

第3章

Visual FoxPro 语言基础

Visual FoxPro 既支持面向过程的结构化程序设计方法开发应用程序,也支持面向对象由事件驱动的程序设计方法开发应用程序。无论采用哪种方法,都需要掌握 Visual FoxPro 的基本命令、程序结构和专用于编程的语句。本章介绍在编写面向对象事件驱动程序中常用的命令格式或编程语句的格式,以及在数据库操作及应用程序开发中经常涉及的一些十分重要的概念和基本知识,包括 Visual FoxPro 的命令格式或语句的格式、数据类型、常量、变量、表达式和函数。只有正确地理解和掌握这些重要的概念和基本知识,才能准确地使用命令和开发数据库应用程序。

3.1 命令规则

Visual FoxPro 以命令的方式执行语言的各种功能(包括数据定义和数据操作功能),它属于命令式语言。它的一条命令相当于一般高级语言中的一段程序,可以完成相当复杂的功能。

3.1.1 命令的一般格式

一般地说,Visual FoxPro 的某些操作(例如表的各种操作),既可以通过菜单操作方式完成,也可以通过命令操作或程序方式完成。无论采用哪种操作方式,都是执行相应的 Visual FoxPro 命令。所以,要使操作准确无误,就必须正确地了解和运用命令。为此,先介绍 Visual FoxPro 命令的一般格式,以便使用户能迅速正确地掌握和运用命令。

Visual FoxPro 命令在程序中通常称做语句,总是由一个称为命令字的动词开头,后随一个宾语和若干子句(称为命令子句),用来说明命令的操作对象、操作结果与操作条件。表 3-1 给出了若干简单 Visual FoxPro 命令的示例。

1. 命令特点

从以上的示例不难看出,Visual FoxPro 的命令具有以下特点。

(1) 采用英文祈使句的形式,命令的各部分简洁规范,初通英语的人都能看懂。

表 3-1

Visual FoxPro 命令示例	说 明
USExs.dbf	打开文件名为 xs 的表文件
LIST	列表显示内容
LIST FOR 入学成绩<550	列 xs 表入学成绩小于 550 分的学生
REPLACE 入学成绩 WHIT 1.3 * 入学成绩 FOR 应届否	应届学生的入学成绩乘以系数 1.3

(2) 操作对象、结果(目的地)和条件均可用命令子句的形式来表示。命令子句的数量不限,顺序不拘。它们使命令的附属功能可以方便地增删,十分灵活。

(3) 命令中只讲对操作的要求,不描述具体的操作过程,言简意赅,所以又称为“非过程化”语言,而常见的高级语言都是“过程化”语言。

另外,Visual FoxPro 的命令既可在 Visual FoxPro 系统的命令窗口中,逐条使用交互的方式执行,又可编写成程序,以“程序文件”的方式执行。

2. 命令分类

Visual FoxPro 拥有数百条命令,大致要分为以下 7 类。

- (1) 建立和维护数据库的命令;
- (2) 数据查询命令;
- (3) 程序设计命令,包括程序控制、输入输出、打印设计、运行环境设置等命令;
- (4) 界面设计命令,包括菜单设计、窗口设计、表单(包括其中的控件)设计等命令;
- (5) 文件和程序的管理命令;
- (6) 面向对象的设计命令;
- (7) 其他命令;

全面介绍这些命令需要很大的篇幅,感兴趣的读者可以参阅相关资料。

3. 命令格式

Visual FoxPro 的命令很多,各有不同的格式和形式,但其中很多命令都具有一种类似的形式,称为命令的一般格式。为了便于以后的讲述,在此介绍这种一般格式,以后在介绍一般命令时,凡与一般格式意义相同的部分不再重复。

格式:

```
<命令字> [<范围>] [FIELDS<表达式表>] [FOR<条件 1>] [WHILE<条件 2>]
```

说明:

- (1) 必须以命令字打头。
- (2) 命令字(必须是英文)之后及各部分之间要用空格隔开。

- (3) 命令字、子句、函数名大多可最少写前 4 个字符,大、小写等效。
- (4) 一行只能写一条命令,一条命令的总长度(包括命令中的所有空格)不能超过 254 个字符(一个汉字以两个字符计算)。若命令较长,可用“;”作为命令或语句的续行符号,后跟一个回车符,转到下一行去继续输入这条命令。系统在执行时,将把它们看作一个整体。当然,也可不管它,连续输入一条命令的全部内容,这样也不影响一条命令的完整性,但决不可在一行快输满时,用 Enter 键换到下一行继续输入本条命令,这样就破坏了命令的完整性。
- (5) 变量名、字段名和文件名应避免与命令字、关键字或函数名同名,以免运行时发生混乱。
- (6) 命令格式中的符号约定:
- 命令中的 []、|……<> 符号是为了表述上的方便,它们都不是命令本身的语法成分,使用时不能照原样输入,相关符号说明见表 3-2。

表 3-2 命令中符号的约定

命令中的符号	符 号 约 定
[]	表示其中内容是可选项,根据具体情况决定是否选用
	表示两边的部分只能选用其中的一个
<>	表示其中内容要以实际名称或参数代入
...	表示可以有任意一个类似参数,各参数间用逗号隔开

3.1.2 命令字句

1. 命令字

命令关键字(命令字)就是一个英文动词,表示要执行的操作。例如 STORE、USE、LIST 等都是命令字。Visual FoxPro 命令在程序中通常称做语句。当一个命令字的字母超过 4 个时,可从第 5 个字母起省略。例如,CREATE 可省略为 CREA、DISPLAY 可省略为 DISP 等。但也有例外,例如后面要介绍的 LOCATE 和 LOCAL 命令。

