.0 为例, 全面介绍数据库管理系统的基本知识及应用程序开发方法。在叙述的方法上一改过去常见的以软件功能介绍为主线的叙述方法, 尽力探索按着人的认知规律, 围绕实际应用系统需求的实现展开论述。书中提供了大量的参考例题, 并在每一章后面给出了许多相关的练习题。

本书以非计算机专业, 特别是高校财经管理类学生为主要读者对象进行编写, 并按着大学教育的规律安排各章节的内容, 因此可以作为普通大学和高职高专学生学习数据库程序设计的首选教材。本书的内容还覆盖了全国计算机等级考试二级数据库考试的内容, 可以作为准备参加计算机等级考试读者的学习参考教材。

21 世纪高等学校文科计算机课程系列规划教材

Visual FoxPro 程序设计与应用开发教程

-
- ◆ 编 著 周山芙
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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京昌平百善印刷厂印刷
 - ◆ 开本: 787×1092 1/16
印张: 20.25 2011 年 1 月第 1 版
字数: 534 千字 2011 年 1 月北京第 1 次印刷

ISBN 978-7-115-23878-8

定价: 37.00 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

Visual FoxPro 是很经典的优秀小型数据库管理系统的杰出代表, Visual FoxPro 6.0 版本是目前应用最成熟的、使用最为广泛的汉字版本, 具有强大的数据库系统管理功能, 提供了可视化操作功能和面向对象的程序设计开发工具。Visual FoxPro 不仅适合独立开发小型应用程序系统, 而且常常作为大型应用程序系统的前台开发软件。本书将以 Visual FoxPro 6.0 版本作为介绍的内容。

我认为利用 Visual FoxPro 学习数据库有几大好处。首先是它本来就是为微机开发的软件, 所以特别适合在微机上使用。其次它是从 FoxBASE、FoxPro 等版本一步一步发展而来的 PC 型数据库管理系统, 所以具有广大的老客户群, 可以继承大量的程序、数据资源。再则它结构简单、小巧, 又具备了典型数据库管理系统的大部分功能和结构, 甚至提供了 SQL 语言。此外, 它可以免费取得, 而又很少占用计算机资源, 便于安装使用。同时通过对它的学习, 还可以比较全面地了解典型数据库管理系统的结构特点以及使用方法, 便于用户将来升级掌握大型数据库系统。

本书没有采用一般书籍按软件功能, 甚至按软件结构展开介绍的方式, 而是将 Visual FoxPro 作为用户解决实际问题的工具, 以一个小型计算机应用系统的创建过程为主线, 将 Visual FoxPro 的功能和操作融合到创建应用系统对象、部件及系统的全过程中。这样的安排可以使一般的读者学习起来有的放矢、学用结合, 能在最短的时间内掌握 Visual FoxPro 的基本内容, 并能够解决一些实际工作中的问题。

本书内容尽量发挥了 Visual FoxPro 软件的可视化操作特点, 使初学者容易接受和掌握, 尽可能介绍怎样使用各种自动生成工具创建对象程序; 但为了那些准备参加等级考试的考生以及对面向过程编程情有独钟的老用户, 本书也介绍了一些过程程序的设计方法、设计技巧以及 SQL 语言的编程实例。读者可以根据自己的实际需要选择性地学习。

为了读者学习的方便和选用该教材老师的使用方便, 在书中每一章节后面都提供了大量的习题, 而且习题尽可能涵盖了等级考试的题型。同时又提供了大量的上机练习, 读者可以先按着章节中的例题上机练习, 然后再独立完成习题中的上机题。另外, 在出版社的相关网站上还将提供本书的习题答案和 PPT 课件。

本书由中国人民大学商学院周山芙编写, 为了使内容更加丰富、完整, 可能篇幅较多, 读者和老师可以根据需要选择内容。由于编者水平和能力有限, 书中会有错误和不足之处, 敬请广大读者批评指正。

编者

2010 年 8 月

第 1 章 Visual FoxPro 概述.....1

1.1 数据库的基本概念.....1	
1.1.1 数据库的基本术语.....1	
1.1.2 数据模型.....3	
1.2 Visual FoxPro 数据库管理系统.....9	
1.2.1 总体介绍.....9	
1.2.2 基本操作.....11	
1.2.3 Visual FoxPro 6.0 系统 环境的配置.....13	
1.2.4 命令格式与规则.....15	
1.2.5 常用命令.....16	
1.2.6 基本数据结构.....19	
1.3 习题.....33	
1.3.1 选择题.....33	
1.3.2 填空题.....34	
1.3.3 简答题.....34	
1.3.4 上机练习.....34	

第 2 章 数据表 and 数据库的 创建与操作.....38

2.1 自由表的创建.....38	
2.1.1 表结构的创建.....38	
2.1.2 追加数据记录.....41	
2.2 设置数据表的索引.....42	
2.2.1 索引表和索引文件.....42	
2.2.2 创建索引文件.....44	
2.2.3 索引的操作.....45	
2.3 数据表的操作与维护.....46	
2.3.1 数据表的基本操作.....46	
2.3.2 数据表的维护.....57	
2.3.3 多表的操作.....58	
2.4 数据库的设计与创建.....61	
2.4.1 创建数据库文件.....61	
2.4.2 数据库文件的操作.....63	
2.4.3 数据库表的创建.....63	

