

21世纪高等教育计算机规划教材

COMPUTER

Visual FoxPro 数据库程序设计

Visual FoxPro Database
Programming

■ 李志强 主编

■ 郭欣 副主编

— 多年教学和实践开发经验的总结

— 符合全国计算机等级考试要求

— 配有大量习题供学生进行练习



人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等教育计算机规划教材

COMPUTER

Visual FoxPro 数据库程序设计

Visual FoxPro Database
Programming

■ 李志强 主编

■ 郭欣 副主编



人民邮电出版社

北京

图书在版编目 (CIP) 数据

Visual FoxPro数据库程序设计 / 李志强主编. --
北京: 人民邮电出版社, 2013. 2
21世纪高等教育计算机规划教材
ISBN 978-7-115-30391-2

I. ①V… II. ①李… III. ①关系数据库系统—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 (2013) 第005006号

内 容 提 要

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最新编制出版的《高等学校计算机基础教学发展战略研究报告暨计算机基础课程教学基本要求》中有关“计算机程序设计基础课程教学基本要求”组织编写。

本书共分9章, 主要讲解关系数据库的基础知识, 中文 Visual FoxPro 6.0 的命令体系, 结构化、过程化和面向对象程序设计方法, SQL 应用技术以及菜单、报表的设计与应用。本书力求为授课教师在教授本门课程时较容易地将面向过程程序设计和面向对象程序设计方法结合在一起, 同时力求学生在学习本门课程时便于掌握和理解。本书贴近全国计算机等级考试二级 Visual FoxPro 考试内容, 读者学习后, 能够为参加计算机等级考试或者从事各类实用数据库应用系统的设计、开发与维护打下坚实的基础。

本书可作为高等院校的学生教材, 也可以作为参加计算机等级考试的考生和计算机专业人员的参考书。

21 世纪高等教育计算机规划教材

Visual FoxPro 数据库程序设计

-
- ◆ 主 编 李志强
 - 副 主 编 郭 欣
 - 责任编辑 李海涛
 - ◆ 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>
北京艺辉印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 14.5 2013 年 2 月第 1 版
字数: 391 千字 2013 年 2 月北京第 1 次印刷

ISBN 978-7-115-30391-2

定价: 32.00 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

计算机信息技术是现代信息技术的核心，其核心与基础就是数据库技术。随着社会信息化程度的不断提高，数据库技术的应用已越来越广泛，越来越深入。理解数据库技术及基本工作原理，掌握数据库技术的基本操作与基本技能是现代社会所必需的知识。因此，现在高校无论是理工医农，还是经济文史几乎全面开设了“数据库程序设计”这门课程。这是因为在现代信息社会，信息都是储存在计算机系统里的，它们的组织形式多为数据库。学会数据库就学会了信息的组织方法，就会利用信息为自己的工作服务。也就是说，学习这门课程，可以使具备信息处理的基本技能，并且为在信息技术应用方向上进一步深入学习打下基础。

本书根据教育部对高等院校计算机公共基础课程中有关对数据库和程序设计方面的基本要求，由多年从事程序设计教学，具有丰富教学与应用项目开发经验的高校教师组织编写。全书从数据库基础理论和数据库技术操作应用两个方面进行了内容的组织和编排。主要内容包括数据库基础、Visual FoxPro 编程语言基础、数据库及数据表的基本操作、查询与视图、结构化查询语言 SQL 基础、结构化程序设计基础、表单设计、菜单设计和报表。本书在编排上从简到繁、由浅入深，围绕各章主题，通过大量示例循序渐进地讲解，做到内容新颖、结构完整、概念清晰、通俗易懂、实用性强。每章都配有习题，并配有大量的上机练习题。另外，本书在编写过程中，参考了全国计算机等级考试的考试大纲，本书基本涵盖了全国计算机等级考试二级 Visual FoxPro 考试的知识点。

本书由李志强任主编，郭欣任副主编，参编人员有张新彩、高丽、辛焦丽。全书由李志强统稿并定稿。在本书的编写和出版过程中得到了中原工学院信息商务学院和人民邮电出版社的大力支持和帮助，在此表示衷心感谢。

本书体现了编者在数据库技术方面教学与开发的经验，但难免存在疏漏与不足之处，望读者批评指正。

编 者
2013 年 1 月

第 1 章 数据库基础与

Visual FoxPro 概述.....1

1.1 数据库基础.....1	
1.1.1 数据库基本概念.....1	
1.1.2 数据处理技术的发展.....2	
1.1.3 数据模型.....3	
1.2 关系数据库.....5	
1.2.1 关系术语.....5	
1.2.2 关系运算.....6	
1.3 Visual FoxPro 简介.....6	
1.3.1 Visual FoxPro 的发展简史.....6	
1.3.2 Visual FoxPro 的特点.....7	
1.3.3 Visual FoxPro 的安装、启动与退出.....7	
1.4 Visual FoxPro 6.0 集成开发环境.....8	
1.4.1 Visual FoxPro 6.0 的主界面.....8	
1.4.2 配置 Visual FoxPro 6.0 的环境.....9	
1.5 Visual FoxPro 的文件类型.....12	
习题.....13	

