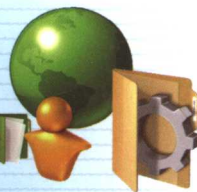


2007年

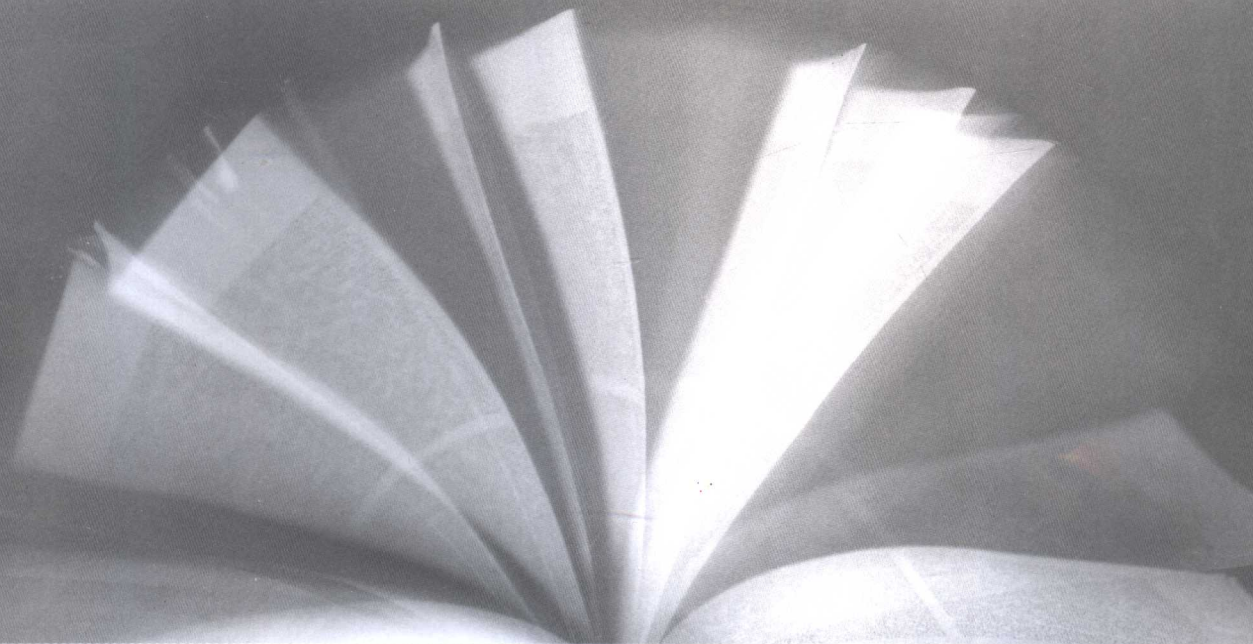


河南省中等职业学校对口升学考试复习指导

计算机类专业（下册）

数据库应用基础——Visual FoxPro 6.0 计算机网络基础

河南省职业技术教育教研室 编



河南科学技术出版社

2007年

河南省中等职业学校对口升学考试复习指导

计算机类专业（下册）

数据库应用基础——Visual FoxPro 6.0
计算机网络基础

河南省职业技术教育教研室 编

河南科学技术出版社

· 郑州 ·

图书在版编目 (CIP) 数据

数据库应用基础: Visual FoxPro 6.0 · 计算机网络基础/河南省职业技术教育教研室编. — 郑州: 河南科学技术出版社, 2007. 2

(2007 年河南省中等职业学校对口升学考试复习指导 · 计算机类专业; 下册)

