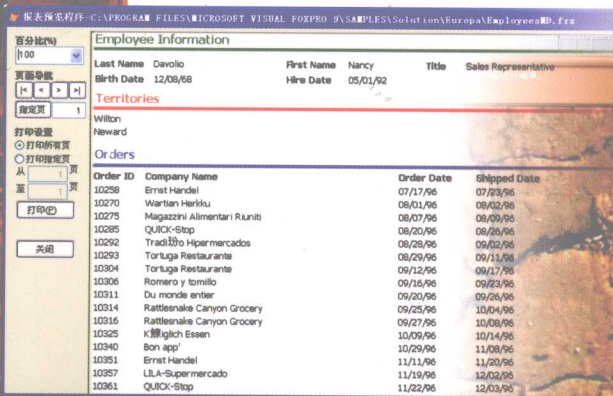
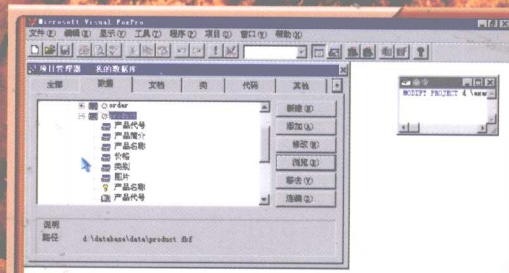


# 计算机应用 系列教材

丁玉琦 主编

# Visual FoxPro 9.0

# 教程



Order ID	Company Name	Order Date	Shipped Date
10258	Ernst Handel	07/17/96	07/23/96
10270	Wartian-Heriku	08/01/96	08/02/96
10275	Magazzini Alimentari Riuniti	08/07/96	08/09/96
10285	QUICK-Stop	08/20/96	08/25/96
10292	Tradição Hipermercados	08/28/96	09/02/96
10293	Toruga Restaurante	08/29/96	09/11/96
10304	Toruga Restaurante	09/12/96	09/17/96
10306	Romero y tomillo	09/16/96	09/23/96
10311	Du monde entier	09/20/96	09/26/96
10314	Rattlesnake Canyon Grocery	09/25/96	10/04/96
10316	Rattlesnake Canyon Grocery	09/27/96	10/08/96
10325	Königlich Essen	10/09/96	10/14/96
10340	Bon app'	10/29/96	11/08/96
10351	Ernst Handel	11/11/96	11/20/96
10357	LEA-Supermercado	11/19/96	12/02/96
10361	QUICK-Stop	11/22/96	12/03/96

中国商业出版社

计算机应用系列教材

# Visual Foxpro 9.0 教程

主 编 丁玉琦  
副主编 赵承祖

中国商业出版社

图书在版编目(CIP)数据

Visual Foxpro 9.0 教程/丁玉琦主编. —北京:中国商业出版社,2008.8

ISBN 978-7-5044-6079-0

I.V… II.丁… III.关系数据库-数据库管理系统, Visual Foxpro 9.0-教材 IV.TP311.138

中国版本图书馆 CIP 数据核字(2008)第 132619 号

责任编辑:刘树林

中国商业出版社出版发行  
(100053 北京广安门内报国寺1号)  
新华书店总店北京发行所经销  
中国石油报社印刷厂印刷

\*  
787×1092毫米 16开 19.75 印张 444 千字  
2008年8月第1版 2008年8月第1次印刷  
定价:32.00元

\* \* \* \*

(如有印装质量问题可更换)

中国商业出版社



# 编审说明

根据当前我国电子计算机普及和发展的新形势,为适应国家技能型、紧缺型、实用型人才培养工程的需要,我们组织商贸系统部分高等职业技术学院和重点中等职业学校中具有丰富理论与实践经验,并多年从事计算机及应用专业教学的教授、高级讲师、讲师、工程师编写了这套计算机应用系列教材。