2.5 数据库表的属性设置.....66	
2.5.1 建立字段级别的有效性.....66	
2.5.2 建立记录级别的有效性.....69	
2.5.3 建立表级别的有效性.....70	
2.6 数据表的关联.....71	
2.6.1 数据表之间联系的分类型.....71	
2.6.2 创建数据库表的关联.....72	
2.6.3 数据库级别的有效性.....74	
2.7 视图操作.....75	
2.7.1 创建单表视图.....76	
2.7.2 创建多表视图.....78	
2.7.3 查看视图程序.....79	
2.7.4 视图设计器的选项卡.....80	
2.8 习题.....84	
2.8.1 选择题.....84	
2.8.2 填空题.....85	
2.8.3 简答题.....86	
2.8.4 上机练习.....86	

第 3 章 程序设计基础.....88

3.1 基本概念和分类.....88	
3.1.1 程序设计方法与语句结构.....88	
3.1.2 程序中常用命令语句与函数.....91	
3.1.3 结构化程序设计.....94	
3.2 面向过程程序的创建与编辑.....100	
3.2.1 使用内部编辑器创建程序.....100	
3.2.2 编辑修改程序.....100	
3.2.3 程序的运行.....101	
3.3 面向过程程序设计原则.....102	
3.3.1 程序设计步骤.....102	
3.3.2 利用公式设计.....102	
3.3.3 描述处理过程.....103	
3.4 面向过程程序的调试.....104	
3.4.1 程序错误的分类.....105	
3.4.2 调试器的构成.....105	
3.4.3 调试程序.....107	

3.4.4 程序错误的捕获·····	108	5.1.3 数据浏览界面·····	189
3.5 过程间的联系·····	109	5.2 用报表创建各种界面·····	203
3.5.1 过程调用·····	110	5.2.1 学生个人信息输出·····	203
3.5.2 自定义函数·····	111	5.2.2 学生数据统计输出·····	204
3.5.3 内存变量的作用区域·····	113	5.3 习题·····	206
3.6 面向对象程序设计概述·····	115	5.3.1 选择题·····	206
3.6.1 基本概念和术语·····	115	5.3.2 填空题·····	207
3.6.2 面向对象程序设计的特点·····	117	5.3.3 简答题·····	208
3.6.3 对象的创建、设置与引用·····	117	5.3.4 上机练习·····	208
3.6.4 面向对象程序的调试·····	122		
3.7 用户子类的创建和使用·····	124	第 6 章 查询处理界面 ·····	211
3.7.1 使用可视化工具·····	124	6.1 使用查询文件查询·····	211
3.7.2 使用过程程序·····	128	6.1.1 用设计器创建查询·····	211
3.8 习题·····	130	6.1.2 查询设计器的选项卡·····	214
3.8.1 选择题·····	130	6.1.3 用向导创建查询·····	216
3.8.2 填空题·····	131	6.1.4 查询输出方式的选择·····	218
3.8.3 简答题·····	132	6.2 使用结构化查询语言 SQL 查询·····	221
3.8.4 上机练习·····	135	6.2.1 数据查询命令·····	222
第 4 章 面向对象程序中的		6.2.2 数据定义命令·····	232
基本对象 ·····	137	6.2.3 数据操纵命令·····	235
4.1 表单对象·····	137	6.3 习题·····	237
4.1.1 表单的基本操作·····	137	6.3.1 选择题·····	237
4.1.2 表单中常用控件对象的设计·····	144	6.3.2 填空题·····	239
4.1.3 表单的整理与调整·····	157	6.3.3 简答题·····	240
4.2 报表对象的设计方法·····	161	6.3.4 上机练习·····	240
4.2.1 创建单表报表布局·····	162		
4.2.2 创建多表报表布局·····	167	第 7 章 接口界面设计 ·····	242
4.2.3 编辑修改报表布局·····	168	7.1 系统菜单界面·····	242
4.2.4 输出报表·····	177	7.1.1 主菜单结构·····	242
4.3 习题·····	179	7.1.2 设置系统菜单·····	243
4.3.1 选择题·····	179	7.2 菜单接口界面的创建·····	243
4.3.2 填空题·····	180	7.2.1 规划与设计·····	243
4.3.3 简答题·····	181	7.2.2 创建菜单文件·····	244
4.3.4 上机练习·····	181	7.2.3 下拉菜单的生成与运行·····	247
第 5 章 创建输入和输出界面 ·····	183	7.3 菜单界面的编辑修改·····	249
5.1 用表单创建各种界面·····	183	7.3.1 一般修改或调整·····	249
5.1.1 封面设计·····	183	7.3.2 添加系统菜单项·····	250
5.1.2 用户登录界面·····	186	7.3.3 增加提示及快捷键·····	251
		7.3.4 增加总体提示·····	252
		7.3.5 设置常规选项·····	253

