

高职高专教材

GAOZHI GAOZHUAN JIAOCAI

# Visual FoxPro 数据库系统及应用教程

何安东 主编

程显林 主审



石油工业出版社  
Petroleum Industry Press

高职高专教材

# Visual FoxPro

## 数据库系统及应用教程

何安东 主 编  
程显林 主 审



石油工业出版社

## 内 容 提 要

本书以 Visual FoxPro 6.0 中文版为基本内容, 在结构设计上, 主要面向高职高专学生。通过大量例题和习题介绍 Visual FoxPro 6.0 中文版的基础知识及关系型数据库的基本概念、程序设计的方法。本书包括数据库基础与 Visual FoxPro 环境、Visual FoxPro 的表达式及组成元素、数据表的创建及基本操作、数据库的操作与查询、视图程序设计、表单设计基础、表单高级设计、报表设计、菜单设计, 并在全书最后一章给出一个系统开发实例, 以帮助读者巩固学习成果, 把学到的知识加以应用和实践。在附录部分给出一套模拟试题, 希望对读者参加全国或省级计算机等级考试有所帮助。

本书可作为职业院校相关课程的教材, 也可以作为各类各级计算机培训班的教材以及计算机应用人员的参考书。

### 图书在版编目 (CIP) 数据

Visual FoxPro 数据库系统及应用教程/何安东主编.

北京: 石油工业出版社, 2008. 2

高职高专教材

ISBN 978-7-5021-6407-2

I. V...

II. 何...

III. 关系数据库-数据库管理系统, Visual FoxPro-高等学校:  
技术学校-教材

IV. TP311.138

中国版本图书馆 CIP 数据核字 (2007) 第 196467 号

---

出版发行: 石油工业出版社

(北京安定门外安华里 2 区 1 号 100011)

网 址: [www.petropub.com.cn](http://www.petropub.com.cn)

编辑部: (010) 64523574 发行部: (010) 64523620

经 销: 全国新华书店

印 刷: 中国石油报社印刷厂

---

2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

787×1092 毫米 开本: 1/16 印张: 14.25

字数: 365 千字

---

定价: 23.00 元

(如出现印装质量问题, 我社发行部负责调换)

版权所有, 翻印必究

# 前 言

高级语言程序设计已由过去的面向过程的程序设计逐步发展成面向对象的程序设计，各种可视化程序设计语言越来越受到广大计算机应用工作者的喜爱。其中在这些高级语言中，Visual FoxPro 是比较理想的入门语言，特别适合初学者使用。在数据库应用领域中，Visual FoxPro 6.0 中文版适用于小型关系型数据库管理系统的开发，用它开发的应用程序在社会上较为流行。

2002 年教育部把 Visual FoxPro 6.0 列入计算机等级考试二级考试范围，部分省市把它作为非计算机专业学生计算机等级考试的重点科目。本书在例题、习题的选编上，充分考虑到这一点，并在附录中给出一套完整考题，以帮助读者掌握难度、题型等。本书列举大量例题以利于读者理解所学内容，并给出一定量的习题。由于本书还另有配套习题、实训教材，所以有关实训内容没做安排，仅仅在最后一章有所体现。

本书按照循序渐进、由浅入深的原则，以 Visual FoxPro 6.0 中文版数据库管理系统为平台，介绍了数据库基础与 Visual FoxPro 6.0 环境、Visual FoxPro 的表达式及组成元素、数据表的创建及基本操作、数据库的操作与查询、视图程序设计、表单设计基础、表单高级设计、报表设计、菜单设计，并在最后给出一个系统开发实例，供读者参考练习。

本书由何安东、庄贵霞、赵丽峰、王敬山、杨楷共同编写。第一、二章由王敬山编写，第三、四章由庄贵霞编写，第五章由杨楷编写，第六、七、八章由何安东编写，第九、十章由赵丽峰编写。本书由程显林主审，由何安东统稿。

