

第 3 章 Visual FoxPro 语言基础

3.1 数据类型

数据是反映现实世界中客观事物属性的记录,它包括两个方面:数据内容与数据形式。数据内容就是数据的值,数据形式就是数据的存储形式和操作使用方式,也称为数据类型。在 Visual FoxPro 中,所有数据都属于某一确定的类型。数据类型决定数据如何存储以及如何使用。Visual FoxPro 6.0 系统为使用户建立和操作数据库更加方便,将系统中所有操作、处理的数据,分为以下不同数据类型:字符型、数值型、单精度浮点型、双精度浮点型、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。其中,单精度浮点型、双精度浮点型、整型、备注型、通用型、二进制字符型和二进制备注型只能用在数据表的字段中。下面主要介绍几种常用的数据类型。

1. 字符型(Character, C)

字符型数据是由中英文字符、数字字符和其他 ASCII 字符组成的字符序列,其长度(即字符个数)范围是 0~254 个字符,每个字符占 1B。注意:如果将阿拉伯数字定义为字符型数据时,它不具备数学上的数值含义,不能参加数学运算,如电话号码、邮编等。

2. 数值型(Numeric, N)

数值型数据是指可以进行算术运算的数据。由 0~9 以及正负号(十和一)和小数点(.)组成。最大长度为 20 位。在内存中,数值型数据占用 8B。一般将需要参加数学运算的数据定义成数值型。

3. 单精度浮点型(Float, F)

与数值型数据完全等价,只是在存储形式上采取浮点格式且数据的精度要比数值型数据高。

4. 双精度浮点型(Double, B)

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。

5. 整型(Integer, I)

只用于数据表中的字段类型的定义,不带小数点的数值类型,占 4B。在表中整型数据以二进制形式存储。

6. 货币型(Currency, Y)

数据的第一个数字前冠一个货币符号(\$)。默认小数位 4 位, 小数位超过 4 个字符的数据, 系统将会按四舍五入原则自动截取, 占 8B。

7. 日期型(Date, D)

表示日期的数据, 包括年、月、日三个部分, 每部分间用规定的分隔符分开。日期的默认格式是{mm/dd/yyyy}, 其中 mm 表示月份, dd 表示日期, yyyy 表示年度, 固定长度 8 位。如{09/01/2010}表示 2010 年 9 月 1 日这一日期数据。

8. 日期时间型(DateTime, T)

描述日期和时间的数据, 除包括日期数据的年、月、日外, 还包括时、分、秒以及上午、下午等内容。日期时间的默认格式是{mm/dd/yyyy hh: mm: ss}, 其中 mm、dd、yyyy 的意义与日期型相同, 而 hh 表示小时, mm 表示分钟, ss 表示秒数。日期时间型数据也是采用固定长度 8 位, 取值范围是: 日期为 01/01/0001~12/31/9999, 时间为 00: 00: 00~23: 59: 59。如{10/01/2010 10: 30: 30}表示 2010 年 10 月 1 日 10 时 30 分 30 秒这一日期时间数据。

9. 逻辑型(Logic, L)

逻辑型数据是描述客观事物真假的数据, 用于表示逻辑判断结果。逻辑型数据只有真(. T. 或. Y.)和假(. F. 或. N.)两种, 固定长度 1 位。为区别其他数据类型, 一般需在表示逻辑值的字母 T、Y、F、N 的前后加圆点符“.”。

10. 备注型(Memo, M)

用于存放较长的字符型数据类型。备注型数据没有数据长度限制, 仅受限于现有的磁盘空间。其字段长度固定为 4 个字符, 用于存储备注文件的地址(指针), 而实际数据被存放在与数据表文件同名的备注文件(*. fpt)中, 长度根据数据的内容而定。

11. 通用型(General, G)

通用型数据是用于存储 OLE 对象的数据。通用型数据中的 OLE(对象链接嵌入)对象可以是电子表格、文档、图片等, 只能用于字段定义, 通用型数据长度固定为 4 个字符, 实际数据长度仅受限于现有的磁盘空间。

3.2 数据存储

在 Visual FoxPro 系统中, 数据可用常量、变量、数组表示, 数据还可以用于字段、记录和对象中, 由它们存储、容纳各种类型的数据。因此, 常把这些供数据存储的常量、变量、数组、字段、记录和对象等称为数据存储容器。用户正是利用不同的数据存储容器, 在 Visual FoxPro 系统中表示、存储、操作、处理各种类型的数据, 实现数据处理的应用。