7.4 快捷菜单的使用	255	8.3.2 补充和完善项目	285
7.4.1 创建快捷菜单	255	8.3.3 连编	285
7.4.2 挂接到所属对象	256	8.4 习题	287
7.4.3 快捷菜单的运行	256	8.4.1 选择题	287
7.5 用工具栏作为接口界面	256	8.4.2 填空题	288
7.5.1 创建表单工具栏	256	8.4.3 简答题	289
7.5.2 屏幕工具栏	259	8.4.4 上机练习	289
7.5.3 优化工具栏	259	第9章 Web 样式的 HTML 图形	
7.6 表单接口界面	260	帮助界面	290
7.6.1 表单的类型	260	9.1 HTML Help Workshop 软件的基本操作	290
7.6.2 创建主从表单	262	9.2 建立 HTML 文件	294
7.6.3 表单之间的连接调用	264	9.3 创建 HTML 帮助	294
7.7 习题	266	9.4 在 Visual FoxPro 中调用帮助	304
7.7.1 选择题	266	9.5 习题	306
7.7.2 填空题	267	9.5.1 选择题	306
7.7.3 简答题	267	9.5.2 填空题	307
7.7.4 上机练习	268	9.5.3 简答题	307
第8章 创建应用程序系统	270	9.5.4 上机练习	308
8.1 项目管理器	270	9.6 系统开发案例	308
8.1.1 项目管理器及其功能	270	9.6.1 开发题目	308
8.1.2 创建项目文件	271	9.6.2 开发思想	308
8.1.3 项目管理器的操作	271	9.6.3 系统功能	308
8.2 用项目管理器创建	275	9.6.4 系统结构	310
8.2.1 创建项目	275	9.6.5 主要操作界面	311
8.2.2 项目的连编	277	参考文献	317
8.3 用应用程序生成器创建	281		
8.3.1 应用向导	281		

第 1 章

Visual FoxPro 概述

在今天的信息时代，信息已经成为最重要的社会资源之一，人们的全部社会活动（生产、交流、生活等）都离不开数据信息。而对数据的采集、储存、分析加工、检索使用和维护工作更是每天自觉或不自觉地进行着的大量繁琐工作。因此建立相应的信息系统是社会、企业、部门，甚至是家庭或个人生存和发展的重要支柱。

目前国内外开发使用的绝大多数信息系统都是以数据库为基础的。数据库的应用范围已经从一般的事务处理扩展到计算机辅助设计、人工智能、软件工程、电子设计自动化（EDA）、办公室自动化、多媒体等计算机应用的各个领域。

1.1 数据库的基本概念

1.1.1 数据库的基本术语

1. 数据和信息

信息至今尚未有一个统一的、确切的定义，在不同的研究领域中对信息的内涵有不同的理解，于是形成了不同的定义和描述。

从广泛意义上说，信息是对事物运动状态和特征的描述，它反映的是关于某一客观系统中，某一事物的属性或表现形式。例如：气象，它是一个客观事物，为了描述这个事物，人们可能说“明天是风和日丽的好天气”。这就是描述气象的信息，由此可以决定是否出游等。为了在计算机中存储和处理这个信息，必须把它的属性和特征抽象出来；当然人们会根据处理的需要每次只提取部分必要的特征，比如：温度、风向风力、相对湿度、降水概率、晴/雨，并且可以描述成：

气象：(12/26, 南转北1~2级, 60%, 30%, 晴转多云)

人们通过这些数据就可以判断出天气的好坏。这些记录信息的物理符号就是数据，这样一组数据的集合叫做一条记录。数据所反映的客观事物属性是它的内容，而表示数据内容的符号则是它的形式。

为了表示不同性质的内容，数据形式通常有3种：数字型数据，即对客观事物进行定量记录的符号，如体重、年龄、价格、温度、湿度等；字符型数据，即对客观事物进行定性记录的符号，如姓名、单位、地址、风向、晴/雨等；特殊型数据，如声音、视频、图形、图像等。

从计算机处理的角度看，数据泛指那些可以被计算机接受，并能够被计算机处理的（数字化）符号。在数据库中，数据是存储和管理的基本对象。

同时，信息也定义为人类认识了的数据，是数据的含义；通俗地讲，信息是经过加工处理并

能对人类客观行为产生影响的有用数据表现形式。比如

气象：(12/26, 南转北1~2级, 60%, 30%, 晴转多云)

是一条天气好坏的信息，而其中 12/26、南转北1~2级等都仅仅是数据。数据表示了信息，而信息必须以数据形式表现才能被人们理解。

2. 数据处理

所谓数据处理，实际上包括对各种类型数据进行加工的处理操作，及把处理过的数据合理组织、存储，随时为用户服务的管理操作。数据处理包括对数据：采集、整理、存储、加工、传输等；数据管理包括：分类、编码、组织、存储、检索、维护等。

数据处理的目的是不但要从大量的、原始的数据中获得我们所需要的资料并提取有用的数据成分，而且要管理好这些数据信息，以便人们能够随时提取和使用它们。所以说数据处理是基础，数据管理是核心。

随着电子计算机软件和硬件技术的发展，数据处理过程发生了划时代的变革，而数据库技术的发展，又使数据处理跨入了一个崭新的阶段。数据处理技术的发展经历了以下3个阶段。

(1) 人工管理阶段

该阶段没有专门的软件对数据进行管理，在程序中既要考虑处理过程，又要考虑数据的定义和组织，程序和数据总是联系在一起。它主要适合早期主要应用于数据量很少的科学计算。其中程序和数据之间的对应关系如图 1-1 所示。

(2) 文件系统阶段

该阶段将数据从程序中分离出来，组成相互独立的数据文件。文件系统建立了数据文件内部的数据结构并可以管理所有数据文件，每个程序通过文件系统和自己相应的数据文件联系。但是数据文件之间没有联系，或者说文件系统在整体上是无结构的。它适用于科学计算及简单数据管理。其中程序和数据之间的对应关系如图 1-2 所示。