由于编者水平有限，书中难免有错误和不妥之处，敬请读者批评指正。

编 者  
2007 年 11 月

# 目 录

<b>第一章 数据库基础与 Visual FoxPro 环境</b> .....	1
<b>第一节 数据库的基本概念</b> .....	1
一、常用术语 .....	1
二、数据模型 .....	1
<b>第二节 关系模型的相关运算与范式</b> .....	4
一、关系模型的相关运算 .....	4
二、范式 .....	4
<b>第三节 Visual FoxPro 环境</b> .....	5
一、Visual FoxPro 的特点 .....	5
二、Visual FoxPro 操作界面 .....	6
三、项目管理器 .....	8
习题 .....	9
<b>第二章 Visual FoxPro 的表达式及组成元素</b> .....	10
<b>第一节 常量</b> .....	10
一、字符型常量 .....	10
二、数值型常量 .....	10
三、日期型常量和日期时间型常量 .....	10
四、逻辑型常量 .....	10
五、货币型常量 .....	10
<b>第二节 变量</b> .....	11
一、内存变量及操作 .....	11
二、字段变量及操作 .....	11
<b>第三节 数组</b> .....	11
一、数组的定义 .....	11
二、数组的赋值和引用 .....	12
<b>第四节 函数</b> .....	13
一、数学函数 .....	14
二、字符串函数 .....	16
三、日期与时间函数 .....	20
四、类型转换函数 .....	21
五、数据库文件操作函数 .....	24
六、其他函数 .....	30
<b>第五节 表达式</b> .....	31
一、数值型表达式 .....	32
二、字符型表达式 .....	32

三、日期（时间）型表达式 .....	33
四、关系型表达式 .....	33
五、逻辑型表达式 .....	34
六、名称型表达式 .....	34
七、宏替换 .....	35
八、表达式的优先级 .....	36
习题 .....	36
<b>第三章 数据表的创建及基本操作 .....</b>	<b>38</b>
<b>第一节 自由表的建立 .....</b>	<b>38</b>
一、表结构的设计与建立 .....	38
二、向表中输入记录 .....	44
<b>第二节 表的编辑修改 .....</b>	<b>46</b>
一、表文件的打开与关闭 .....	46
二、表结构的修改 .....	47
三、记录的显示和修改 .....	48
四、记录指针与定位 .....	50
五、记录的删除与恢复 .....	52
<b>第三节 表的复制 .....</b>	<b>54</b>
<b>第四节 排序与索引 .....</b>	<b>54</b>
一、排序 .....	54
二、索引 .....	55
<b>第五节 统计命令 .....</b>	<b>61</b>
一、记录个数统计 .....	61
二、记录数据求和 .....	61
三、求记录数据平均值 .....	62
四、数据分类汇总 .....	62
习题 .....	63
<b>第四章 数据库的操作、视图与查询 .....</b>	<b>65</b>
<b>第一节 数据库的基本操作 .....</b>	<b>65</b>
一、建立数据库 .....	65
二、数据库的基本操作 .....	66
三、数据字典的使用 .....	69
<b>第二节 使用多个数据表 .....</b>	<b>73</b>
一、多工作区的选择与操作 .....	73
二、数据工作期 .....	75
三、建立表间关联 .....	75
<b>第三节 SELECT - SQL 查询 .....</b>	<b>76</b>
一、用 SELECT - SQL 命令查询 .....	76
二、用查询设计器建立查询 .....	83
三、查询结果的图形处理 .....	87

第四节 视图的建立与使用 .....	89
一、用命令建立视图 .....	89
二、用视图设计器建立本地视图 .....	90
三、创建远程视图 .....	91
四、视图的操作 .....	92
习题 .....	94
<b>第五章 程序设计 .....</b>	<b>96</b>
第一节 程序文件 .....	96
一、程序文件的建立与执行 .....	96
二、程序中的辅助命令 .....	97
三、输入、输出命令 .....	98
第二节 程序的控制结构 .....	100
一、顺序结构 .....	100
二、选择结构 .....	101
三、循环结构 .....	105
第三节 多模块程序设计 .....	108
一、子程序 .....	109
二、过程及过程文件 .....	111
三、自定义函数 .....	113
四、变量的作用域 .....	114
习题 .....	116
<b>第六章 表单设计基础 .....</b>	<b>118</b>
第一节 表单向导 .....	118
第二节 表单设计器 .....	121
一、用表单设计器设计表单的基本步骤 .....	122
二、表单的保存、运行和修改 .....	123
三、表单设计可用的工具栏 .....	125
第三节 面向对象的程序设计方法 .....	125
一、基本概念 .....	125
二、对象引用 .....	128
三、控件的选择、移动、复制、删除及排列 .....	128
第四节 常用控件介绍 .....	131
一、标签控件 (Label) .....	131
二、计时器控件 (Timer) .....	131
三、命令按钮控件 (Command) .....	132
四、图像控件 (Image) .....	134
五、线条控件 (Line) .....	134
六、形状控件 (Shape) .....	134
七、页框控件 (PageFrame) .....	135
八、微调控件 (Spinner) .....	136