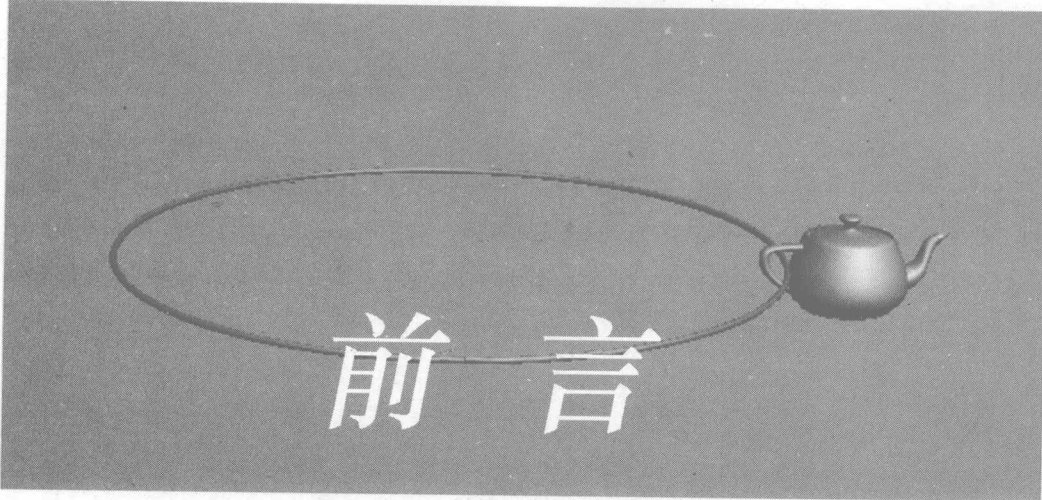
本系列教材面向 21 世纪,突出创新和发展,适应新世纪对人才培养高起点的要求,吸收计算机发展的最新成果。体现了科学性、先进性、理论性与普及性、应用性、操作性相结合的原则,做到了理论联系实际,内容翔实,结构严谨,体系合理,是一套较为实用的计算机应用系列教材。

本系列教材的读者对象,以高职为主,同时兼顾了中职教育和社会培训、等级考核的需要,也可以供职工大学、电视大学和高等专科学校使用。

本系列教材编写过程中,得到了有关院校领导和教师的大力支持,在此一并致谢。鉴于计算机飞速发展的实际,本系列教材难免有疏漏和不足之处,敬请广大读者赐教,使之日臻完善。

计算机应用系列教材编委会

2008 年 5 月



# 前言

Visual FoxPro 9.0 是由 Microsoft 公司发布的一种功能强大的数据库管理系统,它比以前版本更快速、更灵活、更有效。Visual FoxPro 9.0 具有丰富的开发工具,支持面向对象的可视化编程技术,支持客户/服务器结构,具有简单、灵活、多样的数据交换手段,而且增加了 Internet 功能。通过汉化的 Visual FoxPro 9.0 系统为用户提供了更好的开发环境。

本书通过图、表、例题强化了知识的应用性和技能的可操作性,体现了新时期教育的特点,尤其是第 12 章通过介绍一个小型应用系统开发的实例,使得大家在学习了前几章的知识基础上,掌握了开发应用系统的方法。

本书由丁玉琦主编,赵承祖任副主编,参编人员有赵承祖(第 1 章)、程丽(第 2 章)、李梅(第 3 章)、傅岚岚(第 4 章)、付兰平(第 5 章)、么世敏(第 6 章)、郝健妹(第 7 章)、丁玉琦(第 8、10 章)、罗维(第 9 章)、贺贝(第 10、11 章)、陈耀清(第 12 章)。全书最后由丁玉琦统编定稿,陈耀清主审。

本书在编写过程中参考了不少专著和教材,得到了专家学者、学校领导和老师以及编委会的大力支持。在此一并表示感谢!

由于时间仓促,编者水平有限,书中疏漏之处,在所难免,祈望专家学者、使用本书的老师、同学和读者批评指正,帮助我们修改,使此书进一步完善。

编者  
2008.5



# 目 录

<b>第1章 概述</b> .....	(1)
1.1 数据库基础知识 .....	(1)
1.2 Visual FoxPro 9.0 的特点 .....	(4)
1.3 Visual FoxPro 9.0 基础知识 .....	(6)
1.4 Visual FoxPro 9.0 的安装 .....	(13)
<b>第2章 基本操作</b> .....	(16)
2.1 Visual FoxPro 9.0 窗口简介 .....	(16)
2.2 鼠标和键盘的使用 .....	(17)
2.3 对话框的操作 .....	(19)
2.4 工具栏的操作 .....	(21)
2.5 命令窗口的使用 .....	(22)
2.6 文件操作技术 .....	(24)
<b>第3章 项目管理器</b> .....	(27)
3.1 建立项目文件 .....	(27)
3.2 项目管理器界面 .....	(32)
3.3 数据管理 .....	(34)
<b>第4章 数据库的操作</b> .....	(38)
4.1 数据库的建立 .....	(38)
4.2 表的建立 .....	(42)
4.3 表结构的修改 .....	(52)
4.4 数据的输入 .....	(59)
4.5 数据的浏览 .....	(62)
4.6 数据的编辑和修改 .....	(71)
4.7 索引与排序 .....	(79)
4.8 关系型数据库的建立 .....	(86)
4.9 关系型数据库的一致性 .....	(90)

