



FoxPro for Windows

程序设计

陈康 主编

1.138FO



西安电子科技大学出版社

<http://www.xduph.com>

597

TP311.138Fc

C45

FoxPro for Windows

程 序 设 计

陈 康 主 编

杨忠孝 谢膺白

张威虎 李 东 副主编

贾应智 于克倩

西安电子科技大学出版社

2000

内 容 简 介

本书以 FoxPro 2.5b for Windows 为背景,介绍了 FoxPro 程序设计的基本方法和程序设计的技能、技巧。由于本书侧重于对 FoxPro 程序方式下的应用,所以本书也适用于不同操作平台、不同版本的 FoxPro 系统。

语言通俗、流畅却不失严谨、准确,内容深入浅出、循序渐进却不失系统性、完整性,是本书的特色。

本书既可作为普通高校本、专科各专业数据库管理系统程序设计课程的教材,也适于开发办公自动化或管理信息系统程序设计的读者自学。

图书在版编目(CIP)数据

FoxPro for Windows 程序设计/陈康主编. —西安:西安电子科技大学出版社,2000.8

ISBN 7-5606-0891-4

I. F… I. 陈… III. 关系数据库-数据库管理系统, FoxPro IV. TP311.138

中国版本图书馆 CIP 数据核字(2000)第 34482 号

责任编辑 陈宇光 戚文艳

出版发行 西安电子科技大学出版社(西安市太白南路2号)

电 话 (029)8227828 邮 编 710071

<http://www.xduph.com> E-mail: xdupfb@pub.xaonline.com

经 销 新华书店

印 刷 西安长青印刷厂

版 次 2000年8月第1版 2000年11月第2次印刷

开 本 787毫米×1092毫米 1/16 印张 16

字 数 379千字

印 数 4 001~8 000册

定 价 17.00元

ISBN 7-5606-0891-4/TP·0473

* * * 如有印装问题可调换 * * *

本书封面贴有西安电子科技大学出版社的激光防伪标志,无标志者不得销售。

前言

目前,教育部考试中心的计算机等级考试大纲和全国普通高等学校非计算机专业计算机等级考试大纲中,对于数据库管理系统,仍然考核结构化程序设计语言,即暂时未把 Visual FoxPro 列入考试范围。但是,在计算机文化基础的教学和考核已经转向 Windows 平台的情况下,继续讲授 FoxBase 或 FoxPro for DOS,已经不符合新课程体系的要求了。既能在 Windows 下运行,又不是 Visual FoxPro 的数据库系统,只能选择 FoxPro 2.5b for Windows——这就是本书仍然以 FoxPro 2.5b for Windows 为背景编写的原因。

FoxPro 是继 FoxBase 之后,广泛用于办公自动化和管理信息系统开发的一种关系型数据库管理系统。值得说明的是, FoxPro 的版本多种多样, FoxPro 的功能也在不断地扩充,如此丰富的内容,不可能也没必要在本书中全部包容。但是,由于美国微软公司在 FoxPro 的版本更新中,一直遵循向下兼容的原则,更新版本仍然延续了旧版本的基本功能。对于熟悉 FoxPro 某一种版本的用户来说,进一步了解新版本中的扩充功能并不困难,甚至用旧版本开发的应用程序,不需要改动就能在新版本下运行。所以,本书介绍的基本命令和程序设计的基本方法,还是普遍适用的。如果程序设计中需要使用不同操作平台、不同版本的特有功能,或者已经开发的应用程序受当前所使用版本技术指标的限制时,通过参考有关版本的技术资料并不难以解决。

编者在 FoxPro 程序设计课程的教学,曾经选用过各种教材。但是,所选择的教材都不同程度地存在一些不尽如人意的地方。在这种情况下,不得不自行编写授课讲义,本书就是以使用多年的授课讲义为基础编写的。

本书具有以下几个方面的特色:

1. 读者学习 FoxPro 的主要目的在于程序设计。所以本书简化了对 FoxPro 2.5b for Windows 交互式应用环境的介绍,突出了 FoxPro 程序方式下的应用。显然,对于 FoxPro 的这种取舍,更符合 FoxPro 程序设计的教学目标和读者学习 FoxPro 的意愿。

2. 作为一门程序设计的课程,应当重点掌握 FoxPro 基本命令在程序设计中的应用。所以,本书用较少的篇幅介绍 FoxPro 的基础知识和基本操作,用较多的篇幅介绍程序设计的方法和程序设计的技能、技巧;尽可能地把 FoxPro 基本命令及其在程序设计中的应用结合起来,达到使读者通过程序设计,能正确理解和熟练使用 FoxPro 的基本命令的目的。教学实践证明,本书的这种编排,有益于提高读者程序设计的能力,并可取得良好的教学效果。