九、文本框控件 (Text)	137
十、命令按钮组控件 (CommandGroup)	137
十一、编辑框控件 (EditBox)	140
十二、复选框控件 (CheckBox)	141
十三、选项按钮组控件 (OptionGroup)	142
十四、列表框控件 (ListBox)	144
十五、组合框控件 (ComboBox)	145
十六、表格控件 (Grid)	146
十七、容器控件 (Container)	150
十八、超级链接控件 (Hyperlink)	150
习题	151
<b>第七章 表单高级设计</b>	<b>154</b>
第一节 单文档界面与多文档界面	154
一、应用程序界面	154
二、单文档界面与多文档界面的实现	154
三、顶层表单或子表单的设定	154
四、浮动表单的设定	154
五、子表单最大化的样式	154
第二节 MDI 应用程序	155
一、表单的显示与隐藏	155
二、MDI 应用程序的运行	155
三、MDI 窗口之间的参数传递	155
第三节 表单集	157
一、表单集的创建	157
二、表单集的删除	158
三、表单集的释放	158
四、表单集的编辑	158
<b>第八章 报表设计</b>	<b>160</b>
第一节 创建报表	160
一、使用报表向导创建报表	160
二、使用报表设计器创建报表	162
三、使用快速报表创建报表	164
第二节 报表综合举例	165
第三节 打印报表	168
习题	168
<b>第九章 菜单设计</b>	<b>170</b>
第一节 菜单	170
第二节 快速菜单和快捷菜单	173
一、快速菜单	173
二、快捷菜单	174



第三节	常规选项对话框和菜单选项对话框·····	175
一、	常规选项·····	176
二、	菜单选项·····	177
第四节	顶层表单的菜单加载·····	177
习题	·····	178
<b>第十章</b>	<b>系统开发实例</b> ·····	<b>179</b>
第一节	开发 Visual FoxPro 数据库应用系统的一般步骤·····	179
一、	总体规划·····	179
二、	系统分析·····	180
三、	系统设计·····	180
四、	系统实施·····	180
五、	运行与维护·····	182
第二节	“学生管理系统”的开发·····	183
一、	系统分析·····	183
二、	数据库设计·····	184
三、	应用程序设计·····	185
四、	装载数据·····	207
五、	应用程序的建立·····	207
习题	·····	207
<b>附录</b>	<b>2005 年 11 月黑龙江省高校非计算机专业学生计算机等级考试试卷</b> ·····	<b>210</b>
	<b>标准答案</b> ·····	<b>218</b>
<b>参考文献</b>	·····	<b>219</b>

# 第一章 数据库基础与 Visual FoxPro 环境

## 第一节 数据库的基本概念

数据库技术是 20 世纪 60 年代末使用的一种管理方法，是计算机科学的重要分支，近年来得到迅速发展。如今数据库技术已成为先进信息技术的重要组成部分，是现代计算机信息系统和计算机应用系统的基础和核心。本章主要介绍数据库的基本概念、常用术语和数据模型及微机中使用的 Visual FoxPro 数据库管理系统环境操作界面和项目管理器的使用等。