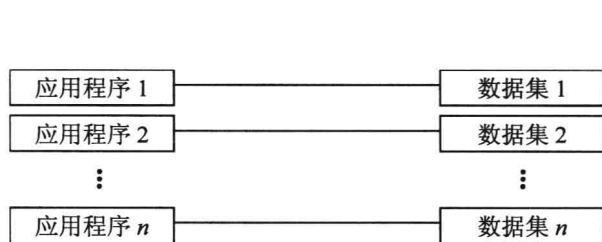


图 1-1 人工管理阶段

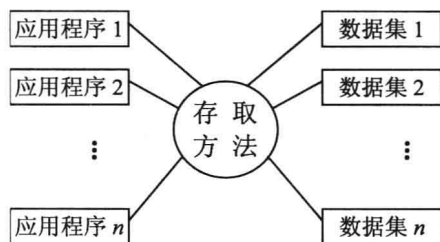


图 1-2 文件系统阶段

(3) 数据库系统阶段

该阶段一方面实现了数据与程序的完全独立，另一方面又实现了数据的统一管理。众多程序或应用需要的各种数据，全部交给数据库系统管理，大大压缩了冗余数据，实现了多用户、多应用的数据的共享。其中程序和数据之间的对应关系如图 1-3 所示。

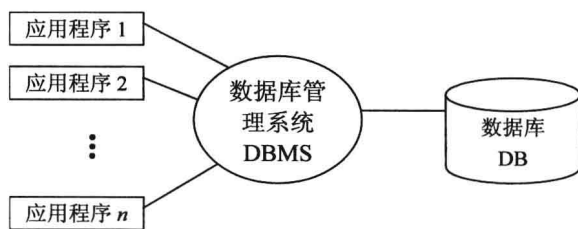


图 1-3 数据库系统阶段

3. 数据库

数据库 (Data Base, DB) 是以一定的组织方式将相关数据组织在一起并存储在计算机外存储器上所形成的能为多个用户共享的、与应用程序彼此独立的一组相互关联的数据集合。

数据库按照信息的自然联系而不是根据某个用户的需要构造数据，它是以最佳的方式，最少的冗余，为多个用户或多个应用共享服务的系统。

4. 数据库管理系统

数据库管理系统是用户建立、使用和管理数据库的软件系统，简称为 DBMS (Data Base Management System)。数据库管理系统是数据库系统的核心，DBMS 通常有下列 3 个基本部分组成。

(1) 数据描述语言 (Data Description Language, DDL)

用来描述数据库、表的结构，供用户建立数据库及表。

(2) 数据操作语言 (Data Manipulation Language, DML)

供用户对数据表进行数据的查询 (包括检索与统计) 和存储 (包括增加、删除与修改) 等操作。

(3) 其他管理和控制程序

实现数据库建立、运行和维护时的统一管理、统一控制，从而保证数据的安全、完整，及多用户并发操作。同时完成初始数据的输入、转换、转存、恢复、监控、通信以及工作日志等管理控制的实用程序。

在小型的数据库管理系统中，DDL 和 DML 通常合二为一，成为一体化的语言。

5. 数据库系统

数据库系统 (Data Base System, DBS) 是指以计算机系统为基础，以数据库技术管理大量共享数据的综合系统。它一般应当由数据库、计算机硬软件系统、数据库管理系统和用户 (最终用户、应用程序设计员和数据库管理员) 4 个部分构成。人们习惯上常常把数据库系统简称为数据库，但是应当注意与前面说的数据库 (相关数据集) 概念相区别。

一个具体的数据库系统也称为数据库应用系统，简称为 DBAS (Data Base Application System)。

在应用系统前面加上“数据库”三字，是为了区别于由普通文件管理系统支持的应用系统。一个数据库应用系统通常由数据库系统、应用程序和实际数据构成。在微机上建立数据库应用系统一般都使用通用的数据库管理系统 (如 Visual FoxPro)，用户只要开发相应的数据库和应用程序就可以了。

随着计算机应用从单机到网络的发展，数据库技术也朝网络应用的方向有了新的发展，主要表现在两个方面：一是从单用户应用扩展到多用户应用，二是从集中式应用扩展到分布式应用。早期的微机 DBMS 都是单用户系统，比如 FoxBASE 等；而 Visual FoxPro 是应用于微机网络上的多用户数据库管理系统。

1.1.2 数据模型

对现实世界事物特征的模拟和抽象就是这个事物的模型。人们对现实世界事物的研究，总是通过对它的模型研究实现的。特别是如果需要用计算机来研究处理现实世界的具体事物，必须先把具体事物转换为抽象的模型，进而转换为计算机可以处理的数据模型，从而以模拟的方式实现对现实世界事物的处理。所以说数据模型是抽象、表示和处理现实世界中事物的基本工具。

建立数据模型应当做到 3 个方面：一是真实反映现实世界，二是容易被人理解，三是便于在计算机上实现。能够真实地反映 (模拟) 现实世界是根本要求，但既要人容易理解，同时又要计算机便于理解实现就很不容易了。因此建模的过程往往采用逐步抽象的方法，并把数据模型划分为两大类；以人的观点模拟现实世界的模型叫做概念模型或信息模型，以计算机系统的观点模拟现实世界的模型叫做数据模型。

概念模型和数据模型属于信息世界和机器世界两个层次，如图 1-4 所示。为了把现实世界中的具体事物抽象成计算机能够处理的数据和信息，一般首先按人的认识观点将现实世界的具体事

物抽象为属于信息世界的概念模型，然后再将概念模型进一步转换为机器（计算机）世界的数据库模型。从图 1-4 可以看出，数据库模型还应当细分为逻辑数据库模型和物理数据库模型。