3. 对 FoxPro 中的许多基本命令, 若仅仅进行文字说明, 则仍然使人难以理解。如果能列举一些和基本概念密切相关的程序例题, 则读者对基本命令、基本概念、注意事项等方面的问题就不难理解了。为此, 本书在正文中引用了数量较多的例题。

4. 本书针对各章节的内容, 附有较多的程序设计练习题, 作为教材, 这是必不可少的。

读者在学习本书之前应掌握 Windows 的基本操作, 本书对所涉及到的 Windows 基本操作未作过多的讲解。

本书将普通高校 1998~2000 年陕西省非计算机专业计算机等级考试的 FoxPro 试题作为附录排在书末, 可供今后参加 FoxPro 语言等级考试的学生参考。本书中的例题和习题, 都已在 Windows 3.X、Windows 98 下调试通过。读者如果需要例题和习题的源程序或用 PowerPoint 制作的教学幻灯片, 可向编者索取。

编 者

2000 年 6 月

数据库和数据库管理系统概述

第 1 章

电子计算机最初被作为一种高速数学运算工具，但是在计算机诞生之后不久，人们便开始探索计算机在事务处理(信息处理)方面的应用。事务处理中的关键技术问题是：怎样把作为事务处理对象的数据存放到计算机的存储设备中，才能使用户对所存放的数据，进行方便、直观的管理、维护和使用？直到 20 世纪 70 年代初，出现了数据库技术之后，才成功地解决了这一问题。

数据库技术使计算机在事务处理领域中的应用产生了一次飞跃，特别是运行于微型计算机上的数据库管理系统问世之后，推动了计算机社会化的普及应用。可以说，当前的微机系统中，几乎都装有数据库管理系统软件；当前使用的计算机，几乎都把事务处理作为主要应用项目。可见，数据库应用技术是每一个计算机用户都需要掌握的基本技术之一。

1.1 数据和数据库的概念

1.1.1 数据、信息及数据处理

客观世界和人类社会是一个数据的世界和数据的社会。无论是企业、事业单位还是行政管理部门，每时每刻都会产生大量的数据，这些数据记录了经营和管理活动的过程。此外，在反映自然现象、生命过程及仪器设备运行状态的数据中，除了我们常规概念上的数据之外，可能还有一些物理量。数值的、非数值的(例如，文字、图形、图像、声音、物理量)数据，从数据处理的角度来看，都统称之为数据。

反映客观世界和人类社会活动的数据，并不是都需要人们采集和处理的，而是根据实际问题的需要，有目的地采集那些最能反映目标问题实质的数据，通过对这种数据的分析、处理得出的结果，叫做信息。因此，可以认为，数据是信息的载体，并不都有利用价值，而信息一般是有利用价值的。

但是，信息也是用数据表达的，因此也把信息称为有意义的数据库。如果对作为数据库的信息进一步分析、处理，又会从中得到新的信息。这样看来，信息和数据库并没有本质上的区别。所以，今后我们把针对实际问题所采集、存储的数据，既可以叫做数据库，也可以叫做信息。

解决实际问题时,往往离不开与实际问题的数据。为此,需要进行数据的采集、整理和存储。在此基础上,利用计算机对存储的数据进行分析,充分挖掘出数据中隐含的信息,这一系列操作过程叫做数据处理。因为数据处理的对象是数据,所以数据的采集、整理和存储是数据处理的基础。特别是如何在计算机中存储数据,既关系到数据的安全性、完整性问题,又关系到数据的存取、管理和使用是否迅速、方便的问题,这是计算机专业人员一直在探索的重要课题。20 世纪 70 年代出现的数据库技术,成功地解决了数据存储、管理和应用中的一系列问题,使计算机在数据处理方面发挥了举足轻重的作用。

1.1.2 数据库的概念

数据库是按一定的组织结构存放数据的磁盘文件,这种特殊的磁盘文件叫做数据库文件,或简称为数据库。

在出现数据库技术之前,数据是附着在数据处理程序之中或者存放在顺序结构或随机结构的磁盘文件之中的。和数据的这种存放方式比较,数据库具有数据和程序可以相互独立,数据可以共享,数据冗余度小,便于管理和检索,并且还可以随时改变数据存储结构等优点。

数据库的形成、管理和使用,是在一种叫做数据库管理系统(DBMS)软件的控制之下进行的。所以,不同的数据库管理系统,所形成的数据库,其结构可能是不同的。目前,商品化的数据库管理系统所对应的数据库结构,主要有三种结构模型,即层次模型、网状模型和关系模型。

1. 层次模型的数据库

层次模型的数据库是按数据的从属关系存放数据的,类似于磁盘上的文件目录结构。例如,使用层次模型可以方便地把一个单位的职工按其隶属关系存放到数据库中;在机械制造业,可以把构成产品的零、部件,按其组装关系存放到数据库中。显然,对层次模型的数据库,如果按隶属关系进行管理和检索是比较直观和方便的。