所有的命令都必须有一个命令字,它决定此命令的性质。

2. 命令子句

前面的命令一般格式可知,命令动词后面的各个项都是命令动词的短语,也称命令子句,如范围子句、条件子句等。为了表述上的方便,在命令子句的左右用方括号“[]”括起来表示可选项,用户可根据实际需要来决定取舍命令子句。

(1) 范围子句。规定命令操作数据表的有效记录的范围。Visual FoxPro 规定如下 5 种范围格式。

- ① NEXT *n*: 范围为从表中当前记录开始的 *n* 个记录,*n* 是一个具体的十进制数。
- ② RECORD *n*: 范围仅为表中的第 *n* 号记录。

- ③ REST: 范围为从表中当前记录起到末记录止。
 - ④ ALL: 范围为从表中首记录开始的所有记录。
 - ⑤ 省略: 若条件项同时省略,使用的默认值不同,命令会有不同的含义,或等价于 ALL,或仅作用于当前记录。条件项以 FOR 开始时,范围为 ALL,以 WHILE 开始时,范围为从当前记录开始的所有记录。
- 范围参数示例如图 3-1 所示。

	学号	姓名	性别	生日	班级	应属否	入学成绩
	20150011	李中华	男	10/01/99	会计1501	T	611
	20150012	肖萌	女	02/04/98	会计1501	T	600
	20150014	李铭	男	12/31/97	会计1501	T	599
当前记录	20150020	张力	男	01/01/95	会计1501	F	585
	20151234	傅丹	男	03/20/98	会计1502	T	630
	20151240	李园	女	01/01/96	会计1502	F	588
	20151255	华晓天	男	07/17/99	会计1502	T	590
NEXT9	20154001	刘冬	女	11/09/98	经济1501	T	633
	20154019	严岩	男	03/13/97	经济1501	T	615
	20154025	王平	男	06/19/98	经济1501	T	600
REST	20156001	江锦添	男	04/27/95	经济1502	F	610
	20156200	李冬冬	女	12/25/98	经济1502	T	570
	20156215	赵天宇	男	08/15/97	经济1502	T	585
	20156345	于天	女	09/10/98	经济1502	T	640
	20156500	梅媚	女	05/20/99	经济1502	T	580

图 3-1 学生表 xs. dbf 范围值示例

说明：表中除第一行是各个数据列的列名以外,其余各行称为该表的记录,共有 15 行,即 15 条记录,第一行称为首记录,最后一行称为末记录,箭头所指为当前记录。

(2) FOR<条件>子句。<条件>是一个返回值为逻辑型的表达式,表示只对逻辑表达式取真值的记录进行规定的操作。如默认范围子句和 WHILE<条件>子句,系统隐含从首记录搜索起直到末记录,凡满足条件的记录都做规定的操作。

(3) WHILE<条件>子句。它表示如默认范围子句,则从当前记录开始测试是否满足其逻辑表达式的条件,如满足便对该记录进行规定操作。接着对下一记录进行测试,如仍满足就再次进行规定操作;否则停止操作,不管其后是否还有满足条件的记录,此时记录指针就指向首先未满足条件的记录。

例如,表文件 kc. dbf 如下：

记录号	课程编号	课程名称	学分
1	A101	高等数学	6.0
2	A201	哲学	4.0
3	A301	大学英语	4.0
4	A401	微观经济学	5.0
5	A402	宏观经济学	5.0
6	A403	计量经济学	5.0
7	B101	数值分析	4.5
8	B102	运筹学	4.5
9	B103	概率与统计	4.0
10	B501	数据结构	4.0

11	B502	操作系统原理	3.5
12	C001	大学语文	3.0
13	C010	法语	3.0
14	C011	德语	3.0

① 用 FOR 子句列清单,则所有满足条件的记录都被列出:

```
LIST FOR 学分<=4.0
```

执行后结果显示如下:

记录号	课程编号	课程名称	学分
2	A201	哲学	4.0
3	A301	大学英语	4.0
9	B103	概率与统计	4.0
10	B501	数据结构	4.0
11	B502	操作系统原理	3.5
12	C001	大学语文	3.0
13	C010	法语	3.0
14	C011	德语	3.0

② 用 WHILE 子句列清单,则遇到第一个不满足条件的记录就停止显示:

```
LIST ALL WHILE 学分>=5.0
```

执行后结果显示如下:

记录号	课程编号	课程名称	学分
1	A101	高等数学	6.0

当一条命令中既有 FOR 子句,也有 WHILE 子句时,在 WHILE 条件首次不满足前,将对满足 FOR 条件的所有记录进行操作;一旦 WHILE 条件不满足,则停止操作。即 WHILE 子句的优先级高于 FOR 子句。为方便理解,本书后面讲到条件时,都以 FOR 子句为准,读者可自行上机实验 WHILE 子句的含义。

命令执行完毕时,表指针一般指向范围内最后一记录。但当范围为 ALL 时,指针值为表文件记录个数加 1,即指向最后一记录的后面(注意,默认条件以 FOR 开始)。

(4) FIELDS<表达式表>。范围、FOR 与 WHILE 子句都能将表中需要操作的记录筛选出来,FIELDS 子句则能确定需要操作的字段。该子句的保留字 FIELDS 在某些命令中可以省略,而<表达式表>用来列出需要的字段,甚至较为复杂的表达式,LIST 命令将按筛选得到的记录依次算出表达式的值,并显示出来。

此项往往是表文件中字段名清单,或是包含字段名的表达式清单。其中各项用英文状态下的半角逗号分隔。省略此项,一般等价于表文件中全部字段。

例如,打开学生信息表文件(xs.dbf),记录指针指向 2 号记录时,执行命令:

```
LIST NEXT 5 FIELDS 学号,姓名,入学成绩
```

执行后结果显示如下:

记录号	学号	姓名	入学成绩
2	20150012	肖萌	600
3	20150014	李铭	599
4	20150020	张力	585
5	20151234	傅丹	630
6	20151240	李园	588

范围子句、FOR 子句与 WHILE 子句用于把表文件中需要操作的记录筛选出来，FIELDS 子句用于确定需要操作的字段。在命令中灵活地使用多个子句，将大大地增强各命令的数据处理能力。命令字必须是一条命令的第一项，而各个命令子句(任选项)的顺序可任意排列。一条命令的各部分之间必须有空格(FOR/WHILE 与<条件>之间也必须有空格)，但空格个数不限。

3.1.3 命令的执行方式

Visual FoxPro 可以支持两类不同的命令执行方式，即交互操作方式与程序执行方式。现分述如下：

1. 交互操作方式

Visual FoxPro 中的交互操作方式包含两种：Visual FoxPro 的命令或菜单操作方式和运用菜单、窗口与对话框技术的图形界面操作方式。

所谓 Visual FoxPro 的命令操作方式是利用图 3-2 所示的 Visual FoxPro 系统中的命令窗口来实现的，用户要记住命令的格式与功能，在 Visual FoxPro 系统的命令窗口中从键盘上输入单个的操作命令和系统命令，完成对数据库的操作管理和系统环境的设置；也可以建立及运行命令文件。

例如，显示学生表 xs.dbf 中入学成绩在 600 分以上的学生的操作命令。

```
USE xs
LIST FOR 入学成绩>630
```

Visual FoxPro 命令窗口键入的命令及在工作区的执行结果，如图 3-2 所示。

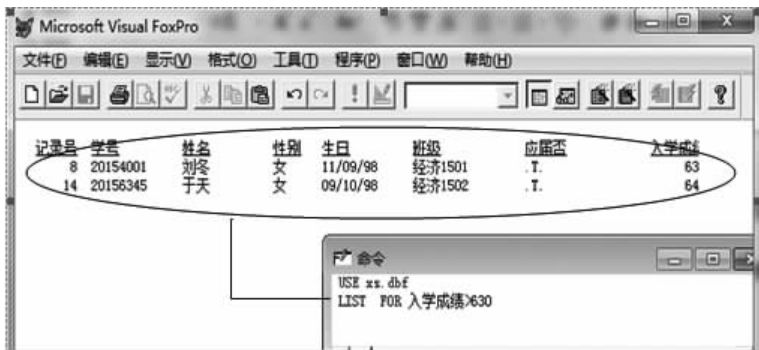


图 3-2 Visual FoxPro 的命令操作窗口

所谓菜单方式,即通过打开不同的菜单选择并完成不同的操作,如图 3-3 所示。

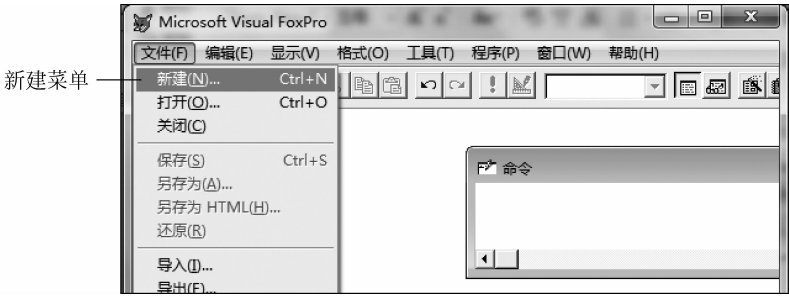


图 3-3 Visual FoxPro 系统主菜单

在 Visual FoxPro 环境下,可以通过系统菜单提供的选项,对数据库资源进行操作管理和对系统环境进行设置。例如,将系统日期格式由 mm/dd/yyyy 格式改为“年月日”的日期格式,可以选择“工具”|“选项”菜单命令,在弹出的“选项”对话框的“区域”选项卡中重设日期格式,如图 3-4 所示。



图 3-4 系统环境设置对话框

交互操作方式的可视化图形界面操作,无须记忆 Visual FoxPro 的命令格式与功能,更为广大用户熟悉和欢迎。

例如,建立学生的必修成绩表 bxcj.dbf 的操作。

选择“文件”|“新建”菜单命令,如图 3-3 所示,在弹出的“新建”对话框中选择“表”,单击“新建”按钮,在“创建”对话框的“输入表名”文本框中输入“bxcj.dbf”,在表设计器中输入学生表的字段名及其类型等操作完成建立表的操作,如图 3-5 所示。

综上所述,在交互方式对数据库资源进行操作时,建议用户以综合运用菜单、窗口和



图 3-5 建立表文件系列对话框

对话框技术的图形界面操作方式为主、命令方式为辅，并结合 Visual FoxPro 提供的向导、各类设计器等辅助设计工具，以多元化操作方式和综合应用为原则，完成 Visual FoxPro 直观的可视化界面操作。

2. 程序执行方式

程序文件(简称程序)也常叫做命令文件，在 Visual FoxPro 环境下，运用程序文件方式进行数据库管理，是通过程序文件编辑工具，将对数据库资源进行操作管理的命令和对系统环境进行设置的命令，集中在一个以 PRG 为扩展名的命令文件中，然后再通过菜单方式或命令方式运行该命令文件。

程序方式就是通过程序文件(或称为命令文件)中的命令完成不同的操作。例如，用程序方式显示 bxcj.dbf 表中的全部数据，并逻辑删除第 6 条记录。操作步骤如下。