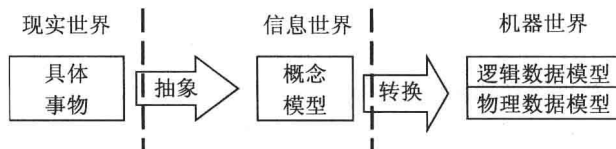


图 1-4 模型层次的划分

1. 概念模型

概念模型是从现实世界到信息世界的抽象，是数据库设计人员设计数据库的工具，同时又是与用户交流的语言。概念模型按用户的观点对现实世界建模，是缺乏计算机知识的基本用户最容易理解的，便于与数据库设计人员进行交流的语言。它独立于任何数据库管理系统，但是又很容易向数据库管理系统支持的逻辑数据库模型转换。描述概念模型最常用的工具就是实体-联系图（Entity-Relationship Approach），也叫 E-R 图。

描述概念模型的基本术语如下。

（1）实体（Entity）

客观存在并且可以互相区别的事物称为实体。它可以是具体的人、物，比如李四、中国人民大学、桌子；也可以是抽象的概念，比如交通法规、ASCII 代码；还可以是某种联系，比如学生的选课、顾客的购物。

（2）实体型（Entity Type）

用实体名及描述它的各属性名可以刻画出全部同质实体的共同特征和性质，它被称为实体型。比如：图书（书号、书名、作者、出版社、价格）就是一个图书实体型。

（3）实体集（Entity Set）

某个实体型下的全部实体叫做实体集。比如某人目前拥有的全部图书就是一个图书实体集。

（4）属性（Attribute）

实体具有的每一个特性都称为一个属性。选择的实体属性越多，刻画出的实体越清晰。属性也有“型”和“值”的区分。属性的名称（说明）就是属性的“型”，对型的具体赋值就是属性的“值”。比如图书实体可以由书号、书名、作者、出版社、价格等属性型的序列来描述，而各属性值（9012-076、英语，陈琳，人民教育出版社，2.00）的集合则描述了一本具体的图书。

（5）码（Key）

在众多属性中能够唯一标识（确定）实体的属性或属性组称为实体的码。比如图书实体的码应当是“书号”。

（6）域（Domain）

属性的取值范围称为该属性的域。比如图书的价格属性域可以是 1~200。

（7）联系（Relationship）

一个实体内部各属性之间是互相联系的，叫做实体内部联系。在各实体之间也存在着联系，称为实体的外部联系。这些联系可以分为一对一、一对多和多对多 3 类。

① 一对一联系。当前实体集中的每一个实体，在另一个实体集中最多只能找到一个可以与它相对应的实体；反过来说，在另一个实体集中的每一个实体，也只能在当前实体集中最多找到一个能够相对应的实体。那么这两个实体之间就存在着一对一的联系，并记做 1:1。

比如企业实体和领导人实体之间的联系就是一对一的联系。因为一个企业只能有一位总经理，而一个总经理又只能领导一个企业。用图形表示如图 1-5 (a) 所示。

再比如在学生实体中，每个学生的学号属性与姓名（假定没有同名同姓）属性之间存在着一对一联系。这是实体内部的联系，可以用图 1-5 (b) 表示。

② 一对多联系。当前实体集中的每一个实体，在另一个实体集中可以找到多个能够与它相对应的实体；反过来说，在另一个实体集中的每一个实体，却只能在当前实体集中找到一个能够相对应的实体。那么这两个实体之间就存在着一对多的联系，并记做 $1:n$ 。

比如：每个教师一般都能开出几门课程，若每门课程只能由一个教师开设，那么“教师”和“课程”实体之间的联系就是一对多的联系。如果反过来看“课程”和“教师”的联系就是多对一的联系了。用图形表示如图 1-6 (a) 所示。

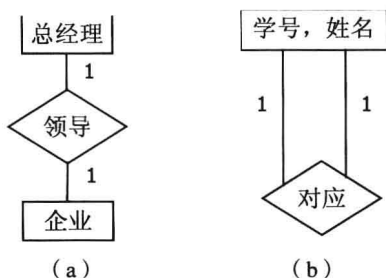


图 1-5 一对一联系

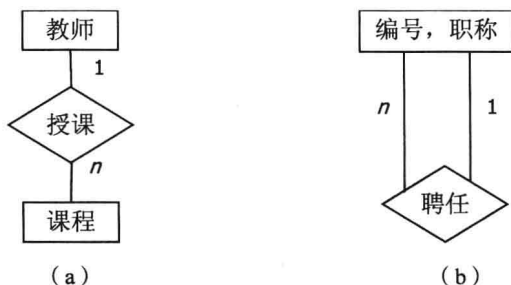


图 1-6 一对多联系

再比如在教师实体中，每一种职称（比如教授）都会有多位老师，所以职称属性与职工编号属性之间就是内部一对多的联系，可以用图 1-6 (b) 表示。

③ 多对多联系。当前实体集中的每一个实体，在另一个实体集中可以找到多个能够与它相对应的实体；反过来说，在另一个实体集中的每一个实体，也能够当前实体集中找到多个能够相对应的实体。那么这两个实体之间就存在着多对多的联系，并记做 $m:n$ 。

比如：每个教师都能开出几门课程，而每门课程又允许由多个教师开设，那么“教师”和“课程”实体之间的联系就是多对多的联系。图形表示如图 1-7 (a) 所示。

再比如，在教师实体中，如果有同名同姓的教师存在，因为每种职称可以聘任多位教师，而同一个人又可以有不同的职称，所以姓名属性与职称属性之间就存在着多对多的联系。这种对应关系如图 1-7 (b) 所示。