2. 网状模型的数据库

网状模型的数据库一般包含若干个数据库文件。如果某一个数据库中的数据和其它数据库中的数据有联系,则除了存放这个数据本身之外,还要存放这个数据指向所要联系数据的“指针”。这样,当检索到一个数据之后,就可以方便地检索到与之有关的其它数据。

3. 关系模型的数据库

关系模型的数据库是把与实际问题的数据,分别归纳成若干个简单的二元关系,每个二元关系可以视为一个二维结构的表格;若干个二维表之间,还可以建立关联关系。

例如,在大家所熟悉的图书馆管理中,涉及到很多数据项目,但是可以把这些数据项目分别归纳成三种二元关系,即三个二维表:有关馆藏图书数据的二维表(见表 1.1),有关读者信息的二维表(见表 1.2),有关图书流通信息的二维表(见表 1.3)。与图书管理工作有关的数据,都可以在这三个表格中找到对应的填写位置。

借助于关系型数据库管理系统,能够按照这三个二维表,分别建立数据库文件,使数据库文件中数据的存放格式和二维表中数据的填写格式完全一致。

表 1.1 馆藏图书二维表

图书编号	书 名	作 者	出 版 社	出版日期	定价	数量	架位
0100001	高等数学	张建国	高教出版社	1999.10	20.5	5	1-1
0200001	有机化学	李西方	科技出版社	1991.5	18.0	4	2-1
0200002	生物化学	周理文	卫生出版社	1987.9	21.5	5	2-2
0300002	大学英语	王家易	西电出版社	1990.1	22.0	6	3-1
0400001	物理学	吴春燕	电子出版社	1986.11	18.5	10	4-1

表 1.2 读者信息二维表

借书证号	读者姓名	职 称	所 在 单 位
9901001	陈海军	助教	数学教研室
9901002	郭树森	教授	化学教研室
9802001	王之光	副教授	物理教研室
9803001	将爱国	讲师	外语教研室
9506001	孙会中	研究员	校长办公室

表 1.3 图书流通信息二维表

借书证号	所借书号	借书日期
9803001	0100001	1998.10.12
9901001	0400001	1999.3.15
9506001	0200001	1996.2.20
9901002	0200002	1999.8.21
9901001	0100001	1999.8.30

二维表中的每一行,是描述一个独立“个体”的有关数据,它对应数据库中的一条“记录”;二维表中的每一列,是各个记录中同一属性的数据,它对应数据库中的一个“字段”。

1) 关系型数据库的特点

- (1) 每条记录长度相同,记录中包含的字段数相同。
- (2) 同属一个字段的数据,其属性、类型和长度相同。
- (3) 可以任意交换两行或两列数据的顺序。

2) 关系型数据库具有三种关系运算

- (1) 选择:按范围或条件选择使用数据库中的一部分记录。
- (2) 投影:按范围或条件选择使用数据库中的一部分字段。
- (3) 连接:可以把多个数据库按条件连接成一个逻辑的或物理的数据库。

数据库技术常用于办公自动化(OA)或管理信息系统(MIS)的应用开发中,即多用于事务处理。开发事务处理的应用课题,需要经过对象业务系统的调查、分析、设计等一些规范化的工作步骤,具体做法可参考系统分析、系统设计方面的资料。只有了解了对象业务的状况和业务部门的需求,才能知道需要利用哪些数据,这些数据应当归纳成几个二维表,哪些二维表之间需要建立联系,用什么字段作为联系字段。这些准备工作,叫做数据库的逻辑设计。逻辑设计将产生若干个二维表,按各个二维表的结构,在计算机的磁盘上建立和分配数据库文件的工作过程,叫做数据库的物理设计。

1.1.3 微机的关系型数据库管理系统

一般计算机用户是通过一种叫做数据库管理系统(DBMS)的软件,使用数据库的。数

数据库管理系统提供了面向用户的数据库定义语言(DDL)、数据库操作语言(DML)、用户程序设计语言和交互式工作环境。

最早在微机上广泛应用的关系型数据库管理系统,是美国 Ashon - Tate 公司于 1981 年开发的 dBASE II。dBASE II 推向市场后,由于其功能丰富、简单易学、使用方便等特点,深受广大用户欢迎,被誉为“大众化数据库”,很快得到了推广应用。之后, dBASE 的版本不断更新,目前国内较为流行的是 1989 年推出的 dBASE IV。

1984 年, Fox Software 公司推出了性能更为优越的 FoxBase, 其版本也在不断更新, 并形成了 Fox 系列。特别是 FoxPro 2.5 版本推出之后, 占据了较大的市场份额, 国内用户开始转向使用 FoxPro。1992 年, 美国微软公司收购了 Fox Software 公司, 继续开发出了 FoxPro 2.5b for DOS 和 FoxPro 2.5b for Windows 两种版本。1995 年, 微软公司把可视化编程技术用于 FoxPro, 开始推出 Visual FoxPro 3.0 for Windows, 目前较新的版本是 Visual FoxPro 6.0 for Windows。