第 2 章 Visual FoxPro 编程语言

基础.....15

2.1 Visual FoxPro 的数据类型.....15	
2.1.1 Visual FoxPro 的基本数据类型.....15	
2.1.2 字段数据类型.....16	
2.2 Visual FoxPro 的常量与变量.....16	
2.2.1 常量.....16	
2.2.2 变量.....18	
2.3 函数.....21	
2.3.1 数值函数.....21	
2.3.2 字符函数.....22	
2.3.3 日期和时间函数.....23	
2.3.4 数据类型转换函数.....24	
2.3.5 测试函数.....25	

2.4 运算符与表达式.....28	
2.4.1 算术运算符与算术表达式.....28	
2.4.2 字符运算符与字符表达式.....29	
2.4.3 日期时间运算符与日期时间表达式.....29	
2.4.4 关系运算符与关系表达式.....29	
2.4.5 逻辑运算符与逻辑表达式.....30	
2.4.6 运算符的优先级.....31	
2.5 Visual FoxPro 的命令结构和书写规则.....31	
2.5.1 命令格式.....31	
2.5.2 修饰子句.....31	
2.5.3 命令书写规则.....32	
习题.....32	
实验 1.....34	

第 3 章 数据库及数据表的基本

操作.....36

3.1 项目管理器.....36	
3.1.1 创建、打开项目.....36	
3.1.2 项目管理器的操作.....38	
3.1.3 定制“项目管理器”.....41	
3.2 数据库的基本操作.....42	
3.2.1 数据库的概念.....42	
3.2.2 数据库的建立方法.....42	
3.2.3 数据库的打开.....43	
3.2.4 数据库的修改.....44	
3.2.5 数据库的添加和移去.....44	
3.3 数据表的基本操作.....45	
3.3.1 相关概念.....45	
3.3.2 自由表的建立.....46	
3.3.3 数据库表的建立.....50	
3.3.4 数据表的浏览.....52	
3.3.5 数据表的维护.....59	
3.4 排序、索引、统计和汇总.....66	
3.4.1 数据表排序.....67	

3.4.2 索引	68	5.4.3 对查询结果排序	120
3.4.3 数据统计	74	5.4.4 使用集合函数查询	122
3.5 多表操作	76	5.4.5 分组查询	124
3.5.1 工作区的基本概念	76	5.4.6 多表查询	125
3.5.2 数据完整性	77	5.4.7 嵌套查询	127
3.5.3 参照完整性与表之间的关联	77	习题	128
3.6 数据字典	79	实验 4	129
3.6.1 设置字段的属性	79		
3.6.2 设置有效性规则	81	第 6 章 结构化程序设计基础	131
3.6.3 设置触发器	82	6.1 VFP 程序文件	131
习题	83	6.1.1 程序的概念	131
实验 2	86	6.1.2 程序文件的建立与运行	132
第 4 章 查询与视图	91	6.2 VFP 程序设计中常用的基本命令	134
4.1 查询	91	6.2.1 程序文件中的辅助命令	134
4.1.1 查询设计器	91	6.2.2 输入/输出语句	135
4.1.2 建立查询	92	6.3 顺序结构程序设计	135
4.1.3 运行查询	95	6.4 分支结构程序设计	136
4.1.4 查询设计器的局限性	96	6.4.1 单分支结构	137
4.2 视图	96	6.4.2 双分支结构	137
4.2.1 建立视图	97	6.4.3 多重分支结构	138
4.2.2 视图与数据更新	98	6.5 循环结构程序设计	140
4.2.3 远程视图与连接	100	6.5.1 FOR 循环语句	140
4.2.4 使用视图	101	6.5.2 DO WHILE 语句	140
习题	102	6.5.3 SCAN 扫描循环	142
实验 3	103	6.5.4 循环嵌套	143
第 5 章 结构化查询语言 SQL		6.6 过程与过程调用	143
基础	106	6.6.1 子程序设计与调用	144
5.1 SQL 概述	106	6.6.2 过程与过程文件	145
5.2 数据定义功能	107	6.7 程序的调试方法	146
5.2.1 建立表结构	107	6.7.1 程序调试的概念与目的	146
5.2.2 修改表结构	109	6.7.2 程序调试方法	147
5.3 数据操纵功能	109	6.7.3 Visual FoxPro 6.0 调试器	148
5.3.1 插入记录	109	习题	149
5.3.2 更新记录	110	实验 5	152
5.3.3 删除记录	112		
5.4 查询功能条件	113	第 7 章 表单设计	154
5.4.1 基本查询	113	7.1 面向对象的概念	154
5.4.2 带条件查询	115	7.1.1 对象	154
		7.1.2 类	155
		7.1.3 Visual FoxPro 中的类	155
		7.1.4 属性、事件和方法	157