ISBN 978 - 7 - 5349 - 3631 - 9

I. 数… II. 河… III. ①关系数据库 - 数据库管理系统, Visual FoxPro 6.0 - 专业学校 - 升学参考资料②计算机网络 - 专业学校 - 升学参考资料 IV. ①TP311.138
②TP393

中国版本图书馆 CIP 数据核字 (2007) 第 005261 号

出版发行: 河南科学技术出版社

地址: 郑州市经五路 66 号 邮编: 450002

电话: (0371) 65737028

责任编辑: 申卫娟

责任校对: 柯 姣

封面设计: 张 伟

版式设计: 采 风

印 刷: 河南第一新华印刷厂

经 销: 全国新华书店

幅面尺寸: 185mm × 260mm 印张: 14 字数: 289 千字

版 次: 2007 年 2 月第 1 版 2007 年 2 月第 1 次印刷

定 价: 19.00 元

如发现印、装质量问题, 影响阅读, 请与出版社联系。



为了做好 2007 年河南省中等职业学校毕业生对口升学考试指导工作,帮助学生有针对性地复习备考,我们组织有关专家和教师编写了这套《2007 年河南省中等职业学校对口升学考试复习指导》。这套资料是以《2007 年河南省中等职业学校毕业生对口升学考纲》为依据,以国家规划教材、省规划教材为参考编写的。每本资料包括复习内容和要求、题型示例、参考答案三部分内容,同时还收录了近两年河南省对口升学考试的试卷、参考答案及评分标准。

在编写过程中,我们以《河南省人民政府贯彻国务院关于大力发展职业教育的决定的实施意见》(豫政〔2006〕20 号)为指导,坚持“以服务为宗旨、以就业为导向”的职业教育办学方针,以基础性、科学性、适应性、指导性为原则,紧扣《考纲》,着重反映了各专业(学科)的基础知识和基本技能,注重培养和考查学生分析问题和解决问题的能力。在内容选择和例题设计上,既适应了高考选拔性能力考试的需要,又注意了对中等职业学校教学工作的引导。在复习时,建议以教材为基础,以资料为指导,二者配合使用,效果更好。

本书是这套书中的一种,其中《数据库应用基础——Visual FoxPro 6.0》部分,主编张平、范吉钰,参编刘南、常征、吴允志;《计算机网络基础》部分,主编张兴华、苗春峰,参编刘诗鹏、陈根长、雷泓霖。本书审稿宋安国。

由于经验不足,书中如有瑕疵之处,恳请广大师生及时提出修改意见和建议,使之不断完善和提高。

河南省职业技术教育教学研究室

2007 年 1 月



第一部分 数据库应用基础——Visual FoxPro 6.0

复习指导	1
第一章 数据库的基本知识和 Visual FoxPro 6.0 最基本的操作 / 1	
第二章 Visual FoxPro 6.0 的数据类型和数据的表示方法 / 6	
第三章 项目管理器的功能和使用方法 / 15	
第四章 数据库和表的创建、编辑及基本操作方法 / 19	
第五章 查询和视图的概念及基本操作方法 / 35	
第六章 报表、标签的设计和创建 / 40	
第七章 表单的基本知识和设计表单的基本方法 / 48	
第八章 菜单的设计和创建 / 58	
第九章 Visual FoxPro 6.0 程序设计基础 / 63	
数据库应用基础——Visual FoxPro 6.0 题型示例	71

第二部分 计算机网络基础

复习指导	123
第一章 计算机网络概述 / 123	
第二章 数据通信基础 / 126	
第三章 计算机网络体系结构与协议 / 130	
第四章 局域网组网技术 / 135	
第五章 网络互联 / 140	
第六章 网络操作系统 / 142	
第七章 Internet 与 Intranet / 147	
第八章 网络安全 / 151	
计算机网络基础题型示例	154





第三部分 近年试题

- 2005 年河南省普通高等学校对口招收中等职业学校毕业生考试计算机类
专业课试卷 / 173
- 2006 年河南省普通高等学校对口招收中等职业学校毕业生考试计算机类
专业课试卷 / 180

第四部分 参考答案

- 数据库应用基础——Visual FoxPro 6.0 题型示例参考答案 / 189
- 计算机网络基础题型示例参考答案 / 204
- 2005 年河南省普通高等学校对口招收中等职业学校毕业生考试计算机类
专业课试卷参考答案及评分标准 / 211
- 2006 年河南省普通高等学校对口招收中等职业学校毕业生考试计算机类
专业课试卷参考答案及评分标准 / 215

第一部分 数据库应用基础—— Visual FoxPro 6.0

复习指导

第一章 数据库的基本知识和 Visual FoxPro 6.0 最基本的操作



复习要求

1. 掌握有关数据库的基本知识。
2. 了解 Visual FoxPro 6.0 的基本功能。
3. 掌握 Visual FoxPro 6.0 的启动和退出方法。
3. 掌握 Visual FoxPro 6.0 主界面的组成和各部分功能。
4. 掌握 Visual FoxPro 6.0 的三种工作方式。
5. 掌握 Visual FoxPro 6.0 命令窗口的使用方法。



复习内容

一、数据库的基本概念

1. 信息

信息是客观世界在人们头脑中的反映，是客观事物的表征，是可以传播和加以利用的一种知识。

2. 数据

数据是信息的载体，是指存储在某一种介质上的可以被识别的物理符号，是对客观存在实体的一种记载和描述。在计算机领域，数据不但包括数字、文字，还包括图形、图像、声音和视频等各种可以被计算机存储和处理的数据。

3. 数据库



数据库是存储在计算机存储介质上的、有组织的、可共享的数据集合。数据库中的数据按一定的数据模型组织、描述和存储，具有较小的冗余度和较高的数据独立性，并可为各种用户共享。

4. 数据库系统

数据库系统是引入了数据库技术的计算机系统。一个数据库系统通常由计算机系统、数据库、数据库管理系统、相关软件、数据库管理员和用户五部分组成。

数据库系统的主要特点为：

- (1) 实现数据共享。
- (2) 实现数据结构化。
- (3) 具有较高的数据独立性。
- (4) 具有统一的数据控制功能。

5. 数据库管理系统

数据库管理系统是对数据库进行统一管理，帮助用户创建、维护和使用数据库的软件系统。数据库管理系统是整个数据库系统的核心，是用户和数据库之间的交互界面，也是用户程序与数据库中数据的接口。

数据库管理系统具有以下功能：

- (1) 数据定义功能。
- (2) 数据操作功能。
- (3) 控制和管理功能。
- (4) 数据字典。

6. 数据模型

目前比较流行的数据模型主要有三种：

(1) 层次模型：层次模型中，各数据对象之间是一种一对一或一对多的联系。层次型数据库如树状结构，具有父子关系，每个父节点可以有很多子节点，但每个子节点仅可有一个父节点。层次型数据库结构简单，易于操作，但搜寻效率很差。

(2) 网状模型：网状模型就像一个网络，可用来表示数据间复杂的逻辑关系。网状型数据库允许子节点有多个父节点，由于子节点之间的关系较接近，容易联系。但由于路径多，不易维护修改。

(3) 关系模型：关系模型是用若干行与若干列构成的表格来描述数据集合以及它们之间的联系，每一个表格被称为一个关系。使用关系模型构造的数据库称为关系型数据库。

对于一个符合关系模型的二维表格，通常将其中的每一列称为一个字段，每一行称为一个记录。构成关系的二维表格，必须满足以下条件：

- ① 表中每一列数据的类型必须相同。
- ② 表中没有内容完全相同的数据行。
- ③ 表中不允许有重复的字段名且每一个字段不可再分解。





④ 表中行或列的顺序可以任意排列，不影响表中各数据项间的关系。

目前比较流行的是关系型数据库，其主要优点为：

- ① 表格式的关系易于理解和建立数据库。
- ② 层次型和网状型结构可容易的转换为二维表格式结构。
- ③ 通过连接与映射表格中的字段，新关系也易于建立。
- ④ 搜寻速度快。
- ⑤ 便于修改和维护。

7. 数据库表和自由表

Visual FoxPro 6.0 的表从处理方法上分为两类：数据库表和自由表。

数据库表属于数据库管理下的表，数据库表的打开、关闭、删除或引用均受到数据库的限制；自由表不受数据库管理和制约，可以自由打开、关闭、删除或引用。数据库表和自由表均是一个二维表，它们的特性和性质完全相同，具有相同的 .DBF 扩展名。数据库表和自由表可以相互转换，数据库表可以退出数据库的管理而成为一个自由表，而自由表也可以加入到数据库中，接受数据库管理，成为一个数据库表。

8. 字段和记录

(1) 字段：在 Visual FoxPro 6.0 的表中，列称为字段。字段名可以是中文或英文，也可以是中文、英文与数字或符号的组合。在数据表中同一个字段具有相同的数据类型。

(2) 记录：在 Visual FoxPro 6.0 的表中，行称为记录。每个记录中不同字段的数据可能具有不同的数据类型，但所有记录的相同字段的数据类型一定是相同的。

(3) 字段与记录的关系：字段和记录是构成数据表的两个要素。字段反映了数据表数据的类型、性质；记录反映了数据表数据的多少，字段名的变化或记录号的变化将直接影响数据表相应数据的变化。

二、Visual FoxPro 6.0 的启动和退出

1. Visual FoxPro 6.0 的启动

启动 Visual FoxPro 6.0 通常有以下三种方法：

(1) 通过“开始”菜单启动：单击“开始”→“程序”→“Microsoft Visual FoxPro 6.0”，即可启动 Visual FoxPro 6.0。

(2) 通过运行 Visual FoxPro 6.0 开发的文件启动：首先在磁盘上找到一个用 Visual FoxPro 6.0 开发的文件，双击该文件图标，即可打开该文件及相关联的 Visual FoxPro 6.0。

(3) 通过建立快捷方式启动：先用 Visual FoxPro 6.0 的启动文件“VFP6.EXE”建立一个快捷方式，放于桌面上，双击该快捷方式启动 Visual FoxPro 6.0。

2. Visual FoxPro 6.0 的退出

退出 Visual FoxPro 6.0 有以下五种方法：





- (1) 单击 Visual FoxPro 6.0 主窗口右上角的“关闭”按钮。
- (2) 选择“文件”菜单中的“退出”命令。
- (3) 双击 Visual FoxPro 6.0 主窗口左上角的系统控制菜单图标。
- (4) 在命令窗口中执行“QUIT”命令。
- (5) 单击 Alt + F4 键。

三、Visual FoxPro 6.0 的主界面

Visual FoxPro 6.0 的主界面由标题栏、菜单栏、工具栏、状态栏和命令窗口组成。

1. 标题栏

标题栏显示目前所使用的系统是 Microsoft Visual FoxPro 6.0。

2. 菜单栏

Visual FoxPro 6.0 通过执行菜单栏中的菜单项实现各项操作。菜单栏包括：“文件”、“编辑”、“显示”、“格式”、“工具”、“程序”、“窗口”、“帮助”八个下拉式菜单。

3. 工具栏

Visual FoxPro 6.0 提供了多个工具栏，每一个工具栏由若干个工具按钮组成，每个按钮对应一个经常使用的特定菜单命令。

4. 窗口显示区

Visual FoxPro 6.0 的窗口显示区用于显示命令或程序的执行结果，同时也用来显示打开的各种窗口和对话框。

5. 命令窗口

主窗口内的命令窗口用来输入和显示各种操作命令。当用户在命令窗口输入正确的命令并按回车键后，系统就会执行该命令。

6. 状态栏

Visual FoxPro 6.0 的状态栏用于显示当前的工作状态。当打开一个数据表之后，将在状态栏上显示该数据表的名称、拥有的记录数目及当前的记录号。

四、Visual FoxPro 6.0 工作方式

Visual FoxPro 6.0 有三种工作方式：菜单方式、命令方式和程序方式。

1. 菜单方式

菜单方式是指利用系统提供的菜单、工具栏、窗口、对话框等进行交互操作。菜单选择方式的优点是操作简单、直观、不需要记忆命令格式，是用户常用的一种工作方式。

2. 命令方式

命令方式是指用户在命令窗口中输入一条命令后按回车键，系统立即执行该命令并显示执行结果。采用命令方式时，用户需要熟悉各种命令格式、功能和使用方法，



采用命令执行方式比采用菜单方式有更高的效率。

3. 程序方式

程序方式是指将所要执行的一批相关命令编写成程序，并存储为程序文件，待需要时执行该程序文件，完成相应任务。程序方式的优点是运行效率高，编制好的程序可以反复执行。对于一些复杂的数据处理问题通常采用程序方式运行。Visual FoxPro 6.0 支持结构化的程序设计方法和面向对象程序设计方法。

五、Visual FoxPro 6.0 的命令窗口

Visual FoxPro 6.0 系统中有两个窗口，一个是“输出窗口”，另一个是“命令窗口”，用户可以在命令窗口中输入各种命令，命令的执行结果在输出窗口中输出。当采用菜单方式操作时，系统会自动把该菜单操作对应的命令显示在命令窗口中。

命令窗口可以放大、缩小、关闭或移动。若显示或隐藏命令窗口可用以下几种方法：

- (1) 选择“窗口”菜单中的“命令窗口”命令，可显示命令窗口；选择“窗口”菜单中的“隐藏”命令，可隐藏命令窗口。
- (2) 单击“常用”工具栏中的“命令窗口”按钮，可显示或隐藏命令窗口。
- (3) 按 Ctrl + F2 组合键可以显示命令窗口；按 Ctrl + F4 组合键可以隐藏命令窗口。





第二章 Visual FoxPro 6.0 的数据类型和数据的表示方法



复习要求

1. 了解 Visual FoxPro 6.0 常用的数据类型。
2. 理解常量、变量、函数和表达式的概念。
3. 掌握常用函数的功能和使用方法。
4. 掌握运算符的运算规则和运算的优先级。
5. 能够熟练进行各种表达式的运算。



复习内容

一、数据及类型

数据有型和值之分，型是数据的分类，值是数据的具体表示。Visual FoxPro 6.0 的数据类型主要有以下几种：

1. 字符型

字符型数据是指一切可显示的字符和字符串。它由中文字符、英文字符、数值字符、空格和其他专用符号组成，通常表示姓名、地址和无需计算的数字等文本信息。1 个字符占 1 个字节，1 个汉字占 2 个字节。

2. 数值型

数值型数据是指可以进行数学运算的数据。它只能由正负号、小数点、0~9 组成，一个数值型数据的长度最多达 16 位，小数最多 15 位。数值型数据还可分为整数型、实数型等。

3. 日期型

日期型数据是用来表示日期的特殊数据。日期型数据有固定的格式，如美国格式 mm/dd/yy（月/日/年），其中月、日、年都是两位数字，可用 SET DATE 和 SET CENTURY 命令来改变日期型数据的格式。日期型数据固定长度为 8 个字节。

4. 逻辑型

逻辑型数据是表示逻辑判断结果的值。逻辑型数据固定长度为 1 个字节。逻辑型



数据通常表示一个条件。

5. 备注型

用于存储指向一个数据块的指针，在表中用 4 个字节表示。只要内存足够大，其能够存储的数据理论上可以无限大，实际存储的数据量受内存空间的限制。Visual FoxPro 6.0 的备注可以包含任意的数据，只要适用于字符串的所有内容，都可以写入备注中。由于备注型数据只用于表中，因此内存中没有备注型的变量和数组元素。

6. 通用型

用于存储指向一个 OLE 对象的指针，在表中用 4 个字节表示。同备注型一样，只要内存足够大，其能够存储的数据理论上可以无限大，实际存储的数据量受内存空间的限制。每个 OLE 对象的具体内容可以是电子表格、文档和图片等。

7. 货币型

用于存储货币值。使用货币型时，对字段或者变量赋初值时要在前面加上一个 \$ 符号。另外，货币型至多只能保留到小数点后 4 位，超过 4 位则会四舍五入，少于 4 位则系统自动在后面补 0。

8. 日期时间型

用于存储日期和时间值。日期时间型字段或变量的存储格式为“yyyymmddhhmmss”，其中，yyyy 表示日期中的年号，mm 表示月份，dd 表示日子，hh 表示时间中的小时（占用 2 个字节），mm 表示时间中的分钟（占用 2 个字节），ss 表示时间中的秒（占用 2 个字节）。日期时间型数据既可以只包含一个日期或只包含一个时间值，也可以同时包含一个日期和一个时间值。如果省略日期，则将自动加上 1899 年 12 月 30 日这个日期；如果省略时间，则自动加上午夜零点这个时间。

对于日期时间型的常量，必须用一对花括号“{}”括起来。

日期时间型中的日期部分的取值介于 {1/1/100} 到 {12/31/9999} 之间，时间部分的取值介于 00:00:00am 到 11:59:59pm 之间。

二、常量和变量

1. 常量

常量是在操作过程中其值保持不变的数据。

(1) 字符型常量：字符型常量是用半角的单引号、双引号或方括号等定界符括起来的一串字符。字符型常量可由文字或符号构成，包括大小写的英文字母、数字、空格等所有合法的 ASCII 码字符以及汉字等。某个字符串所含字符的个数称为该字符串的长度，Visual FoxPro 6.0 允许字符型数据的最大长度为 254 个字节。

如果某一种定界符本身又是字符串的组成部分，就应选择另一种定界符来界定该字符串。例如：[“傻子”的选择]。

注意，只有定界符而不含任何字符的字符串也是一个字符型常量，表示一个长度为零的空字符串。显然，空字符串和包含空格的字符串是不同的。





(2) 数值型常量: 数值型常量是由阿拉伯数字、小数点和正负号构成的各种整数、小数或实数。数值型常量最多为 20 位, 其中小数点占 1 位。对于绝对值很大或很小的数值型常量还可以用科学计数法表示。

注意: 在 Visual FoxPro 6.0 中, 分数 (包括百分数) 不是一个数值型常量。

(3) 逻辑型常量: 逻辑型常量用来表示某个条件的成立与否。逻辑型常量只有逻辑真与逻辑假两个值, 逻辑真值用 .T. 表示, 逻辑假值用 .F. 表示。逻辑型数据固定用 1 个字节表示。

注意: 逻辑值前后的小圆点是不能缺少的。

(4) 日期型常量: 日期型常量是用来表示一个具体的日期。Visual FoxPro 6.0 默认的日期格式为 {^YYYY-MM-DD} 或 {^YYYY/MM/DD}, 例如: {^2004-02-29} 用来表示 2004 年 2 月 29 日。

(5) 日期时间型常量: 日期时间型常量用来表示一个具体的日期与时间。日期时间型常量包括日期和时间两部分内容, 中间用逗号隔开。日期时间型常量默认的格式为: {^YYYY-MM-DD, [hh[:mm[:ss]] [a | p]]}。

(6) 货币型常量: 货币型常量用来表示一个具体的货币值, 货币型常量在数值前添加一个货币符号 “\$”, 是一种特殊的数值常量。货币型数据在存储和计算时保留 4 位小数, 多于 4 位小数时将自动对其后的小数作四舍五入处理。

2. 变量

(1) 变量及变量名: 变量是在命令执行过程中其值可以改变的量。变量名的命名规则是必须以字母或汉字开头, 后面可以是字母、汉字、数字、下划线, 不超过 10 个字符。

变量的数据类型是由存储在其内的数据的类型决定的。

在 Visual FoxPro 6.0 中, 变量的种类可分为内存变量、字段变量、数组变量、系统变量。

(2) 内存变量: 内存变量是一种临时变量, 是在程序或命令状态时用于存放临时数据的内存单元。根据存放数据的数据类型的不同, 又可分为字符型变量、数值型变量、日期型变量、日期时间型变量、货币型变量和逻辑型变量。

通常用赋值的方法来建立内存变量, 所赋值的数据类型决定了该内存变量的类型。如果一个内存变量与一个字段变量同名, Visual FoxPro 6.0 将优先考虑字段变量。这时可以在内存变量前面加上 M. 或 M-> 以示区别。

① 内存变量的赋值命令。

格式 1: STORE <表达式> TO <内存变量表>

功能: 将 <表达式> 的值赋值给 <内存变量表> 中的各变量。

格式 2: <内存变量> = <表达式>

功能: 将 <表达式> 的值赋值给 <内存变量>。

② 内存变量的显示。



格式: ? | ?? 〈表达式表〉

功能: 计算并输出〈表达式表〉中各表达式的值。

(3) 字段变量: 字段变量是用来描述数据表中记录属性的量。在创建数据表时所定义的一个字段就对应一个字段变量, 数据表中的字段名即字段变量名。

字段变量可分为字符型、数值型、日期型、逻辑型、备注型和通用型等。

Visual FoxPro 6.0 规定, 一个数据表文件至多有 255 个字段变量, 且它们的总长不得超过 4 000 个字节。字段变量是一种多值变量, 字段变量的当前值就是当前记录中该字段的值。

备注型字段变量用来存放超过 254 个字符且不等长的文本数据。备注型字段变量的长度固定为 4 个字节, 仅用来存放一个指针, 该指针指向一个与数据表同名而扩展名为 .FPT 的表备注文件内的某一个信息块, 实际的备注文本数据就存放在这个信息块中, 且只受磁盘空间大小的限制。备注型字段可以被编辑、显示或打印, 但不能进行任何形式的运算。

通用型变量用来存放各种 OLE 对象 (可以是电子表格、文档和图片) 的信息, 与备注型字段相似, 通用型字段的长度同样固定为 4 个字节, 也是仅存放一个指针, 该指针指向一个与数据表同名的 .FPT 文件内的某一个信息块。扩展名为 .FPT 的表备注文件是独立于数据表文件而存储在磁盘上的, 专门用来存放那些长度较大且又不等长的信息块。

说明: 一个数据表不管有几个备注型或通用型字段, 其对应的表备注文件只有一个。

字段变量的名、类型与长度是在创建数据表结构时定义的。

(4) 数组变量: 数组也是一种内存变量, 是一组有序数据的集合, 数组中的每个值称为数组元素, 每个数组元素在数组中的位置是固定的, 只能通过称为下标的编号来进行访问。Visual FoxPro 6.0 允许用户定义一维或二维的数组变量, 并支持数据表记录 and 数组之间的数据交换。

三、函数

Visual FoxPro 6.0 中的每一个函数都代表了一种特定的数据操作功能。

使用函数时, 只要写出函数名并给出相应参数, 便能得到它的函数值。

注意: 函数只能出现在表达式中, 不能作为单独的命令使用。

1. 数值函数

(1) 取整函数: INT(数值表达式)

功能: 截去数值表达式的小数部分, 取整数部分, 函数值的类型为数值型。

(2) 取余数函数: MOD(被除数, 除数)

功能: 取被除数和除数相除后得到的余数, 函数值的类型为数值型。

(3) 随机数函数: RAND([〈种子数值〉])





功能：产生一个 0 到 1 之间的随机数，函数值的类型为数值型。

(4) 四舍五入函数：ROUND(〈数值表达式〉, 〈小数点后位数〉)

功能：对小数点后面的数字进行四舍五入，函数值的类型为数值型。

(5) 求绝对值函数：ABS(数值表达式)

功能：求出数值表达式的绝对值，函数值的类型为数值型。

(6) 求最大值函数：MAX(〈表达式 1〉, 〈表达式 2〉, [〈表达式 3〉...])

功能：求出所有表达式中的最大值，函数值的类型可以为：字符型、数值型、浮点型、日期型。

(7) 求最小值函数：MIN(〈表达式 1〉, 〈表达式 2〉, [〈表达式 3〉...])

功能：求出所有表达式中的最小值，函数值的类型可以是：字符型、数值型、浮点型、日期型。

(8) 求平方根函数：SQRT(〈数值表达式〉)

功能：求数值表达式的平方根，数值表达式不能为负，函数值的类型为数值型。

(9) 求自然对数函数：LOG(〈数值表达式〉)

功能：求数值表达式的自然对数，函数值的类型为数值型。

(10) 求指数函数：EXP(〈数值表达式〉)

功能：求出 e^x 的值， x 是给定的数值表达式。函数值的类型为数值型。

2. 字符函数

(1) 宏代换函数：& 〈内存变量〉 [. 〈字符串〉]

功能：执行宏替换，用内存变量的值来代替 & 及变量名。

(2) 子串搜索函数：AT(〈字符表达式 1〉, 〈字符表达式 2〉 [, N])

功能：返回字符表达式 1 在字符表达式 2 首次出现的位置。

(3) 求字符串长度函数：LEN(〈字符表达式〉)

功能：测试字符串的长度，返回的值为数值类型。

(4) 取子串函数：SUBSTR(〈源字符表达式〉, 〈起始位置〉, [〈子串长度〉])

功能：从源字符串中的起始位置处取子串长度个字符，返回的值为字符串。

(5) 左取子串函数：LEFT(〈字符表达式〉, 〈N〉)

功能：从字符表达式的最左边取 N 个字符，返回的值为字符串。

(6) 右取子串函数：RIGHT(〈字符表达式〉, 〈N〉)

功能：从字符表达式的最右边取 N 个字符，返回的值为字符串。

(7) 删除尾部空格函数：TRIM(〈字符表达式〉)

功能：截掉指定字符表达式的尾部空格，返回字符串。

(8) 生成空格字符串函数：SPACE(〈数值表达式〉)

功能：生成指定数值表达式的空格串，返回的值为空格串。

(9) 重复生成字符串函数：REPLICATE(〈字符表达式〉, 〈重复次数〉)

功能：返回的值为一个字符串，这个字符串是将指定字符表达式重复指定次数后



得到的。

(10) 字符串的替换函数: CHRTRAN(<字符表达式 1>, <字符表达式 2>, <字符表达式 3>)

功能: 在字符表达式 1 中, 把与字符表达式 2 字符相匹配的字符替换为字符表达式 3 中的字符, 返回的值为字符串。

(11) 查找替换函数: STRTRAN(<字符表达式>, <查找串> [, <替换串>] [, <始点>] [, <次数>])

功能: 在字符表达式中, 搜索查找串, 并用替换串替换每次出现的查找串, 返回字符串。

3. 日期时间函数

(1) 取系统日期函数: DATE()

功能: 求系统当前日期, 返回的值为日期型。

(2) 取系统时间函数: TIME([<数值表达式>])

功能: 求取系统当前时间, 返回的值为 HH: MM: SS 格式字符。

(3) 取年函数: YEAR(<日期表达式>)

功能: 返回指定日期表达式的年份, 返回的值为数值型。

(4) 取月函数: MONTH(<日期表达式>)

功能: 返回指定日期表达式的月数, 返回的值为数值型。

(5) 取日函数: DAY(<日期表达式>)

功能: 求取日期表达式的日号, 返回的值为数值型。

(6) 求星期几函数: CDOW(<日期表达式>)

功能: 求取日期表达式是星期几, 返回的值为字符型。

4. 转换函数

(1) 数值转换为字符函数: STR(<数值表达式 1> [, <数值表达式 2> [, <数值表达式 3>]])

功能: 将数值转换成字符, 返回的值为字符串。

(2) 字符转换为数值函数: VAL(<字符表达式>)

功能: 提取字符串里的数值, 返回的值为数值型。

(3) 字符转换为 ASCII 码函数: ASC(<字符表达式>)

功能: 返回字符表达式最左边字符的 ASCII 码, 返回值为数值型。

(4) ASCII 码转换为字符函数: CHR(<数值表达式>)

功能: 将 ASCII 码转换为字符, 返回值为字符型。

(5) 将小写字母转换为大写字母函数: UPPER(<字符表达式>)

功能: 将指定字符表达式转换成大写字母形式, 返回值为字符型。

(6) 将大写字母转换为小写字母函数: LOWER(<字符表达式>)

功能: 将字符表达式转换为小写字母形式, 返回值为字符型。