Visual FoxPro 提供了快速开发用户应用程序的向导、生成器和设计器, 简化了诸如窗口、菜单、按钮、查询和数据库关联等方面的程序设计。Visual FoxPro 支持面向对象编程, 具有继承、封装、多态及子类等功能。利用 Visual FoxPro 的对象与事件模型, 用户可以快速地建立应用程序的原型。

但是, Visual FoxPro 的“可视化”和“面向对象”这两部分内容, 在应用环境和程序设计方法等方面和面向过程程序设计的数据库管理系统有很多不同之处。本书作为数据库应用程序设计的教程, 一方面考虑到一些学校的教学设施和师资状况, 目前尚不具备“可视化”和“面向对象”程序设计的条件, 特别是可视化方面的内容必须具备大屏幕投影机的教学设施和 Pentium II 以上档次的微机供学生实习; 另一方面, 对于已经习惯使用 dBASE、FoxBase 和 FoxPro for DOS 的用户来说, 可能更容易接受这种沿袭了结构化程序设计的 FoxPro for Windows, 这就是本书仍以面向过程程序设计的数据库管理系统为背景编写的原因。

1.2 FoxPro 2.5b for Windows 简介

1.2.1 FoxPro 2.5b for Windows 的特点

和 FoxBase 及 FoxPro for DOS 比较, FoxPro 2.5b for Windows(以下简称 FoxPro)具有以下特点。

1. 具有 Windows 风格的用户界面

不仅 FoxPro 系统本身具有和 Windows 操作平台类似的窗口、菜单、对话框等图形操作界面, 而且支持用户程序生成具有 Windows 风格的窗口、菜单系统。系统和用户程序均支持鼠标操作。

2. 支持多媒体数据

FoxPro 数据库的字段数据类型中, 增加了一种叫做“通用型”(General)的数据类型, 可用于存放声音、图形、图像等多媒体数据。

3. 先进的数据处理技术

FoxPro 采用了压缩索引、复合索引、结构化查询语言(SQL)、快速查询技术(Rushmore), 提高了多库关联查询和对大型数据库的查询速度。

4. 具有丰富的辅助设计工具

FoxPro 具有屏幕生成器、报表生成器和项目管理器, 使繁琐的屏幕格式、报表设计, 变得十分简单、快捷, 大大减少了这方面的编程工作量。

5. 提供了多种使用方式

使用 FoxPro 系统时, 除了传统的命令方式和程序方式之外, 还提供了菜单驱动方式, 即在交互方式下, 对于同一任务, 既可以用命令方式完成, 也可以用系统功能菜单完成。

1.2.2 FoxPro 的主要技术指标

1. 有关数据库文件的技术指标

- 数据库中的记录数可多达 10 亿条。
- 一条记录中的字段数不超过 225 个。
- 记录的最大长度为 65 000 个字节。
- 字符型字段最大长度为 254 个字节(可容纳 254 个半角字符或 127 个全角字符)。
- 数值型字段最大长度为 20 个字节, 最多表达 16 个有效数字。

2. 有关内存变量的技术指标

- 简单内存变量最多为 65 000 个。
- 数组总数最多为 65 000 个。
- 每个数组中的下标变量总数最多为 65 000 个。

3. 其它技术指标

- 最多可使用 225 个工作区(可分别打开 225 个 .DBF 文件)。
- 复合索引关键字最大长度为 240 个字节。
- 命令行中最多可以使用 2048 个字符。
- 循环嵌套最多为 32 层。
- READ 命令嵌套最多为 5 层。
- 数据库文件逻辑关联的个数不受限制。

1.2.3 FoxPro 的常用文件类型

FoxPro 系统所生成和使用的文件类型较多, 这里仅介绍一些常用的文件类型:

- (1) 扩展名为 .DBF 的文件, 是数据库文件。
- (2) 扩展名为 .FPT 的文件, 是存储数据库文件中 M、G 型字段数据的文件。
- (3) 扩展名为 .TXT 的文件, 是对应数据库文件的文本文件。
- (4) 扩展名为 .IDX 的文件, 是对数据库进行单索引时生成的文件。
- (5) 扩展名为 .CDX 的文件, 是对数据库进行复合索引时生成的文件。
- (6) 扩展名为 .PRG 的文件, 是 FoxPro 的源程序文件。
- (7) 扩展名为 .FXP 的文件, 是执行 .PRG 文件时所生成的目标文件。

