



普通高等教育“十二五”规划教材

新编 Visual FoxPro

数据库程序设计实验指导

[经管类]

金一宁 韩雪娜 杨 俊 主编



科学出版社

普通高等教育“十二五”

新编 Visual FoxPro 数据库程序设计

实验指导（经管类）

金一宁 韩雪娜 杨 俊 主 编
杨 玉 王宏伟 关绍云 副主编

科学出版社

北 京

内 容 简 介

本书是《新编 Visual FoxPro 数据库程序设计实用教程（经管类）》一书的配套教材，主要分为3章：第1章是《新编 Visual FoxPro 数据库程序设计实用教程（经管类）》一书的各个章节概要；第2章包含18个上机实验，上机实验与授课计划同步，讲述了实验目的、实验范例和实验内容，使学生明确上机实验任务，在实验过程中加深对课堂所学知识的理解，并提高实际动手能力；第3章是习题与参考答案，所涉及的题型丰富，便于学生巩固所学知识。

本书既可作为高等院校经管类专业 Visual FoxPro 数据库程序设计的参考教材，也可作为全国计算机等级考试二级 Visual FoxPro 的参考用书。

图书在版编目(CIP)数据

新编 Visual FoxPro 数据库程序设计实验指导（经管类）/金一宁，韩雪娜，杨俊主编. —北京：科学出版社，2015
（普通高等教育“十二五”规划教材）
ISBN 978-7-03-046299-2

I. ①新… II. ①金… ②韩… ③杨… III. ①关系数据库系统—程序设计—高等学校—教学参考资料 IV. ①TP311.138

中国版本图书馆 CIP 数据核字（2015）第 267652 号

责任编辑：宋 丽 袁星星 / 责任校对：刘玉靖
责任印制：吕春珉 / 封面设计：东方人华平面设计部

科学出版社 出版

北京东黄城根北街 16 号

邮政编码：100717

<http://www.sciencep.com>

百善印刷厂 印刷

科学出版社发行 各地新华书店经销

*

2015年12月第 一 版 开本：787×1092 1/16

2015年12月第一次印刷 印张：13 1/2

字数：317 000

定价：26.00 元

（如有印装质量问题，我社负责调换〈百善〉）

销售部电话 010-62134988 编辑部电话 010-62135319-2047

版权所有，侵权必究

举报电话：010-64030229；010-64034315；13501151303

前 言

Visual FoxPro 是小型关系数据库系统的杰出代表, Visual FoxPro 程序设计是高等院校开设范围最广泛的一门程序设计语言课程,同时也是教育部考试中心指定的全国计算机等级考试二级考试的科目之一。

本书的特色是针对性强,突出应用,重视培养学生的实际动手能力。在章节概要中结合教师实际授课经验给出了主教材各章的要点,便于学生理解与掌握。上机实验部分按照授课计划同步给出了 18 个实验,每个实验明确了实验目的和实验内容,并提供一个有代表性的实验范例供学生参照,使学生及时消化所学内容,其中第 18 个实验介绍了一个实际的小型数据库应用系统的开发,可以使学生理论联系实际,融会贯通所学课程知识,积累程序调试经验,提高分析问题、解决问题的能力。习题与参考答案提供了大量的习题,可以使学生加强对课程的理解,从容应对期末考试和全国计算机等级考试二级 Visual FoxPro 考试。