3.2.1 常量

常量是在命令或程序中可直接引用、具有具体值的命名数据项, 其特征是在整个操作

过程中它的值和表现形式保持不变。Visual FoxPro 按常量取值的数据类型,将常量分为 6 种类型:数值型常量、浮点型常量、字符型常量、逻辑型常量、日期型常量、日期时间型常量。

1. 数值型常量

由数字、小数点和正负号组成,表示整数或实数值。如:110、-43、68.99、-21052.5741 等分别是数值型常量中的整数和实数。

2. 浮点型常量

以浮点格式表示的数值常量。通常用来表示那些绝对值很大或很小、而有效位数不太长的一些数值,对应于日常应用中的科学记数法。如中国的人口数接近 13 亿,以实数形式表示是 1,300,000,000,以科学记数法的乘幂形式表示是 1.3×10^9 。显然用科学记数法形式表示比较简短、方便、适用,对应在 Visual FoxPro 中采用浮点型常量表示为 1.3E+9,其中,1.3 称为尾数,表示有效数字,9 称为阶码,表示幂指数。阶码只能是整数,尾数可以是整数,也可以是小数,尾数与阶码均可正可负。

3. 字符型常量

由汉字和 ASCII 字符集中可打印字符组成的字符串。为与其他类型常量、变量和标识符相区别,Visual FoxPro 要求将字符串中所有字符,用一对双引号“”或单引号“'”或方括号“[]”作为定界符括起来。如“ABC”、‘西南大学’、[321]都是字符串。

4. 逻辑型常量

逻辑型数据只有逻辑真和逻辑假两个值。逻辑真的常量表示形式有 .T.、.t.、.Y. 和 .y.; 逻辑假的常量表示形式有 .F.、.f.、.N. 和 .n.。

注:前后两个句点作为逻辑型常量的定界符是必不可少的,否则会被误认为变量名。

5. 日期型常量

日期型常量是表示日期值的数据,其默认格式是 {mm/dd/[yy]yy}。如 {10/01/12} 和 {10/01/2012} 均表示 2012 年 10 月 1 日这一日期常量值。

6. 日期时间型常量

Visual FoxPro 系统中增加了一种表示日期和时间值的日期时间型常量,其默认格式如下:

```
{mm/dd/[yy]yy [,] hh:mm:ss[a|p]}
```

其中 a 和 p 分别表示 AM(上午)和 PM(下午)。

日期值和日期时间值的输入格式与输出格式并不完全相同,特别是输出格式受系统环境设置的影响,用户可根据应用需要进行相应调整、设置。

(1) 设置分隔符。

格式:

```
SET MARK TO [日期分隔符]
```

命令功能:用于设置显示日期型数据时使用的分隔符,如“-”、“.”等。

(2) 设置日期显示格式。

格式：

```
SET DATE [TO] AMERICAN | ANSI | BRITISH | FRENCH | GERMAN | ITALIAN | JAPAN | USA | MDY | DMY  
| YMD
```

功能：设置日期显示的格式。命令中各个短语所定义 的日期格式如表 3.1 所示。

表 3.1 日期设置命令短语所对应的日期格式

短 语	格 式	短 语	格 式
AMERICAN	mm/dd/yy	BRITISH FRANCE	dd/mm/yy
MDY	mm/dd/yy	DMY	dd/mm/yy
USA	mm-dd-yy	GERMAN ITALIAN	dd. mm. yy dd-mm-yy
JAPAN	yy/mm/dd	ANSI	yy. mm. dd
YMD	yy/mm/dd		

(3) 设置世纪。

格式：

```
SET CENTURY ON/OFF
```

功能：用于设置显示日期型数据时是否显示世纪，即是否用 4 位数字显示年份。

3.2.2 内存变量

在命令执行过程中，其值可变化的量称为变量。变量值在使用过程中可改变。内存变量则是存储于内存中的变量，常被用来存储所需要的常数、中间结果或最终结果。确定一个变量，需要确定其 3 个要素：变量名、数据类型和变量值，其中变量名由汉字、字母、数字和下划线组成，必须由字母或汉字开头，最长不超过 254 个字符（系统内存变量通常以下划线开始），建议一般不用汉字作为变量名。