- (8) 扩展名为 .MEM 的文件, 是内存变量的文件。
- (9) 扩展名为 .QPR 的文件, 是关联查询文件。
- (10) 扩展名为 .SCX 的文件, 是屏幕生成器生成的屏幕格式文件。
- (11) 扩展名为 .SPR 的文件, 是 .SCX 的源程序文件。
- (12) 扩展名为 .FRX 的文件, 是报表书写器生成的报表定义文件。
- (13) 扩展名为 .LBX 的文件, 是标签书写器生成的标签定义文件。

1.2.4 FoxPro 的启动和交互式应用

1. FoxPro 的启动和退出

1) 启动

双击 Windows 桌面上的 FoxPro 的图标, 或从 Windows 开始菜单中 (Windows 95 以上版本) 选择 FoxPro 程序项, 都可以启动 FoxPro for Windows。启动 FoxPro 后, 屏幕显示出 FoxPro 的系统窗口 (见图 1.1), 该系统窗口内有一个叫做命令窗口的子窗口, 用于打入 FoxPro 的单条命令。

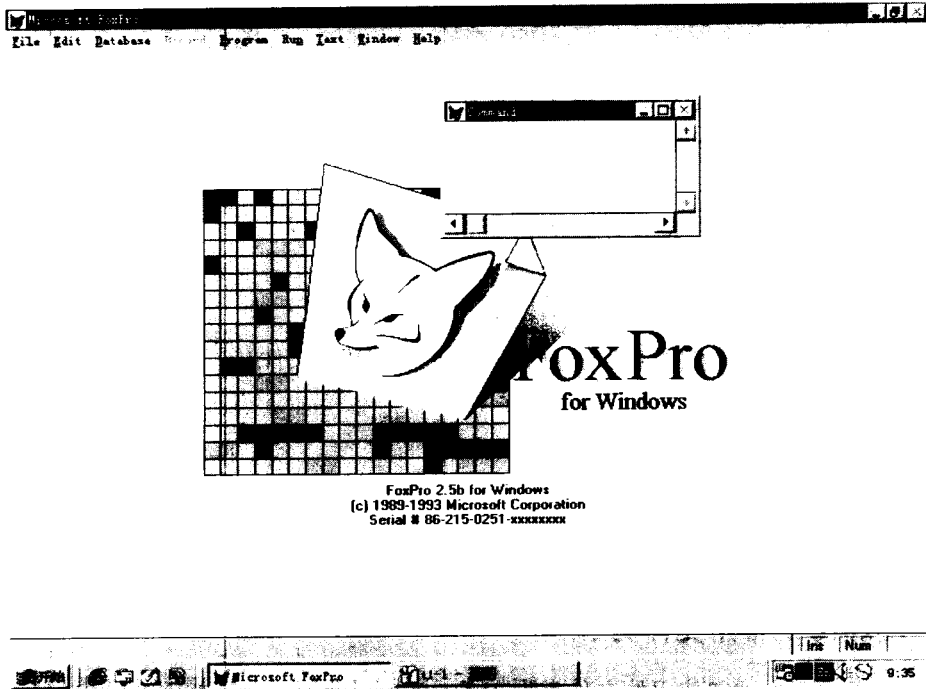


图 1.1 FoxPro 的系统窗口

2) 退出

下列三种操作都能退出 FoxPro 并返回到 Windows 系统:

- (1) 单击位于 FoxPro 窗口标题左端的控制菜单按钮 (小狐狸图标), 从控制菜单中选择“关闭”;
- (2) 单击 File 菜单项, 从中选择“Exit”;

(3) 在命令窗口打入命令: QUIT<回车>。

2. FoxPro 的交互式应用

FoxPro 有交互式 and 程序式两种应用方式。

FoxPro 的交互应用方式包括利用 FoxPro 的系统菜单或在命令窗口打入 FoxPro 的单条命令。

1) 使用 FoxPro 的菜单系统

FoxPro 的系统菜单(主菜单)位于窗口标题下面,系统菜单包括 File、Edit、Database、Record、Program、Run、Text、Window、Help 等。每一个系统菜单项的功能,分列在对应的下拉菜单中。

在交互方式下使用 FoxPro 时,可以利用 FoxPro 功能丰富的菜单系统。为了使用菜单系统中的某一功能,首先要激活系统菜单。激活系统菜单的操作方法主要有以下两种:

(1) 使用鼠标激活。单击要使用的某一系统菜单项,则激活并显示出该系统菜单项的下拉菜单。单击下拉菜单中所要使用的子功能菜单项,则转入子功能菜单的操作。

(2) 使用键盘激活。按下 Alt 键,系统菜单项 File 被选中,此后用左、右箭头键,可以选择其它系统菜单项,选中后按回车键,显示出对应的下拉菜单。用上、下箭头键选择子功能菜单后回车,则转入子功能菜单的操作。