### 一、常用术语

#### 1. 数据

在日常生活中，数据是用来记录、描述事物特征的一种符号化表示，它是可存储在媒体上并能够被识别的物理符号。

#### 2. 信息

信息是以数据为载体的对事物、事件和概念的抽象反映。它是被加工为特定形式的数据。信息也是事物属性与事物间关系的表现，通过信号、消息、情报与数据等形式进行传播与交流。只有当信息的接收者觉得信息具有新内容时，信息才发生作用。

#### 3. 数据处理

数据要经过处理才能变为信息。数据处理是将数据转变成信息的过程，是指对数据进行收集、整理、存储、加工及传播等一系列活动的总和。

#### 4. 数据库

数据库 (Data Base, DB) 是以一定的组织方式存储在计算机存储设备上的相关的数据的集合。它不仅包括描述事务特征的数据，而且还包括相关事务之间的关系。

#### 5. 数据库管理系统

数据库管理系统 (Data Base Management System, DBMS) 是介于用户与操作系统之间的用于进行数据库管理的系统软件。借助于数据库管理系统，用户能够方便地建立、使用和维护数据库。

#### 6. 数据库系统

数据库系统是指在计算机系统中引入了数据库后的系统构成，是实现有组织地、动态地存储大量相关数据，提供数据处理和信息资源共享的便利手段。数据库系统一般由数据库、数据库管理系统 (含有关开发工具、环境)、应用系统、数据库管理员和用户共同构成。

### 二、数据模型

现实世界中客观存在并且相互区别的事物称为实体。实体可以是具体的人、事、物 (例

如：一名学生，一个班级，或一台计算机等），也可以是抽象的概念或事件（例如：学习，供求）。

同类型实体的集合称为实体集。例如，学校全体教职工构成一个学校的教职工实体集，学校全体学生构成一个学校的学生实体集。

实体的特征称为属性，属性是实体之间相互区别的标志，一个实体可以由若干个属性来刻画。例如，教职工实体可以用职工编号、姓名、性别、出生日期和职称等属性来描述。

实体之间的对应关系称为联系，它反映了现实世界各个事物之间的相互关系。实体之间的联系有以下 3 种类型：

(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（也可以没有）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，一个班级只有一个班长，而班长只在一个班级中任职，则班级与班长之间具有一对一联系，联系名设定为“班级班长”。

(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，一个班级中有若干名学生，而每名学生只在一个班级中，则班级与学生之间具有一对多联系，联系名设定为“班级学生”。

(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个教师讲授，而一个教师可以同时讲授多门课程，则教师与课程之间具有多对多联系，联系名设定为“授课”。