本书由哈尔滨商业大学金一宁、韩雪娜、杨俊担任主编,由杨玉、王宏伟、关绍云担任副主编,由张洪瀚担任主审。

在编写本书的过程中,编者得到了哈尔滨商业大学各级领导的大力支持和帮助,同时得到了哈尔滨商业大学计算机与信息工程学院教师们的支持和关心,在此表示衷心感谢。

由于编者水平有限,书中不足之处在所难免,恳请读者提出宝贵意见和建议。

编 者

2015 年 10 月

目 录

第 1 章 主教材各章节概要	1
1.1 Visual FoxPro 基础概要	1
1.1.1 关系数据库基础知识	1
1.1.2 Visual FoxPro 概述	2
1.2 Visual FoxPro 的数据及其运算概要	6
1.2.1 常量和变量	6
1.2.2 表达式	8
1.2.3 常用函数	10
1.3 Visual FoxPro 数据库基本操作概要	16
1.3.1 数据库的建立及其管理	16
1.3.2 数据库表的建立及操作	17
1.3.3 索引的建立及使用	18
1.3.4 数据完整性	20
1.3.5 多工作区的概念及多表的使用	20
1.4 面向对象程序设计基础概要	21
1.4.1 面向对象的基本概念	21
1.4.2 创建与管理表单	21
1.4.3 常用表单控件	23
1.4.4 自定义类	26
1.4.5 综合总结	27
1.5 程序设计基础与应用概要	27
1.5.1 程序与程序文件	27
1.5.2 分支结构	28
1.5.3 循环结构	30
1.5.4 自定义函数及内存变量的作用域	31
1.6 结构化查询语言概要	32
1.6.1 SQL 概述	32
1.6.2 SQL 的数据查询功能	32
1.6.3 SQL 的数据定义功能	37
1.6.4 SQL 的数据操纵功能	40
1.7 查询与视图概要	41
1.7.1 查询	41
1.7.2 视图	43



1.8 菜单设计与应用概要	46
1.8.1 设计菜单	46
1.8.2 下拉式菜单的操作	47
1.8.3 为顶层表单添加菜单	48
1.8.4 快捷菜单设计	49
1.9 报表设计与应用概要	49
1.9.1 创建报表	49
1.9.2 报表的编辑与输出	49
第 2 章 上机实验	50
2.1 建立项目、数据库和表	50
2.2 常变量、运算符和表达式	55
2.3 常用函数	58
2.4 库、表的编辑	63
2.5 索引、表之间联系和多表操作	65
2.6 表单及其控件应用（一）	68
2.7 表单及其控件应用（二）	70
2.8 分支结构	71
2.9 循环结构	73
2.10 循环嵌套	74
2.11 数组	75
2.12 自定义函数及循环习题巩固	76
2.13 SQL 语言（一）	78
2.14 SQL 语言（二）	81
2.15 SQL 查询与表单控件结合综合应用	82
2.16 查询设计器与视图设计器	85
2.17 菜单设计与报表设计	89
2.18 小型数据库应用系统的设计与实现	91
第 3 章 习题与参考答案	102
3.1 单选题	102
3.2 程序填空题	125
3.3 程序改错题	145
3.4 程序设计题	161
3.5 窗体设计题	167
参考答案	171
附录 A Visual FoxPro 6.0 常用命令一览表	182



附录 B Visual FoxPro 6.0 常用属性一览表	187
附录 C Visual FoxPro 6.0 常用函数一览表	196
附录 D Visual FoxPro 6.0 常用文件类型一览表	206
参考文献	208

第1章 主教材各章节概要

1.1 Visual FoxPro 基础概要

1.1.1 关系数据库基础知识

1. 关系模型

一个关系就是一张二维表,用二维表的形式来表示实体和实体之间联系的数据模型称为关系数据模型。常用的关系术语如下。

(1) 关系

一个关系就是一张二维表,每个关系有一个关系名。在 Visual FoxPro 中,一个关系存储为一个文件,文件扩展名为.dbf,称为“表”。

(2) 元组

一个关系中水平方向的行称为元组,每一行是一个元组,对应的 Visual FoxPro 二维表中行称为记录。

(3) 属性

一个关系中垂直方向的列称为属性,每一列有一个属性名,对应的 Visual FoxPro 二维表中列称为字段,每个字段有一个字段名。每列中具体的数据是属性值,在 Visual FoxPro 中称为字段值。