例如,使用主菜单项 Help 可以获得使用 FoxPro 系统的帮助功能;使用主菜单项 Window,可以隐藏当前窗口、清除当前窗口所显示的内容和直接对 225 个工作区中的某一个数据库进行操作,等等。

一方面,几乎所有的子功能菜单都有 FoxPro 的基本操作命令与之对应,完全可以用 FoxPro 的命令代替系统菜单的操作;另一方面,本书的重点是介绍如何在程序方式下使用 FoxPro。所以,这里对 FoxPro 的系统菜单不作全面介绍,在今后介绍某些 FoxPro 的基本操作命令时,将顺便提及对应的菜单操作方法。其实,读者完全可以自行练习系统菜单的使用,从中了解各个菜单项的功能。

2) 使用命令窗口

命令窗口专用于打入 FoxPro 的单条命令,每打入一条命令并回车后,所打入的命令会立即执行。所执行的命令如果有输出信息,则立即显示在屏幕上。

将命令窗口所打入的命令,逐条保留并显示在命令窗口上,当打入的命令长度超过命令窗口的显示宽度时,窗口中显示的命令会向左滚动;当打入了多条命令时,窗口中已经执行过的命令会向上滚动。借助位于命令窗口边框上的滚动条,可以阅读已经打入的各条命令。用鼠标或用箭头键指向命令窗口中已经执行过的某一条命令并回车,可以重复使用该命令或进行编辑后再使用,这样可以减少键盘打入命令的工作量。

为了说明命令窗口的使用方法,这里先简单介绍以下命令:

(1) 把数据存入变量的命令(赋值命令)。例如,为了把数据 123.45 送到一个名为 abc 的变量(关于变量的概念在下一章中讲述)中保存,应当在命令窗口打入命令: abc=123.45<回车>,则变量 abc 被定义并存储了数据 123.45。

(2) 显示变量中所存储的数据(显示命令)。例如,为了显示变量 abc 中所存储的数据,在命令窗口打入命令: ? abc<回车>,则在屏幕上立即显示出 123.45。

此时,命令窗口显示的是已经执行过的两条命令。如果把数据 456.78 存入变量 abc,

可以重新打入命令: $abc=456.78$ (回车), 也可以用鼠标或箭头键指向原来已经打入的命令: “ $abc=123.45$ ”, 将 123.45 改为 456.78 并回车。

在 FoxPro 系统环境下, 除了命令窗口和其它系统窗口外, 用户还可以自行定义窗口 (参见第 12 章)。当屏幕上有多个窗口时, 用户只能使用其中的一个窗口, 这个窗口叫做当前窗口。例如, 启动 FoxPro 之后, 命令窗口就是当前窗口。在一些系统菜单或命令的操作中, 又会出现其它的当前窗口。但是, 用户可以通过使用系统菜单或有关命令改变当前窗口。FoxPro 的系统菜单 Window 主要功能就是窗口操作: 使用子功能菜单 Cycle, 可以改变当前窗口; 使用子功能菜单 Hide, 可以隐藏当前窗口; 使用子功能菜单 Clear, 可以清除当前窗口中所显示的内容; 使用子功能菜单 Command, 不论当前窗口是哪个窗口, 立即把命令窗口作为当前窗口; 使用子功能菜单 View, 可以进行多个数据库的操作。当前窗口的移动、改变尺寸等操作和 Windows 系统中的操作方法相同, 这里不再赘述。

习 题 1

1. 什么是数据和信息?
2. 和其它存放数据的方法相比较, 用数据库管理数据的优点有哪些?
3. 数据库有几种结构模型? 各种结构模型的特点是什么?
4. 了解数据库技术指标的含义。
5. 练习:
 - (1) 启动 FoxPro for Windows。
 - (2) 试用 FoxPro 的 Help 功能。
 - (3) 利用 Window 菜单, 隐藏命令窗口及取消命令窗口的隐藏。
 - (4) 使用鼠标, 练习改变命令窗口的大小和位置。
 - (5) 在命令窗口练习赋值命令和显示命令。
 - (6) 退出 FoxPro。

使用 FoxPro 的基础知识

第 2 章

在学习 FoxPro 的基本操作和 FoxPro 的程序设计之前,首先介绍与基本操作和程序设计有关的一些基本概念和基础知识。对常量、变量、函数、表达式及 FoxPro 基本操作命令中常用的短语等概念已经有所了解的读者,可以越过本章,从第 3 章开始学习。

2.1 常量、变量、函数和表达式

FoxPro 的常量、内存变量、函数和数据库中的字段变量都有类型属性。“数据类型”是一个非常重要的概念,因为数据库管理系统规定:只有类型相同的常量、内存变量、字段变量、函数之间才能进行操作。