7.1.5 创建对象	158	8.1.3 创建菜单系统的步骤	190
7.2 表单的基本操作	160	8.2 菜单设计器	191
7.2.1 创建表单	160	8.3 下拉菜单设计	193
7.2.2 保存表单	161	8.3.1 菜单设计的基本过程	193
7.2.3 运行表单	161	8.3.2 菜单设计实例	194
7.2.4 修改表单	161	8.4 快捷菜单设计	196
7.3 表单设计器	162	8.4.1 快捷菜单设计的基本过程	196
7.3.1 表单设计器环境	162	8.4.2 快捷菜单设计实例	196
7.3.2 设置数据环境	164	习题	199
7.3.3 控件的基本操作	166	实验 7	199
7.4 表单的属性、事件和方法	167	第 9 章 报表	201
7.4.1 表单的属性	167	9.1 报表向导与快速报表	201
7.4.2 表单的事件	168	9.1.1 报表向导的使用	201
7.4.3 表单的方法	170	9.1.2 快速报表	204
7.5 常用控件的设计	170	9.2 利用“报表设计器”创建报表	206
7.5.1 标签控件	170	9.2.1 调用“报表设计器”	206
7.5.2 文本框控件	171	9.2.2 “报表设计器”介绍	206
7.5.3 编辑框控件	172	9.2.3 报表工具栏	207
7.5.4 命令按钮控件	172	9.2.4 修改报表布局	208
7.5.5 命令按钮组控件	173	9.2.5 报表的数据环境	208
7.5.6 选项按钮组控件	175	9.2.6 报表控件	209
7.5.7 复选框控件	176	9.3 设计分组报表	211
7.5.8 列表框控件	177	9.4 设计多栏报表	212
7.5.9 组合框控件	178	9.5 输出报表	214
7.5.10 图像控件	179	9.6 标签设计	215
7.5.11 表格控件	180	9.6.1 标签向导	215
7.5.12 页框控件	182	9.6.2 标签输出	215
7.5.13 定时器控件	183	习题	216
习题	185	实验 8	218
实验 6	187	附录 A Visual FoxPro 数据库操作	
第 8 章 菜单设计	189	命令常用命令表	219
8.1 菜单系统基础知识	189	附录 B 计算机等级考试二级 Visual	
8.1.1 菜单的基本概念	189	FoxPro 考试大纲	224
8.1.2 菜单的类型	190		

第 1 章

数据库基础与 Visual FoxPro 概述

在信息化的社会里，信息大都存储在计算机系统中，它们的组织形式多为数据库。所以，想要利用信息为自己的工作、学习服务，就有必要学习数据库的基础知识。本章主要介绍数据库系统的一些基本概念和 Visual FoxPro（简称 VFP）的集成开发环境，使读者对关系数据库及 VFP 大致有所了解，为今后的学习奠定基础。

1.1 数据库基础

1.1.1 数据库基本概念

数据库技术涉及许多基本概念，主要包括数据、数据处理、数据库、数据库管理系统以及数据库系统。

1. 数据

数据（Data）是指存储在某一种媒介上能够被识别的物理符号。数据不仅包括数字、文字、字母和其他特殊字符组成的文本形式，还包括图形、图像、动画、声音等多媒体形式。例如，某个同学的姓名、出生日期等都表示数据。

2. 数据处理

数据处理是指对各种形式的数据进行收集、存储、加工、传播等一系列活动，即对数据进行查找、统计、分类、修改等处理过程。其目的之一是从大量的、原始的数据中抽取、推导得到对人们有价值的信息以作为决策的依据；目的之二是为了借助计算机，科学地保存和管理复杂的、大量的数据，以便人们能够方便而充分地利用这些宝贵的信息资源。

3. 数据库

数据库（DataBase，DB）可以直观地理解为存放数据的仓库。只不过这个仓库是在计算机的大容量存储器上，如硬盘，而且数据是按一定的格式存放的，以便于查找。

4. 数据库管理系统

数据库管理系统（DataBase Management System，DBMS）是对数据库进行管理的系统软件，它的主要功能包括数据定义、数据操纵、数据库运行控制、数据库的建立和维护。

5. 数据库系统

数据库系统（DataBase System，DBS）是数据库技术支持的计算机系统，它由以下几部分组成：计算机硬件系统、数据库、数据库管理系统、相关软件（包括操作系统、编译系统和应用系

统开发工具等)、各类人员(包括数据库管理员、系统分析员、应用程序员和最终用户)。

1.1.2 数据处理技术的发展

利用计算机进行数据处理,使得数据处理技术不断丰富和发展,到目前为止大致经历了人工管理、文件管理、数据库管理、分布式数据库管理等不同发展阶段。

1. 人工管理阶段

20 世纪 50 年代前,计算机主要用于数值计算。从当时的硬件看,外存只有纸带、卡片、磁带,没有直接存取设备;从软件看(实际上,当时还未形成软件的整体概念),没有操作系统,也没有数据管理方面的软件;从数据看,数据的组织和管理完全依靠程序员手工完成,数据管理的效率很低,且数据完全依赖于特定的应用程序,缺乏独立性。

2. 文件管理阶段

从 20 世纪 50 年代后期到 20 世纪 60 年代中期,计算机软硬件技术发展到了一个新阶段,在硬件方面,已经出现了磁盘等直接存取数据的存储设备;软件方面出现了操作系统,并在操作系统中包含了专门管理数据的文件系统。

在这一时期,数据与程序在存储位置上完全分开,数据被单独组织成文件保存到外部存储器上,数据文件可以被任何程序重复利用;当程序读写外存文件时,需要在程序中给出数据的存取格式和方法,程序设计仍然受到数据存储格式和方法的影响,即程序不完全独立于数据。

3. 数据库管理阶段

20 世纪 60 年代中期以来,由于计算机管理的数据量急剧增加,对数据共享性要求更强。原有的文件系统远远满足不了需要,因此在文件管理系统的基础上发展到数据库管理阶段。