1. 内存变量的数据类型

内存变量的数据类型包括：数值型、字符型、逻辑型、日期型和日期时间型。

2. 内存变量的建立

建立内存变量就是给内存变量赋值。内存变量赋值既可定义一个新的内存变量，也可改变已有内存变量的值或数据类型。

格式 1：

```
STORE <表达式> TO <内存变量表>
```

功能：计算表达式的值并赋值给各个内存变量。

格式 2：

```
<内存变量>=<表达式>
```

功能：计算表达式的值并赋值给指定内存变量。

3. 内存变量的输出

可以用两个命令在屏幕上输出内存变量的值。

格式：

? | ?? <表达式表> [AT <列号>]

功能：计算表达式表中各表达式的值，并在屏幕上指定位置输出各式的值。

?: 先回车换行，再计算并输出表达式的值；

?: 在屏幕上当前位置，计算并直接输出表达式的值；

表达式表：多个用逗号两两分隔的表达式，各表达式的值输出时，以空格分隔；

AT 列号子句指定表达式值从指定列开始显示输出。AT 的定位只对它前面的一个表达式有效，多个表达式必须用多个 AT 子句分别定位输出。

【例 3.1】 变量赋值与输出。

在命令窗口中输入命令：

```
ab='university'
STORE 'computer' TO ab
STORE 5 TO s1,s2,s3
?s1, s2, s3,ab
```

主窗口显示：

```
5 5 5 computer
```

4. 内存变量的显示(显示变量名,作用域,类型,值)

格式：

LIST|DISPLAY MEMORY

LIST MEMORY 一次显示与通配符匹配的所有内存变量，DISPLAY MEMORY 分屏显示与通配符匹配的所有内存变量，如内存变量多，显示一屏后暂停，按任意键之后再继续显示下一屏。

5. 内存变量的清除

格式 1：

CLEAR MEMORY

功能：清除所有内存变量。

格式 2：

RELEASE <内存变量名表>

功能：清除指定的内存变量。

格式 3：

RELEASE ALL [LIKE<通配符> |EXCEPT<通配符>]

功能：选用 LIKE 短语清除与通配符相配的内存变量，选用 except 短语清除与通配符不相匹配的内存变量。

Visual FoxPro 中有两个通配符：? 代表任意一个字符

* 代表任意多个字符

格式 4：

```
CLEAR ALL
```

功能：清除所有的内存变量，并关闭所有打开的库文件及相关文件，设置 1 区为当前工作区。

3.2.3 数组变量

数组变量是一种有组织的内存结构变量，它是具有相同名称而下标不同的一组有序内存变量。Visual FoxPro 允许定义一维和二维数组，数组在使用之前需要先定义。

1. 定义数组

Visual FoxPro 中的数组和其他高级语言中的数组有所不同，数组本身没有数据类型，而各个数组元素的数据类型跟最近一次被赋值的类型相同。创建数组的命令如下：

格式：

```
DIMENSION|DECLARE|PUBLIC<数组名 1>(<数值表达式 1>[,<数值表达式 2>])[,<数组名 2>(<数值表达式 3>[,<数值表达式 4>])...
```

2. 数组元素及其引用

数组中每个有序变量构成数组的成员，称之为数组元素。数组元素的名称使用数组名和用括号括起来的下标组成。如 X(1)，表示一维数组的第一个元素。Y(2,2)表示二维数组的第 2 行、第 2 列元素。

关于数组元素的引用说明：

- (1) 数组下标使用圆括号，二维数组的下标之间使用逗号隔开；
- (2) 数组的下标可以是常量、变量和表达式，如 X(1), X(b1), X(a+b)；
- (3) 数组的第一个下标是 1，也就是说数组下标是从 1 开始的；
- (4) 数组元素的类型为最近一次被赋值的类型；
- (5) 数组元素和简单内存变量一样都可以被赋值和引用。

3. 数组赋值

数组的赋值是按数组元素进行的，其规则遵循内存变量赋值规则。

【例 3.2】 定义数组、赋值并输出。

```
DIMENSION X(2), Y(2,2), Z(5)
X(1) = "ABCD"
X(2) = .T.
Y(1,2) = X(1)
```

```
Y(2,2)=123  
Y(2,1)=11.1  
Z=8  
?X(1),X(2),Y(1,2),Y(2,2),Z(1),Z(3)
```

主窗口显示:

```
ABCD .T. ABCD 123 8 8
```

3.2.4 系统变量

系统变量是系统自己定义的一些变量。这些变量的名称是系统已经定义好的,都是以“_”(下划线)字符开头。故在定义内存变量名称时,最好不要以“_”字符开头,以避免重名。学会使用系统变量会带来许多方便,系统提供的系统变量很多,在使用 DISPLAY MEMORY 命令显示内存变量时,可以看到这些系统变量的当前值。

