第 3 章　数据与数据运算

数据是计算机进行操作和处理的主要对象。数据有多种类型，为了方便各类数据在计算机中存储和处理，计算机必须使用某种数据类型以表达和描述相应的数据。一旦确定了用于表示数据的数据类型，则确定了该数据的存储空间、能够参加的运算集合和取值范围。因此，使用计算机处理数据时，必须根据实际情况为其选择合适的数据类型。

3.1　数 据 类 型

Visual FoxPro 的所有数据都有一个特定的数据类型，它定义了各种数据的取值范围。在定义了数据类型后，Visual FoxPro 就可有效地存储和操作该数据。下面介绍常用的 7 种数据类型。

1. 字符型

字符型(Character)数据是表示不具计算能力的文字字符序列，简称 C 型。字符型数据是由中文字符、英文字符、数字字符和其他可打印的 ASCII 字符组成的字符序列，称为字符串，其长度不超过 254 个字节。

2. 数值型

数值型数据是表示数量并可以进行数值运算的数据，简称 N 型。数值型数据由数字 0～9、小数点和正负号组成，最大长度为 20 位(包括正负号和小数点)。数值型数据又可分为整型数据、浮点型数据和双精度型数据。

① 整型(Integer)数据是不包含小数点部分的数值型数据。

② 浮点型(Float)数据只是在存储形式上采取浮点格式(占用 8 个字节)，增设浮点型数据的主要目的是使计算有较高的精度。浮点格式是指小数点位置可任意浮动，由尾数、阶数及字母 E 组成。例如－0.211E＋04，其中－0.211 是尾数，一般采用纯小数形式；＋04 是阶数，它必须是整数。尾数和阶数可正可负，之间用 E 分隔。

③ 双精度型(Double)数据是更高精度的数值型数据。

3. 日期型

日期型(Date)数据是表示日期的数据，简称 D 型。日期数据的默认输出格式是{mm/dd/yy}，其中 yy 表示年度，mm 表示月份，dd 表示日期，固定长度 8 个字节。

4. 日期时间型

日期时间型(Date Time)数据是表示日期和时间的数据，简称 T 型。日期时间数据

的默认格式是{mm/dd/yy hh：mm：ss}，其中 mm、dd、yy 的意义与日期型相同，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 个字节，取值范围是：日期为 01/01/0001～12/31/9999，时间为 00：00：00～23：59：59。

5. 逻辑型

逻辑型(Logic)数据是描述表示逻辑判断的结果的数据，简称 L 型。逻辑型数据只有真和假两种取值，存储时占 1 个字节。

6. 备注型

备注型(Memo)数据用于存放较长字符数据的数据类型，简称 M 型。备注型数据没有数据长度限制，仅受限于磁盘空间。它只用于数据表中字段类型的定义，在表中字段长度固定为 10 位，实际数据存放在与表文件同名的备注文件(.fpt)中，长度根据数据的内容而定。

7. 通用型

通用型(General)数据是用于存储 OLE 对象的数据，简称 G 型。通用型数据中的 OLE 对象可以是电子表格、文档和图片等内容。它只用于数据表中字段类型的定义，在表中长度固定为 4 个字节。

在数据表中存放的通用型数据实际内容、类型和数据量取决于链接或嵌入 OLE 对象的操作方式。如果采用链接 OLE 对象方式，则数据表中只包含对 OLE 对象的引用说明，以及对创建该 OLE 对象的应用程序的引用说明。如果采用嵌入 OLE 对象方式，则数据表中除包含对创建该 OLE 对象的应用程序的引用说明外，还包含 OLE 对象中的实际数据。其实际数据也是存放在与表文件同名的备注文件中，长度根据数据的内容而定。

3.2 常量与变量

3.2.1 常量

常量是指在程序运行过程中值不发生变化的量。Visual FoxPro 有 6 种类型的常量，分别为字符型、数值型、日期型、日期时间型、货币型和逻辑型常量。

1. 字符型常量

字符型常量是用定界符括起来的字符串。在 Visual FoxPro 中，定界符有 3 种：'(单撇号)、"(双撇号)和[ ](方括号)。例如'abc'、"学生"、[计算机 1 班]等都是字符型常量。