<b>第5章 查询</b> .....	(96)
5.1 使用“查询向导”建立查询 .....	(96)
5.2 使用查询设计器建立简单查询 .....	(100)
5.3 查询文件的操作 .....	(108)
5.4 建立复杂条件查询 .....	(111)
5.5 加入计算字段查询 .....	(114)
5.6 多个数据表的查询 .....	(115)
5.7 交叉表查询 .....	(119)
5.8 图表向导 .....	(122)
<b>第6章 视图操作</b> .....	(127)
6.1 使用视图向导建立本地视图 .....	(127)
6.2 使用视图设计器建立本地视图 .....	(132)
6.3 设置字段属性 .....	(140)
6.4 视图数据的更新 .....	(141)
6.5 参数化视图 .....	(145)
<b>第7章 结构化查询语言—SQL</b> .....	(149)
7.1 数据库的建立和删除 .....	(149)
7.2 表结构的建立、修改和删除 .....	(150)
7.3 表的建立、修改和删除 .....	(154)
7.4 视图的建立、修改和删除 .....	(155)
7.5 SQL 查询 .....	(156)
<b>第8章 报表的设计</b> .....	(160)
8.1 报表设计基础 .....	(166)
8.2 使用“极表向导”创建报表 .....	(167)
8.3 创建一对多报表 .....	(176)
8.4 快速报表 .....	(181)
8.5 使用报表设计器制作报表 .....	(183)
8.6 标签的设计 .....	(203)
8.7 报表/标签的预览和打印 .....	(208)
<b>第9章 表单设计</b> .....	(212)
9.1 面向对象的概念 .....	(212)
9.2 Visual FoxPro 基类介绍 .....	(214)
9.3 表单向导 .....	(216)
9.4 表单设计器 .....	(218)
9.5 创建单一表单 .....	(223)
9.6 添加表单控件 .....	(227)
<b>第10章 菜单设计</b> .....	(251)
10.1 创建菜单的方法及步骤 .....	(251)
10.2 快速菜单 .....	(252)

---

10.3	定制菜单	(256)
10.4	创建快捷菜单	(259)
<b>第 11 章</b>	<b>程序设计基础</b>	<b>(263)</b>
11.1	程序的基本概念	(263)
11.2	顺序结构	(265)
11.3	分支结构	(268)
11.4	循环结构	(272)
11.5	过程与过程文件	(276)
<b>第 12 章</b>	<b>小型应用系统开发</b>	<b>(281)</b>
12.1	应用系统开发概述	(281)
12.2	需求分析	(283)
12.3	概念模型、逻辑模型和物理模型设计	(286)
12.4	外模式设计	(290)
12.5	创建菜单和工具栏	(296)
<b>附录:</b>	<b>常用函数</b>	<b>(301)</b>





# 第1章 概述

本章主要介绍数据与信息、数据类型、常量与变量、表达式和空值处理的概念与内容,关系模型、关系型数据库、数据库管理系统和数据库系统的概念,关系运算的概念与内容, Visual FoxPro 9.0 的特点、安装。这些内容是学习 Visual FoxPro 9.0 的基础。

## 1.1 数据库基础知识

### 1.1.1 数据与信息

为了研究和把握客观事物,人们总是希望能及时地、准确地描述其状态、发展趋势和相互联系。这种描述,通常是以采集有关数据为基础的。例如,要了解某学校的基本情况,就需要通过采集该学校的名称、地点、教师人数、学生人数、教学设备数等。这些情况都是以图形、数码、文字等物理符号来描述的。这些用来记录或标记事物的物理符号的集合就是数据。在计算机处理中,数据是通过一组能被计算机识别、处理的字符来描述的。

信息是从原始数据中经过筛选、提炼等加工过程后,产生的对决策有影响的数据。信息和数据在概念上既有联系,又有区别。不经过加工的数据只是一种原始材料,难以影响人们的决策,其价值只是在于记录了客观世界的事实。只有经过提炼和加工,已经发生了质的变化的数据才能成为信息。因此,信息源于数据。信息在记载、传送、整理中,仍然是以数据的形式出现的。在计算机中,信息也是以数据的形式存贮、处理的。所以,在某些情况下,常把信息处理和数据处理当作同一概念使用。

### 1.1.2 数据处理及其发展阶段

人类社会的政治、经济、军事、文化等活动总是在信息的支配和指导下进行的,信息对于人类社会的发展具有重要作用。特别是在当今的信息社会里,数据信息量急剧膨胀,对数据处理的精度、速度提出了更高的要求。为此,我们不但需要更先进的计算机,而且需要更先进的数据组织与管理技术。