① 利用程序编辑工具建立一个程序文件 program1.prg，内容如下：

```
USE d:\vfp98\bxcj.dbf EXCLUSIVE
BROWSE LAST
DELETE RECORD 6
```

② 利用菜单选项或命令运行程序文件 program1. prg。

程序会逐个地执行每一个操作命令,完成显示学生必修成绩 bxcj. dbf 表中的全部数据,并逻辑删除第 6 条记录的操作。

如果用户比较熟悉 Visual FoxPro 系统,用交互操作方式则比较简单方便。若频繁操作数据库,用户操作与计算机的执行互相交叉,则计算机执行的高度自动化将难以体现。为此在解决某些实际问题时,会将 Visual FoxPro 的命令编成特定的序列,并将它们存入程序文件。用户需要时,只需通过特定的命令(例如 DO 命令)调用程序文件,Visual FoxPro 就能自动执行这一程序文件,自动完成序列命令的执行。

程序执行方式不仅运行效率高,而且可重复执行。要执行几次就调用几次,何时调用便何时执行。另一个好处是,虽然程序员需熟悉 Visual FoxPro 的命令和掌握编程的方法,而使用程序的人员却只须了解程序的运行步骤和运行过程中的人机交互要求,对程序的内部结构和其中的命令可不必知道。还需指出,开发 Visual FoxPro 应用程序要求同时进行结构化程序设计与面向对象程序设计,其庞大的命令集往往令初学者望而生畏。幸运的是 Visual FoxPro 提供了大量的辅助设计工具,不仅可直接产生应用程序所需要的界面,而且能自动生成 Visual FoxPro 的程序代码。因此在一般情况下,仅有少量代码需要由用户手工编写。这些工具充分体现了“可视化程序设计”的优越性。

3.2 数据类型

记载信息的符号组合称为数据,人们用数据来描述实体的对象及其属性。数据类型是数据的基本属性,只有相同类型的数据才可以直接进行运算,否则就会发生数据类型不匹配的错误。

Visual FoxPro 是一种关系型数据库管理软件,在关系型数据库中把描述每一实体集合的数据表示成一张二维表。描述学生信息的一张二维表如图 3-6 所示。

记录号	学号	姓名	性别	生日	班级	应届否	入学成绩	照片	曾获奖励
1	20150011	李中华	男	10/01/1999	会计1501	.T.	611	Gen	memo
2	20150012	肖萌	女	02/04/1998	会计1501	.T.	600	Gen	memo
3	20150014	李铭	男	12/31/1997	会计1501	.T.	599	Gen	memo
4	20150020	张力	男	01/01/1995	会计1501	.F.	585	Gen	memo
5	20151234	傅丹	男	03/20/1998	会计1502	.T.	630	Gen	memo
6	20151240	李园	女	01/01/1996	会计1502	.F.	588	Gen	memo
7	20151255	华晓天	男	07/17/1999	会计1502	.T.	590	Gen	memo
8	20154001	刘冬	女	11/09/1998	经济1501	.T.	633	Gen	memo
9	20154019	严岩	男	03/13/1997	经济1501	.T.	615	Gen	memo
10	20154025	王平	男	06/19/1998	经济1501	.T.	600	Gen	memo
11	20156001	江锦添	男	04/27/1995	经济1502	.F.	610	gen	memo
12	20156200	李冬冬	女	12/25/1998	经济1502	.T.	570	gen	memo
13	20156215	赵天宁	男	08/15/1997	经济1502	.T.	585	gen	memo
14	20156345	于天	女	09/10/1998	经济1502	.T.	640	gen	memo
15	20156500	梅媚	女	05/20/1999	经济1502	.T.	580	gen	memo

图 3-6 学生信息表

图 3-6 中,共有 15 名学生的记录数据。记录有 9 个字段。第一行是描述实体集合的记录型,即记录结构。其中 9 个字段名分别为学号、姓名、性别、出日、班级、应届否、入学成绩、照片及曾获奖励。

在 Visual FoxPro 系统中,定义了 13 种字段类型和 7 种数据类型。13 种字段类型是字符型、数值型、浮点型、双精度型、整型、货币型、日期型、日期时间型、逻辑型、备注型、通用型、二进制字符型和二进制备注型;而 7 种数据类型是字符型、数值型、货币型、日期型、日期时间型、逻辑型和通用型。表 3-3 列出了学生信息表的字段类型及数据类型。

表 3-3 学生信息表的字段类型及数据类型

字段	字段名	字段类型	数据(字段值)	数据类型
1	学号	字符型	20150011	字符型
2	姓名	字符型	李中华	字符型
3	性别	字符型	男	字符型
4	生日	日期型	10/01/99	日期型
5	班级	字符型	会计 1501	字符型
6	应届否	逻辑型	T	逻辑型
7	入学成绩	整型	611	整型
8	照片	通用型	照片	通用型
9	曾获奖励	备注型	获奖项目	字符型

字段为表文件所特有,而数据既可做表文件中的字段内容,也可以做内存变量内容或做常量使用。下面比较详细地介绍常用字段和数据类型。

3.2.1 字符型字段和字符型数据

字符型字段用作存放字符型数据。字符型(Character)数据描述不具有计算能力的文字数据类型,是最常用的数据类型之一。字符型数据由汉字和 ASCII 码字符集中可打印字符(英文字母、数字字符、空格及其他专用符号)组成,字符型字段中的字符数据长度为 0~254 个字符,每个字符占 1B 空间。上述学生信息表中的学号和姓名等字段就属于字符型字段,而存储的学号和姓名数据就属于字符型数据。