实际上，一对一联系是一对多联系的特例，一对多联系又是多对多联系的特例。在实际处理中，常常将两个一对一的实体合并成一个实体，而将一个多对多联系分解为一对多联系。比如：对于教师实体内部多对多联系实例，可将“职称”属性信息从教师实体中抽出来单独建立一个实体集，将所有职称实体都集中在这里。这样教师被拆分成两个实体，其内部的多对多联系就不存在了，同时“职称”实体和“教师”实体之间成了一对多联系，如图 1-8 (a) 所示。

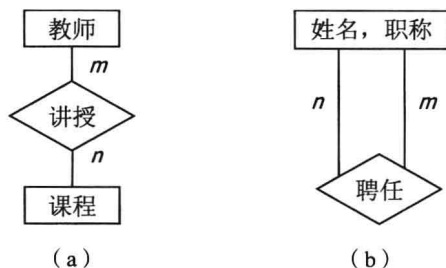


图 1-7 多对多联系

对于外部多对多联系，可以用增加一个中间实体的方法，使得一个多对多联系变成两个一对多联系。比如：先为教师开课联系创建一个新的“开课”实体，其中包含了学校所有老师开设的

全部课程记录；再让“教师”实体和“课程”实体分别都与“开课”实体联系。那么“教师”实体与“开课”实体之间，以及“课程”实体与“开课”实体之间就分别存在着一对多联系，如图 1-8 (b) 所示。在这个转换的过程中，“开课”实体起到了一种中间纽带的作用，所以就把它叫做纽带实体。

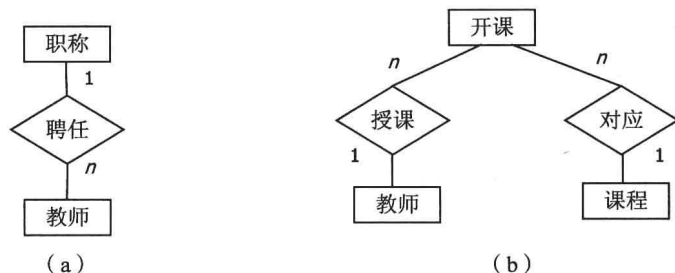


图 1-8 转换结果

2. 逻辑数据模型

逻辑数据模型是用户通过计算机系统中数据库管理系统看到的现实世界，它描述了数据库数据的整体结构。逻辑数据模型是数据库系统的核心和基础，它通常由数据结构、数据操作和数据完整性约束 3 部分概念组成。数据结构是对系统静态特性的描述，它是逻辑数据模型中最重要的部分，所以人们一般以数据结构的类型来命名该数据模型。其中常见的有：层次、网状、关系 3 类模型。不同的数据库管理系统提供了不同的逻辑数据模型。

在数据库中的数据结构如果依照层次模型定义，则该数据库为层次数据库；如果依照网状模型定义，则该数据库为网状数据库；如果依照关系模型定义，就是关系数据库系统。

(1) 层次模型

层次模型是最早出现的逻辑数据模型。层次模型以树型结构来表示实体之间联系，数据结构从树根开始向树枝、树叶逐层展开。实体用记录（类型）表示，实体的属性对应记录的数据项，实体之间的联系用有向连线表示。图 1-9 所示是某出版社组织机构的层次模型例子。

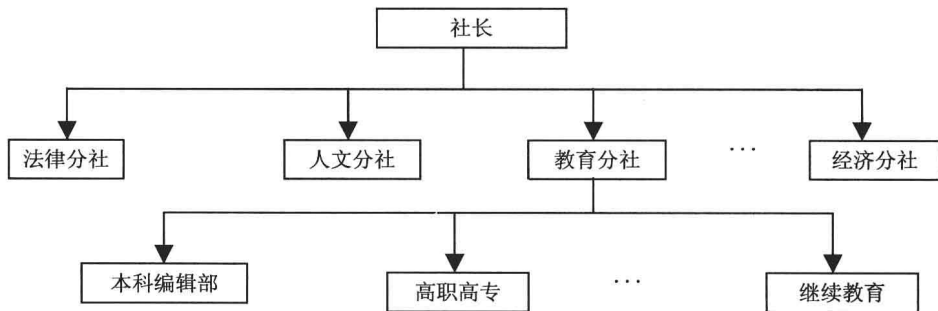


图 1-9 层次模型

层次模型以每个实体为节点，上层结点叫做父节点，下层结点叫做子节点。层次模型像一棵倒置的树，父节点在上，子节点在下，逐层排列。最上层的节点没有双亲，称为根节点；最下层的节点没有子女，称为叶节点。所以层次模型是一种以记录类型为节点的有向树结构。各节点之间的联系靠路径实现。

(2) 网状模型

层次模型表达实体间的复杂关系（多对多联系）比较困难，在此基础上产生了网状模型。它既可以表示数据间的纵向关系，又可以表示其横向关系，呈现一种交叉联系的网络结构。网状模型有一个以上的节点无双亲，但至少有一个节点有多个双亲。图 1-10 是网状模型的示意图。