随着计算机技术的发展,数据处理技术经历了以下几个阶段。

### 1. 人工管理

20世纪50年代中期以前,这是计算机用于数据处理的初期阶段,对数据的管理是由程序员个人考虑和安排的,数据与程序不具有独立性。

### 2. 文件系统

20世纪50年代后期至60年代中后期,在这一阶段中,将数据组织成文件的形式,数据与程序有了一定的独立性,这就使得计算机数据管理方法得到了极大的改善。

### 3. 数据库系统

20世纪60年代后期,数据库方式在文件系统方式的基础上迅速发展起来。数据库的实质是一个所有存储在计算机存储介质上的相关数据所构成的集合,其基本思想就是要对所有的数据实行统一的、集中的、独立的管理,数据库中的数据独立于程序而存在并且可以提供给各类不同用户共享使用。

### 4. 分布式数据库系统

是数据库技术和网络技术紧密结合的产物。开放式数据库连接(ODBC, Open Database Connectivity)是用于数据库服务器的一种标准协议。可以使 Visual Foxpro 与该数据库相连,访问库中的数据。

### 5. 面向对象数据库系统

20世纪80年代,是数据库技术与面向对象程序相结合的产物。Visual Foxpro 不但支持标准的过程化程序设计,而且还提供了面向对象程序设计的强大功能。

#### 1.1.3 关系型数据库

##### 1. 关系模型

只有把数据按一定的形式组织起来,才便于快速地、准确地处理、存储和传递数据。各个数据对象及它们之间存在的相互关系的集合,称为数据模型。常见的数据模型有层次模型、网络模型和关系模型。其中,关系模型对数据的组织方式可以看成是一张有若干行、若干列的二维表格,如表1.1.1所示。目前,关系模型以其简洁明了的数据组织形式和具有严密、成熟的数学理论基础而被人们广为接受和应用。可以说,关系模型是三种数据模型中最重要的模型。微型计算机数据库管理系统绝大部分都是关系数据库管理系统。

表1.1.1共有6行,8列(栏)。其中,每一列为一个栏目,称为一个字段。表中的第一行为栏目的名称,称为字段名。该表共有8个字段,字段名分别是货号、商品名、进货日期、单价、数量、缺损否、样品和商品简介。除第一行外,向下还有5个连续的行。每一行都说明了一种商品的基本情况,称为一个记录。该表共有5条记录。

表 1.1.1 商品进货表

货号	商品名	进货日期	单价	数量	缺损否	样品	商品简介
10001	手挽杯	09/18/05	58.0	59	T	gen	备注
20001	八宝粥	10/05/05	42.0	300	F	gen	备注
20002	营养粥	10/05/05	30.0	150	F	gen	备注
30001	糖水蜜桔	11/12/05	72.6	500	T	gen	备注
30002	糖水椰果	11/12/05	84.0	200	F	gen	备注

由此可以看出,关系模型将数据元素内部各数据项间的联系(局部逻辑结构)和各数据元

素间的联系(整体逻辑结构)都表示成满足一定条件的二维关系表的形式。用这种关系模型设计的数据库就是关系型数据库。

一份二维表格称为一个关系,它必须满足以下条件:表中不能有完全相同的字段或记录;表中列间(或除第一行外的行间)的次序相互调换,不影响表所描述的物理实质。

## 2. 关系模型中的数据层次

关系模型中的数据,可分为四个层次。

### (1) 字段

字段是关系数据库文件中,由若干字符组成的,具有名称的最小数据单位。字段用来描述某个实体对象的属性,相当于二维表格中的一列,该列各值都具有相同的数据类型。一般,我们就以属性名作为字段名。例如表 1.1.1 中第一行的各列为字段名,其余每行中的各列都是相应字段中具体的数据,称为字段值。组成一个字段的字符个数称为字段宽度。

### (2) 记录

记录是描述一个实体(例如一种商品)的数据集合。一个记录由若干个字段值组成,相当于二维表中的一行。在一个关系中,各个记录都具有相同的字段个数。

### (3) 文件

文件也称库文件,是具有相同性质的记录的集合。它相当于一张二维关系表。同一文件中的记录必定是同格式、等长度的。数据库文件中包含了数据和有效数据的结构。字段名、字段类型、字段宽度、记录的排列,决定了数据的有效方式,称为数据库文件的结构。

### (4) 数据库

描述实体集的所有文件的集合称为数据库,它由若干个二维表的数据库文件组成。数据库不是某一方面数据文件的简单集合,它按照一定法则对文件进行重新组织,以便使数据具有最大的独立性和最小冗余度,并实现数据共享。

以上四个层次中,低一层次的数据是组成后一层次的元素。为了对关系型数据库有更清晰的了解,现将二维表格与关系型数据库文件作以下对照:

表 1.1.2

二维表格与关系型数据库文件的对照

二维表格	数据库文件
栏目名	字段名
列	字 段
行	记 录
二维表	数据库文件
多个二维表	

## 1.1.4 数据库管理系统、数据库系统

数据库管理系统(DBMS)是为数据库的建立、使用和维护而配置的软件。它建立在操作系统的基础上,对数据库进行统一的管理和控制。用户使用的多种数据库命令以及应用程序的运行,都要通过数据库管理系统来执行。例如,对数据库定义结构,输入记录,对记录完成显示、修改、增删、排序、索引、检索,以及对数据库的维护和加密等。

数据库管理系统、数据库、应用程序、数据库管理人员和相应的硬件设备共同组成了数据库系统。其中,数据库管理系统是数据库系统的核心部分,它提供了对数据库中的数据资源进

行统一管理和控制的功能,是应用程序和数据库中的数据间的接口,它由一系列软件组成。应用程序由用户编写,用来调用数据库中所存储的数据的各种命令文件。数据库管理系统自备有编写应用程序的语言,它提供了一系列与高级语言中相类似的命令,用户可以直接使用这些命令编写应用程序。

### 1.1.5 关系运算

关系数据模型的理论基础是集合论,每一个关系就是一个笛卡尔积的子集。在关系数据库系统中对数据的各种处理都是以集合运算为根据的。基本的关系运算有三种:选择(筛选)、投影和连接。关系数据库系统至少应当具有这三种操作。比较完善的关系数据库管理系统具有结构化查询语言(SQL),可以实现更多的关系运算。

关系是具有相同性质的记录的集合,可以用一张二维表来表示。二维表中的一行叫元组,二维表中的一列称为属性。各属性名的集合叫做关系框架,相当于二维表的表头。

对于关系这样一个二维表来说,总是由若干行元组和若干列属性组成的。如果我们要从全部元组行中挑选一部分满足一定条件的元组出来,这种对元组行的操作运算叫做“选择”。“选择”操作是横向操作,它只对关系行作用。这种运算的结果并不改变关系的框架,只改变关系中元组的多少。

如果我们要从关系框架的全部属性序列中选取一部分我们需要的属性及属性值,那么这种操作运算就叫做“投影”。“投影”操作是纵向操作,它对属性列操作,但有时也会影响到元组行。这一运算的结果将改变关系的框架,等于从原关系框架的属性序列中选出部分属性的子集而构成一个新的关系。当新关系中的一些元组由于舍去了一些属性而变得完全相同时,只保留其中一条元组。

“连接”运算是关系模型有别于其他模型的一个显著特点。其作用就是把两个关系通过比较、投影运算而连接生成一个新的关系。这个新的关系反映出来两个关系之间的联系。

## 1.2 Visual FoxPro 9.0 的特点

为了更好地理解 Visual FoxPro 9.0 的特点,有必要大致了解一下 Visual FoxPro 9.0 产生的背景。

### 1.2.1 数据库技术的历史

数据库理论研究的逐步深化,为数据库技术的广泛应用奠定了良好的基础。随着 20 世纪 70 年代后期数据库理论趋于成熟和 80 年代初期微型计算机的普及、性能的大幅度提高,数据库系统应运而生。其中,简单方便、功能较强、交互性好的 dBASE 关系数据库逐渐进入微机世界,并在 PC 平台关系数据库应用方面取得了成功,逐渐成为微机数据库的主导产品。以 dBASE 为基础,随后又开发、研制出功能不断增强的 dBASEX 系列产品。

80 年代中期,Fox 公司推出了与 dBASE III plus 完全兼容的 FoxBase,以后又推出功能更强大的两个版本,成为 PC 平台关系数据库产品系列中新的一员。随着图形用户界面的广泛使用和面向对象程序设计技术、网络技术、多媒体技术的飞速发展,PC 平台上的关系数据库管理系统(DBMS)需要有一个质的飞跃。正是在这种背景下,推出了 FoxPro 系列软件。Microsoft 公司在 1993 年推出了 FoxPro 2.5 等多个适用于不同平台的 FoxPro 版本,之后又推出了支持面向对象程序设计的 FoxPro 2.6;1995 年和 1997 年成功地推出了引入可视化编程技术的 Visual Fox-

Pro 3.0 和 Visual FoxPro 5.0。1998 年推出了全面支持 Internet 和 Intranet 的 Visual FoxPro 6.0, 2001 年推出的 Visual FoxPro 7.0、和 2003 年推出的 Visual FoxPro 8.0 版本增加了许多新功能, 2004 年由 Microsoft 公司发布的 Visual FoxPro 9.0 是一种功能强大的数据库管理系统, 它对以前版本的 FoxPro 提供完全的兼容性。

### 1.2.2 Visual FoxPro 9.0 的特点

与其他各种关系数据库管理系统和低版本的 Visual FoxPro 相比, Visual FoxPro 9.0 具有如下特点:

#### 1. Visual FoxPro 9.0 是 PC 平台上检索速度最快的数据库

这主要有三方面的原因: 第一, 采用了 Rushmore 技术, 使得长达数小时的查询可以降低到数秒内完成; 第二, 广泛采用了复合索引技术, 改变了传统的单一入口的索引文件结构, 使得索引文件中可以包含更多的索引; 第三, 引入了 SQL 选择命令, 使得用户能以最少的代码和最快的速度从一个和多个表中检索数据。

#### 2. Visual FoxPro 9.0 具有全自动性

Visual FoxPro 9.0 是一个典型的关系型数据库管理系统, 几乎所有的数据库操作都可以由各种向导 (Wizard)、生成器 (Builder) 和设计器 (Designer) 自动完成。

#### 3. Visual FoxPro 9.0 具有丰富的开发工具

开发人员能充分利用功能强大的编程器、设计器、跟踪与调试器、项目管理等工具开发功能齐全的应用程序。

#### 4. Visual FoxPro 9.0 支持真正的面向对象的可视编程技术

Visual FoxPro 9.0 不仅保留了传统的面向过程的程序设计, 而且支持真正的面向对象的可视编程技术, 能使用户快速、方便地建立和修改应用程序, 具有支持类、子类、对象、继承、封装、多态性等面向对象编程设计的多种特征。

#### 5. Visual FoxPro 9.0 在 Windows X 等系统环境的支持下具有工具条、对话框功能

Visual FoxPro 9.0 在 Windows X 等系统环境下运行时, 用户可以在工具条上设置一些图形化的按钮来代表一些经常使用的对象。此外, Visual FoxPro 9.0 的工具条还可以进行用户化的处理, 即允许用户从 Visual FoxPro 9.0 的工具条中选择一些按钮组成用户自己的工具条, 而且用户还可以在应用程序中通过编写代码来定义工具条和按钮。

#### 6. Visual FoxPro 9.0 具有简单、灵活、多样的数据交换手段

Visual FoxPro 9.0 支持更多与其他应用程序进行数据交换的文件格式, 如文本文件、电子表格、表等, 其外部的数据不但可以轻而易举地添加到 Visual FoxPro 9.0 的表中, 而且 Visual FoxPro 9.0 还可以将它的表转换成其他格式的数据文件交付给其他的应用程序。

#### 7. Visual FoxPro 9.0 支持客户/服务器结构

这一特点使得用户在开发客户/服务器应用程序时可以把 Visual FoxPro 9.0 作为客户机终端使用的软件, 使用 SQL 直接访问服务器, 同时 Visual FoxPro 9.0 综合了对服务器数据的更新视图技术。所有这些都为用户提供了一个坚实的基础, 以开发功能齐全的客户/服务器应用程序。系统中提供的数据库、远程视图、空值、事务处理以及对 ODBC 数据源的访问都支持客户/服务器结构应用程序。

#### 8. Visual FoxPro 9.0 中的数据库 (.DBC) 文件

不但用于组织和管理 Visual FoxPro 9.0 中的表 (.DBF), 而且还可以作为一个数据字典来

存储和管理有关表、记录和字段的规则、缺省值、触发器、视图、表间关联以及表到数据库的连接等。

9. 在 Visual FoxPro 9.0 中,用户可以生成真正的事件驱动的应用程序。摆脱了传统上的 READ 层次嵌套。用户不但可以利用 Visual FoxPro 9.0 提供的标准时间处理代码,而且可以编写自己的事件处理代码来响应全部的标准 Windows 事件。

10. 在 Visual FoxPro 9.0 中引入了事务处理技术、数据缓冲技术。大大简化了多用户环境下开发者对数据更新问题的处理。

### 1.3 Visual FoxPro 9.0 基础知识

Visual FoxPro 9.0 是开发应用程序的强有力工具,能很好地实现信息的分类、追踪和处理。与其他的程序设计语言一样,Visual FoxPro 9.0 提供了多种数据类型,并可以将数据存入各种类型的表、数组、变量和其他存储容器中。Visual FoxPro 9.0 的数据分为两大类:一类是变量和数组,另一类是表中的字段。

#### 1.3.1 数据类型

##### 1. 字符类型(Character)

字符数据类型由字母、数字、空格、符号和标点等组成。字符型的字段,内存变量、数组元素等存储的是诸如名、地址、提示信息以及不用于算术运算的数字等形式的文本信息。字符类型字段或者内存变量的长度介于 1 到 254 字节之间,并且每个字符占一个字节。

