



Visual Foxpro 6.0

教程

于凌云 王计堂 主编

中国矿业大学出版社



Visual Foxpro 6.0

Visual FoxPro 6.0 教程

主 编 于凌云 王计堂
副主编 董红卫 胡军利

中国矿业大学出版社

图书在版编目(CIP)数据

Visual Foxpro 6.0 教程/于凌云,王计堂主编.

徐州:中国矿业大学出版社,2002.12

ISBN 7-81070-621-7

I. V... I. ①于...②王... III. 关系数据库—数据库管理系统, Visual Foxpro 6.0—教材

IV. TP311.138

中国版本图书馆 CIP 数据核字(2003)第 007086 号

- 书 名 Visual Foxpro 6.0 教程
主 编 于凌云 王计堂
责任编辑 李士峰
出版发行 中国矿业大学出版社
(江苏省徐州市中国矿业大学内 邮编 221008)
排 版 中国矿业大学出版社排版中心
印 刷 中国矿业大学印刷厂
经 销 新华书店
开 本 787×1092 1/16 印张 14.5 字数 350 千字
版次印次 2002 年 12 月第 1 版 2002 年 12 月第 1 次印刷
印 数 1~4000 册
定 价 19.80 元
(图书出现印装质量问题,本社负责调换)

前 言

Visual FoxPro 6.0 关系型数据库系统是新一代数据库管理系统的杰出代表,它以其完美的性能、丰富的工具、超高的运行速度、友好的界面和完备的兼容性等特点,吸引了众多的用户,成为当前 PC 机上最流行的数据库管理系统。

Visual FoxPro 6.0 是为数据库结构和应用程序开发而设计的、功能强大的、面向对象的环境。无论是组织信息、运行查询、创建集成的关系型数据库系统,还是为最终用户编写功能全面的数据管理应用程序,Visual FoxPro 6.0 都可以提供管理数据所需的工具。它采用了可视化的、面向对象的程序设计方法,大大简化了应用系统的开发过程,并提高了系统的模块性和紧凑性。Visual FoxPro 6.0 提供了大量的系统开发工具和向导工具,使以往费时费力的开发工作变得轻松自如。

本书是掌握 Visual FoxPro 6.0 关系数据库管理系统及利用 Visual FoxPro 6.0 进行应用系统开发的一本较全面的教科书和实验指导书。本书可以作为高职、高专、中职学校学生学习的基本教材,也适合于计算机用户和计算机技术初学者使用。

本书共 12 章,第 1 章介绍了数据库的基本概念及 Visual FoxPro 6.0 概述;第 2 章至第 5 章介绍了数据的操作及组织管理;第 6、7 章介绍了结构化程序设计及面向对象的程序设计方法;第 8、9 章介绍了表单的创建及设计;第 10、11 章介绍了报表、标签和菜单的设计;第 12 章介绍了应用系统的开发。

本书由中国矿业大学夏阳老师审定。在本书编写的过程中,得到了王维峰、张格余等同志的大力支持,在此表示诚挚的谢意。由于时间仓促,加上水平有限,书中不当之处,恳请广大读者批评指正。

编者
2002 年 10 月

目 录

前言	1
第 1 章 数据库系统及 Visual FoxPro 6.0 概述	1
1.1 数据库系统及相关概念	1
1.1.1 数据和数据处理	1
1.1.2 数据库系统	1
1.1.3 数据模型	2
1.2 数据存储与操作	3
1.2.1 数据库类型	3
1.2.2 数据存储	4
1.3 函数	5
1.4 表达式	9
1.4.1 算术表达式	9
1.4.2 字符表达式	9
1.4.3 时间日期表达式	9
1.4.4 关系表达式	10
1.4.5 逻辑表达式	10
1.5 Visual FoxPro 6.0 概述	11
1.5.1 Visual FoxPro 6.0 的安装	11
1.5.2 Visual FoxPro 6.0 的启动	11
1.5.3 Visual FoxPro 6.0 的退出	11
1.5.4 Visual FoxPro 6.0 的窗口组成	11
1.6 定制 Visual FoxPro 6.0 环境	18
1.6.1 使用“选项”对话框	18
1.6.2 使用 SET 命令配置 Visual FoxPro 6.0	19
1.6.3 使用配置文件	19
思考题与实验	19
第 2 章 数据表的建立及操作	21
2.1 数据表的概述	21
2.1.1 数据表的概念	21

2.1.2	数据表的组成	21
2.1.3	数据表的结构	22
2.2	数据表结构的建立与修改	23
2.2.1	利用表设计器建立数据表	23
2.2.2	利用表向导建立数据表	25
2.2.3	利用表设计器修改表结构	25
2.3	输入数据	27
2.3.1	建立数据表时立即输入数据	27
2.3.2	追加数据	27
2.3.3	备注型字段的数据输入	28
2.3.4	通用型字段的数据输入	30
2.4	数据的显示与修改	31
2.4.1	数据的显示	31
2.4.2	记录的条件显示	32
2.4.3	字段的筛选	34
2.4.4	记录的定位	35
2.5	记录的删除	38
2.5.1	逻辑删除	38
2.5.2	恢复数据	39
2.5.3	物理删除	41
2.6	索引的创建与操作	41
2.6.1	索引概述	41
2.6.2	创建索引	42
2.6.3	利用表达式建立索引排序	43
2.6.4	索引的使用	44
	思考题与实验	46

第3章	数据库的建立及操作	49
3.1	数据库概述	49
3.1.1	数据库表的关系	49
3.1.2	数据库的设计步骤	50
3.2	数据库的建立与使用	51
3.2.1	数据库的建立	51
3.2.2	数据库的打开	52
3.2.3	向数据库添加数据表	53
3.2.4	在数据库新建数据库表	54
3.2.5	删除数据库中的数据库表	55
3.3	数据库表字段的属性	56
3.3.1	设置字段的标题	56

3.3.2 设置字段的注释.....	58
3.3.3 设置字段的默认值.....	58
3.3.4 设置字段的有效性规则.....	59
3.4 数据库表的表属性.....	60
3.4.1 长表名.....	60
3.4.2 表记录的验证规则.....	60
3.4.3 表的触发器.....	61
3.5 数据库中数据库表间的关联.....	61
3.5.1 建立数据库表间的关联.....	61
3.5.2 设置参照完整性.....	63
3.6 数据工作期.....	64
思考题与实验	66
第4章 查询和视图	67
4.1 建立查询.....	67
4.1.1 利用查询向导创建单表查询.....	67
4.1.2 利用查询设计器创建多表查询.....	71
4.2 定制并运行查询.....	74
4.2.1 定制查询.....	74
4.2.2 运行查询.....	76
4.3 建立视图.....	77
4.3.1 利用视图向导创建本地视图.....	77
4.3.2 利用视图设计器创建本地视图.....	80
思考题及实验	85
第5章 项目管理器	87
5.1 建立项目管理器.....	87
5.2 项目管理器对话框组成.....	88
5.2.1 选项卡说明.....	88
5.2.2 查看详细内容.....	89
5.2.3 命令按钮的功能.....	90
5.3 项目管理器的使用.....	90
5.3.1 添加文件.....	90
5.3.2 移去文件.....	91
5.3.3 创建和修改文件.....	92
5.4 定制项目管理器.....	93
5.4.1 改变显示外观	93
5.4.2 改变选项卡的位置.....	93
思考题与实验	94

第 6 章 Visual FoxPro 6.0 程序设计基础	96
6.1 程序文件的建立与执行.....	96
6.1.1 程序文件的建立与编辑.....	96
6.1.2 程序的运行.....	97
6.1.3 程序中命令的书写规则.....	99
6.1.4 基本命令语句	100
6.2 程序文件的基本结构	101
6.2.1 顺序结构	101
6.2.2 分支结构	101
6.2.3 循环结构	108
6.3 过程设计	112
6.3.1 过程	112
6.3.2 过程文件	113
6.3.3 过程的调用	113
6.4 程序调试	114
思考题与实验.....	117
 第 7 章 面向对象的编程概述.....	 121
7.1 面向对象编程的相关概念	121
7.1.1 面向对象的程序设计方法	121
7.1.2 对象	121
7.1.3 类	121
7.1.4 属性	122
7.1.5 事件	122
7.1.6 方法	122
7.2 基 类	122
7.2.1 类的特性	122
7.2.2 类的分类	122
7.3 处理对象	124
7.3.1 引用对象	124
7.3.2 设置对象属性	124
7.3.3 调用对象的方法程序	124
7.3.4 对象对事件的响应	125
7.4 相关事件	125
7.4.1 核心事件	125
7.4.2 事件激发的顺序	126
7.5 方 法	126
7.6 类的创建	127

7.6.1 创建类	127
7.6.2 类属性的定义	128
7.6.3 定义类的方法和事件	130
思考题	131
第8章 表单设计	132
8.1 建立表单	132
8.1.1 利用“表单向导”建立表单	132
8.1.2 利用“表单向导”建立一对多表单	135
8.1.3 利用“表单设计器”建立表单	140
8.2 修改表单	141
8.3 运行表单	143
思考题与实验	144
第9章 表单控件	145
9.1 表单控件及工具栏概述	145
9.1.1 “表单设计器”工具栏	145
9.1.2 “表单控件”工具栏	146
9.1.3 属性窗口	147
9.1.4 “布局”工具栏	148
9.1.5 “调色板”工具栏	150
9.2 表单控件的使用	150
9.3 增强控件的易用性	166
9.3.1 设置控件的“Tab 键次序”	166
9.3.2 设置访问键	167
9.3.3 设置工具提示文本	167
9.3.4 启用和停用控件	168
思考题与实验	168
第10章 报表和标签设计	170
10.1 创建报表	170
10.1.1 利用“报表向导”创建报表	170
10.1.2 利用“报表设计器”修改报表	174
10.2 标签设计	176
10.2.1 利用“标签向导”创建标签	176
10.2.2 利用“标签设计器”修改标签	179
思考题与实验	180

第 11 章 菜单设计	182
11.1 建立菜单系统.....	182
11.1.1 建立菜单.....	182
11.1.2 指定菜单项任务.....	186
11.1.3 生成并执行菜单.....	186
11.2 建立快捷菜单.....	187
11.3 快速菜单.....	189
思考题与实验.....	190
第 12 章 应用系统的开发	192
12.1 系统开发的一般过程.....	192
12.2 建立数据库.....	193
12.3 设计用户界面.....	193
12.4 测试与调试.....	194
12.5 构造应用系统框架.....	194
12.6 应用系统连编及运行.....	195
附录 1 成绩表和课程表的结构及数据	197
附录 2 常用函数一览表	198
附录 3 操作命令一览表	205
附录 4 控件类名称及功能	210
附录 5 对象类名称及功能	211
附录 6 属性语法与功能	212

第1章 数据库系统及 Visual FoxPro 6.0 概述

数据库是现代计算机系统的一个重要部分,它使用数据库技术对数据资源进行统一管理,使数据能为各类用户和应用程序所共享,数据库技术已成为当今信息技术中应用最广泛的技术之一。

1.1 数据库系统及相关概念

1.1.1 数据和数据处理

数据是反映客观事物属性的记录。从计算机的角度看,数据泛指那些可以被计算机接受并能够被计算机处理的符号。数据一般归纳为三种表现形式:数值型数据,是对客观事物进行定量记录的符号,如年龄、工资等;字符型数据,是对客观事物进行定性记录的符号,如姓名、性别、政治面貌等;特殊型数据,如声音、视频、图像等。

数据处理是指对各种类型的数据进行采集、存储、分类、计算、检索、加工和传输的过程。数据处理的目的是从大量的、原始的数据中获得人们所需要的资料并提取有用的数据成分,作为行为和决策的依据。随着电子计算机软件和硬件技术的发展,数据处理过程也发生了巨大的变化。

1.1.2 数据库系统

1. 数据库

数据库(DB)是以一定的组织方式将相关的数据组织在一起存放在计算机存储器中,并能为多个用户共享、与应用程序彼此独立的一组相关数据的集合。如把一个学校学生的基本情况等数据有序地组织并存储起来,可以构成一个数据库,并随时可以查询该数据库的有关信息。

2. 数据库管理系统

数据库管理系统(DBMS)是对数据库进行管理的系统软件,是用户与数据库之间的接口,它提供了用户对数据库进行操作的各种命令、工具及方法。Visual FoxPro 6.0 就是目前广为流行的数据库管理系统之一,它改变了以往的 XBase 系列的面貌,功能更强,操作更直观,使用更灵活。

3. 数据库系统

数据库系统(DBS)是实现有组织地、动态地存储大量关联数据,方便用户访问计算机的软、硬件资源而组成的具有管理数据库功能的计算机系统。它由数据库、数据库管理系统、操

作系统、计算机硬件资源等组成。

数据库、数据库管理系统、数据库系统是数据库的基本概念,三者之间既有联系又有区别。数据库和数据库管理系统是数据库系统的组成部分,用户通过数据库管理系统可以建立和使用数据库。使用数据库是目的,而数据库管理系统是实现这一目的的手段和工具。

1.1.3 数据模型

现实世界中的客观事物是彼此相互联系的。客观事物的这种联系性决定了作为事物属性记录符号的数据与数据之间也存在着一定的联系性。具有联系性的相关数据总是按照一定的组织关系排列,从而构成一定的结构,对这种结构的描述就是数据模型。

客观事物是千变万化的,各种客观事物的数据模型也是千差万别的,但也有其共性。常用的数据模型有层次模型、网状模型和关系模型。

1. 层次模型

层次模型表示数据间的从属关系结构,是一种以记录某一事物的类型为根结点的有向树结构。层次模型像一棵倒置的树,根结点在上,层次最高;子结点在下,逐层排列。其主要特征如下:

- 仅有一个无双亲的根结点。
- 根结点以外的子结点,向上仅有一个父结点,向下有若干子结点。

层次模型表示的是从根结点到子结点的一个结点对多个结点,或从子结点到父结点的多个结点对一个结点的数据间的联系。

2. 网状模型

网状模型是层次模型的扩展,表示多个从属关系的层次结构,呈现一种交叉关系的网络结构。网状模型是以记录为结点的网络结构。其主要特征如下:

- 有一个以上的结点无双亲。
- 至少有一个结点有多个双亲。

网状模型可以表示较复杂的数据结构,即可以表示数据间的纵向关系与横向关系。这种数据模型在概念上、结构上都比较复杂,操作上也有很多不便。

3. 关系模型

关系模型的所谓“关系”,是有其特定含义的,广义地说,任何数据模型都描述一定事物数据之间的关系。关系模型的主要特点有:

- 关系中每一数据项不可再分,是最基本的单位。
- 每一竖列数据项是同属性的。列数根据需要而设,各列的顺序可以任意。
- 每一横行记录由一个事物的诸多属性项构成,记录的顺序可以任意。
- 一个关系是一张二维表,不允许有相同的字段名,也不允许有完全相同的记录行。

关系模型既可以反映客观事物数据间的一对一的联系,又能反映客观事物数据间的一对多的联系,还能反映客观事物数据间的多对多的联系。Visual FoxPro 6.0 所管理的数据,就是依照关系模型进行存储的,是新一代的关系型数据库管理系统。

1.2 数据存储与操作

1.2.1 数据类型

数据是反映客观事物属性的记录。Visual FoxPro 6.0 系统为了使用户能够更加方便地建立和使用数据库,将数据细化为以下几种类型。

1. 字符型(Character)

字符型数据是由汉字和 ASCII 字符集中可打印字符(英文字符、数字字符、空格及其他专用字符)组成,长度范围是 0~254 个字符,使用时必须用定界符双引号(")或单引号(')或方括号([])括起来(注意,必须是英文半角的单引号、双引号和方括号)。如"DJS"、'Z001'、[会计电算化专业]。

2. 数值型(Numeric)

数值型数据是由数字(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)。

3. 浮点型(Float)

浮点型数据是数值型数据的一种,与数值型数据完全等价。它只是在存储形式上采取浮点格式。

4. 双精度型(Double)

双精度型数据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。

5. 整型(Integer)

整型数据是不包含小数点部分的数值型数据。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。

6. 日期型(Date)

日期型数据是用于表示日期的数据,用默认格式{mm/dd/yyyy}来表示。其中,mm 代表月,dd 代表日,yyyy 代表年,长度固定为 8 位。

7. 日期时间型(DateTime)

日期时间型数据是描述日期和时间的数据。其默认格式为{mm/dd/yyyy hh:mm:ss}。其中,yyyy 代表年,前两个 mm 代表月,dd 代表日,hb 代表小时,后两个 mm 代表分钟,ss 代表秒,{hh:mm:ss}长度固定为 8 位。

8. 逻辑型(Logic)

逻辑型数据是描述客观事物真和假的数据,用于表示逻辑判断结果。逻辑型数据只有真(.T. 或 .t.)和假(.F. 或 .f.)两种值,长度固定为 1 位。

9. 备注型(Memo)

备注型数据是用于存放较长的字符型数据的数据类型,可以把它看成是字符型数据的特殊形式。它只用于数据表中的字段类型的定义,其字段长度固定为 10 位,而实际数据被存放在与数据表文件同名且扩展名为 .FPT 的备注文件中,文件长度根据数据的内容而定。

10. 货币型(Currency)

当涉及货币时,可使用货币类型代替数值类型。在货币表达式中,如果小数位数超过 4 位,则系统将在处理表达式之前把它四舍五入到 4 位。

11. 通用型 (General)

通用型数据是用于存储 OLE 对象的数据,可以是电子表格、文档、图片等。它只用于数据表中的字段类型的定义。数据长度固定为 4 位,实际数据长度受限于现有的磁盘空间。

12. 二进制字符型 (Binary Character)

在各种代码页间保持不变的字符型数据。如保存在数据表中的用户口令,可用于不同国家。

13. 二进制备注型 (Binary Memo)

在各种代码页间保持不变的备注型数据。它可用于不同国家的登录脚本。

1.2.2 数据存储

一般程序设计语言可以使用常量、变量和数组,而在 Visual FoxPro 6.0 系统环境下,数据还可以存入到字段、记录 and 对象中。我们把这些供数据存储的常量、变量、数组、字段、记录和对象称为数据存储容器。

1. 常量

在数据处理过程中其值不发生变化的量叫常量。常用的常量有以下几种:

- (1) 数值型:由数字(0~9)、小数点和正负号组成,如 5.18、-660。
- (2) 字符型:由汉字和 ASCII 字符集中可打印字符组成的字符串,使用时必须用定界符双引号(" ")或单引号(' ')或方括号([])括起来,如"ASDF"、[张小明]。
- (3) 日期型和日期时间型:用于表示日期或日期和时间,可以用定界符({^})括起来,如{^ 2001/05/19}、{^ 1990/12/27 20:30:00}。
- (4) 逻辑型:表示逻辑判断的结果“真”、“假”,如.T.(.t.)或.F.(.f.)。

2. 变量

变量是在命令操作、程序运行过程中其值可以变化的量,包括:内存变量(简单变量和数组)、字段变量。变量以变量名来标识,其一般命名规则如表 1-1 所示。

表 1-1 变量名的命名规则

内存变量	字段变量
以字母或汉字开头	以字母或汉字开头
至多 254 个字符	至多 10 个字符
由字母、汉字、数字、下划线组成	由字母、汉字、数字、下划线组成
不可与系统保留字同名	不可与系统保留字同名

(1) 内存变量(简单变量)

内存变量是在数据运算中用于存放临时性数据或中间运算结果的变量。内存变量的数据类型由其值的数据类型决定,并随着值的类型的改变而改变,主要有数值型、字符型、日期型、逻辑型、屏幕型等几种类型。内存变量的赋值、显示、删除命令如下:

- 赋值:〈内存变量〉=〈表达式〉或 STORE 〈表达式〉 TO 〈内存变量〉

例如:

```
STORE "VFP6.0" TO SJ1
```

```
SJ1="VFP6.0"
```

- 显示:LIST|DISPLAY MEMORY [LIKE <通配符>][TO PRINT]

例如:

```
LIST MEMORY LIKE * &&. 显示所有的内存变量
```

```
DISPLAY MEMORY LIKE * &&. 分屏显示所有的内存变量
```

- 删除:RELEASE [<内存变量表>][ALL][LIKE|EXCEPT<通配符>]]

(2) 数组

数组是按一定顺序排列的一组内存变量,数组中的各个变量称为数组元素。数组必须先定义后使用。

数组的定义格式为:

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

例如:

```
DIMENSION m(3),该数组是一维数组,其中包括三个数据元素:m(1),m(2),m(3)
```

```
DIMENSION a(2,3),该数组是二维数组,其中包括六个数据元素:a(1,1),a(1,2),a(1,3),a(2,1),a(2,2),a(2,3)
```

数组定义后,初值为逻辑假(.F.),重新赋值后,数据类型由其值决定。数组中各元素的数据类型可以相同,也可不同。

(3) 字段变量(简称字段)

字段变量是数据库管理系统中的一个重要概念,是指数据表中已定义的任意一个字段,有数值型、字符型、逻辑型、日期型、时间日期型、备注型和通用型等。字段变量的定义需要在“表设计器”中进行。

当内存变量与数据表中的字段变量同名时,用户在引用内存变量时,需要在其名字前加一个“m.”,用于强调这一变量是内存变量。

3. 记录

记录是数据表中一组数据项的集合。在同一个数据表中可以有若干个记录,每一个记录具有相同的字段个数,记录的顺序不影响对记录的操作。在 Visual FoxPro 6.0 系统中,许多操作都是通过对记录的操作来完成的。

1.3 函 数

函数,是针对一些常见问题预先编好的一系列子程序。在 Visual FoxPro 6.0 系统中,一个函数可以完成一个指定的功能。当用户应用时遇到此类问题就可以调用相应的函数,也就是执行相应的函数子程序。

函数的一般形式:函数名(<自变量表>)

其中,函数名是系统规定的,自变量表可以是一个变量或多个自变量,也可以为空,即:函数名()。用户需要注意的是函数名后面的一对圆括号不能省略。

Visual FoxPro 6.0 提供了许多功能丰富的函数(常用函数见附录 2),下面给出几个常

用函数的应用举例。

1. 求整函数 INT

格式:INT(<数值表达式>)

功能:取数值表达式的整数部分。

例如:

? INT(56.18)→56

? INT(-28.5)→-28

2. 四舍五入函数 ROUND

格式:ROUND(<数值表达式>,<保留小数位>)

功能:对数值表达式按保留小数位数进行四舍五入。

说明:当<保留小数位>为正整数或零时,系统将对小数位数进行四舍五入处理;当<保留小数位>为负数时,系统对整数部分进行四舍五入处理。

例如:

? ROUND(3.14159,3)→3.142

? ROUND(3.14159,0)→3

? ROUND(536.56,-1)→540

? ROUND(536.56,-2)→500

3. 求子串函数 SUBSTR

格式:SUBSTR(<字符表达式>,<起始位置>[,<长度>])

功能:对<字符表达式>从给定的<起始位置>开始,截取指定<长度>的字符,生成一个新的字符串。

说明:若无<长度>或<长度>大于剩余的字符个数,则截至末尾。若<起始位置>大于字符串表达式长度,则输出空串。

例如:

? SUBSTR("ABCDEFGHIJK",4,6)→DEFGHI

? SUBSTR("ABCDEFGHIJK",4)→DEFGHIJK

? SUBSTR("ABCDEFGHIJK",12)→(空字符串)

4. 日期和时间函数

(1) 系统日期函数 DATE

格式:DATE()

功能:返回系统的当前日期。

例如:

? DATE()→05/01/02

(2) 系统时间函数 TIME

格式:TIME()

功能:返回系统的当前时间。

例如:

? TIME()→19:50:43

(3) 年、月、日函数 YEAR|MONTH|DAY

格式:YEAR|MONTH|DAY(<日期表达式>)

功能:从<日期表达式>中求出年、月、日的数值。

例如:

? DATE(),YEAR(DATE())→05/01/02 2002

? DATE(),MONTH(DATE())→05/01/02 05

? DATE(),DAY(DATE())→05/01/02 01

5. 类型转换函数

(1) 数值转换为字符函数 STR

格式:STR(<数值表达式>[,<长度>][,<小数位数>])

功能:把<数值表达式>的运算结果转换为指定长度的字符串。

说明:

- <长度>包括小数点在内,不指定长度时,系统默认整数位 10 位。
- 若指定<长度>小于<数值表达式>整数位数,则用一串“*”号表示数据溢出;若<长度>为 0,则返回空字符串。

- 若省略<小数位数>,则只转换整数位;若指定<小数位数>,则对下一位四舍五入。

例如:

? STR(1357.2468,7,2)→1357.25

? STR(1357.2468)→1357

? STR(1357.2468,3)→* * *

? STR(1357.2468,0)→(空字符串)

(2) 字符转换为日期函数 CTOD

格式:CTOD(<字符表达式>)

功能:将符合 yy/mm/dd、mm/dd/yy 日期格式的字符串转换为相应日期。

例如:

? CTOD("05/11/02")→05/11/02

(3) 日期转换为字符函数 DTOC

格式:DTOC(<日期表达式>[,1])

功能:将<日期表达式>转换为相应的字符串。

说明:不选用 1 时,按 mm/dd/yy 格式转换;选用 1 时,按 yyyymmdd 格式转换。

例如:

? DTOC(DATE())→05/01/02

? DTOC(DATE(),1)→20020501

6. 测试函数

(1) 记录号测试函数 RECNO

格式:RECNO(<数值表达式>)

功能:给出<数值表达式>指定的工作区中打开的数据库的当前记录号。

说明:

- 不选用<数值表达式>时,给出当前工作区数据库的当前记录号。
- 若<数值表达式>指定的工作区没有打开数据库文件,则返回值为 0。