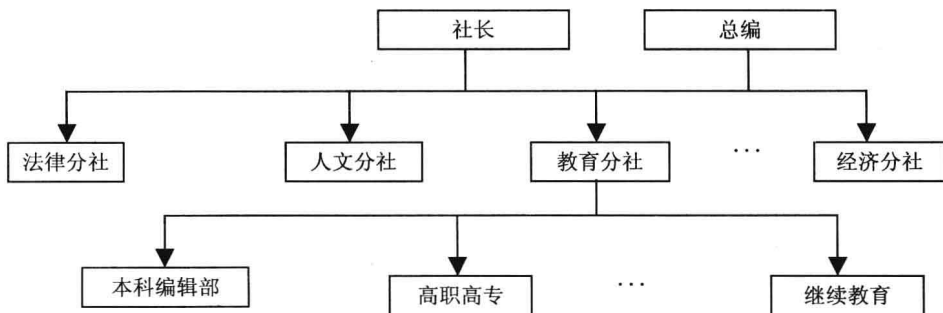


图 1-10 网状模型

网状模型也有致命的弱点，就是当需求扩展时，对原有数据结构及应用程序的修改会产生严重的后果。

(3) 关系模型

1970 年，美国的 E.F.Codd 提出了关系模型的理论，首次运用数学方法来研究数据结构和数据操作，将数据库的设计从以经验为主提高到以理论为指导。不仅如此，关系模型以人们经常使用的表格形式作为基本的存储结构，通过相同关键字段实现表格间的数据联系。关系模型简单明了、易学易用，使它从一开始就吸引了公众的注意，成为广大用户特别是微机用户乐于接受的数据模型，是目前使用最广泛的数据模型。

关系数据库系统总是由许多不同的关系构成，其中每个关系就是一个实体，可以用一张二维表表示。例如一张“职工”数据表就是一个关系，如图 1-11 所示。

编号	职称	姓名	性别	年龄
1101	教授	王国庆	男	42
1203	讲师	张小露	女	28
3019	副教授	李雯	女	33
2356	教授	黄大强	男	38

图 1-11 关系二维表

① 基本术语。

- 关系 (Relation)：一个实体对应一个关系，每个关系都有一个唯一的名字。
- 属性 (Attribute)：表中每一列叫做一个属性，属性有名和值的区别。
- 元组 (Tople)：由属性值组成的每一行叫做一个元组（记录值）。
- 框架 (Framework)：由属性名组成的表头称为框架（关系型）。
- 分量：表中每一个属性值叫做一个分量。

- 域 (Domain): 每个属性的取值范围为域。
- 候选码 (Candidate Key): 可以唯一确定一个元组的属性或属性组称为候选码 (简称码), 比如上表中的“姓名”属性。
- 主码 (Primary Key): 一个关系中往往有多个候选码, 当前指定的候选码叫主码。
- 主属性 (Primary Attribute): 包含在码中的属性叫做主属性。
- 非码属性 (Non-key Attribute): 不包含在任何候选码中的属性叫做非码属性 (也叫非主属性)。
- 关系模式: 对关系的描述称为关系模式, 关系模式常常记做:

关系名 (属性 1, 属性 2, 属性 3, …… 属性 n)

如图 1-11 所示的职工关系可以记作: 职工 (编号, 职称, 姓名, 性别, 年龄)

在关系模型中, 不但实体用关系表示, 而且实体之间的联系也用关系来表示。比如用关系模式描述图 1-7 (a) 中实体“教师”、“课程”和联系“讲授”, 其中的实体和联系分别可以记做:

教师 (教师号, 姓名, 系别)

课程 (课程号, 课程名, 学分)

讲授 (教师号, 课程号, 教室)

② 约束条件。关系模型要求关系必须是规范的, 也就是要求关系必须满足一定的规范条件, 其中最基本的一条就是, 关系中每个分量必须是不可再分的基本项。换一句话说就是在一张表中不能嵌套另一张表。

并不是所有平面表格都可以满足关系的这个约束条件, 例如图 1-12 所示的两张平面表格就不是关系二维表。

要成为一张关系二维表, 必须具备下面几个约束条件:

- ① 表格中每个数据项都不可再分, 是基本项;
- ② 每一列数据都有唯一的属性名, 并且具有相同的数据类型;
- ③ 每列的顺序是任意的;
- ④ 每一行元组是实体各属性值的集合, 表格中不允许有完全相同的元组存在;
- ⑤ 各行顺序可以是任意的。

学院		姓名	性别	年龄
系别	专业			
企业系	工商管理	李四	男	20
会计系	会计	田小雅	女	19
贸易系	贸易经济	马倩倩	女	20

单位	中国人民大学			
姓名	赵堡岛	电话	12345678	
通信地址		北京市海淀区中关村路 95 号		

图 1-12 不规范表格

3. 物理数据模型

物理数据模型用来描述数据的物理存储结构和存储方法, 它不但受数据库管理系统控制, 而且与计算机存储器、操作系统密切相关。作为一般的用户, 在数据库设计时不需要过多地考虑物理结构, 所选定的数据库管理系统总会自动地加以处理。用户只需要选择合适的数据库管理系统, 以及用该数据库管理系统提供的语句命令实现数据库。

1.2 Visual FoxPro 数据库管理系统

1.2.1 总体介绍

Visual FoxPro 是一个适用于微型计算机的关系数据库管理系统,通过它能够让用户以最有效率的方式管理与处理大量的数据,可以实现数据的新增、修改与删除,报表与标签的制作,数据的查询,甚至让用户快速自行开发一个简单的应用系统。Visual FoxPro 6.0 是目前市场上最成熟的、免费下载的软件,它不仅拥有良好的整体式开发环境,更提供了易学易用的设计器 (Designer)、向导 (Wizard)、生成器 (Builder); 用户能够通过菜单、工具栏以及对话框等可视化工具操纵和使用它,同时也可以利用语句命令来使用它。