若字符串本身包含有定界符，就应选择另一种定界符来定义该字符型常量。例如，"I'm a student!"表示字符常量 I'm a student!，含有 14 个字符。定界符在使用时必须匹配，即如果一个字符串左边是用方括号作为定界符，右边也必须以方括号作为定界符。

2. 数值型常量

数值型常量是可以带正负号的整数或小数(正号通常省略),可以用科学记数法表示。

3. 逻辑型常量

逻辑型常量也称布尔型常量。逻辑型常量只有逻辑真和逻辑假两种值。逻辑真有.T.、.Y.、.t.、.y.这 4 种表示方法,逻辑假有.F.、.N.、.f.、.n.这 4 种表示方法。

必须注意的是,用于表示逻辑值的字母两端必须紧靠有小圆点,圆点与字母之间不能有空格。

4. 日期型常量

日期型常量要用一对花括号作为定界符,花括号内包括年、月、日 3 部分内容,各部分内容之间用分隔符分隔。分隔符可以是/、-、.等。Visual FoxPro 的默认日期格式是{mm/dd/[yy]yy},其中 yy 表示年度,mm 表示月份,dd 表示日期。

5. 日期时间型常量

日期时间型常量也要用一对花括号作为定界符,其中既包含日期又包含时间。日期的格式与日期型常量相同,时间包括时、分、秒,时分秒之间用":"分隔。日期时间型常量的默认格式是{mm/dd/[yy]yy [,] [hh[:mm[:ss]][a|p]]},其中 mm、dd、yy 的意义与日期型相同,hh、mm、ss 的默认值分别为 12、0 和 0。a 和 p 分别表示 AM(上午)和 PM(下午),默认为 AM。如果指定时间大于等于 12,则自然为下午的时间。

注意:不管是日期型常量,还是日期时间型常量,在命令窗口均不能直接输入。例如,若在命令窗口输入以下命令则会显示错误信息。

```
? {05/04/2008}
```

Visual FoxPro 增加了一种所谓严格的日期格式。严格的日期格式是:

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

在命令窗口中可以直接输入严格的日期格式数据。例如:

```
? {^2009/10/1}
```

在主窗口中显示为:

```
10/01/09
```

6. 货币型常量

货币型常量以符号$开头,后面接数值型常量,占 8 个字节。货币型常量不能用科学记数法表示。如果货币型常量是负数,一放在"$"的前面和后面都可以。货币型数据在存储和计算时采用 4 位小数。如果一个货币型常量多于 4 位小数,那么系统会自动将多

余的小数位四舍五入。例如,货币型常量 $2.258696 将存储为 $2.2587。

3.2.2 变量

变量是指在程序运行过程中其值可能发生变化的量。Visual FoxPro 系统包含字段变量和内存变量两大类,而内存变量又含有一种特殊的类型——数组。

1. 命名规则

为了便于变量的使用与管理,每个变量都有一个名称,叫做变量名。Visual FoxPro 系统通过名字在内存中提取相应的数据,即通过变量名来引用变量的值。为变量取名应遵守以下规则:

(1) 使用字母、汉字、下划线和数字命名。

(2) 命名以字母或下划线开头。除自由表中字段名、索引的 TAG 标识名最多只能 10 个字符外,其他的命名可使用 1～128 个字符。

(3) 变量名不要使用 Visual FoxPro 的系统保留字。所谓系统保留字,是指 Visual FoxPro 语言使用的命令名、函数名、命令短语中的关键字、系统内存变量名等,如命令 USE、LIST 和 DISPLAY 等就是系统的保留字。

2. 字段变量

每个数据库表中都有多个数据字段,每个数据字段的字段名就是字段变量。字段变量是一种多值变量。当用某一字段名做变量时,它的值就是表记录指针所指的那条记录对应字段的值。例如,设某个数据库表中包括“姓名”字段,“姓名”是字段名,是一个字段变量,若当前表记录指针所指的那条记录的姓名为“李三”,则此时“姓名”字段变量的值就为“李三”。字段变量的类型可以是 Visual FoxPro 的任意数据类型。字段变量的名字、类型和长度等是在定义表结构时定义的。要使用字段变量,必须首先打开包含该字段的表。

3. 内存变量

内存变量是用户通过命令或程序临时定义的变量,每一个内存变量占用一定的存储单元。内存变量的数据类型不用事先定义,其类型取决于所存放数值的类型,可以通过对内存变量重新赋值来改变其值和类型。

(1) 内存变量的赋值

给内存变量赋值的命令有两种格式:

格式 1:

```
<内存变量>=<表达式>
```

例如:

```
a=8.88        &&a 为数值型
a="abc123"    &&a 变为字符型
```

“=”是 Visual FoxPro 的命令，其功能是将“=”右边表达式的值赋给“=”左边的变量。而常量是一种特殊的表达式。

格式 2：

```
STORE <表达式>TO <内存变量表>
```

例如：

```
STORE 5+3 TO A1,A2,A3  && 将计算结果 8 存入变量 A1,A2 和 A3
```

该命令先计算表达式的值，然后将表达式的值赋给一个或几个内存变量。第一种格式只能给一个内存变量赋值。第二种格式可以同时给多个内存变量赋相同的值，各内存变量名之间用逗号分隔。

(2) 内存变量的清除

内存变量需要占据一定的内存空间，若要释放相应的内存空间，可以采用以下命令：

格式 1：

```
CLEAR MEMORY
```

该命令的作用是清除所有的内存变量。

格式 2：

```
RELEASE [<内存变量表>]
```

该命令可清除指定的内存变量。如：

```
RELEASE A1,A2
```

有一类特殊的内存变量叫系统变量，它是 Visual FoxPro 自身提供的用于设置、保存系统状态信息的内存变量。这类变量不需要预先定义，系统启动后就已经存在。系统变量名都是以下划线开始的，使用方法同一般内存变量一样。为了避免与系统变量名冲突，在定义内存变量名时，尽量不要以下划线开始。

注意：内存变量和字段变量可以同名，此时，将优先存取字段变量，屏蔽同名的内存变量。若要明确指定访问内存变量，应在内存变量名前加上符号 M. 或 M—>。

4. 数组变量

数组是内存中一片连续的存储区域，每个数组都要取个合法的名字，这个名字叫数组名。数组中的数据称为数组元素，每个数组元素可通过数组名以及相应下标来访问。每一个数组元素相当于一个内存变量，可以给各个元素分别赋值，赋值方法与内存变量相同。

只有一个下标的数组称为一维数组，含有两个下标的数组称为二维数组。Visual FoxPro 允许用户定义一维和二维数组。

数组在使用之前必须用命令来声明，包括数组名和数组的大小。数组的大小由下标值的上、下限决定。其中，下限值系统隐含为 1，不需要用户指定。声明数组时，用户需要指定数组的上限值。显然，数组的大小是由数组的上限决定的。

声明数组可以用以下两条命令之一来实现，这两条命令的功能完全相同，用于定义一个或多个一维或二维数组。为了说明的方便，后面的例子都用格式 1。

格式 1：

```
DIMENSION <数组名 1>(<下标 1>[,<下标 2])[, <数组名 2>(<下标 1>[,<下标 2])][,...]
```

格式 2：

```
DECLARE <数组名 1>(<下标 1>[,<下标 2])[, <数组名 2>(<下标 1>[,<下标 2])][,...]
```

数组一经定义，它的每个元素都可当作一个内存变量来使用，因此它具有与内存变量相同的性质。Visual FoxPro 命令行中可以使用内存变量的地方都能用数组元素代替。但数组变量有其本身特殊的地方，在使用时应注意以下几个方面：

① 数组名后面的括号，既可以用圆括号也可以用方括号。以下两条命令是等价的：

```
DIMENSION a(5),b(3,2)
DIMENSION a[5],b[3,2]
```

该命令定义了一维数组 a 和二维数组 b。其中，数组 a 有 5 个元素，分别为 a(1)，a(2)，a(3)，a(4)，a(5)；数组 b 有 6 个元素，分别为 b(1,1)，b(1,2)，b(2,1)，b(2,2)，b(3,1)，b(3,2)。

② 数组定义后，系统自动将每个数组元素定义为逻辑型，初值为逻辑假.F.。

③ 数组中的元素位置是有序而固定的。其中，二维数组各元素在内存中是按行的顺序存储，因此，二维数组可以按一维数组来使用。例如，上述数组 b 中元素 b(1,1)排在第 1 行第 1 列，即 b 数组的第 1 个元素，它又可以表示为 b(1)。同样，b(1,2)排在第一行第 2 列，即 b 数组的第 2 个元素，可以表示为 b(2)；b(3,1)是第 5 个元素，可表示为 b(5)。

例如，在命令窗口输入以下命令序列：

```
DIMENSION b(3,2)
b(3,1)=9
?b(5)
```

主窗口显示的结果为：

```
9
```

④ 给数组变量赋值时，如果未指明下标(即未指明第几个元素)，则对该数组中所有元素赋同一个值。

例如，对上述数组 b 赋值：b＝88，则为二维数组 b 的 6 个元素都赋了相同的值 88。

⑤ 数组中各个元素的数据类型可以不同。

⑥ 在引用数组时，如果未指明下标，则引用该数组的第一个元素。

⑦ 内存变量和数组不能重名。

⑧ 引用数组元素时，下标不能超界。

例如，对上面定义的数组 b，若引用 b(7)就是非法的。

3.3　运算符和表达式

在 Visual FoxPro 中，运算符就是用来进行运算的符号，表达式是用运算符将变量、常量和函数连接起来的式子。表达式分为算术表达式、字符表达式、关系表达式、逻辑表达式和日期时间表达式 5 类。

3.3.1　算术运算符和算术表达式

算术表达式由数值型字段、返回数值型的函数、数值型数据组成的内存变量和数组元素、数值型常量、算术运算符组成。

Visual FoxPro 6.0 中的算术运算符有：

- ()：括号。
- **或^：乘方。
- %：模数(除法的余数)。
- *、/：乘、除。
- +、-：加、减。

算术运算符的优先级依次为括号、乘方、乘除和加减，同级运算从左到右依次进行。

例如，在命令窗口输入如下命令：

```
?((3+4)^2-5*2)%4
```

在主窗口显示表达式的值为：

```
3.00
```

余数的正负号与除数相同。当两个同符号数求余运算(%)时，结果为第一个数除以第二个数的余数；当两个异符号数求余时，结果为第一个数除以第二个数的余数再加上第二个数。

例如，在命令窗口输入以下命令：

```
?8%3, 8%-3, -8%3, -8%-3
```

主窗口显示的结果为：

```
2  -1  1  -2
```

3.3.2　字符串运算符和字符串表达式

字符串表达式由字符串运算符、字符型常量、变量和函数组成，其运算结果是字符串或逻辑值。

Visual FoxPro 6.0 中的字符串运算符有：

- +：字符串连接符，将运算符两边的字符串连接起来，形成一个新字符串。
- -：串尾空格移位连接符，在将两个字符串连接时，把第一个字符串的尾部空格

移到结果字符串的尾部。

例如，在命令窗口输入如下命令：

```
?"ab "-"cd"+"ef"
```

在主窗口显示表达式的值为：

```
abcd ef
```

- $：字符串包含运算符，若第一个字符串包含在第二个字符串之中，其表达式值为真，否则为假。

例如，在命令窗口输入如下命令：

```
?"学生"$"大学生"
```

在主窗口显示表达式的值为：

```
.T.
```

在写字符串表达式时，运算符两边操作数的数据类型一定要一致，否则会产生错误。

3.3.3 关系运算符和关系表达式

关系表达式是由关系运算符将两个同类型的数据连接起来的式子。关系表达式用于表示两个数据之间的关系，返回逻辑值。

Visual FoxPro 中的关系运算符有：

- < 小于。
- <= 小于等于。
- <>或 # 或!= 不等于。
- > 大于。
- >= 大于等于。
- = 等于。
- == 精确等于。

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代数值的大小进行比较。

例如，在命令窗口输入如下命令：

```
?8>6
```

在主窗口显示表达式的值为：

```
.T.
```

② 日期型和日期时间型数据进行比较时，离现在日期或时间越近的日期或时间越大。

例如，在命令窗口输入如下命令：

```
?{^2009/4/8}>{^2006/5/8}
```

在主窗口显示表达式的值为：

```
.T.
```

③ 逻辑型数据比较时，.T. 比 .F. 大。

例如，在命令窗口输入如下命令：

```
? (5>3) > (5<3)
```

在主窗口显示表达式的值为：

```
.T.
```

④ 单个字符型数据按其 ASCII 码值的大小比较。

例如，在命令窗口输入如下命令：

```
?"a"<"b"
```

在主窗口显示表达式的值为：

```
.T.
```

注意：默认情况下，汉字按拼音(一级字库)或部首(二级字库)比较。排列次序可以重新设置，方法为选择“工具”→“选项”→“数据”→“排序序列”命令。

也可以用命令设置字符的排序次序：

```
SET COLLATE TO "<排序次序名>"
```

排序次序名可以是 Machine(机器内码)、PinYin(拼音)或 Stroke(笔画)。

⑤ 比较字符串时，按从左到右的顺序，依次比较每个位置上的字符，直到得出比较结果为止。

例如，在命令窗口输入如下命令：

```
?"abcd">"abca"
```

在主窗口显示表达式的值为：

```
.T.
```

⑥ 在系统默认状态下，关系运算符“=”比较数值型数据时，与数学上的等号意义相同。而用于比较字符串时，当“=”号右边的串与“=”号左边的串的前几个字符相同时，运算结果即为真。

例如，在命令窗口输入如下命令：

```
?"Visual FoxPro"="visual","Visual FoxPro"="foxpro"
```

在主窗口显示表达式的值为：

```
.T. .F.
```

⑦ “= =”用于更精确的比较。如果用它比较字符型数据，只有两个字符串完全相

同时，结果才为逻辑真。

例如，在命令窗口输入如下命令：

```
?"abc"=="aBc"
```

在主窗口显示表达式的值为：

```
.F.
```

可以用命令 SET EXACT ON 来设置字符串精确比较，此时，=和==的作用相同；用命令 SET EXACT OFF 可设置字符串非精确比较，此时，=和==的作用是不相同的，==为精确比较，=为非精确比较。

3.3.4 逻辑运算符和逻辑表达式

逻辑型表达式是由逻辑运算符将逻辑型数据连接起来的式子，其结果仍然是逻辑值。

Visual FoxPro 6.0 中的逻辑运算符有：

- .NOT. 或 NOT 或！　逻辑非。
- .AND. 或 AND　逻辑与。
- .OR. 或 OR　逻辑或。

逻辑运算符的优先级依次为 NOT、AND、OR，同级运算按从左到右的顺序运算。

例如，在命令窗口输入如下命令：

```
?!((5>3) and (5<3))
```

在主窗口显示表达式的值为：

```
.T.
```

3.3.5 日期时间运算符和日期时间表达式

日期时间型表达式是指含有日期型或日期时间型数据的表达式。日期型数据或日期时间型数据是一种特殊的数据，它们的运算有以下 6 种情况。

① 两个日期型数据可以相减，结果为数值，表示两个日期之间相差的天数。

例如，在命令窗口输入如下命令：

```
?{^2009-5-16}-{^2009-5-11}
```

在主窗口显示表达式的值为：

```
5
```

② 日期型数据加上一个整数，其结果是将来的某个日期。该整数代表天数。

例如，在命令窗口输入如下命令：

```
?{^2009-5-16}+3
```

在主窗口显示表达式的值为：