##### 2. 数值类型(Numeric)

数值类型用于表示数量的一种数据类型,它由数字 0 到 9、一个符号(+和-)和一个小数点组成。数值类型既可以用于表中对字段进行定义,也可以用于内存变量和数组元素。在表中,数值类型的长度介于 1 到 20 字节之间,对于变量和数组元素,则占用 8 个字节。存储时要转换成 ASCII 字符再存储的。

##### 3. 整数类型(Integer)

整数类型用于无小数数据的存取。整型类型仅用于表中的字段类型说明,占 4 个字节,且用二进制形式表示的。因此,它比数值类型的字段占用的空间要少得多,而且整数类型的值不需要转换成 ASCII 字符来存储。

##### 4. 双精度类型(Double)

双精度数据类型用于取代数值类型,以便能提供更高的数值精度。双精度数值类型只用于表中字段的定义,它采用固定存储长度的浮点数形式。双精度类型占 8 个字节。它的小数点位置是由输入的数据来决定的。

##### 5. 浮点类型(Float)

浮点类型与数值类型完全等价,它主要用于 Visual FoxPro 9.0 的兼容处理。

##### 6. 货币类型(Currency)

当涉及到使用金钱时,可以使用货币数据类型以代替数值类型。如果在货币类型字段或者变量中指定的值小数位超过四位。那么在计算之前将对这个货币值进行舍入处理。因为 Visual FoxPro 9.0 中的货币类型只允许最多有四位小数。货币类型字段和变量的值占用 8 个字节的存储空间。例如,下面的两个语句定义了两个货币类型的变量,并且其中的一个还进行

了舍入处理：  
`Money = $100.35`

`Moremoney = $ 645.74557`

### 7. 日期类型(Date)

日期数据类型是用于存储有关年、月和日的一种数据类型。日期类型的字符格式为‘yyyymmdd’，其中，yyyy表示年号，占用四个字节，mm表示月份，占用两个字节，dd表示日子，占用两个字节。日期类型共占用八个字节，取值不大于9999年12月31日。日期类型字段和变量中表达日期的日期格式有许多种，最常用的格式为mm/dd/yyyy，如：

`mdate = {'09/19/1998'}`

对于日期类型常量，必须用花括号将一个日期括起来，而对于空值的日期常量，则可以用{}或者{}或者{/}表示。

注意：SET DATE、SET MARK、SET CENTURY命令的设置值将影响日期格式。

### 8. 日期时间类型(DateTime)

日期时间数据类型用于存储日期和时间值。日期时间数据类型字段或者变量的字符存储格式为‘yyyymmddhhmmss’，分别表年、月、日、小时、分和秒，共占16个字节。日期时间类型的值既可以只包含一个日期或者只包含一个时间值，但也可以同时包含一个日期和一个时间值。若只包含两者之一，则系统会自动加上缺省值。时间部分的取值介于00:00:00am到11:59:59pm之间。如：

`mdatetime = {'9/19/1998 10:00am'}`

`mtimeonly = {'10:10:20pm'}`

对于，日期时间类型常量而言，与日期类型的常量一样，需要用一对花括号来加以说明。而描述一个空的日期时间类型值，则需要用{:}来表示。同时，日期时间类型中的日期部分与日期类型完全一样，具有多种显示格式，并受SET DATE、SET MARK、SET CENTURY命令设置值的影响，而时间部分的显示格式则受SET HOURS和SET SECONDS命令设置的控制。

### 9. 逻辑类型(Logical)

逻辑类型用于存储只有两个值的数据。即只有真(.T.)和假(.F.)两个值。只占一个字节。

### 10. 备注类型(Memo)

备注类型只用于表中，它用于数据块的存储。在表中，Memo字段只包含4个字节，并用这4个字节来引用备注的实际内容。由于备注字段的实际内容变化很大，不能直接将备注内容存储在表.DBF文件中。因此，需要一个相对独立的文件来存储这些备注内容，该文件的扩展名是.DBT。

### 11. 通用类型(General)

通用数据类型用于存储OLE对象。该通用类型字段包含了对OLE对象的引用，而一个OLE对象的具体内容则可以是一个电子表格、一个字处理器的文档、图片等。这些OLE对象是其他的应用软件建立的。

通用数据类型字段在表中的长度为4个字节，而OLE对象的实际内容、类型和数据量则取决于建立该OLE对象的服务器以及是连接或者嵌入OLE对象。如果采用连接OLE对象方式，那么表中只包括对OLE对象中数据的一个引用说明，以及到创建该OLE对象的应用软件

之间的引用说明。如果采用嵌入 OLE 对象方式,那么,表中在包含到创建该 OLE 对象的应用软件之间的引用之外,还包括了 OLE 对象中的实际数据。这时,通用数据类型字段的长度仅受限于硬盘的可用空间。

另外还有:二进制字符类型和二进制备注字段类型。它们与字符型、备注型相类似。

### 1.3.2 常量

一个常量是一个命名的数据项,在所有的操作期间,它的值都将保持不变。例如,3.14159 是一个数值常量,而字母“ABC”就是一个字符常量。字符型常量要用引号括起;逻辑型常量只有两个,即真(.T.)和假(.F.)。日期型常量和日期时间型常量要用花括号来定义。在 Visual FoxPro 9.0 中,可以创建一种特殊的常量——编译时常量。这种常量只存在于 Visual FoxPro 9.0 应用程序的编译时,编译之后,将用常量的具体内容换该常量在源代码中的位置,而且常量可以包含任意类型的数据。编译时常量是需要用伪编译命令 # DEFINE 进行定义,如:

```
# DEFINE PI 3.14159
```

定义了 PI 常量后,在源代码中就可以使用 PI。而在编译之后,凡是出现 PI 的地方都用数值 3.14159 取代。如果编译时常量不再用了,则可用伪编译指令 # UNDEF 取消定义,如:

```
# UNDEF PI
```

### 1.3.3 变量

变量的值是可以随时更改的,Visual FoxPro 9.0 包括字段变量和内存变量两种类型。表中的一个字段名可以取不同的字段值,它就是一个变量,称为字段变量。

内存变量,是内存中的一个存储区域,它的类型是由变量中的值来决定的。

#### 1. 内存变量的赋值

在 Visual FoxPro 9.0 中,它通常有两种定义方式:

格式 1: STORE <表达式> TO <变量名表>

功能:计算表达式的值并赋值给一个或多个内存变量。

如: STORE 2 + 6 TO A1, A2, M, N

格式 2: <内存变量名> = <表达式>

功能:计算表达式的值并赋值给一个内存变量。

如 n = 8。其中, n 就是变量,此时,该变量的值是 8,类型是数值型。

#### 2. 内存变量的显示

格式 1: LIST MEMORY [LIKE <通配符>] [TO PRINTER] | TO FILE <文件名>

格式 2: DISPLAY MEMORY [LIKE <通配符>] [TO PRINTER] | TO FILE <文件名>

功能:显示内存变量的当前信息。

说明:

DISPLAY MEMORY 为显示一屏暂停,按任一键继续显示下一屏。

通配符:“?”代表变量名中当前位置的任一个字符。

“\*”代表变量名中从当前位置开始的一个或多个字符。

选用 LIKE 短语只显示与 <通配符>相匹配的内存变量。

TO PRINTER: 显示内存变量的同时送往打印机。

TO FILE <文件名>: 显示内存变量的同时存入文件名指定的文本文件中,扩展名 .TXT。

例 1.3.1 先给以下变量赋值,再显示。



## CLEAR MEMORY

清除内存变量

STORE 2 \* 3 TO A1, A12, B1, B11, CD

给变量赋值

DA = "123456"

D1 = DATE()

C12 = .T.

C1 = "ABCD"

## DISP MEMORY

显示内存变量

显示内存变量如下:

变量名

作用域

类型

值

变量名	作用域	类型	值
A1	Priv	N	6
A12	Priv	N	6
B1	Priv	N	6
B11	Priv	N	6
CD	Priv	N	6
DA	Priv	C	"123456" al
D1	Priv	D	08/03/07 al
C12	Priv	L	.T. al
C1	Priv	C	"ABCD" al

已定义 9个变量, 占用了24个字节  
16375个变量可用

## 3. 内存变量的清除

## 格式 1: CLEAR MEMORY

功能: 清除所有的内存变量。

## 格式 2: RELEASE &lt;内存变量名表&gt;

功能: 清除指定的内存变量。

## 格式 3: RELEASE ALL[EXTENDED]

功能: 清除所有的内存变量。若在程序中则应选 EXTENDED, 否则不能删除公共内存变量。

## 格式 4: RELEASE ALL[LIKE &lt;通配符&gt;][EXCEPT &lt;通配符&gt;]

功能: 选用 LIKE 短语清除与 <通配符>相匹配的内存变量。选用 EXCEPT 短语清除与 <通配符>不匹配的内存变量。

## 例 1.3.2 清除变量名以 B 为第一个字符的内存变量。

RELEASE ALL LIKE B \*      \* 代表从 B 后面位置开始的一个或多个字符

## DISP MEMORY

显示内存变量

变量名	作用域	类型	值
A1	Priv	N	6
A12	Priv	N	6
CD	Priv	N	6
DA	Priv	C	"123456" al
D1	Priv	D	08/03/07 al
C12	Priv	L	.T. al
C1	Priv	C	"ABCD" al

已定义 7个变量, 占用了24个字节  
16377个变量可用