



教育部 高职高专规划教材
Jiaoyubu Gaozhi Gaozhuan Guihua Jiaocai

Visual FoxPro 6.0 简明教程

魏绍谦 陈万里 编著

1.138FO-4

高等教育出版社

HIGHER EDUCATION PRESS



341
2004-10-21
0.02

教育部高职高专规划教材

Visual FoxPro 6.0 简明教程

魏绍谦 陈万里 编著



A0938158

高等教育出版社

内容提要

本书是“教育部高职高专规划教材”之一,按教育部最新制定的《高职高专教育数据库应用技术课程教学基本要求》编写。

本书主要包括:Visual FoxPro 6.0 软件的安装方法;项目管理器的使用;数据库和数据表的建立与使用;向导和视图设计器的使用;利用向导或设计器设计查询、表单、报表和标签;菜单和工具栏的设计;过程化程序设计以及面向对象程序设计等。

本书可作为高等专科学校、高等职业学校、成人高等学校及本科院校举办的二级职业技术学院经管类各专业的通用教材,同时也可作为其他专业及各类培训班的教材或自学参考书。

图书在版编目(CIP)数据

Visual FoxPro 6.0 简明教程/魏绍谦,陈万里编著.

—北京:高等教育出版社,2000

教育部高职高专规划教材

ISBN 7-04-008716-2

I.V… II.①魏…②陈… III.关系数据库-数据库管理系统,Visual FoxPro-高等教育-教材
IV.TP311.138

中国版本图书馆 CIP 数据核字(2000)第 62356 号

Visual FoxPro 6.0 简明教程

魏绍谦 陈万里 编著

出版发行 高等教育出版社

社 址 北京市东城区沙滩后街 55 号

电 话 010-64054588

网 址 <http://www.hep.edu.cn>

邮政编码 100009

传 真 010-64014048

经 销 新华书店北京发行所

印 刷 国防工业出版社印刷厂

开 本 787×1092 1/16

印 张 15.5

字 数 350 000

版 次 2000 年 7 月第 1 版

印 次 2000 年 7 月第 1 次印刷

定 价 19.70 元

本书如有缺页、倒页、脱页等质量问题,请到所购图书销售部门联系调换。

版权所有 侵权必究

第 1 章 Visual FoxPro 6.0 概述

本章主要介绍 Visual FoxPro 6.0 的特点、安装、系统界面以及 Visual FoxPro 6.0 的基本语法。

1.1 Visual FoxPro 6.0 简介

Visual FoxPro 6.0 (中文版, 简称 VFP 6.0) 是由 Microsoft 公司于 1998 年推出的新一代软件系统, 它将面向对象的程序设计技术与关系型数据库系统有机地结合在一起, 是具有更强大功能的可视化程序设计的关系数据库系统。

Visual FoxPro 6.0 集数据库和程序设计语言于一体, 可以设计许多小型数据库系统。它在 VFP 5.0 的基础上更加重了项目管理器、向导、生成器、查询与视图、OLE 连接、Active 集成、帮助系统制作、数据的导入和导出以及面向对象的程序设计等方面的技术力度。

Visual FoxPro 6.0 与 Windows 98 操作系统以及 Office 办公软件都可以很好地交流, 可以制作出更加专业化的软件。它在客户机/服务器应用技术、远程数据共享、数据安全管理和文档管理等方面, 具有很强的优势, 非常适合于制作各种数据库应用程序。

1.2 Visual FoxPro 6.0 的安装

1.2.1 安装的系统环境

1. Visual FoxPro 6.0 所需要的硬件环境

- 80486/50MHz 或更高的计算机系统。
- 鼠标器。
- 16MB 以上的内存存储器。
- 最小化安装需要 15MB 硬盘空间; 典型安装需要 100MB 硬盘空间; 最大安装需要 240MB 硬盘空间。
- VGA 或更高分辨率的显示器, 1MB 以上显示缓存。

2. Visual FoxPro 6.0 所需要的软件环境

Windows 95/98 操作系统 (中文版) 或 Windows NT 3.51 (中文版) 或更高版本。

3. Visual FoxPro 6.0 所需要的网络环境

服务器: SQL Server for Windows NT。

客户机: 包括 ODBC 组件的 Visual FoxPro 6.0。

网络: Novell Netware; Windows NT。

1.2.2 Visual FoxPro 6.0 的安装

直接启动 CD-ROM 的安装方法是:

(1) 将 Visual FoxPro 6.0 系统光盘插入到 CD-ROM 驱动器中, 安装程序自动运行, 进入“Visual FoxPro 6.0 安装向导”窗口。

(2) 在“Visual FoxPro 6.0 安装向导”窗口, 用户可以选择安装 Visual FoxPro 6.0 有关选项, 单击“下一步”按钮进入“Visual FoxPro 6.0 安装程序”窗口。

(3) 在“Visual FoxPro 6.0 安装程序”窗口, 系统提供了三种可以选择的安装方式:

1) 选择“添加/删除”按钮, 为当前安装添加新组件, 或删除已安装的组件。

2) 选择“重新安装”按钮, 重复上一次安装, 恢复丢失的文件和重新设置系统。

3) 选择“全部删除”按钮, 删除已有的全部组件。

如果是第一次安装 Visual FoxPro 6.0 或想添加一些新组件, 可选择“添加/删除”按钮; 如果只是想恢复丢失的文件, 可选择“重新安装”按钮。

(4) 确定了安装方式后, 系统开始安装。在安装过程中, 用户要回答安装程序所提出的各种问题, 按步骤选择相应的选项完成安装过程。

1.3 Visual FoxPro 6.0 系统界面简介

1.3.1 Visual FoxPro 6.0 的菜单系统

Visual FoxPro 6.0 也是 Windows 95/98 的应用程序, Windows 95/98 窗口的所有操作方法(如移动、拉伸、缩小等功能)对它都适用。

Visual FoxPro 6.0 主要由菜单、工具条、命令窗口以及各种对话框组成, 用户既可以在命令窗口中输入命令, 也可以使用菜单和对话框来完成所有的操作。

Visual FoxPro 6.0 的菜单系统遵循 Windows 95/98 的方式: 菜单条横跨在窗口顶部, 菜单项出现在菜单条中, 单击菜单项时弹出一子菜单, 此时选择其中一项就可以执行一个功能。

选择菜单有几种不同的方法, 最简单的方法是使用鼠标。将鼠标光标定位在某一菜单项上, 然后单击左键打开子菜单, 从弹出的命令列表中选择要执行的命令, 再单击左键。如果不想选命令而退出菜单, 只需在菜单外面单击左键或按一下 F10 键。使用鼠标选择菜单的另一种方法是打开弹出列表后, 键入带有下划线的字符。欲激活菜单条还可以使用 Alt 键或 F10 键, 菜单条激活后就可用上述方法进行选择, 或者使用四个光标移动键

(←、↑、→、↓) 移动菜单，在光标移到要选择的命令上后按回车键即可执行该命令，用 Esc 键取消对菜单的选择。

如果菜单项后面带有组合键，比如“编辑”菜单中的“撤消 Ctrl+Z”，这表明该项功能可以通过快捷键实现，使用快捷键可以不激活菜单条而直接执行该项命令，从而节省用户选择菜单的时间；如果菜单项右边有一个黑三角时，说明该菜单项尚有子菜单，可提供更详细的选择项目；如果菜单项右边有省略号(…)，这说明该项命令的执行将调出一个对话框，在对话框中可更方便地进行信息输入以及各种选择；如果菜单系统中某些项目呈灰色，表示该项目在当前状态下不具备使用条件，不能使用。

菜单分为动态菜单和弹出菜单。所谓动态菜单是指当程序执行某项功能时系统主菜单及其主菜单下的子菜单的增减；而弹出菜单则是指当用户处于某特定区域时单击鼠标右键而弹出的一个菜单项。

一般情况下，Visual FoxPro 6.0 仅含若干菜单项及其对应的子菜单，在程序运行过程中用到某些功能时，系统将会动态地增加或修改一些菜单项。

Visual FoxPro 6.0 中众多工具条、对话框、窗口、设计器等都具有弹出菜单。当鼠标光标移至某一区域，然后单击鼠标右键即可将其打开并可从中选择。如果要关闭弹出菜单，只需要将鼠标移到弹出的菜单之外，然后单击鼠标左键即可。实际上，弹出菜单的选项大多数都可在相应主菜单或对话框的按钮上找到，但使用它更为快捷。

1.3.2 Visual FoxPro 6.0 的工具栏

Visual FoxPro 6.0 一旦启动，在主窗口画面上方，首先呈现标准工具栏，如图 1.3.1 所示的是标准工具栏。

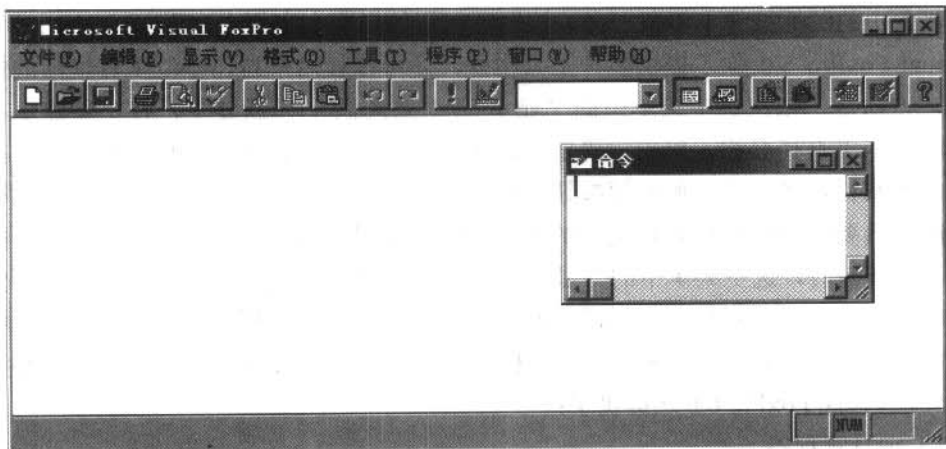


图 1.3.1 标准工具栏

在工具栏显示的按钮往往代表了最为常用的命令，有效地利用工具栏，能使程序的开发工作更加方便、快捷。除了标准工具栏外，Visual FoxPro 6.0 还为用户提供了十几种工具栏，在编辑相应的文档和窗口时，可选择所需要的工具栏。

选择所需要的工具栏的方法是：从“显示”菜单中选择“工具栏”，如 1.3.2 所示。

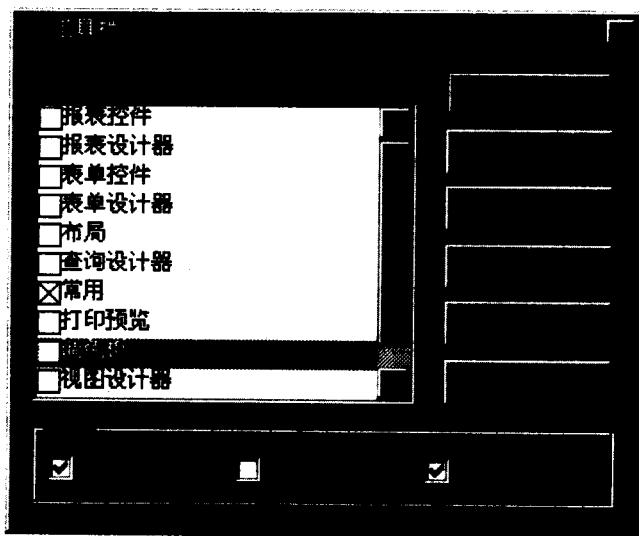


图 1.3.2 工具栏对话框

在这个工具栏对话框选项中有报表控件、报表设计器、表单控件、表单设计器、布局、查询设计器、常用、打印预览、调色板、视图设计器和数据库设计器。当选择要使用的工具栏选项时，点击该项并使其前面出现“☒”，然后单击“确定”按钮，这时就会在屏幕上弹出相应的工具条，将鼠标放在工具栏上，按住鼠标左键拖动，可将工具栏对话框或工具条移到屏幕上的任何位置，以便腾出更大的屏幕空间，用于编辑和窗口操作，将光标在工具条中的图标按钮上停留一会，该图标按钮的说明文字便会出现。

Visual FoxPro 6.0 除了提供多种工具栏外，还可以让用户自己定制工具栏内的快捷按钮图标，方法是：依次选主菜单中的“显示”/“工具栏”/“新建”，在“新建工具栏”对话框中输入工具栏名称，单击“确定”，画面上便会出现新的工具栏，但这个工具栏是一个空的工具栏，未建立任何快捷按钮图标，此时画面上出现“定制工具栏”对话框，让用户拖动所需要的图标按钮到自定义的工具栏中。如果要对已经设置好的工具栏图标按钮进行修改或增减时，选主菜单中的“显示”/“工具栏”，再选取要修改的工具栏，然后单击“定制”，进行工具栏修改。增减工具栏中的图标按钮时，用鼠标将要移出的图标按钮拖回定制工具栏对话框中，或把定制工具栏对话框中将要加入的图标按钮拖到工具栏中。

此外，Visual FoxPro 6.0 还提供了：

- (1) 项目管理器。
- (2) 设计器。
- (3) 向导。
- (4) 生成器。

以上这些工具我们将在后面用到时再作介绍。

1.3.3 命令窗口

在 Visual FoxPro 6.0 中, 菜单中的命令也能够通过命令窗口的输入来执行, 命令窗口是一个可编辑的窗口, 就像其他的文本窗口那样, 可在命令窗口中进行各种插入、删除、块拷贝等操作并用光标或滚条在整个命令窗口中上下移动, 这些特性极大地方便了命令的使用。当选择菜单命令时, 对应的命令行在命令窗口中显示出来, 运行进入 Visual FoxPro 6.0 系统后, 用户从菜单或命令窗口输入的命令到退出系统之前都具有有效性, 用户只需要将光标移到命令行上, 然后按回车键, 所选命令将再次执行。用户也可以在命令窗口中将本次进入 Visual FoxPro 6.0 系统后的任何一条已执行的命令加以修改, 然后再次执行。

在 Visual FoxPro 6.0 中, 命令与函数仅识别前四个字母, 即命令和函数只需输入前四个字母, 不过一旦输入多于四个字母, 则必须将该命令输入完整, 否则将是错误命令。如果在操作过程中, 命令窗口被覆盖或者隐藏起来了, 可以在主菜单的“窗口”项中选择“命令窗口”, 使之重新显示出来。通过右下角的中英文操作方式转换钮, 可选择中文或英文的输入方式。

用户可以利用鼠标移动和改变命令窗口的大小。当要移动窗口时, 将鼠标指向窗口的标题栏, 然后按住鼠标左键, 拖拉窗口到希望的位置, 放开左键。当要改变窗口的大小时, 只需要向内或者向外移动一条或多条边框。移动边框的方法是, 将鼠标的光标移到该边框, 当鼠标的光标箭头变为双箭头时, 按住鼠标左键将该边框拖拉到所需要的位置, 然后放开左键。

用鼠标单击命令窗口右边滚动的上、下箭头, 可查看窗口中的前后内容, 拖动中间的滑动块也可以达到同样的目的。当然, 使用窗口下边滚动条的左、右箭头也可以查看窗口中的左右内容。

若不需要命令窗口时, 用户可以用鼠标单击窗口右上角的按钮, 取消命令窗口。

1.4 Visual FoxPro 6.0 基本方法

1.4.1 Visual FoxPro 6.0 数据类型

Visual FoxPro 6.0 的数据有以下几种类型:

1. 字符型

字符型数据由汉字和字母、数字、空格等 ASCII 字符组成, 可用于存储字符串, 最大长度可达 254 个字符, 使用时必须用定界符 (“ ” , ‘ ’) 括起来。

2. 数值型

数值型数据是由数字字符 (0~9)、小数点和正负号 (+或-) 组成, 最大长度为 20 个字符。

3. 浮点型

浮点型数据与数值型数据完全等价，它由一个尾数、阶数及字母 E (e) 组成，它的主要目的是使计算机具有较高的精度。

4. 双精度型

双精度型数据是更高精度的数值型数据。它只用于数据表中字段类型的定义。

5. 整型

整型数据是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。

6. 日期型

日期型数据由默认格式 {mm/dd/yyyy} 来表示，其中“mm”代表“月”，“dd”代表“日”，“yyyy”代表“年”。

7. 日期时间型

日期时间型是由默认格式 {mm/dd/yyyy hh:mm:ss} 来表示，“hh”代表“小时”，后面的“mm”代表“分”，“ss”代表“秒”。

8. 逻辑型

逻辑型数据只有真 (T.) 和假 (F.) 两种值。

9. 通用型

通用型数据是用于存储 OLE 对象的数据。

10. 货币型

在使用货币值时，可以使用货币型数据。

11. 备注型

备注型用于字符型数据块的存储，只能用于数据表中字段的定义。

1.4.2 常量、变量及数组

1. 常量

常量是在所有的命令操作或程序运行过程中其值始终保持不变的量。常用的有：

- (1) 字符型常量：“ABC”或‘人民’。
- (2) 数值型常量：678 或 3.14。
- (3) 浮点型常量：123e+16 或 -356e-12。
- (4) 逻辑型常量：.T. 或 .F.。
- (5) 日期型常量：{10/17/1997}。
- (6) 时间日期型常量：{06/17/1997 08:16:36}

2. 变量

变量是命令操作和程序运行过程中其值允许变化的量。常用的有：

(1) 内存变量

内存变量存在内存之中，独立于数据库和表。内存变量是一种临时工作单元，通常用来保存中间结果或对数据库和表进行某种分析处理后得到的数据结果，或者用于控制流程，用到时可以随时定义，不用时又可以释放。定义内存变量时需要为它取名并赋初值。

内存变量名应以字母或汉字开头，由字母、数字、下划线组成（内存变量名建议不要超过8个字符，且不要与系统保留字同名）。

内存变量赋值命令格式：

<内存变量>=<表达式> 或

STORE<表达式> TO <内存变量表>

功能：计算<表达式>，然后将计算结果赋给内存变量。

说明：表达式可以是常量、变量、函数和运算符的组合，表达式具有计算、判断和数据类型转换等功能。

【例 1.4.1】定义内存变量 A, B, C。

A="人民" && 字符串“人民”赋值给变量 A，结果 A 值为“人民”，并成为字符型变量

STORE 2*3 TO B, C && 计算 2*3 得 6，结果两个变量值都是 6，且成为数值型变量

说明：

1) 命令后面的符号“&&”，表示该符号后跟随的内容是本命令行的注释，它只对命令起注释作用，与命令执行无关，常用于解释命令功能。

2) 内存变量在赋值时定义了它的值及类型，其类型与所赋的值的类型相同。

3) STORE 命令的<内存变量表>可包括多个内存变量，但需用逗号分隔。该命令可将同一值赋给多个内存变量，而“=”命令仅为一个内存变量赋值。

(2) 字段变量

字段变量指数据表中已定义的任意一个数据项，每个数据表中都包含若干字段变量。

记录是数据表中一组字段变量值的集合。

通常，内存变量简称为变量，而字段变量简称为字段。

3. 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。

数组定义的命令格式：

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

功能：定义一维或二维数组，及其下标的上界。

说明：

(1) 系统规定各下标的下界为 1。

例如命令“DIMENSION A(3),B(2,3)”分别定义了数组名为 A 的一维数组与数组名为 B 的二维数组。数组 A 的下标为 3，所以有 A(1)，A(2)，A(3)3 个组元素。对于二维数组，常将第 1 个下标称为行标，第 2 个下标称为列标，故二维数组 B 具有 2 行 3 列共 6 个元素，分别表示为 B(1,1)，B(1,2)，B(1,3)，B(2,1)，B(2,2)，B(2,3)。

(2) 在 Visual FoxPro 6.0 中最多可定义 65 000 个数组，每个数组最多可包含 6 500 个元素。系统将各数组元素的初值设置为 F。

(3) 二维数组各元素在内存中按行的顺序存储，而且也可按一维数组来表示其数组元素。例如上述二维数组元素 B(2,2)也可用 B(5)来表示。

1.4.3 运算符

运算符包括算术运算符、逻辑运算符和关系运算符，分别如表 1-1、表 1-2 和表 1-3 所示。

表 1-1 算术运算符

符号	名称
()	圆括号
^或**	乘方
*	乘号
/	除号
%	取模（即取两数相除的余数）
+	加号
-	减号

表 1-2 逻辑运算符

符号	名称
.NOT.或!	逻辑非
.AND.	逻辑与
.OR.	逻辑或

表 1-3 关系运算符

符号	名称
<	小于
<=	小于等于
>	大于
>=	大于等于
=	相等
==	完全相等
<>或!=或#	不相等
\$	包含（左串是右串子串时才是真）

1.4.4 表达式

符合 Visual FoxPro 规定, 用运算符将常量、变量、字段、函数连接起来的式子称为 Visual FoxPro 表达式。作为特例, 单个的常量、变量、字段和函数均为最简单的表达式。每个表达式都有确定的值, 按照值的数据类型, 可把表达式分为: 数值表达式、字符表达式、日期表达式、关系表达式和逻辑表达式。

14.5 命令格式

Visual FoxPro 的命令格式为:

<命令动词>[<表达式表>][<范围>][FOR<条件>][WHILE<条件>][TO FILE<文件名>/TO PRINTER/TO ARRAY<数组表>/TO <内存变量>][ALL[LIKE/EXCEPT<通配符>]][IN <别名>]

说明:

(1) 命令动词是 FoxPro 的命令名, 用来批示计算机要完成的操作。

(2) 表达式表是一个或多个由逗号分隔开的表达式, 用来批示计算机执行该命令所操作的结果参数。

(3) 范围表明命令可以操作的记录。范围可有以下四种:

RECORD<n>——仅对指定的 n 号记录进行操作

NEXT<n>——对从当前记录开始的 n 条记录进行操作

ALL——对当前数据表中所有记录进行操作

REST——对从当前记录开始到数据表结尾的记录进行操作

(4) FOR<条件>让命令仅对满足条件的记录进行操作。

(5) WHILE<条件>是从当前记录开始, 按记录顺序依次从上向下处理, 直到不满足条件为止。

(6) TO...可以控制命令操作结果的输出。

(7) ALL[LIKE/EXCEPT<通配符>]可以操作包括或不包括与通配符相匹配的文件、字段或内存变量。通配符有“?”和“*”。 “?”代表任何一个字符; “*”代表一组任意字符。

(8) IN...允许在当前工作区下操作其他工作区中的数据表文件。

可选用选项参数用 “[]” 分隔, 如果是必选项则用 “< >”, 用 “/” 隔开的命令选项, 表示可在若干项参数中选其中之一。

关于命令的应用及命令中子句的作用, 将在以后的章节中陆续介绍。

1.4.6 一些常用命令

1. 数据显示命令

命令格式:

?<表达式>

??<表达式>

功能：上述两个命令可以显示表达式和常数的值。

说明：它们具有计算和输出显示的双重功能，在遇到表达式时，它们先计算后显示输出计算结果。单问号（?）命令从下一行开始显示输出；双问号（??）命令从当前光标位置开始显示输出。

【例 1.4.2】?和??命令的应用。

```
STORE 10 TO X
?X
10                && 在主窗口显示 10
Y=6
?(X+Y)/2
8
A="CD"
B="EF"
?A+B
CDEF
??"30*20=",30*20
30*20=600
```

2. 内存变量的显示命令

命令格式：

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

功能：显示当前已定义的内存变量名、作用范围、类型和值。

说明：

(1) LIKE 子句表示将选出与通配符相匹配的内存变量，若要显示例 1.4.2 中建立的内存变量，执行命令“LIST MEMORY LIKE ?”后，主窗口中显示下列内容：

X	Pub	N	10	(10.00000000)
Y	Pub	N	6	(6.00000000)
A	Pub	C	C	"CD"
B	Pub	C	C	"EF"

缺省该选项则选出全部内存变量（包括系统内存变量），并同时显示当前内存变量的总数、字节数等。

(2) 选项 TO PRINTER 能将屏幕显示内容输出到打印机，使用[PROMPT]则能提供是否打印的提示窗口。选项 TO FILE <文件名>能将显示内容存入文件。

3. 内存变量的存储

命令格式：

SAVE TO <内存变量文件名>[ALL LIKE/EXCEPT<内存变量名结构>]

功能：该命令把内存变量以文件的形式存放到磁盘上，以备需要时调用。

说明:

- (1) 在内存变量名结构中可以用通配符来表达具有公共字符的一类内存变量。
- (2) 当选用 ALL LIKE 时, 则指定存储同类结构的所有内存变量。
- (3) 当选用 ALL EXCEPT 时, 则存储指定同类结构以外的所有内存变量。
- (4) 缺省选项时, 存储当前全部的内存变量。

4. 内存变量的恢复

命令格式:

RESTORE FROM <内存变量文件名>[ADDITIVE]

功能: 该命令把指定的内存变量文件中的内存变量从磁盘读入内存中。

说明: 当选择可选项[ADDITIVE]时, 则在保留内存中原有内存变量的情况下, 从磁盘上将指定的内存变量读入内存, 若读入的内存变量与内存中原有的变量同名时, 则以新值取代原有内存变量的值; 若缺省该选项时, 则内存变量读入内存时, 先清除原内存中的内存变量。

5. 内存变量的释放

命令格式:

RELEASE <内存变量名表>

或

RELEASE ALL [LIKE/EXCEPT<内存变量名结构>]

功能: 该命令删除当前内存中所有或部分内存变量。

还有一些常用命令将在涉及到有关内容时再分别介绍。

1.5 常用函数

1. 数值型函数

(1) 求绝对值函数

格式: ABS(<数值表达式>)

功能: 用于求数值表达式的绝对值。

例如: ?ABS(-17.88)

结果: 17.88

(2) 平方根函数

格式: SQRT(<数值表达式>)

功能: 求非负数值表达式的算术平方根。

例如: ?SQRT(2)

结果: 1.41

(3) 指数函数

格式: EXP(<数值表达式>)

功能: 求 e 的<数值表达式>次方的值。

例如: ?EXP(2)

结果: 7.39

(4) 对数函数

格式: LOG(<数值表达式>)

功能: 求<数值表达式>的自然对数值。

例如: ?LOG(1.9876)

结果: 0.6869

(5) 取整函数

格式: INT(<数值表达式>)

功能: 求<数值表达式>值的整数部分。

例如: ?INT(-1.27)

结果: -1

(6) 取最大值函数

格式: MAX(<数值表达式 1>,<数值表达式 2>)

功能: 求两个数值表达式中较大的值。

例如: ?MAX(6,9)

结果: 9

(7) 取模函数

格式: MOD(<数值表达式 1>,<数值表达式 2>)

功能: 求<数值表达式 1>除以<数值表达式 2>所得的余数。

例如: ?MOD(5,3)

结果: 2

(8) 舍入函数

格式: ROUND(<数值表达式 1>,<数值表达式 2>)

功能: 对<数值表达式 1>的值进行四舍五入, 保留的小数由<数值表达式 2>确定。

例如: ?ROUND(5.1329,2)

结果: 5.13

又如: ?ROUND(123.45,-2)

结果: 100

(9) 随机函数

格式: RAND([数值表达式])

功能: 求随机数。

例如: ?RAND()

结果: 0.16

2. 字符处理函数

(1) 子串查找函数

格式: AT(<字符表达式 1>,<字符表达式 2>[,<数值表达式>])

功能: 查找字符串<字符表达式 1>在<字符表达式 2>中第<数值表达式>次出现的起始位置。

例如: ?AT("B","ABCBBD",2)

结果: 4

(2) 左截子串函数

格式: LEFT(<字符表达式>,<数值表达式>)

功能: 截取<字符表达式>左起<数值表达式>个字符的子串。

例如: ?LEFT("ABCD",3)

结果: "ABC"

(3) 右截子串函数

格式: RIGHT(<字符表达式>,<数值表达式>)

功能: 截取<字符表达式>右起<数值表达式>个字符的子串。

例如: ?RIGHT("ABCDE",3)

结果: "CDE"

(4) 截子串函数

格式: SUBSTR(<字符表达式>,<数值表达式 1>[,<数值表达式 2>])

功能: 截取<字符表达式>中<数值表达式 1>位起的长度为<数值表达式 2>的子串。

例如: ?SUBSTR("ABCD",3,2)

结果: "CD"

(5) 字符串长度函数

格式: LEN(<字符表达式>)

功能: 求字符串的长度。

例如: ?LEN("ABC")

结果: 3

(6) 生成空格函数

格式: SPACE(<数值表达式>)

功能: 生成<数值表达式>个空格。

例如: ?SPACE(3)

结果: " "

(7) 字符转数值函数

格式: VAL(<字符表达式>)

功能: 将字符串转换成数值。

例如: ?VAL("678")

结果: 678

(8) 数值转字符函数

格式: STR(<数值表达式 1>[,<数值表达式 2>[,<数值表达式 3>]])

功能: 将数值<数值表达式 1>转换为长度为<数值表达式 2>位, 具有<数值表达式 3>位小数的字符串。

例如: ?STR(1234.56,6,1)

结果: 1234.6

(9) 字符转 ASCII 码函数

格式: ASC(<字符表达式>)

功能: 求字符的 ASCII 码值。

例如: ?ASC("a")

结果: 97

(10) ASCII 码转字符函数

格式: CHR(<数值表达式>)

功能: 把<数值表达式>表示的 ASCII 码转换成相应的字符。

例如: ?CHR(97)

结果: a

3. 测试函数

(1) 数据表文件头函数

格式: BOF([<工作区>])

功能: 记录指针指向首记录之前时, 函数值为.T., 否则为.F.。

(2) 数据表文件尾函数

格式: EOF([工作区])

功能: 记录指针指向末记录之后时, 函数值为.T., 否则为.F.。

(3) 记录计数函数

格式: RECCOUNT([<工作区>])

功能: 测定工作区中打开的数据表文件中记录的个数。

(4) 记录号函数

格式: RECNO([<工作区>])

功能: 测定工作区中打开的数据表文件中指针所指向记录的记录号。

(5) 类型测试函数

格式: TYPE(<表达式>)

功能: 测定<表达式>值的类型, 其结果可以是 C (字符型)、N (数值型)、L (逻辑型)、D (日期型)、M (备注型) 等中的任意一种。

例如: ?TYPE("XY")

结果: C