(4) 域

属性的取值范围。例如,性别属性(字段)取值只能是“男”或“女”。

(5) 关键字

属性或属性的组合,关键字的值能够唯一地标识一个元组,即关键字值必须是唯一的。在 Visual FoxPro 中关键字表示为字段或者字段的组合。

(6) 外部关键字

如果表中的一个字段不是本表的关键字,而是另一个表的关键字,这个字段(属性)就称为外部关键字。

2. 关系的特点

在关系模型中对关系有一定的要求,关系必须具有以下特点。

1) 关系必须规范化,关系中的每个属性必须是不可分割的数据单元,即表中不能再包含表。

2) 在同一个关系中不能出现相同的属性名,Visual FoxPro 中不允许同一个表中有相同的字段名。

3) 关系中不允许有完全相同的元组,即不允许有冗余。

4) 在一个关系中元组的次序无关紧要,任意两行交换位置也不影响数据的实际含义。

5) 在一个关系中属性的次序无关紧要, 任意两列交换位置也不影响数据的实际含义。

3. 关系运算

专门的关系运算包括选择、投影和连接。

(1) 选择

从关系中找出满足给定条件的元组的操作称为选择。例如, 在学生表中查找男同学的记录, 就属于选择运算。选择是从行的角度进行的运算, 即从水平方向抽取记录。经过选择运算得到的结果可以形成新关系, 其关系模式不变, 但其中的元组是原关系的一个子集。

(2) 投影

从关系模式中指定若干个属性组成新的关系称为投影。投影是从列的角度进行的运算, 相当于对关系进行垂直分解。经过投影运算可以得到一个新关系, 其关系模式所包含的属性个数往往比原关系少, 或者属性的排列顺序不同。

(3) 连接

连接是关系的横向结合。连接运算将两个关系模式拼接成一个更宽的关系模式, 生成的新关系中包含满足连接条件的元组。在连接运算中, 按照字段值对应相等为条件进行的连接操作称为等值连接, 去掉重复属性的等值连接称为自然连接, 自然连接是最常用的连接运算。

1.1.2 Visual FoxPro 概述

1. Visual FoxPro 6.0 的安装与启动

(1) 安装 Visual FoxPro 6.0

使用 Visual FoxPro 6.0 系统光盘, 找到 setup.exe 文件, 双击该文件运行安装向导, 按照安装向导提示逐步操作即可。默认的安装文件夹是 C:\...\Microsoft Visual Studio\VFP98, 其中的狐狸头像图标文件 VFP6.EXE 是启动 Visual FoxPro 6.0 系统的可执行文件。

(2) 启动 Visual FoxPro 6.0

在 Windows 中启动 Visual FoxPro 6.0, 可以单击“开始”按钮, 依次选择“所有程序”|“Microsoft Visual FoxPro 6.0”|“Microsoft Visual FoxPro 6.0”命令, 也可以在桌面上创建快捷方式图标, 双击该图标也可以方便地启动 Visual FoxPro 6.0。

(3) 退出 Visual FoxPro 6.0

退出 Visual FoxPro 6.0 有多种方法。常用的包括:

- 1) 单击 Visual FoxPro 6.0 标题栏最右面的“关闭”按钮。
- 2) 选择“文件”|“退出”命令。
- 3) 按 Alt+F4 组合键。
- 4) 在 Visual FoxPro 6.0 “命令”窗口中输入 QUIT, 然后按 Enter 键。

2. Visual FoxPro 6.0 的主界面

(1) 标题栏

标题栏位于屏幕最上端, 主要包括标题、“最小化”按钮、“最大化(还原)”按钮和“关闭”按钮。单击“关闭”按钮可以退出 Visual FoxPro 6.0。

(2) 菜单栏

菜单栏包括 8 个下拉式菜单项：文件、编辑、显示、格式、工具、程序、窗口、帮助。选择任何一个菜单命令或者按 Alt+相应菜单热键都会弹出下拉菜单，其中包含 Visual FoxPro 的操作命令。使用菜单栏可以方便地进行各种操作。在 Visual FoxPro 6.0 菜单栏里的各个选项并不是一成不变的，随着当前运行程序的不同，所显示的主菜单和下拉菜单的选项也不尽相同。

(3) 工具栏

使用工具栏中的按钮比通过菜单栏方便快捷。工具栏中的按钮都有文本提示功能，当鼠标指针停留在某个按钮图标上时，系统将提示其功能。选择“显示”|“工具栏”命令，打开“工具栏”对话框，选择想要显示的工具栏。