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。

模型是现实世界特征的模拟和抽象，而数据模型 (Data Model) 是现实世界数据特征的抽象。在数据库中用数据模型来抽象、表示和处理现实世界中的数据和信息。

数据模型应满足三方面要求：①能比较真实地反映现实世界；②容易为人所理解；③便于在计算机上实现。

不同的数据模型实际上是提供模型化数据和信息不同工具。根据模型应用的不同目的，可以将这些模型划分为两类，分属于两个不同的层次。

第一类：概念模型，也称信息模型，按用户的观点对数据和信息建模，主要用于数据库设计。

第二类：数据模型，主要包括网状模型、层次模型、关系模型等，它是按计算机系统的观点对数据建模，主要用于 DBMS 的实现。

层次模型用树形结构来表示各类实体以及实体间的联系。满足下面两个条件的基本层次联系的集合为层次模型：

(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) 根结点以外的其他结点有且只有一个双亲结点。

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。

在层次模型中，同一双亲的子女结点称为兄弟结点，没有子女结点的结点称为叶结点。

网状模型用网形结构来表示各类实体间的联系。把满足以下两个条件的基本层次联系集合称为网状模型：

- (1) 允许一个以上的结点无双亲；
- (2) 一个结点可以有多个的双亲。

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（称之为复合联系）。

网状模型中每个结点表示一个实体集，结点间的连线表示实体集之间一对多的父子联系。

在网状模型中，子女结点与双亲结点的联系可以不唯一。因此，要为每个联系命名，并指出与该联系有关的双亲记录和子女记录。

关系模型中数据的逻辑结构是一张二维表，它由行和列组成。关系模型是目前最重要的一种模型。

关系 (Relation)：一个关系对应通常说的一张二维表。例如，表 1-1 中的这张学生成绩表就是一个关系，可以命名为“成绩”关系，也可称为成绩表。

表 1-1 学生成绩表

学号	姓名	班级	计算机基础	英语	高等数学	C 语言
111111	赵洋	电子商务 02-02	68	79	98	77
111112	李林	电子商务 02-02	76	66	88	69
111178	关新	电子商务 02-02	61	87	70	81
111765	王平南	电子商务 02-01	87	89	94	90
111777	张彦强	电子商务 02-01	62	65	50	48
111345	李丰瑞	电子商务 02-01	71	75	37	63

元组 (Tuple)：表中的一行即为一个元组，在 Visual FoxPro 等数据库管理系统中被称为一个记录。例如，表 1-1 有 6 行，对应 6 个元组，即 6 条记录。

属性 (Attribute)：表中的一列即为一个属性，给每一个属性起一个名称即属性名，在 Visual FoxPro 等数据库管理系统中被称为字段名。例如，表 1-1 中有 7 列，对应的 7 个属性分别为学号、姓名、班级、计算机基础、英语、高等数学、C 语言。

主码 (Key)：主码是表中的某个属性组，它可以唯一确定一个元组，在 Visual FoxPro 等数据库管理系统中被称为关键字段。例如，表 1-1 中的学号可以唯一确定一个学生，也就可以作为“成绩”关系的主码。

域 (Domain)：属性的取值范围。例如，各科成绩的域是 0~100，班级的域是一个学校所有班级名称集合。

分量：元组中的一个属性值，在 Visual FoxPro 等数据库管理系统中被称为字段值。例如，表 1-1 中的第一个元组在学号属性上的取值为 111111，则 111111 就是第一个元组的一个分量。

关系模式：对关系的描述，一般表示为：

关系名 (属性 1, 属性 2, ……，属性  $n$ )

例如，表 1-1 中的学生关系可描述为：

成绩(学号,姓名,班级,计算机基础,英语,高等数学,C语言)在关系模型中,实体以及实体间的联系都是用关系来表示的。

## 第二节 关系模型的相关运算与范式

### 一、关系模型的相关运算

关系数据库是采用关系模型构造的一种数据库,是若干相关的二维表的集合。在数据库设计完成后,我们最终目的是要对其进行使用,同时也需要对数据进行处理和维护,而所特有对数据库的操作都是基于关系数据库的3种基本关系运算:即选择、投影和联接。

选择是指依据一定的条件来选取若干条记录,即选取二维表格中的若干行的操作。在 Visual FoxPro 中可通过“FOR<条件>”、“WHILE<条件>”等形式来实现。

投影是指从一个二维表中选取若干列(字段)的操作。在 Visual FoxPro 中可通过“FIELD<字段1,字段2,……>”等形式来实现。

联接是从两个或多个二维表中按某种条件提取部分或全部记录及部分或全部字段的操作。联接可以实现两个或多个二维表的横向合并。在 Visual FoxPro 中可通过带有 JOIN 的 SQL 语言来实现。

### 二、范式

为使数据库发挥更优化的性能,设计构造数据库时要遵循一定的准则。在关系数据库中,这种准则称为范式。范式是符合某一种级别关系模式的集合。关系数据库中的关系必须满足一定的要求,满足不同级别要求的就属于不同级别的范式。我们只介绍第一范式(1NF)、第二范式(2NF)和第三范式(3NF)。

#### 1. 第一范式(1NF)

第一范式是指数据库表的每一列都是不可分割的基本数据项。同一列中不能有多值,即实体中的某个属性不能有多值或者不能有重复的属性。如果出现重复的属性,就可能需要定义一个新的实体,新的实体由重复的属性构成,新实体与原实体之间为一对多关系。在第一范式中表的每一行只包含一个实例的信息。简而言之,第一范式就是无重复的列。