以 Visual FoxPro 作为数据库课程的讲授和实践的平台是一个不错的选择。虽然它不是一个完善的大型数据库管理系统,真正用它建立数据库系统还不太理想,但是它却包含了典型数据库管理系统的基本组成,便于学员从此入门,进一步学习大型数据库;同时因为它是自立型软件,不需要嵌入到其他高级语言中使用,而且短小精悍、相对比较简单,所以一方面它很容易在 PC 上使用,另一方面它又很容易学习掌握,使用起来也灵活、方便。

Visual FoxPro 是关系数据库管理系统,它一方面能够创建、管理和操纵数据库数据;另一方面它又提供了高级程序设计语言和 SQL 标准查询语言,可以直接创建应用程序及数据库应用系统。

1. 基本概念和术语

(1) 数据表

在 Visual FoxPro 中,数据表简称为表,是最重要的数据存储和操作对象。一个表就是一个关系,它总是以文件的形式存放在计算机的外存储器中。表文件的存取名称就是关系名,它应当符合 Windows 操作系统对文件名的规定。一个表文件由 3 个同名文件组成,主文件的扩展名是 DBF,索引文件的扩展名是 CDX,备注文件的扩展名是 FPT。图 1-13 所示就是这样一个数据表。

编号	职称	姓名	性别	年龄
1101	教授	王国庆	男	42
1203	讲师	张小露	女	28
3019	副教授	李雯	女	33
2356	教授	黄大强	男	38

图 1-13 数据表

关系中的属性叫做数据表文件的字段,属性名叫字段名,属性值叫字段值;关系框架叫做数据表文件的结构;关系元组叫做数据表文件的记录。

(2) 数据库

在 Visual FoxPro 中,数据库也作为文件存放在计算机外存储器中,数据库文件一般由 3 个同名文件组成,其主文件扩展名是 DBC,备注文件扩展名是 DCT,索引文件扩展名是 DCX。在数据库中不能直接存放任何用户数据,它的任务只是组织和管理其中包含的数据表及视图,并且建立数据字典和数据安全保护。

(3) 索引

索引是按数据表中某个关键字或关键字表达式值的顺序,使数据表中的记录有序排列的一种技术,在 Visual FoxPro 系统中是借助于索引文件实现的。索引实际上并不改变数据表中数据的物理顺序,而是另外建立一个索引对应列表。索引表的功能和结构与图书馆的索引卡片基本相同,一种图书卡片规定了图书的一种排列顺序,并通过书号指明了该图书;而一个索引表也决定了数据表的一种排列的顺序,并通过记录序号联系数据表中该记录。

一般情况下,表中记录的顺序是由用户建立该表时数据记录输入的先后次序决定的,并用记录号予以标识。除非有记录插入或者有记录删除,否则表中记录的物理顺序不会改变。当用户有不同需求时,为了加快数据的检索、显示、查询和打印速度,需要对文件中的记录顺序重新组织,索引技术则是实现这些目的的最好办法。除此以外,要在数据库内建立数据表之间的关联也必须靠索引实现,而且 Visual FoxPro 中 SQL 语言也是靠索引技术支持的。

(4) 关键字

做为创建索引的表达式在 Visual FoxPro 中一般叫做关键字,而关键字表达式可以由数据表的一个字段,或者由若干个字段及变量、函数等表达式组成。

(5) 视图

视图是一种特殊类型的表,它往往由一个或多个表(或视图)中的部分字段或部分记录组成。但是视图并不会被作为一个完整的数据集合存放在存储器中,虽然从用户的角度看,视图也像表一样有自己的名字,相应的字段、记录,具备了一般数据表的特征,可是实际上,它只是在数据库中存放了相应的关系和操作的要求。所以视图可称为“虚表”或逻辑表。

视图分为本地视图和远程视图两类。本地视图是利用本地数据库表、自由表及视图建立在本地服务器上的视图。远程视图是利用远程服务器中的数据建立的视图。

(6) 存储过程

存储过程实际上是封装在数据库中的一个命令(程序)文件。其中可以包含系统为实现数据库中数据的保护、安全、联接、查询、视图及关联等功能而自动或人为设计的若干程序过程段、函数等。它相当于 FoxBASE 中的过程文件,用户可以随时打开查看和编辑修改。

2. 基本操作方式

Visual FoxPro 是关系数据库管理系统,其中包含了完善的数据操纵和管理的语言,通过这些内含的操纵管理命令,可以实现 Visual FoxPro 的全部功能。

(1) 人机交互方式

传统意义上的人机交互过程,是用户事先从键盘输入一条完整的操作命令后,系统接受并理解命令含义,再执行该命令并显示出执行结果的过程。虽然这种方式快捷,为数据库的老用户所熟悉和欢迎,目前系统仍然保留了这种命令操作的方式,但是在 Visual FoxPro 系统已经能够提供五六百条命令的今天,这种操作方法确实是有些强人所难。

在 Windows 的图形平台上,Visual FoxPro 系统提供了新的人机交互操作方式。这主要是通过菜单、窗口、对话框、工具栏等图形界面,直接操纵数据库的工作方式。显然,这种图形界面的操作要直观和方便得多,虽然有时候为了实现某个目的,需要反复选择很麻烦,也不快捷,但是免除了记忆大量命令之苦,必然最受新用户的欢迎。Visual FoxPro 也主要支持和提倡这种可视化的人机交互方式。

(2) 程序执行方式

交互方式虽然简单、方便,但是只适用于解决简单问题。要解决复杂问题,或建立需要反复操作的应用系统,必然需要创建程序,并用程序执行方式实现用户的要求。