(4) 工作显示区

工具栏下面的空白区域是工作显示区，用于显示命令或者程序执行的结果。

(5) “命令”窗口

在“命令”窗口中可以输入 Visual FoxPro 的命令语句，按 Enter 键执行。“命令”窗口自动保留已经执行过的命令，如果要执行前面输入过的命令，只需将光标定位到这条命令的任意位置，按 Enter 键即可，还可以对命令进行修改、删除、剪切、复制、粘贴等操作。单击“命令”窗口右上角的“关闭”按钮可以关闭它，选择“窗口”|“命令窗口”命令可以重新打开。

(6) 状态栏

状态栏显示和当前操作相关的状态信息。例如，当前打开的数据表的名字、表中的记录数、当前记录号等信息。

3. Visual FoxPro 6.0 命令概述

在“命令”窗口中可以使用 Visual FoxPro 提供的命令进行相应操作，也可以用命令编写程序来解决实际问题。

(1) Visual FoxPro 命令的一般格式

<命令动词> [范围子句] [FIELDS 子句] [FOR 子句] [WHILE 子句]

第一部分是命令动词，表示命令的功能；第二部分是各种子句，是命令在执行过程中的各种限制或说明。格式中的<>表示必选项，[]表示可选项，|是或者的意思，输入命令的时候不要输入这些符号。Visual FoxPro 命令以命令动词开头，其余子句排列顺序任意，命令动词与子句，子句与子句之间用空格分隔。例如，LIST FOR 性别="男"，可以显示已打开的数据表中所有男生记录。

(2) Visual FoxPro 命令中的子句

1) 范围子句。范围子句表示本命令对数据表进行操作的记录范围，有四种选择：

- ① ALL：表示对表的全部记录进行操作。
- ② NEXT N：表示对包括当前记录在内的后续 N 条记录进行操作。
- ③ RECORD N：表示只对第 N 条记录进行操作。
- ④ REST：表示对包括当前记录在内的后续所有记录进行操作。

范围子句是可选项，在某些命令中省略此项时按 ALL 处理，也有些命令省略此项时只

对当前记录进行操作。

2) **FIELDS** 子句。**FIELDS** 子句是指定要处理的字段, **FIELDS** 后面给出字段名表列, 对于指定字段进行操作, 多于一个字段用英文半角状态的逗号分隔。 **FIELDS** 命令本身也可以省略。例如, **LIST FIELDS** 姓名, 出生日期也可以写成 **LIST** 姓名, 出生日期, 都表示显示表中姓名和出生日期两列数据。

3) **FOR** 子句和 **WHILE** 子句。**FOR** 和 **WHILE** 后面的子句都是条件表达式, 用于对表中满足条件的记录进行操作。

① **FOR** 子句的功能是在指定的记录范围内, 从第一条记录开始检查, 遇到满足条件的记录就执行命令, 若不满足条件就直接跳过此记录, 检查下一条, 直到指定的记录范围最后一条记录为止。

② **WHILE** 子句的功能是在指定的记录范围内, 从第一条记录开始检查, 遇到满足条件的记录就执行命令, 一旦遇到不满足条件的记录则停止执行, 即使此记录后面还有满足条件的记录也不再执行此命令了。

(3) Visual FoxPro 命令的书写规则

1) 命令行中使用的英文字母不区分大小写, 可以混合使用。

2) 命令动词和子句中的英文单词可以用前四个字母表示。

3) 一个命令行只能书写一条命令, 每条命令输入完毕后按 **Enter** 键执行。

4) 一条命令最大长度是 254 个字符。如果一行写不下, 可以使用英文半角状态的分号, 然后按 **Enter** 键换行后, 在下一行继续书写。

(4) Visual FoxPro 的几个基本命令

1) 输出命令。

格式:

?|??[表达式表列]

功能: 在主窗口显示表达式的运算结果。其中, ?命令是主窗口光标先换行, 然后显示表达式运算结果; ??命令是直接在主窗口的光标当前位置进行输出。

2) 清屏命令。

格式:

CLEAR

功能: 用于清除主窗口中的所有内容。

3) 赋值语句。

格式 1:

<变量名>=<表达式>

格式 2:

STORE <表达式> TO <变量名表列>

功能: 两种赋值语句都是给指定变量赋值, 格式 1 常用于给一个变量赋值, 如 **X=7**; 格式 2 一般用于同时给多个变量赋初值, 如 **STORE 0 TO X, Y, Z**。

4. 向导、设计器、生成器

(1) 向导

向导是一种交互式程序，它通过一组对话框依次与用户对话，引导用户分步完成 Visual FoxPro 的某项任务。用户需要在一系列向导对话框中回答问题或者选择选项，向导则会根据回答生成文件或者执行任务，帮助用户快速完成一般性的任务，如创建表单、创建报表、建立查询等。

(2) 设计器

Visual FoxPro 的设计器是用来创建或修改应用系统中各种组件的可视化工具。各种设计器使得初学者可以轻松创建表、表单、查询和报表等文件。在新建各种文件的时候会打开相应的设计器。

(3) 生成器

生成器是带有选项卡的对话框，用于简化对表单、复杂控件等的创建和修改，对设计器起到辅助作用。一般右击操作对象然后在弹出的快捷菜单中选择生成器可以打开相应的生成器对话框。

5. 项目管理器

项目管理器是 Visual FoxPro 处理数据和对象的主要组织工具，为系统开发者提供了简便的、可视化的方法来组织和处理表、数据库、表单、报表、查询及和其他一切项目相关的文件，最后可以在项目管理器中把应用系统编译成一个扩展名为.app 的应用文件或者.exe 的可执行文件。

项目管理器窗口包括 6 个选项卡，其中“数据”“文档”“类”“代码”“其他”5 个选项卡用于分类显示各种文件，“全部”选项卡用于集中显示该项目中的所有文件。

(1) “数据”选项卡

“数据”选项卡管理项目中的所有数据：数据库、数据库表、自由表、视图和查询。

(2) “文档”选项卡

“文档”选项卡管理表单、报表和标签。

(3) “类”选项卡

“类”选项卡用于管理和编辑用户自定义类。

(4) “代码”选项卡

“代码”选项卡管理三大类程序：扩展名为.prg 的程序文件、函数库 API Libraries 和应用程序.app 文件。

(5) “其他”选项卡

“其他”选项卡管理文本文件、菜单文件和其他文件（如.bmp 位图、.ico 图标文件等）。

(6) “全部”选项卡

“全部”选项卡用于集中显示以上各类文件。

项目文件的扩展名是.pjx。

1.2 Visual FoxPro 的数据及其运算概要

数据类型决定了数据的存储方式和运算方式。在 Visual FoxPro 中,常用的数据类型主要包括数值型、字符型、货币型、逻辑型、日期型、日期时间型、整型、备注型和通用型。Visual FoxPro 有常量、变量、表达式和函数四种形式的数据,常量和变量是数据运算和处理的基本对象,而表达式和函数则体现了 Visual FoxPro 对数据进行运算和处理的能力和功

1.2.1 常量和变量

1. 常量

常量用以表示一个具体的、不变的值。

(1) 数值型常量

数值型常量就是平时所讲的常数,由数字、小数点和正负号组成。

(2) 字符型常量

字符型常量也称为字符串。在 Visual FoxPro 中有三种定界符:单引号、双引号和方括号。

(3) 货币型常量

货币型常量用来表示货币值,书写格式与数值型常量类似,但要加上一个前置的\$。货币型数据在存储和计算时,采用 4 位小数。

(4) 日期型常量

日期型常量的定界符是一对花括号。花括号内包括年、月、日三部分内容,各部分内容之间用分隔符分隔。分隔符可以是/、-、.等,其中/是系统在显示日期型数据时使用的默认分隔符。Visual FoxPro 的默认日期格式是{mm/dd/[yy]yy},Visual FoxPro 增加了严格的日期格式,其格式是{^yyyy-mm-dd}。

影响日期格式的设置命令主要有以下几种形式:

设置日期格式中的世纪值命令格式:

```
SET CENTURY ON/OFF|TO[nCentury]
```

设置日期显示格式命令格式:

```
SET DATE [TO] MDY/DMY/|YMD/...
```

设置日期分隔符命令格式:

```
SET MARK TO[日期分隔符]
```

设置是否检查日期格式命令格式:

```
SET STRICT DATE TO[0/1/2]
```

(5) 日期时间型常量

日期时间型常量的定界符也是一对花括号,其中既含日期又含时间。日期时间型常量的

默认格式是: {mm/dd/[yy]yy[.][hh[:mm[:ss]][a|p]]}。

(6) 逻辑型常量

逻辑型常量表示逻辑判断的结果, 只有“真”和“假”两种值。在 Visual FoxPro 中, 逻辑真用.T.、.t.、.Y.和.y.表示, 逻辑假用.F.、.f.、.N.和.n.表示。

2. 变量

变量是在操作过程中可以改变其值的数据对象。用户在 Visual FoxPro 中主要使用的变量是字段变量和内存变量两大类。确定一个变量, 需要确定其三个要素: 变量名、数据类型和变量值。

(1) 字段变量

字段变量就是表中的字段名, 是表中最基本的数据单元。字段变量是一种多值变量, 一个表有多少条记录, 那么该表的每一字段就有多少个值, 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。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、长度等是在定义表结构时定义的。

(2) 内存变量

Visual FoxPro 中还有一种变量, 它独立于表, 是内存中的一个存储区域, 称为内存变量, 其变量值就是存储在这个存储区域里的值。内存变量的类型有字符型、数值型、货币型、逻辑型、日期型和日期时间型等。

通常, 可以直接用内存变量名对简单内存变量进行访问, 但若它与字段变量同名时, 则同名的字段变量优先。简单内存变量赋值不必事先定义, 赋值命令有以下两种格式:

<内存变量>=<表达式>

STORE <表达式> TO <内存变量表>

在 Visual FoxPro 中, 数组是内存中连续的一片存储区域, 数组由数组元素组成, 每个数组元素就相当于一个内存变量, 通过数组名和相应的下标来访问。一个数组中各数据元素的数据类型可以不同, 数组元素的初值均是逻辑假.F.。与简单内存变量不同, Visual FoxPro 规定数组在使用之前必须定义, 即定义数据名、维数和大小。其命令格式为:

DIMENSION <数组名>(<下标上界 1>[, <下标上界 2>])

DECLARE <数组名>(<下标上界 1>[, <下标上界 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。

3. 内存变量的常用命令

(1) 内存变量的赋值

格式 1:

<内存变量>=<表达式>

格式 2:

STORE <表达式> TO <内存变量表>

功能：格式 1 只能给一个内存变量赋值，格式 2 可以同时给多个内存变量赋相同的值。

(2) 表达式值的显示

格式 1:

?[<表达式表>]

格式 2:

??[<表达式表>]

功能：计算表达式表中各表达式并输出各表达式值。格式 1 代表换行输出，格式 2 可以在同行光标处输出。

(3) 内存变量的显示

可以用命令显示当前已定义的内存变量的有关信息，包括变量名、作用域、类型和取值。

格式 1:

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

格式 2:

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

其中，LIKE 选项表示显示与通配符相匹配的内存变量，在<通配符>中允许使用符号?和*，分别代表单个字符和多个字符。

(4) 内存变量的清除

格式 1:

CLEAR MEMORY

格式 2:

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

功能：清除内存变量并释放相应的内存空间。其中，格式 1 是清除所有的内存变量，格式 2 是清除指定的内存变量。

1.2.2 表达式

表达式是由常量、变量和函数通过特定的运算符连接起来的式子。在 Visual FoxPro 编写的程序中，表达式被广泛应用。表达式的形式既包括单一的运算对象（如常量、变量或函数），也包括由运算符将运算对象连接起来形成的式子。根据运算对象的数据类型不同，表达式可以分为算术表达式、字符表达式、日期和时间表达式、关系表达式和逻辑表达式等。

1. 算术表达式

用算术运算符将数值型数据连接起来的式子叫算术表达式。算术运算符包括：()（括号）、**或^（乘方）、*（乘）、/（除）、%（求余数）、+（加）、-（减）。各运算符运算的优先顺序和一般算术运算规则完全相同。