数据库管理阶段的主要特点如下。

① 数据结构化。文件管理阶段只考虑了同一文件内的数据联系,而不同文件的数据之间是没有联系的。因此,从整体来说数据之间是没有结构的。这样的文件是有局限性的,不能反应现实世界各种事物之间错综复杂的联系。数据库管理阶段则实现了整体数据的结构化,在描述数据的时候,不仅要描述数据本身,还要描述数据之间的联系。数据结构化是数据库的一个主要特征。

② 数据共享。在文件系统中,由于数据文件之间是相互独立的,不同的文件通常是不同的应用服务的,所以同一单位的不同部门通常需要各自建立不同的数据文件。例如,一个单位的人事部门需要建立人事文件,财务部门需要建立职工工资文件,这些文件都含有每个职工的有关信息,如职工号、姓名、性别等信息,从单个文件来说都是必需的,但从总体来说,出现了数据重复存储,即出现了数据冗余。在数据库系统中,数据库的组织是针对整个单位,而不是每个部门,整个应用中的一种性质的数据尽量只出现在一个数据文件中,其他文件通过使用公共数据项(如职工号)与之连接,这样可提高数据的共享度,减少数据重复和冗余。

③ 数据与程序的独立性较高。在数据库管理阶段,应用程序不是直接与数据库打交道,而是通过数据库管理系统对数据库进行操作的。数据的定义、查询、删除、插入和修改都由数据库管理系统负责,应用程序员不需要在程序中定义数据,当然也就不需要定义数据的逻辑结构和物理结构了。因此,当数据的逻辑结构和物理结构发生改变时,一般不需要修改程序,数据和程序的独立性很强。

④ 数据存取粒度小。文件系统中,数据存取的最小单位是记录;而在数据库系统中,数据存取的粒度可小到记录中的一个数据项。因此,数据库中数据存取的方式非常灵活,便于对数据的管理。

⑤ 数据库管理系统(DBMS)可对数据进行统一的管理和控制。

4. 分布式数据库管理阶段

分布式数据库管理是数据库技术、通信技术和网络技术相结合的产物，并随着这3种技术的发展而得到不断的发展和进步。它通过计算机网络和通信线路可以把不同地域的、不同局域网环境下的、不同类型的数据库系统连接和统一管理起来。它除了具有一般数据库的优点之外，还具有系统的兼容性强、可靠性高、地域范围广、数据量大、客户数多等优点，缺点是由提高可靠性而带来的数据冗余，由不同局部数据库使用平台相连接和统一管理所带来的系统复杂性。

1.1.3 数据模型

数据库系统在管理数据时，可以根据数据特点的不同采取不同的数据模型来管理数据，因而能充分发挥数据库系统的功能。

1. 实体

从数据处理角度来看，实体就是现实世界中的客观事物。它可以指人，如一个学生、一名教师等；也可以指物，如一本书、一台电视机等；也可以指抽象的事件，如一次旅游、一场球赛等；还可以指事物与事物之间的联系，如学生选课、客户订货等。

一个实体有不同的属性，如学生实体可以用学号、姓名、性别、出生日期、籍贯等属性来描述。每个属性可以有不同的值。属性值的变化范围称作域，如“性别”这个属性值的域为（男，女）。属性集的一次取值表示一个实体。同类型的实体的集合称为实体集。

2. 实体间的联系

实体之间的对应关系反映了现实世界事物之间的相互关联。两个实体间的联系可以归结为以下3种类型。

（1）一对一联系

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对应，而对于实体集B中的每一个实体，实体集A中有且只有一个实体与之对应，则称实体集A与实体集B具有一对一联系。例如，考查公司和总裁这两个实体集，如果一个公司只有一个总裁，一个总裁不能同时在其他公司再兼任总裁，在这种情况下公司和总裁之间是一对一的联系。

（2）一对多联系

如果对于实体集A中的每一个实体，实体集B中有多个实体与之对应，而对于实体集B中的每一个实体，实体集A中至多只有一个实体与之对应，那么我们称实体集A与实体集B具有一对多联系。例如，考查学生和系这两个实体集，一个学生只能在一个系里注册，而一个系有很多学生，系和学生是一对多的联系。

（3）多对多联系

如果对于实体集A中的每一个实体，实体集B中有多个实体与之对应，而对于实体集B中的每一个实体，实体集A中也有多个实体与之对应，那么我们称实体集A与实体集B具有多对多联系。例如，考查学生和课程这两个实体集，一个学生可以选修多门课程，一门课程可由多个学生选修，所以，学生和课程间存在多对多的联系。

3. 数据模型

模型：现实世界特征的模拟和抽象。

数据模型（Data Model）：是对客观事物及其联系的描述，反映实体内部和实体之间的联系。根据采用的数据模型不同，相应的数据库管理系统也不同。在数据库系统中，常用的数据模型有层次模型、网状模型、关系模型和面向对象模型。

(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，层次数据库系统采用层次模型作为数据的组织方式。层次数据库系统的典型代表是 IBM 公司的 IMS (Information Management System) 数据库管理系统。

层次模型是用树状结构来表示各类实体以及实体间的联系。在数据库中，满足以下 3 个条件的数据模型被称为层次模型。

① 有且只有一个结点没有双亲结点，这个结点称为根结点。

② 根以外的其他结点有且只有一个双亲结点。

③ 每个结点可以有若干个孩子结点。

④ 在层次模型中，每个结点表示一个实体集，实体集之间的联系用结点之间的连线(有向边)表示，这种联系是父子之间的一对多的联系。层次模型只能处理一对多的实体联系，如图 1-1 所示。