数据可分为:字符型、数值型、浮点型、日期型、逻辑型、屏幕型、窗口型、备注型和普通型等 9 种类型。其中,常量只能使用前 5 种类型;内存变量只能使用前 7 种类型;字段变量只能使用前 5 种类型及备注型和普通型。

2.1.1 常量

在命令或程序语句中直接使用的数据叫做常量,程序中的常量在程序运行过程中,其值始终保持不变。为了使人和计算机能够区分常量的类型和宽度,有的常量需要加类型界定符。

1. 数值型常量

数值型常量由数字、小数点和正负号组成,是能够进行算术运算的数据。数值型常量的整数部分和小数部分合起来最多能表示 16 位有效数字。

数值型常量除了常规的表示方法外,还可以用“科学计数”法表示。科学计数法表示数值常量的一般格式是:〈尾数〉E〈±整数〉。其中,尾数是用常规方式表达的数值数据,E〈±整数〉相当于以 10 为底的整数次方,正或负的〈整数〉决定了小数点在〈尾数〉中向左或向右移动的位数。例如:

$$0.123456E4=1234.56 \quad 123456E-4=12.3456$$

2. 字符型常量

字符型常量是由 ' '、" "或 [] 等符号界定起来的字符串。当上述某一种或两种界定符出

现在字符串中时，应改用其它符号作为字符串的界定符。例如，

'abc123'、"abc123"、[abc123] 是三个相同的字符型常量。

'123.45'、"abc123"、[aa'aa] 是正确的字符型常量。

"123.45"、'abc123'、'aa'aa' 是错误的字符型常量。

3. 日期型常量

日期型常量用花括号界定，年、月、日各用两位数字表示。其中的月、日应当是合理的 1~2 位数，公元年号超过两位时只取末两位。空日期常量表示为{}/{}。

可以用日期格式设置命令改变日期格式。例如，在命令窗口打入并执行命令

SET DATE TO YMD

之后，日期格式改变为{年/月/日}，这种格式符合我们的日期表达习惯。命令中的参数 YMD 还可以改写为 MDY、DMY，它们分别对应{月/日/年}、{日/月/年}格式。在 YMD 格式下：

{99/01/01}、{99/1/1}、{1999/1/1} 是正确的日期常量。

{990101}、{99/13/35}、"99/01/01" 是错误的日期常量。

目前销售的微机，其硬件系统已经用 4 位数表示公元年号了，即硬件已经不存在“千年虫”问题了。早期推出的应用软件，例如 FoxPro 2.5b for Windows 仍然沿用两位数字表示年号。但是，FoxPro 2.5b for Windows 提供了可以将公元年号设置为 4 位表示的命令（即 SET CENTURY ON，将在第 5 章介绍）。

4. 逻辑型常量

逻辑型常量只有两个，用于表达真和假。逻辑常量用圆点符界定，被界定的常量可以是：T、t、Y、y、F、f、N、n 之一。其中，.T.、.t.、.Y.、.y. 表示真值；.F.、.f.、.N.、.n. 表示假值。

2.1.2 变量

为了识别和使用计算机中存储的数据，需要用标识符标记数据的存储区域，这样的标识符叫做变量名，简称变量。一个变量名既代表数据的存储区域，又代表该存储区域中存放的数据。由于存储区域中的数据是可以随时改变的，所以一个变量名所代表的数据也是可以变化的。

向变量中存或从变量中取数据的原则是：从变量所代表的存储区域取数，不改变原来的存储内容；向变量所代表的存储区域存放数据，将取代原来存储的数据。

FoxPro 的变量可分为内存变量和字段变量。这里先介绍内存变量，下一章再介绍字段变量。

内存变量不依赖于数据库文件，是独立存在于内存中的变量，使用时可临时定义。内存变量可用于存放常量、程序执行过程中的中间结果或最终结果，也可以作为控制变量控制程序的运行等。

内存变量又分为用户内存变量和系统内存变量。系统内存变量是由系统定义并赋值的，用户内存变量是用户定义并赋值的。

用字母打头的字母数字串作为用户内存变量的名称。如果构成内存变量名称的字符超过 10 个半角字符，则只使用前 10 个 ASCII 字符或前 5 个汉字。因为系统内存变量的名称

都是以下划线作为首字符的，所以用户命名的内存变量最好不以下划线为首字符。

内存变量有 6 种类型，分别为 C 型、N 型、D 型、L 型、W 型和 S 型。其中，W 型内存变量是窗口型变量，待介绍窗口技术(第 12 章)时再作说明；S 型内存变量是屏幕型变量，用于保存当前屏幕所显示的内容。