3.2.5 字段变量

字段变量是定义在表中的变量,随表的存取而存取,因而是永久性变量。字段名就是变量名;字段变量的数据类型为 Visual FoxPro 中的任意数据类型,字段值就是变量值。

3.2.6 字段、记录和数据表

- (1) 字段是数据表中存储、处理数据的基本元素。
- (2) 记录是数据表中一组字段的集合,同一个数据表中所有记录拥有相同的字段名,且同名字段在每个记录中的数据类型、数据长度都是相同的。
- (3) 数据表是一系列相关数据的集合,由多条记录构成。

3.2.7 对象、类和属性

- (1) 对象(object)是对一事物属性及行为特征的描述,它将某一数据和使用该数据的一组基本操作或过程封装在一起,从而构成一个统一体。
- (2) 类(class)是一组对象的数据和行为特征的抽象描述,是具有共同属性、共同操作性质的对象的集合。
- (3) 属性(attribute)是用来描述对象特征的参数。

3.3 运算符及表达式

在 Visual FoxPro 系统中,运算符是对数据对象进行加工处理的符号,而表达式是由常量、变量、函数及其他数据容器单独或与运算符组成的有意义的运算式子。

根据其处理数据对象的数据类型,运算符分为算术(数值)运算符、字符运算符、日期时间运算符、关系运算符、逻辑运算符和类与对象运算符 6 类,相应的,表达式也分为算

术表达式、字符表达式、日期时间表达式、关系表达式、逻辑表达式和类与对象表达式 6 类。

常量、变量和函数可以作为表达式的特例。今后涉及表达式的描述，除特别指明，均可包含这些特殊的形式。

在一个表达式中可能包含多个由不同运算符连接起来的、具有不同数据类型的数据对象，但任何运算符两侧的数据对象必须具有相匹配数据类型，否则运算将会出错。由于表达式中有多种运算，不同的运算顺序可能得出不同结果，甚至出现运算错误，因此当表达式中包含多种运算时，必须按一定顺序施行相应运算，才能保证运算的合理性和结果的正确性、唯一性。用户也可以通过给表达式加圆括号的方式，改变其默认运算顺序。在 Visual FoxPro 系统中，各类运算的优先顺序如下：

圆括号>算术和日期运算>字符串运算>关系运算>逻辑运算

同一类运算符也有一定的运算优先顺序，这在各类表达式中分别介绍。如果多个同一级别的运算，则按在表达式中出现的先后顺序进行运算。

3.3.1 算术表达式

算术表达式由算术运算符与数值型常量、变量、函数构成，运算结果仍为数值型，又称数值表达式，其运算对象和运算结果均为数值型数据。数值运算符的功能及运算优先顺序，如表 3.2 所示。

表 3.2 算术运算符及其优先级

运 算 符	功 能	表达式举例	运算结果	优先级别
()	圆括号	(6-3)*(3+2)	15	最高 ↓ 最低
-	取相反数	-(4-8)	4	
,^	乘幂	32、3^3	9、27	
*,/	乘、除	6*10、15/5	60、3	
%	取余数	21%4	1	
+, -	加、减	26+18、30-50	44、-20	

注：求余运算符“%”的运算结果(即余数)的符号与除数一致。如表达式 22%-4 的值是-2，表达式 -22%4 的值是 2。

3.3.2 字符表达式

字符表达式是由字符运算符将字符型数据对象连接起来进行运算的式子。字符运算的对象是字符型数据对象，运算结果是字符常量或逻辑常量。表 3.3 为字符运算符的功能。

表 3.3 字符运算符及其表达式

运算符	功 能	表达式举例	运算结果
+	串 1+串 2：两串顺序相连接	'中国 '+'重庆'	'中国 重庆'
-	串 1-串 2：串 1 尾空格移到串 2 尾后再顺序相连接	'中国 '-'重庆'	'中国重庆 '
\$	串 1\$串 2：串 1 是否为串 2 子串	'12a3'\$ 'x12a345' '重庆'\$ '中国'	. T. . F.