(2) 网状模型

网状模型系统采用网状模型作为数据的组织方式。网状数据模型的典型代表是 DBTG 系统。网状模型是用网状结构来表示各类实体以及实体间的联系。满足以下两个条件的数据模型称为网状模型。

① 可以有一个以上的结点无父结点。

② 至少有一个结点有一个以上父结点。

网状模型是一种比层次模型更具有普遍性的结构，它去掉了层次模型的两个限制，允许多个结点没有双亲结点，允许一个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。因此，网状模型可以更直接地描述现实世界。网状模型如图 1-2 所示。

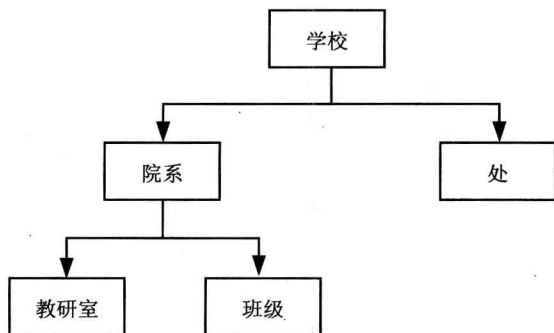


图 1-1 学院组织机构的层次模型

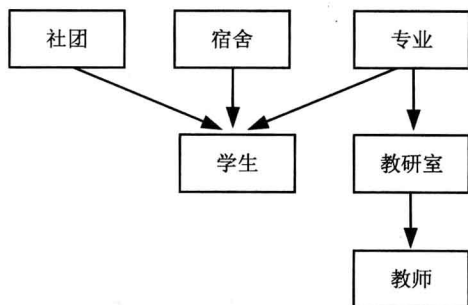


图 1-2 学院管理网状模型

(3) 关系模型

关系模型是目前最常用的一种数据模型。关系数据库系统采用关系模型作为数据的组织方式。关系模型是用二维表结构来表示各类实体以及实体间的联系，如表 1-1 所示。

表 1-1

学生关系

学 号	姓 名	性 别	出 生 日 期	籍 贯
2009260001	张丽	女	03/01/89	河南省郑州市
2009260002	李伟	男	08/12/90	山东省济南市
2009260003	张文斌	男	10/20/92	广东省中山市
2009260004	赵露	女	11/25/89	湖北省武汉市
2009260005	宁静	女	05/08/92	湖南省长沙市

在关系模型中,操作的对象和结果都是二维表,它由行和列组成。一张二维表就对应关系模型中的一个关系,满足下列条件的二维表称为一个关系。

- ① 表中每一列数据属于同一个域,不同列允许对应同一个域。
- ② 表中列的顺序可以是任意的。
- ③ 表中行的顺序可以是任意的。
- ④ 表中的任意两行不能完全相同。
- ⑤ 表中的分量是不可再分割的最小数据项,即表中不允许有子表。

(4) 面向对象模型

面向对象模型是数据库系统中继层次、网状、关系等传统数据模型之后得到不断发展的一种新型的数据模型。它是数据库技术与面向对象程序设计技术相结合的成果。

面向对象模型中基本的概念是类和对象。对象具有封装性、继承性和多态性。

面向对象数据模型比层次、网状、关系数据模型具有更加丰富的表达能力。但正因为其丰富的表达能力,模型相对复杂,实现起来较困难。因此,面向对象数据库尚未达到关系数据库的普及程度。

1.2 关系数据库

关系数据库系统是支持关系模型的数据库系统,它是目前应用最广泛、最重要、最流行的数据库。关系模型的数据结构非常简单,只包含单一的数据结构——关系。在关系模型中,实体及其实体间的联系均由关系来表示。

1.2.1 关系术语

在关系数据库中,经常会提到关系、域、码等基本概念。为了进一步了解关系数据库,首先给出一些关系术语的定义。

1. 关系 (Relation)

一个关系就是一张二维表,每个关系都有一个关系名。在 VFP 中,一个关系存储为一个文件,文件扩展名为.dbf,称为“表”。二维表中水平方向的行称为元组,有时也叫做一条记录;二维表中垂直方向的列称为属性,有时也叫做一个字段。

2. 域 (Domain)

域是一组具有相同数据类型值的集合,又称为值域,通常用字母 D 来表示。在数据库中是指指数据表列(或属性)的取值范围。例如,自然数、整数、实数、长度小于 25 的字符串集合、 $D_1 = \{\text{张三, 李四, 王五}\}$ 、 $D_2 = \{\text{男, 女}\}$ 等,都可以是域。

3. 码

① 候选码 (Candidate key): 能唯一标识元组的一个属性或属性集。例如,“学生关系”中的“学号”、“身份证号”;“选课关系”中“学号+课程号”是候选码。候选码可以有多个。

② 主码 (Primary key): 又称主键。从候选码中选出一个作为主码,不同时刻可选用不同的候选码做主码。例如,“学生关系”中的“学号”、“身份证号”是候选码,则可选“学号”为主码,或选“身份证号”为主码。

③ 备用码 (Alternate key): 除主码之外的候选码是备用码,可以没有,也可以有一个或多个。

④ 外码 (Foreign key): 若关系 1 中的属性在关系 2 中作主码, 则称该属性为关系 1 的外码。

⑤ 主属性 (Prime Attribute) 和非主属码: 包含在任意候选码中的属性为主属性, 其他为非主属性。

4. 关系模式

关系模式又称为关系结构, 是对关系的描述。

其格式为: 关系名 (属性名 1, 属性名 2, ..., 属性名 n)

一个关系模式对应一个关系的结构。例如, 学生关系 (见表 1-1) 可以表示为学生 (学号, 姓名, 性别, 出生日期, 籍贯)。表中的每一行是一条学生记录, 是关系的一个元组, 学号、姓名、性别、出生日期、籍贯均是属性。其中学号是唯一标识一条记录的属性, 因此称为主码。对于学号属性, 域是 0000000001~9999999999; 对于姓名属性, 域是由 2~4 个汉字组成的字符串; 对于性别属性, 域是男、女。

一个关系模式在某一时刻的内容 (称为相应模式的状态) 是元组的集合, 称为关系。在不至于引起混淆的情况下, 常常将关系模式和关系系统称为关系。

1.2.2 关系运算

从一个关系中访问所需要的数据时, 就需要对这个关系进行一定的关系运算。关系运算的操作对象是关系, 运算的结果仍为关系。关系数据库支持的 3 种基本关系运算为: 选择、投影和连接。

1. 选择运算

选择是对一个关系表的元组进行选择, 把符合某个条件的记录集选择出来, 并重新构建新的关系表, 新表的结构与原表一样。也就是说, 选择运算是在二维表中选择满足指定条件的行。例如, 在学生表中, 若要找出所有女生的元组, 就可以使用选择运算来实现。

2. 投影运算

投影是对一个关系表的属性进行选择, 消去表中某些属性, 并可以按要求重新安排属性的次序, 新关系中的属性个数要少于或等于原关系的属性个数。例如, 在学生表中, 若要显示所有学生的学号、姓名和性别, 就可以使用投影运算来实现。

3. 连接运算

连接是按照两个关系表中相同属性间的一定条件对两个关系表中的记录进行有选择的连接, 从而形成新的关系表。表的结构可取原来两个表中的属性, 记录取原来两个表中满足连接条件的记录。连接运算将两个关系模式拼接成一个更宽的关系模式。

选择和投影运算的操作对象只是一个表, 相当于对一个二维表进行切割。连接运算需要至少两个表作为操作对象。如果需要连接两个以上的表, 应当两两进行连接。

1.3 Visual FoxPro 简介

1.3.1 Visual FoxPro 的发展简史

Visual FoxPro 6.0 的发展起源于 XBASE 系列数据库管理系统。早期有 Dbase、FoxBASE 和 Foxpro 等数据库管理系统, 这些系统都曾在我国得到了广泛的应用。

Dbase 系列是 1983 年, 由美国 Ashton-Tate 公司推出的产品。

FoxBASE 系列是 1987 年, 由美国 Fox Software 公司推出的产品, 并于 1989 年推出了 Foxpro 产品。

1992 年美国 Microsoft 公司收购 Fox Software 公司后, 相继推出 Foxpro 升级系列产品, 并于 1995 年开始推出 Visual FoxPro 系列产品。Visual FoxPro 6.0 版数据库管理系统是美国 Microsoft 公司 1998 年正式推出的新一代高级数据库管理系统, 并迅速以功能强大、操作简便而征服了大量用户, 在我国数据处理应用范畴内也得到了广泛的应用。本书的所有程序都是以 Visual FoxPro 6.0 为开发平台。目前 Visual FoxPro 的最高版本是 9.0。

1.3.2 Visual FoxPro 的特点

1. 支持面向对象的程序设计

VFP 6.0 不但支持面向过程的程序设计, 同时也支持面向对象的程序设计。不但允许用户自定义“对象”和“类”, 同时还提供了大量的预先定义的基类, 因此大大减少了用户的编程工作量。

2. 支持结构化查询语言 (SQL)

在 VFP 6.0 中 SQL 的命令已得到较大的扩充, 可以非常有效地访问索引文件中的数据, 使查询数据变得更简单、更迅速。

3. 提供多种可视化设计工具

在 VFP 6.0 中提供了向导、设计器和生成器三类可视化设计工具, 使用户能非常方便地完成各种查询和设计工作。

4. 提供 OLE (对象链接与嵌入) 功能

在 VFP 6.0 中可以与其他应用软件共享数据, 实现应用集成, 编辑其他软件的对象, 与其他软件之间进行数据传输等较为方便。

1.3.3 Visual FoxPro 的安装、启动与退出

1. VFP 6.0 的安装

VFP 需要运行在 Windows 95 及以上版本操作系统的计算机上。

VFP 系统一般都存放在光盘上, 并在系统盘上的 VFP 文件夹中有一个安装程序 Setup.exe, 只要启动该程序则进行系统安装向导, 根据安装向导完成整个安装过程。

2. VFP 6.0 的启动与退出

(1) VFP 6.0 的启动

启动 VFP 6.0 有多种方法, 通常采用以下方式之一。

① 从“开始”菜单启动: 选择“程序 | Microsoft Visual FoxPro 6.0”选项, 进入 Microsoft Visual FoxPro 系统。