3.2.2 数值型、浮动型、双精度型和整型字段与数值型数据

数值型数据是描述数量的数据类型,是最常用的数据类型之一。在 Visual FoxPro 系统中它被细分为以下 4 种类型。

1. 数值型

数值型字段按每位数 1B 的长度存放数值数据,数值型(Numeric)数据是由数字(0~

9)、一个符号(十或一)和一个小数点(.)组成,最大长度为 20 位(包括十、一号和小数点)。

2. 浮点型

浮点型字段存放浮点型(Float)数据,是数值型数据的一种,与数值型数据完全等价。浮点型数据只是在存储形式上采取浮点格式。增设浮点型数据的主要目的是使计算有更大的数据表示范围。

3. 双精度型

双精度型字段用于存放双精度型(Double)数据,常用于科学计算,是更高精度的数值型数据。它只用于数据表中的字段类型的定义,并采用固定长度浮点格式存储。

4. 整型

整型字段存放整型(Integer)数据(不包含小数部分的数值型数据)。它只用于数据表中的字段类型的定义,整型数据以二进制形式存储,上述学生信息表中的入学成绩字段就属于整型字段,而存储的入学成绩就属于整型数据。

3.2.3 货币型字段和货币型数据

货币型字段用于存放货币型(Currency)数据。在使用货币值时,可以使用货币型来代替数值型,货币型数据取值的范围是-922 337 203 685 477.5807~922 337 203 685 477.5807。

3.2.4 日期型字段和日期型数据

日期型字段用于存放表示日期的日期型(Date)数据。常用日期格式为:“yyyymmdd”或“mm-dd-yy”等格式。在学生信息中的生日字段就属于日期型字段,其出生的年月日数据就是日期型数据。日期型字段有固定宽度,占用 8B 空间,其中,yyyy 为年,mm 为月,dd 为日。

3.2.5 日期时间型字段和日期时间型数据

日期时间型字段用于存放日期和时间值。日期时间型(Date Time)数据的格式为:yyyy-mm-dd hh:mm:ss am/pm,其中,yyyy 为年,前两个 mm 代表月;dd 为日;hh 为时间中的小时;后两个 mm 代表时间中的分钟;ss 为时间中的秒;am/pm 是每天时间设定为 12 小时格式,时间设定为 24 小时不需设定。

3.2.6 逻辑型字段和逻辑型数据

逻辑型字段用于存放逻辑型(Logical)数据。

逻辑型字段用于存储只有两个值的数据。存入的值只有真(.T.)和假(.F.)两种状态。常用于做逻辑判断或用于描述只有两种状态的数据,例如,婚否只有已婚和未婚,常用“真”表示已婚,而用“假”表示未婚。逻辑型字段有固定宽度,占用 1B 空间。在输入逻

辑型数据时可以用 T、t、Y、y 中的任何一个字符代表“真”，而用 F、f、N、n 中的任何一个字符代表“假”。学生信息中的“应届否”字段就被定义为逻辑型字段，可用“真”表示应届毕业生，而用“假”表示往届毕业生。

3.2.7 备注型字段

备注型字段可以存放字符型 (Memo) 信息，如文本、源程序代码等，使其得到了广泛应用。它常用于记录信息可有可无，可长可短的情况，如学生信息表中的简历一项，有些人的简历内容可能长一些，而有些人的个人简历内容可能短一些。此外，备注型字段还可以用于提供运行时的帮助信息。

记录在备注项中的信息，实际上并不存放在表文件中，而是存放在与表文件同名，但扩展名为 .FPT 的文件中。当创建文件时，如果定义了备注型字段，则相应的备注文件就会自动生成，当其建成后也会随表文件一起打开。

3.2.8 通用型字段和通用型数据

通用型字段是用于存储 OLE 对象的数据。通用型数据中的 OLE 对象可以是照片、电子表格、声音、设计分析图及字符型数据等，它只能用于数据表中字段的定义，有了这种类型字段就使得 Visual FoxPro 成为全方位数据库。和备注字段一样，通用型字段数据也存入与表文件同名而扩展名为 FPT 的文件中。

3.3 常量与变量

常量是一个命名的数据项，在整个操作过程中其值保持不变，常量在程序和 Visual FoxPro 命令中的表示有一些规定。而变量在操作过程中允许随时改变其值。Visual FoxPro 定义了常量、用户内存变量、系统内存变量和字段变量。

3.3.1 常量

Visual FoxPro 定义了 5 种类型的常量：数值型常量、字符型常量、日期型和日期时间型常量、逻辑型常量和货币型常量。

1. 数值型常量

数值型常量可以是带小数的常量 (实数) 和不带小数的常量 (整数)。例如 5、28.23、-3 等在程序或命令中都是数值型常量。

2. 字符型常量

必须用英文半角状态下的单引号、双引号括或方括号括起，这些符号总称为字符括号。给出三种字符括号是为了当字符括号本身是字符型常量的组成部分时，就应选用另一种字符括号。例如 '你好'、"你好"、[你好]、['你好'] 等在程序中都是正确的表示了“你好”和“'你好'”的字符型常量。

3. 逻辑型常量

逻辑型常量有两个,一个是“真”,另一个是“假”。在 Visual FoxPro 中用. T. \. t. \. Y. \. y. 表示“真”,用. F. \. f. \. N. \. n. 表示“假”,注意两边的小圆点是逻辑值的定界符不能丢掉但可以用空格代替。

4. 日期型常量和日期时间型常量

(1) 日期型常量必须用花括号括起来,正确格式是{^yyyy-mm-dd}。例如,{^2015-12-02}、{^2015/12/02}或{^2015. 12. 02}都是日期型常量的正确写法。空白的日期可表示为{}或{/}。