“+”与“-”都是字符连接运算符，都将两字符串顺序连接，但“+”是直接连接，“-”则将串 1 尾部所有空格移到串 2 尾部后再连接；“\$”运算实质上是比较两个串的包含关系，因此有些书籍中将其归于关系运算，其作用是比较、判断串 1 是否为串 2 的子串，如果串 1 是串 2 的子串，运算结果为“真”，否则为“假”。所谓子串，如果串 1 中所有字符均包含在串 2 中、且与串 1 中排列方式与顺序完全一致，则称串 1 为串 2 的子串。

3.3.3 日期及日期时间表达式

由日期运算符将一个日期型或日期时间型数据与一个数值型数据连接而成的运算式称为日期表达式，如表 3.4 所示。日期运算符分为“+”和“-”两种，其作用分别是在日期数据上增加或减少一个天数，在日期时间数据上增加或减少一个秒数。两个运算的优先级别相同。

表 3.4 日期及日期时间运算符

运算符	功 能	表达式举例	运算结果
+	<日期>+<天数>：计算指定日期若干天后的日期	{01/01/2010}+20	{01/21/2010}
	<天数> + <日期>：计算指定日期若干天后的日期	20+{01/01/2010}	{01/21/2010}
	<日期时间>+<秒数>：计算指定日期时间若干秒后的日期时间	{10/10/2010 9：15：20}+200	{ 10/10/2010 9：18：40}
	<秒数>+<日期时间>：计算指定日期时间若干秒后的日期时间	200+{10/10/2010 9：15：20}	{ 10/10/2010 9：18：40}
-	<日期>-<天数>：计算指定日期若干天前的日期	{05/30/2010}-10	{05/20/2010}
	<日期>- <日期>：计算两个指定日期相差的天数	{05/30/2010} - {05/20/2010}	10
	<日期时间>- <秒数>：计算指定日期时间若干秒前的日期时间	{10/10/2010 9：18：40}-200	{ 10/10/2010 9：15：20}
	<日期时间>- <日期时间>：计算两个指定日期时间相差的秒数	{10/10/2010 9：18：40}-{10/10/2010 9：15：20}	200

3.3.4 关系表达式

由关系运算符连接两个同类数据对象进行关系比较的运算式称为关系表达式，如表 3.5 所示。关系表达式的值为逻辑值，关系表达式成立则其值为“真”，否则为“假”。

表 3.5 关系运算符及其表达式

运算符	功 能	表达式举例	运算结果
<	小于	10<3 * 6	. T.
>	大于	'X'>'C'	. T.
=	等于	4+4 = 3 * 4	. F.
<>、#、!=	不等于	8 <> -10	. T.
<=	小于或等于	'xyz' <= 'XY'	. F.
>=	大于或等于	{10-25-10}>={10/01/10}	. T.
==	字符串恒等	'abc'== 'abcabc'	. F.

注：运算符==（精确比较）仅适用于字符型数据，其他运算符适用于任何类型的数据，但前后两个运算对象的数据类型要一致。

各种类型数据的比较规则如下。

- (1) 数值型和货币型数据根据其数值的大小进行比较。
- (2) 逻辑型数据比较时，. T. 比 . F. 大。
- (3) 单个字符的比较是以字符 ASCII 码的大小作为字符的“大小”，也就是先后顺序；但字母则是按“先小写后大写”的顺序排列，即'a'<'A'<'b'<'B'<'c'<'C'<…<'z'<'Z'。
- (4) 字符串比较时，先将 2 个字符串的第一个字符比较，若两者不等，其大小就决定了 2 个字符串的大小。若相等，则再将第 2 个字符比较，以次类推，直到最后，若每个字符都相等，则 2 个字符串相等。
- (5) 对于汉字字符，系统默认按汉字的拼音排列汉字的顺序，也就是以汉字的拼音顺序比较“大小”，因此，汉字比较实质上是以字母的顺序进行比较；但 Visual FoxPro 系统可以设置汉字按笔画排列顺序，因而，汉字的“大小”就决定其笔画数的多少。用菜单设置汉字排列顺序方式的操作步骤为：选择“工具”|“选项”菜单命令，将打开“选项”对话框，在“数据”选项卡的“排序序列”下拉列表框中选择 Stroke 项并确定，系统将按汉字的笔画数进行汉字的排序、比较运算。

注意：当运算对象为字符型时，可用命令 SET EXACT ON/OFF 来设置“=”是否为精确比较。在非精确比较时，在关系表达式的格式中，只要后一个表达式是前一个表达式的前缀，其结果便为真。

【例 3.3】 关系运算示例。

SET EXACT OFF	&& 设置字符串为非精确比较
? 102.35>60	&& .T.
? x>y	&& .T. 或者 .F., 取决于变量 x 和 y 的内容
? 'A' > 'B'	&& .F.
? '乡长' > '市长'	&& .T.