② 以快捷方式启动: 在 Windows 操作系统的桌面上建立 VFP 快捷方式 (图标), 双击 VFP 图标启动。

(2) 退出 VFP 6.0

退出 VFP 6.0 系统可以使用以下方法之一。

① 在 Microsoft Visual FoxPro 主菜单下, 单击“文件 | 退出”选项。

② 按 Alt+F4 组合键。

③ 在 Microsoft Visual FoxPro 系统环境窗口, 单击标题栏右上角“关闭”按钮。

④ 在命令窗口中, 输入命令“QUIT”并按回车键。

1.4 Visual FoxPro 6.0 集成开发环境

1.4.1 Visual FoxPro 6.0 的主界面

启动 VFP 6.0 后的主界面如图 1-3 所示, 从图中可看出, 主界面主要由标题栏、菜单栏、工具栏、命令窗口、工作区、状态栏等组成。

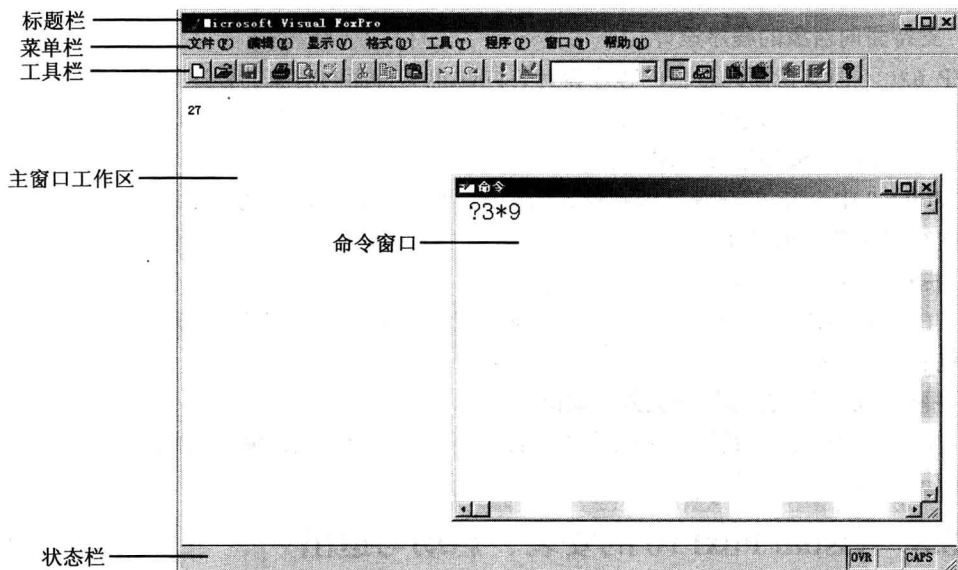


图 1-3 VFP 6.0 的主界面

1. 标题栏

标题栏的左端显示 Microsoft Visual FoxPro 系统特有的图标和使用软件名称, 标题栏的右上方依次排列有最小化按钮、还原/最大化按钮和关闭按钮。

2. 菜单栏

VFP 6.0 主窗口的菜单栏主要由“文件”、“编辑”、“显示”、“格式”、“工具”、“程序”、“窗口”和“帮助”8个下拉菜单组成。

3. 工具栏

VFP 6.0 系统提供了 11 种工具栏, 用户可通过单击“显示 | 工具栏...”选项, 决定哪些工具栏需要在主窗口中显示。在默认情况下, “常用”工具栏会随系统启动时一起打开, 并显示在菜单栏下面。同时, VFP 6.0 能够记忆工具栏的位置, 再次进入 VFP 6.0 时, 工具栏将位于前次退出时所在的位置上。

4. 主窗口工作区

主窗口工作区位于菜单栏和工具栏下面, 主要用于显示命令或程序执行的结果, 同时也用来显示打开的各种工具栏、对话框和其他窗口。例如, 当用户在“命令”窗口中输入“? 3*9”并按下 Enter 键之后, 在主窗口工作区中将会显示其结果“27”。

5. 命令窗口

“命令”窗口位于主窗口工作区内,“命令”窗口不仅可用来显示用户从键盘发出的命令,还可显示在界面操作时与之对应的操作命令。

若用户想清除主窗口工作区的内容,只需在“命令”窗口键入并执行“CLEAR”命令;若用户通过界面操作创建一个表单时,在“命令”窗口中将会显示对应的操作命令“CREATE FORM”。同时,系统提供了对“命令”窗口所有显示文本命令的剪切、复制、粘贴、清除等操作。另外,“命令”窗口还具有命令记忆功能,即用户可通过使用“命令”窗口右侧的滚动条,或用键盘中的上、下光标移动键能把光标移至曾执行过的某个命令上。这不仅可用于命令的查看、重复执行,而且对纠正错误、调试程序是非常有用的。