(2) 日期时间型常量也必须用花括号括起来,例如,{^2015-08-11 2:57:00 pm}、{^2015. 09. 01 9:15 am}或{^2015/12/02 6:50:00 pm}都是日期时间型常量的正确写法。注意:日期和时间之间必须用半角的逗号或空格分隔开。

严格的日期格式:

```
{yyyy-mm-dd [,][hh[:mm[:ss]]][an/pm]}
```

格式中的符号^是为了解决计算机 2000 年问题,又叫“千年虫”而加入的,表明该日期格式是严格的,并按照 YMD 的格式来解释日期,其中分隔符-可用/或“.”代替,例如{^2015/12/02}和{^2015. 12. 02}。

5. 货币型常量

货币型常量以\$符号开头,并四舍五入到小数点后 4 位。例如,货币型常量\$ 147. 72706,实际只按 147. 7271 算。

3.3.2 变量

在 Visual FoxPro 中有 4 种变量:内存变量、数组变量、字段变量和系统变量。内存变量是存放单个数据的内存单元,数组变量是存放多个数据的内存单元组,字段变量则是存放在数据表中的数据项。

变量名的命名规则如下:

每个变量都有一个名称,叫做变量名,Visual FoxPro 通过相应的变量名来使用变量。

变量名的命名规则是,由字母、数字及下划线组成,以字母或下划线开头,长度为 1 ~ 128 个字符。但要注意,不能使用 Visual FoxPro 的保留字。中文 Visual FoxPro 中,可以使用汉字作变量名,可以汉字开头,每个汉字占 2 个字符。

例如,定义合法的变量名示例。

ABCD	P0000	_ymmei	姓名	是合法的变量名
7ab	IF	A[b]5	-ymmei	89DFF 是不合法的变量名

1. 字段变量

字段变量是指数据表中的字段,它是建立数据表时定义的一类变量。例如,表文件

xs.dbf 中的学号、姓名、生日等都是字段变量。说字段是变量,是由于对于某一个字段,它的值因记录而异。例如,学生表文件有多条记录,则表的各个字段就有多个值,移动记录指针的位置到所需记录,就可以找出各字段变量的当前值。

2. 内存变量

内存变量是内存中的一些临时工作单元,独立于数据库各个表文件,常用来保存所需要的常数、中间结果或对数据表 and 数据库进行某种处理后的结果等。Visual FoxPro 定义了 6 种类型的内存变量,即字符型、数值型、逻辑型、日期型、日期时间型和屏幕型内存变量。对屏幕型内存变量,可用 SAVE SCREEN TO <内存变量>命令存放当前屏幕上的信息,用 RESTORE SCREEN <内存变量>命令从屏幕内存变量恢复屏幕信息。Visual FoxPro 中允许定义数万个内存变量。

(1) 内存变量的赋值。在命令窗口中或程序语句中均可以对内存变量赋值。

格式 1:

```
STORE<表达式>TO<内存变量名表>
```

格式 2:

```
<内存变量名>=<表达式>
```

功能: 用表达式的值给内存变量赋值。

例如以下赋值命令:

```
STORE 6 TO a1,a2
STORE"非诚勿扰"TO 电影
m=6
rq={^2015-10-02}
性别=.t.
```

说明:

① 两种命令的主要区别在于,格式 1 可以对多个内存变量同时赋相同的值,而格式 2 仅对一个内存变量赋值。注意,当内存变量名表中有多个变量时,变量与变量之间应用逗号分开。

② 内存变量应当“先定义后使用”,格式 1 和格式 2 均是在赋值给内存变量的同时定义内存变量,确定其数据类型

③ 上述命令分别给内存变量 a1、a2 和 m 赋予数值型常量 6,使其成为数值型的内存变量;给内存变量“电影”赋予字符型常量“非诚勿扰”,使其成为字符型的变量;给内存变量 rq 和“性别”分别赋予日期型常量 2015-12-02 和逻辑型常量,使其分别成为日期型的内存变量和逻辑型的内存变量。

(2) 显示内存变量的值。如果只是查看内存变量的值,也可用表达式显示命令?/?实现。

格式:

?/??<内存变量名表>

功能：换行或在当前位置显示内存变量名表中每个内存变量的值。

例如，显示上例内存变量 a1 和 rq 赋值后的值。

```
? a1,rq
```

结果为：

```
6 2015-12-02
```

(3) 显示内存变量的定义信息。若想了解内存变量的定义信息，如内存变量的名称，类型和值等，可用下面的显示命令。

格式：

```
DISPLAY/LIST MEMORY [LIKE<通配符>] [TO PRINTER/TO FILE<文件名>]
```

功能：显示内存变量的当前内容，可选择打印或者将这这些内容送到一个文本文件中。

说明：

DISPLAY MEMORY 表示命令字，显示内存变量的定义信息，如名称，类型和值；

LIST MEMORY 表示命令字，显示内存变量的定义信息，如名称，类型和值；

LIKE <通配符>表示命令子句，通配符用 * 和“?”，代表多个字符和单个字符；

TO FILE<文件名>表示定义信息输出到扩展名是 TXT 的文本文件中。

例如，定义内存变量 a 的值为 2，a1 的值为“多媒体计算机”，并在屏幕上显示出来。

```
a=2
```

```
a1="多媒体计算机"
```

```
LIST MEMORY LIKE "a * "
```

命令的显示结果为：

```
A      Pub  N   2              (      2.00000000)
```

```
A1     Pub  C   "多媒体计算机"
```

(4) 内存变量的保存与恢复。当退出 Visual FoxPro 系统后，用户所建立的内存变量将不会存在，如果希望保存这些内存变量的定义，可用下面的命令将它们保存到内存变量文件中。