在任何一个关系数据库中,第一范式是最基本的要求。

#### 2. 第二范式(2NF)

第二范式是在第一范式的基础上建立起来的,即满足第二范式必须先满足第一范式。第二范式要求数据库表中的每个实例或行必须可以被唯一地区分。为实现区分通常需要为表加上一个列,以存储各个实例的唯一标识。这个唯一属性列被称为主关键字或主键、主码。

第二范式要求实体的属性完全依赖于主关键字。所谓完全依赖是指不能存在仅依赖主关键字一部分的属性,如果存在,那么这个属性和主关键字的这一部分应该分离出来形成一个新的实体,新实体与原实体之间是一对多的关系。为实现区分通常需要为表加上一个列,以存储各个实例的唯一标识。简而言之,第二范式就是非主属性非部分依赖于主关键字。

### 3. 第三范式 (3NF)

满足第三范式必须先满足第二范式。简而言之，第三范式要求一个数据库表中不包含在其他表中已包含的非主关键字信息。例如，存在一个学生成绩表，其中每个学生有专业编号 (dept\_id)、专业名称、专业简介等信息。那么在学生成绩表中列出专业编号后就不能再将专业名称、专业简介等与部门有关的信息再加入学生成绩表中。如果不存在专业信息表，则根据第三范式也应该构建它，否则就会有大量的数据冗余。简而言之，第三范式就是属性不依赖于其他非主属性。

## 第三节 Visual FoxPro 环境

### 一、Visual FoxPro 的特点

Visual FoxPro 是为数据库结构和应用程序开发而设计的功能强大的面向对象的环境。无论是组织信息、运行查询、创建集成的关系型数据库系统，还是为最终用户编写功能全面的数据管理应用程序，Visual FoxPro 都可以提供管理数据所需的工具，可以在应用程序或数据库开发的任何一个领域中提供帮助。Visual FoxPro 所具有的速度、能力和灵活性，是普通数据库管理系统无法比拟的，它把我们带入一个 XBase 新时代。

#### (1) 更简便、快速、灵活的应用程序开发。

Visual FoxPro 添加了新的“应用程序向导”，其提供的新的 ProjectHook 对象和改进了的应用程序框架功能可以使我们的应用程序更有效率。Visual FoxPro 6.0 中还添加了一些功能来增强开发环境，以便更容易地向应用程序中添加有效的功能。Visual FoxPro 6.0 提供了更多更好的生成器、工具栏和设计器等，在它们的帮助下，我们可以快速开发应用程序。在 Visual FoxPro 6.0 中，我们可以更简便地调试及监控应用程序的组件：可以跟踪事件以及记录执行代码的工具，它看起来与 Visual C 的调试环境很相似；利用它可以深入程序，查看属性设置值、对象以及数组元素的值；为显示交互的信息或代码中的信息提供了方便的途径，同时可以把结果输出到应用程序窗口之外的另一个窗口；可以用来分析程序、应用程序或实际运行的项目代码。

#### (2) 不用编程创建界面。

应用程序组件的实例中收集了一系列应用程序组件，它们显示了如何利用 Visual FoxPro 的特性解决现实世界的问题。在应用程序中可以直接使用示例组件及它们的代码。

#### (3) 提供真正的面向对象程序设计。

Visual FoxPro 仍然支持标准的面向过程的程序设计方式，但更重要的是它现在提供真正的面向对象程序设计的能力。借助 Visual FoxPro 的对象模型，可以充分使用面向对象程序设计的所有功能，包括继承性、封装性、多态性和子类。

#### (4) 使用优化应用程序的 Rushmore 技术。

Rushmore 是一种从表中快速地选取记录集的技术，它可将查询响应时间从数小时或数分钟降低到数秒，可以显著地提高查询的速度。

#### (5) 可以与多个开发者一齐工作。

如果是几个开发者开发一个应用程序，可以使用 Visual FoxPro 允许同时访问数据库组

件的能力。同时，若要跟踪或保护对源代码的更改，还可以使用带有“项目管理器”的源代码管理程序。

(6) 充分利用已有数据。

如果有 Visual FoxPro 先前版本的文件，只要打开它们，就会出现 Visual FoxPro 转换对话框。可以把其他数据源移到 Visual FoxPro 表中，如果有电子表格或文本文件中的数据，比如 Microsoft Excel 及 Microsoft Word，使用 Visual FoxPro 可以方便地实现数据共享。

(7) 可以开发客户/服务器解决方案，增强客户/服务器性能。

(8) 多语言编程。

由于 Visual FoxPro 支持英语、冰岛语、日语、朝鲜语、繁体汉语以及简体汉语多种语言的字符集，因此 Visual FoxPro 能在几个领域提供对国际化应用程序开发的支持。

和以往的 XBase 系列数据库相比，在 FoxPro 系列数据库管理系统中，完善了关系型数据库的概念，采用了 Rushmore 技术，引入了 SQL 命令；支持多种数据交换格式；采用了可视化的面向对象的程序设计方式；提供了功能完善的集成环境和丰富的开发工具。

## 二、Visual FoxPro 操作界面

启动 Visual FoxPro 6.0 后，我们会看到如图 1-1 所示的画面。

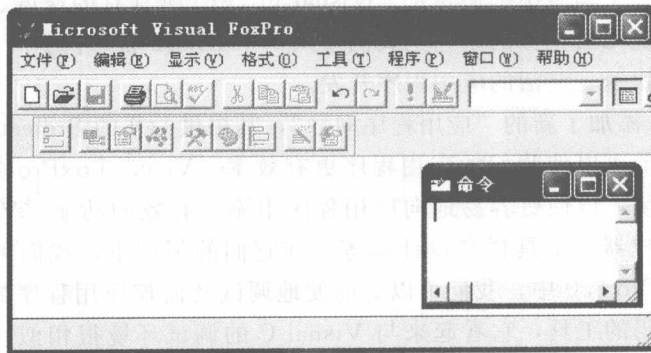


图 1-1 Visual FoxPro 用户界面

界面基本组成元素：标题栏、主菜单栏、工具栏、主窗口、命令窗口、状态栏。Visual FoxPro 6.0 提供了众多的工具，方便使用者使用和开发，所提供的工具见表 1-2。

表 1-2 系统工具一览表

菜 单	工 具 栏	窗 口	设 计 器	生 成 器	向 导
文件菜单	常用工具栏	命令窗口	数据库设计器	文本框生成器	文件菜单
编辑菜单	表单控制工具栏	浏览窗口	表设计器	组合框生成器	编辑菜单
显示菜单	布局工具栏	代码窗口	表单设计器	命令组生成器	显示菜单
格式菜单	调色板工具栏	调试窗口	菜单设计器	编辑框生成器	格式菜单
工具菜单	打印预览工具栏	编辑窗口	报表设计器	表达式生成器	工具菜单
程序菜单	报表控制工具栏	查看窗口	标签设计器	表单生成器	程序菜单
窗口菜单	查询设计器工具栏	跟踪窗口	类设计器	表格生成器	窗口菜单
帮助菜单	表单设计器工具栏	属性窗口	连接设计器	列表框生成器	帮助菜单
菜单菜单	报表设计器工具栏	通用字段窗口	查询和视图设计器	参照完整性生成器	菜单菜单

续表

菜 单	工 具 栏	窗 口	设 计 器	生 成 器	向 导
数据环境菜单	数据库设计器工具栏			自动格式生成器	数据环境菜单
表单菜单				选项组生成器	表单菜单
项目菜单					
查询菜单					
报表菜单					
表菜单					
数据库菜单					
类菜单					

Visual FoxPro 6.0 的工作方式有 3 种：

(1) 菜单操作方式。

根据所需的操作从菜单中选择相应的命令（与 WORD 类似）。每执行一次菜单命令，命令窗口中一般都会显示出与菜单对应的命令内容。利用工具菜单中的向导可以很方便地完成常规任务。

(2) 命令交互方式。

根据所要进行的各项操作，采用人机对话方式在命令窗口中按格式要求逐条输入所需命令，按回车后，机器逐条执行。

(3) 程序执行方式。

先在程序编辑窗口中编完程序，再从程序菜单中选择执行，或从命令窗口中输入 DO 命令，让机器执行。

Visual FoxPro 6.0 常用文件类型如表 1-3 所示。

表 1-3 Visual FoxPro 6.0 常用的文件扩展名及其关联的文件类型

扩 展 名	文 件 类 型	扩 展 名	文 件 类 型
.APP	生成的应用程序	.FRX	报表
.EXE	可执行程序	.FRT	报表备注
.PJX	项目	.LBX	标签
.PJT	项目备注	.LBT	标签备注
.DBC	数据库	.PRG	程序
.DCT	数据库备注	.FXP	编译后的程序
.DCX	数据库索引	.ERR	编译错误
.DBF	表	.MNX	菜单
.FPT	表备注	.MNT	菜单备注
.CDX	复合索引	.MPR	生成的菜单程序
.IDX	单索引	.MPX	编译后的菜单程序
.QPR	生成的查询程序	.VCX	可视类库
.QPX	编译后的查询程序	.VCT	可视类库备注
.SCX	表单	.TXT	文本
.SCT	表单备注	.BAK	备份文件



新建各种类型的文件时，可以利用系统提供的相应工具，以提高工作效率。新建文件时也可用相应的设计器和向导。

### 三、项目管理器

在 Visual FoxPro 中，项目是一种文件，用于跟踪创建应用系统所需要的所有程序、表单、菜单、库、报表、标签、查询和一些其他类型的文件。

“项目管理器”是 Visual FoxPro 中处理数据和对象的主要组织工具，是 Visual FoxPro 的“控制中心”。项目是文件、数据、文档和 Visual FoxPro 对象的集合，其保存文件带有 .PJX 扩展名，项目备注的扩展名为：.PJT。通过项目管理器，我们可以对项目及数据有更强的控制，能够使源代码得到更有效的管理，同时可以在“项目管理器”中看到组件的状态。数据库容器允许几个用户在同一个数据库中同时创建或修改对象。利用“数据库设计器”可以迅速更改数据库中对象的外观。我们可以借助“项目管理器”创建和集中管理应用程序中的任何元素；可以访问所有向导、生成器、工具栏和其他易于使用的工具。它提供了一个进行集中管理的环境。可以对 Visual FoxPro 系统本身定制工具栏，也能够为编写的应用程序定义工具栏。

项目管理器窗口：由数据项（全部、数据、文档、类、代码、其他）和命令按钮（新建、添加、修改、运行、移去、连编）组成（图 1-2）。

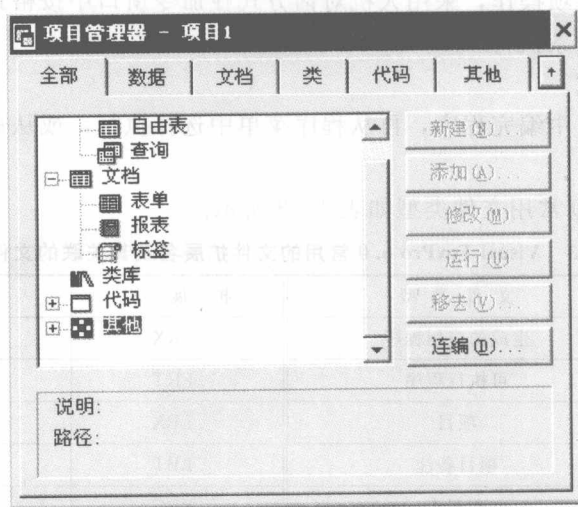


图 1-2 Visual FoxPro 项目管理器

项目菜单由 13 个子菜单（其中一部分与项目管理器中的命令按钮功能相同）组成。

项目管理器的功能是组织和管理所有与应用系统有关的各种类型的文件。

创建项目文件可以用菜单方式：打开文件菜单→新建→项目→新文件→给出合适的项目文件名及保存位置→进入项目管理器→开发应用系统→完成后关闭窗口退出。

创建项目文件也可以用命令方式：

语法：CREATE PROJECT<项目文件名>。

功能：打开项目管理器窗口进行应用系统的开发。

修改项目文件的菜单方式为：打开文件菜单→打开→找到所需的项目文件并确定→进入项目管理器→修改应用系统→完成后关闭窗口退出。