在默认情况下,“命令”窗口会随系统启动时一起打开,有下列 3 种方法可以显示或隐藏“命令”窗口。

① 单击命令窗口右上角的关闭按钮可将其关闭,通过单击“窗口 | 命令窗口”命令可以重新打开。

② 单击“常用”工具栏上的“命令窗口”按钮。按下则显示,弹起则将其隐藏。

③ 按 Ctrl+F4 组合键隐藏命令窗口,按 Ctrl+F2 组合键显示命令窗口。

6. 状态栏

状态栏是位于 VFP 6.0 窗口底部的水平区域,用来提供系统在不同操作下对应的状态。例如,单击“文件 | 打开”命令时,状态栏将显示“打开已有文件”。

状态栏右边有 3 个方格,左格表示当前是否处于插入方式,若是为空白,否则显示 OVR,由 Insert 键控制;中格表示小键盘是否处于数字方式,若是显示 Num,否则为空白,由 NumLock 键控制;右格表示键盘是否处于大写字母方式,若是显示 Caps,否则为空白,由 CapsLock 键控制。

1.4.2 配置 Visual FoxPro 6.0 的环境

VFP 6.0 的配置决定其外观和行为。安装完 VFP 6.0 之后,系统自动用一些默认值来设置环境,为了使系统能满足个性化的要求,也可以定制自己的系统环境。环境设置包括主窗口标题、默认目录、项目、编辑器、调试器及表单工具选项、临时文件存储、拖放字段对应的控件和其他选项等内容。例如,可以建立 VFP 6.0 所用文件的默认位置,指定如何在编辑窗口中显示源代码及日期和时间的格式等。

VFP 6.0 可以使用“选项”对话框或 SET 命令进行附加的配置设定,还可以通过配置文件进行设置。在此仅介绍使用“选项”对话框进行设置的方法。

1. 使用“选项”对话框

单击“工具 | 选项”命令,弹出“选项”对话框。“选项”对话框中包括有一系列代表不同类别环境的选项卡(共 12 个)。各个选项卡及其设置功能如表 1-2 所示。

表 1-2 “选项”对话框中的选项卡及其功能

选 项 卡	设 置 功 能
显示	设置界面,如是否显示状态栏、时钟、命令结果或系统信息等
常规	设置数据输入和程序设计选项,如设置警告声音,调色板使用的颜色等
数据	对数据库进行设置,如是否使用 Rushmore 优化,是否使用索引强制唯一性,备注块大小,查找的记录计数器间隔以及使用什么锁定选项等
远程数据	设置对远程数据的访问,如连接超时限定值、一次获取记录数目以及如何使用 SQL

续表

选项卡	设置功能
文件位置	设置文件的位置, 如 VFP 默认目录、帮助文件以及辅助文件的存储位置
表单	设置表单设计器, 如网格面积, 所用的刻度单位等
项目	设置项目管理器, 如是否提示使用向导, 双击时运行或修改文件以及源代码管理等
控件	设置“表单控件”工具栏中的“查看类”按钮所提供的可视类库和 ActiveX 控件选项
区域	设置日期、时间、货币及数字的格式
调试	设置调试器显示及跟踪选项, 如使用什么字体与颜色
语法着色	设置程序元素所用的字体及颜色, 如注释与关键字
字段映像	设置从数据环境设计器、数据库设计器或项目管理器向表单拖放表或字段时创建何种控件类型

在各个选项卡中均可以采用交互的方式来查看和设置系统环境, 下面仅举几个常用的例子来说明。

(1) 设置日期和时间的显示格式

在图 1-4 所示的“区域”选项卡中, 可以设置日期和时间的显示方式, VFP 6.0 中的日期和时间有多种显示方式可供选择。例如, “美语”显示方式为 08/28/10 16:45:36 PM; “年月日”显示方式为 05/4/23 04:45:36 PM; “汉语”显示方式为 2010 年 08 月 28 日, 16:45:36。默认情况下, 系统显示的日期格式为“美语”。

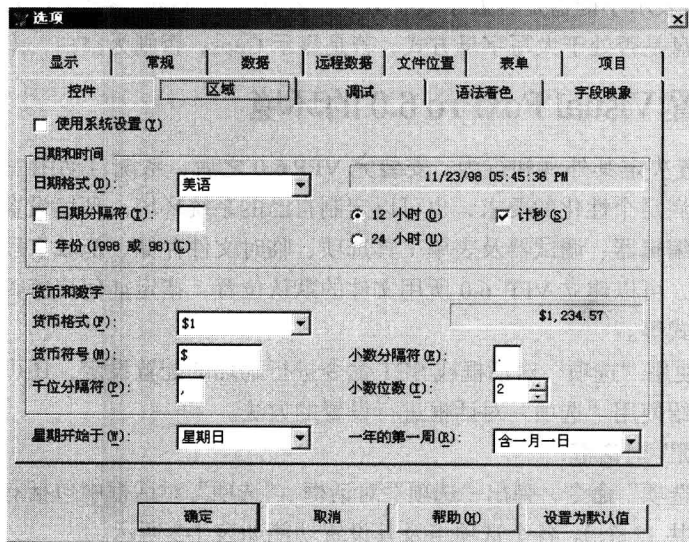


图 1-4 “选项 | 区域”选项卡

(2) 设置默认目录

为便于管理, 用户开发的应用系统应当与系统自有的文件分开存放, 需要事先建立自己的工作目录。在“选项”对话框中选择“文件位置”选项卡, 如图 1-5 所示。在文件类型中选中“默认目录”, 单击“修改”按钮, 或者直接双击“默认目录”, 弹出如图 1-6 所示的“更改文件位置”对话框。选中“使用默认目录”复选框, 激活“定位默认目录”文本框。然后直接输入路径, 或