格式：

```
SAVE TO <内存变量文件名> [ALL LIKE <通配符>/ALL EXCEPT<通配符>]
```

功能：将当前内存中的内存变量存放到内存变量文件中。

说明：内存变量文件的扩展名为 MEM；此项省略，则将所有内存变量（系统变量除外）存放到内存变量文件中。

如果要重新使用已保存在内存变量文件中的内存变量，可用下列命令进行恢复，将内存变量重新调入内存。

RESTORE FROM<内存变量文件名> [ADDITIVE]

(5) 内存变量的删除。

为节省存储空间,不用的内存变量应使用删除命令来释放其所占的内存空间。

格式 1:

CLEAR MEMORY

格式 2:

RELEASE<内存变量名表>

格式 3:

RELEASE ALL [LIKE/EXCEPT<通配符>]

功能: 格式 1 删除内存中所有内存变量;格式 2 删除内存变量名表中指定的内存变量;格式 3 利用通配符选择删除内存变量或全部。

3. 数组变量

数组变量(数组)是按一定顺利排列的一组内存变量的集合,在内存中占用一组连续的存储单元。数组中的变量称为数组元素,每一数组元素用数组名以及该元素在数组中排列的序号一起表示,因此,数组也称为下标变量。例如 x(1),x(2),y(1,1),y(1,2),y(2,2)等。因此数组也看成名称相同、而下标不同的一组变量。

下标变量的下标个数称为维数,只有一个下标的数组叫一维数组,有两个的叫二维数组。数组的命名方法和一般内存变量的命名方法相同,如果新定义的数组和已经存在的内存变量同名,则数组取代内存变量。

总的来说,Visual FoxPro 中数组的引入是为了提高程序的运行效率、改善程序结构。

(1) 数组的定义。数组使用前一般须先定义后使用,Visual FoxPro 中可以定义一维数组和二维数组。

格式:

DIMENSION/DECLARE<数组名 1> (<下标表达式 1> [,<下标表达式 2>]
[,<数组名 2> (<下标表达式 3> [,<下标表达式 4>])...]

功能: 定义一个或多个一维或二维数组。

例如,命令:

DIMENSION mz (3),mt (2,3) && 定义了一维数组 mz 和二维数组 mt。
DISPLAY MEMORY LIKE m* && 显示数组 mz,mt 的内容

执行后的显示结果为:

MZ	Pub	A
(1)	L	.F.
(2)	L	.F.
(3)	L	.F.

MT	Pub	A
(1, 1)	L	.F.
(1, 2)	L	.F.
(1, 3)	L	.F.
(2, 1)	L	.F.
(2, 2)	L	.F.
(2, 3)	L	.F.

说明：

① 该命令定义了两个数组，一个是一维数组，它有 3 个元素，分别为 mz(1)、mz(2)和 mz(3) ；另一个是二维数组 mt，它有 6 个元素，分别是 mt(1,1)、mt(1,2)、mt(1,3)、mt(2,1)、mt(2,2)和 mt(2,3)。

② 此命令执行后，所建数组的所有数组元素都赋了一个逻辑型的初值. F.（默认值），但以后可以给各数组元素赋不同类型的值。

③ 结果中的“MZ Pub A”，表明 MZ 为数组(Array)且为全局型(Public)，内存变量的全局性概念参见 6.4 节。

(2) 数组的赋值。数组定义好后，数组中的每个数组元素自动被赋予逻辑值. F. 。

当需要对整个数组或个别数组元素进行新的赋值时，与一般内存变量一样，可以通过 STORE 命令或赋值号=在命令窗口或程序中进行赋值。对数组的不同元素，可以赋予不同数据类型的数据。

例如，在命令窗口，依次执行以下命令序列：

```
DIMENSION mt(2,3)
STORE "BOY" TO mt(1,2)
STORE "GIRL"TO mt(2,3)
mt(1,1)=50
STORE {^2015-12-02} TOmt(1,3)
mt(2,1)=.t.
DISPLAY MEMORY LIKE "m* "
```

显示结果为：

MT	Pub	A
(1, 1)	N	50 (50.00000000)
(1, 2)	C	"BOY"
(1, 3)	D	12/02/2015
(2, 1)	L	.T.
(2, 2)	L	.F.
(2, 3)	C	"GIRL"

说明：

① 在定义二维数组 mt 后，对数组赋值，mt 中各元素分别赋予了不同类型的数据。

② 二维数组可以用一维数组来表示，如上例中数组 mt 中的元素 mt(1,2)也可以用 mt(2)来表示，下标 2 是 mt(1,2)在数组中的排列序号，因为二维数组在内存中按行存放。

(3) 数组与数据表间数据的传递。在 Visual FoxPro 系统中,使用命令 SCATTER、GATHER 或 COPY TO ARRAY 命令,可以实现数据表与数组间数据的传递。

① 当前记录数据送入变量命令——SCATTER。

格式:

```
SCATTER [FIELDS<字段名清单>] [MEMO] TO<数组名> /TO<数组名>  
BLANK/MEMVAR/MEMVAR BLANK
```

功能: 表文件的当前记录内容,以字段为单位复制到数组。若数组不存在,则自动生成。

说明:

- FIELDS<字段名清单>: 若无此项,将复制所有字段;否则只复制列出的字段。
- MEMO: 若有此项,将同时复制备注型字段。否则忽略备注型字段。
- TO<数组名>: 按顺序当前记录各字段内容依次复制该数组对应次序的下标变量中。若下标变量多于需复制的字段个数,则排在后面的下标变量的内容不变。若数组不存在或数组的下标变量不足,则自动建立或重新建立该数组。
- TO<数组名>BLANK: 选择此项,则仅建立数组,但不复制内容。
- MEMVAR: 选用此项,则建立与当前记录各字段同名、同类型、同宽度的内存变量,并把字段内容复制过去,注意,此项前无 TO。
- MEMVAR BLANK: 此项将为当前记录的要复制字段建立一同名、同类型、同宽度的内存变量,但并不把字段内容复制过去。

② 变量内容送当前记录命令——GATHER。

格式:

```
GATHER FROM<数组名>/MEMVAR [FIELDS<字段名清单>] [MEMO]
```

功能: 把数组或一组内存变量内容存入当前打开表文件的当前记录。

说明:

- FIELDS<字段名清单>: 若含此项,数组或内存变量的内容替代<字段名清单>中所列字段内容;否则,所有字段内容都被替代。
- MEMO: 包含此项,备注型字段内容被替代;否则忽略备注型字段。
- FROM<数组名>: 指明数据来源于数组。数组中第一个下标变量对应当前记录的第一个字段,第二个下标变量对应第二个字段,以此类推,直至结束。若下标变量数少于字段数,则后面的字段内容不变;若下标变量数多于字段数,则后面的下标变量内容不被复制。
- MEMVAR: 若有此项,则从与字段同名的内存变量中为字段取数据。

例如,以学生表 xs.dbf 为例,假定当前记录为 3 号记录,依次执行以下命令:

```
SCATTER FIELDS 学号,姓名,生日 TO mm  
DISPLAY MEMO LIKE mm
```

执行结果显示:

MM Pub A
(1) C "20150014"
(2) C "李铭 "
(3) D 12/31/1997

由结果可见,表 xs.dbf 当前记录的学号、姓名和生日 3 个字段,依次复制来数组 mm 的 3 个下标变量: mm(1)、mm(2)和 mm(3),如图 3-7 所示。



图 3-7 学生表当前记录送入数组 mm 的下标变量示意图

接着以上操作,依次执行下列命令:

```
APPEND BLANK                                &&xs.dbf 添加一个空记录
GATHER FROM mm FIELDS 学号,姓名,生日
DISPLAY ALL
```

执行结果显示如下:

记录号	学号	姓名	性别	生日	班级	应届否	入学成绩
1	20150011	李中华	男	10/01/99	会计 1501	.T.	611
2	20150012	肖萌	女	02/04/98	会计 1501	.T.	600
3	20150014	李铭	男	12/31/97	会计 1501	.T.	599
4	20150020	张力	男	01/01/95	会计 1501	.F.	585
5	20151234	傅丹	男	03/20/98	会计 1502	.T.	630
6	20151240	李园	女	01/01/96	会计 1502	.F.	588
7	20151255	华晓天	男	07/17/99	会计 1502	.T.	590
8	20154001	刘冬	女	11/09/98	经济 1501	.T.	633
9	20154019	严岩	男	03/13/97	经济 1501	.T.	615
10	20154025	王平	男	06/19/98	经济 1501	.T.	600
11	20156001	江锦添	男	04/27/95	经济 1502	.F.	610
12	20156200	李冬冬	女	12/25/98	经济 1502	.T.	570
13	20156215	赵天宁	男	08/15/97	经济 1502	.T.	585
14	20156345	于天	女	09/10/98	经济 1502	.T.	640
15	20156500	梅媚	女	05/20/99	经济 1502	.T.	580
16	20150014	李铭		12/31/97		.F.	

由结果可见: 数组 mm 的 3 个下标变量: mm(1)、mm(2)和 mm(3)的值,依次复制来学号、姓名和生日三个字段(当前记录为 16 号记录)。

③ 当前打开表文件内容送数组命令——COPY TO ARRAY。

格式:

COPY TO ARRAY<数组名> [FIELDS<字段名清单>] [<范围>] [FOR<条件>]

功能：将指定范围内符合条件的所有记录的有关字段内容全部送入数组。

说明：

- 本命令只对数值型、货币型、日期型、日期时间型、字符型、逻辑型字段进行操作，忽略备注型字段。默认范围为 ALL。
- TO ARRAY <数组名>：指定接收传送数据的数组名。若不存在，系统将自动建立接收数据的数组。
- FIELDS<字段名清单>：选用此项，当前表文件<字段名清单>中指定的字段送入数组；否则，所有字段都将送入数组。

若只复制一条记录，用一维数组即可；否则，用二维数组。数组的第一下标（数组行数）决定了能存放的记录数；第二下标（数组列数）决定了能存放的字段数。

表文件中的第一条记录复制到数组的第一行，第一字段进第一列，第二字段进第二列，依次类推。复制结束后，数组中每一行为一条记录内容，每一列为同一字段内容。

例如，下列命令将 xs.dbf 表的当前记录内容传送给数组 x。

```
USE xs                                && 打开学生信息表文件
DIMENSION x(8)
COPY TO ARRAY x                      && 当前记录传送 x 数组
DISPLAY MEMO LIKE x                  && 显示 x 数组内容
```

显示结果：

X	Pub	A
(1)	C	"20150011"
(2)	C	"李中华 "
(3)	C	"男"
(4)	D	10/01/99
(5)	C	"会计 1501 "
(6)	L	.T.
(7)	N	611 (611.00000000)
(8)	L	.F.

④ 数组内容追加入表文件命令——APPEND FROM ARRAY。

格式：

```
APPEND FROM ARRAY<数组名> [FOR<条件>] [FIELDS<字段名清单>]
```

功能：由数组为当前表指定字段追加满足条件的新记录。

4. 系统变量

系统变量是由 Visual FoxPro 自动生成和维护的内存变量，以下划线“_”开头，分别用于控制外部设备（如打印机、鼠标等）、屏幕输出格式，或处理有关计算器、日历、剪贴板等方面的信息，其名称由系统规定。