2. 字符表达式

字符表达式由字符串运算符将字符型数据连接起来形成。在 Visual FoxPro 中字符运算有两类：连接运算和包含运算。

(1) 连接运算

连接运算符有完全连接运算符“+”和不完全连接运算符“-”两种。“+”运算的功能是将两个字符串连接起来形成一个新的字符串。“-”运算的功能是去掉字符串 1 尾部的空格，然后将两个字符串连接起来，并把字符串 1 末尾的空格放到结果串的末尾。

(2) 包含运算

格式：

$\langle \text{字符串 1} \rangle \$ \langle \text{字符串 2} \rangle$

功能：包含运算的结果是逻辑值。若 $\langle \text{字符串 1} \rangle$ 包含在 $\langle \text{字符串 2} \rangle$ 之中，其表达式值为.T.，否则为.F.

3. 日期和时间表达式

日期和时间表达式是指含有日期型或日期时间型数据的表达式。使用的运算符有“+”和“-”两种。与日期型表达式有关的格式有三种，日期时间型表达式的格式与之类似。

格式 1：

$\langle \text{日期型数据} \rangle + \langle \text{天数} \rangle$

$\langle \text{天数} \rangle + \langle \text{日期型数据} \rangle$

格式 2：

$\langle \text{日期型数据} \rangle - \langle \text{天数} \rangle$

格式 3：

$\langle \text{日期型数据 1} \rangle - \langle \text{日期型数据 2} \rangle$

4. 关系表达式

关系运算符包括： $<$ （小于）、 $\leq$ （小于等于）、 $>$ （大于）、 $\geq$ （大于等于）、 $=$ （等于）、 $=$ （精确等于）、 $<>$ 或 $\#$ 或 $\neq$ （不等于）。关系运算符的作用是比较两个表达式的大小或前后，它们的运算优先级相同。

格式：

$e1 \langle \text{关系运算符} \rangle e2$

其中， $e1$ 、 $e2$ 可以同为数值型表达式、字符型表达式、日期型表达式或逻辑型表达式。但 $=$ 仅适用于字符型数据。关系表达式表示一个条件，条件成立时值为.T.，否则为.F.

5. 逻辑表达式

逻辑表达式是由逻辑运算符将逻辑型数据连接起来的式子，其运算结果仍是逻辑值。逻辑

辑运算符包括: NOT 或.NOT.或! (逻辑非)、AND 或.AND. (逻辑与)、OR 或.OR. (逻辑或)。其运算优先级顺序由高到低依次是 NOT、AND、OR。

1.2.3 常用函数

函数是用程序来实现的一种数据运算或转换。每个函数都有一个运算结果,称为函数值或返回值。Visual FoxPro 提供了许多内部函数,每个函数实现某项功能或完成某种运算。函数调用的一般形式为

函数名([参数表])。

1. 数值函数

数值函数是函数值为数值的一类函数,它们的自变量和返回值通常都是数值型数据。

(1) 求绝对值函数

格式:

ABS(<数值型表达式>)

功能: 求数值型表达式的绝对值。函数值为数值型。

(2) 求平方根函数

格式:

SQRT(<数值型表达式>)

功能: 求数值型表达式的算术平方根,数值型表达式的值应不小于零。函数值为数值型。

(3) 取整函数

格式:

INT(<数值型表达式>)

CEILING(<数值型表达式>)

FLOOR(<数值型表达式>)

功能: INT()取数值型表达式的整数部分。

CEILING()取大于或等于指定表达式的最小整数。

FLOOR()取小于或等于指定表达式的最大整数。

(4) 余数函数

格式:

MOD(<数值型表达式 1>,<数值型表达式 2>)

功能: 求<数值型表达式 1>除以<数值型表达式 2>所得出的余数,所得余数的符号和表达式 2 相同。

(5) 四舍五入函数

格式:

ROUND(<数值型表达式 1>,<数值型表达式 n>)

功能: 对<数值型表达式 1>求值并保留 n 位小数,从 n+1 位小数起进行四舍五入,若 n