不论在交互方式或程序方式下，经常要使用内存变量。为此，下面先介绍几个对内存变量进行操作的命令。

1. 内存变量的定义和存取内存变量数据的命令

为内存变量指定名称和类型，叫做内存变量的定义。FoxPro 的很多操作命令，都具有定义内存变量的功能。例如，下面就是两个定义内存变量并赋值的命令：

赋值命令：〈内存变量名〉=〈表达式〉

存数命令：STORE 〈表达式〉 TO 〈内存变量名表〉

这两个命令的功能都是将〈表达式〉的值赋给〈内存变量〉。不同的是，赋值命令只能给一个变量赋值，而存数命令可以把一个〈表达式〉的值同时赋给多个变量。关于“表达式”的概念待后述，这里可以先把表达式理解为常量、内存变量或由常量和内存变量组成的式子。两条命令都给所使用的内存变量指定了名称，而内存变量的类型遵从〈表达式〉的数据类型。

注意，赋值和数学中的等式是不同的概念，例如， $N=N+1$ 是常见的一类赋值命令，但是如果把它理解成等式，那么不论 N 取什么值，这个等式都是不能成立的。

例如，在命令窗口依次打入下列命令，观察被赋值后的内存变量输出的内容。

```
专业="数学"
? 专业
班级="9501 班"
? 班级
STORE "张兰" TO 姓名
?? 姓名
STORE 2 TO a,b,c
? 2*a,b+c
```

说明：

(1) ? 和 ?? 都是输出命令，其命令格式分别是：

? 〈输出项目表〉 和 ?? 〈输出项目表〉

其中 ? 是换行输出命令，?? 是不换行输出命令。例如，在上例中，执行了换行输出命令：? 专业〈回车〉和 ? 班级〈回车〉之后，在屏幕上显示的输出内容“数学”和“9501 班”各占一行；而执行了不换行输出命令 ?? 姓名〈回车〉之后，在前面已输出的“9501 班”后面输出“张兰”。

当〈表达式表〉中包含有多个表达式时，各表达式之间要用逗号分隔。

(2) 规定在算术运算中，用符号 * 表示乘号。

2. 显示内存变量的命令

显示内存变量的命令除了上面已经介绍过的 ? 和 ?? 命令之外，还有两个能同时显示多个内存变量的命令，其命令格式分别是：

LIST MEMO [LIKE <内存变量通配名>] [TO PRINTER|TO FILE <文件名>]
DISPLAY MEMO [LIKE <内存变量通配名>] [TO PRINTER|TO FILE <文件名>]

说明:

(1) LIST MEMO 是连续显示的, DISPLAY MEMO 是分屏显示的。

(2) LIKE <内存变量通配名>表示可以用通配符号指定所要显示的内存变量名。通配符号有“?”和“*”,如果变量名称中某一个字符的位置上使用了通配符“?”,则该位置上不论是什么字符,都是本命令中所指定的变量;如果变量名称中使用了通配符“*”,则该位置上不论是什么字符,也不论是几个字符,都是本命令中所指定的变量。

(3) TO PRINTER|TO FILE <文件名>用于将显示的内存变量在打印机上打印出来或存入指定的磁盘文件。当选择 TO FILE <文件名>时,则系统将会自动附加文件的扩展名 MEM。

例如,下面的各条命令都是显示内存变量的命令:

```
STOR 123 TO a1,a2,a3,a4
LIST MEMO                DISPLAY MEMO
LIST MEMO LIKE a?        DISPLAY MEMO LIKE a?
LIST MEMO LIKE a*        DISPLAY MEMO LIKE a*
LIST MEMO TO FILE m1     DISPLAY MEMO TO FILE m1
```

上面最后一条命令将产生名为 M1.MEM 的磁盘文件。

3. 将内存变量保存到磁盘文件的命令

当退出 FoxPro 时,用户内存变量被清除。为了能“永久”保存内存变量,可以把需要保留的内存变量保存到磁盘的 .MEM 文件中。其保存命令是:

SAVE TO <内存变量文件名> [ALL LIKE <内存变量通配名>]

例如:

```
STORE 123.0000 TO a1,a2,a3,a4,a5 (定义内存变量并赋值)
SAVE TO m1 ALL LIKE a? (保存内存变量的命令)
```

4. 从 .MEM 文件中恢复内存变量的命令

命令格式: RESTORE FROM <内存变量文件名> [ADDI]

选择 ADDI,则保留当前用户内存变量,追加所恢复的内存变量。否则,当前用户内存变量被清除。例如,执行命令 RESTORE FROM m1<回车>后,原来已经定义过(或者说使用过)的所有内存变量都被清除。

5. 删除指定的内存变量的命令

命令格式一: RELEASE <内存变量名 1[,...]>

命令格式二: RELEASE ALL [LIKE <内存变量通配名>]

例如:

```
RELEASE a1,a2 (删除内存变量 a1、a2)
RELEASE ALL LIKE a? (删除首字符是 a、第二个是任意字符的变量)
```

6. 有关屏幕型内存变量的命令

(1) 把当前屏幕显示内容存入 S 型内存变量的命令: