

FoxBASE

通用数据库管理 程序设计技术

萧蔚等编著

- 从入门到精通的捷径
- 内有完整的FoxBASE
管理程序清单
- 从dBASE到FoxPro
都适用的技术
- 配有通用的数据库管
理软件包



上海科学普及出版社

FoxBASE

通用数据库管理程序 设计技术

萧蔚等编著

上海科学普及出版社

内 容 提 要

本书向读者介绍 FoxBASE 程序设计基础,并进一步介绍新的面向数据结构的通用数据库管理程序设计方法。以传统方法设计的管理程序只能管理特定数据库,如结构改变,程序也须随之变动。而此系统可处理任意一个数据库。本书围绕一个完整实用的通用数据库管理工具软件的编制过程,以大量实例说明程序,还介绍宏代换、计算公式翻译、超宽表格输出、表格自动生成等编程技巧。本书部分程序曾在《软件报》上连载,现将完整的系统奉献给读者。为了方便读者应用,本书另附有《通用数据库管理工具》软件。

本书适用范围广,不仅可作为培训教材供初、中级学员使用,而且也可供有经验的程序设计人员参考。

(沪)新登字第 305 号

责任编辑:郭子安

FoxBASE 通用数据库管理程序设计技术

萧 蔚 等编著

上海科学普及出版社出版

(上海曹杨路 500 号 邮政编码 200063)

新华书店上海发行所发行 常熟高专印刷厂印刷

开本 787×1092 1/16 印张 23.25 字数 588000

1995 年 5 月第 1 版 1995 年 5 月第 1 次印刷

ISBN 7-5427-0914-3/TP·219 定价: 21.00 元
(附软件) 88.00 元

前 言

FoxBASE 是目前国内微机上使用较为广泛的数据库管理系统。从 dBASE 到 FoxBASE, 国内已有十多年的应用历史, 程序设计人员积累了丰富的编程经验和技巧。但从目前的程序设计方法上看, 程序设计人员普遍还采用一个管理程序只管理一个特定数据库的传统的专用程序设计方法。这种专用程序在处理某项特定事务上虽然有其优点, 但在实际应用中并不方便, 原因在于专用程序一则不能管理与程序员设计思想不符的数据库; 二则是当处理的特定数据库结构稍有改动, 相应的管理程序也需随之改动。这些弊病与当前各部门大量信息处理的需要不相适应, 因此微机用户有必要学习用一个程序来处理任意一个数据库的通用程序设计方法。

本书在介绍通用程序设计方法前, 首先向初学者介绍 FoxBASE 数据库基本知识及其程序设计语言, 然后引导读者循序渐进, 从传统程序设计方法入手, 进而学习到较为复杂的通用程序设计方法。考虑到数据库语言已普及多年, 目前需要的是进一步提高, 因此本书重点围绕一个完整实用的《数据库管理工具包 DT》软件的编制过程, 系统讲解用 FoxBASE 编制通用管理程序的方法。书中介绍的宏代换应用、数据库间的数据互录、计算公式翻译、超宽表格输出、复杂表头建立、表格自动生成等大量技巧应用, 充分反映了当前 FoxBASE 程序设计的新概念和新方法, 可供读者在程序设计中参考采用。为减轻各部门程序设计人员的编程工作量和开发一般操作人员的微机应用技能, 本书另附有《数据库管理工具包 DT》软件磁盘, 可供各部门一般微机操作人员用于人事、工资、材料等各类项目管理和各类报表制作。

全书分为两个部分, 第一章到第五章介绍 FoxBASE+ (2.00) 的数据库操作命令和函数, 从第六章到第十章主要介绍程序设计命令和方法。书中的部分程序曾在《软件报》上进行过软件交流和连载, 现汇集成册, 以感谢读者对该软件的关心和支持。

参加本书编著和程序编制工作的有萧蔚、马晓佳、肖翎、罗松、杨元穗、王士华、黄永坤等, 全书由萧蔚同志负责主编。

由于编著者水平所限, 书中难免存在错误和疏漏, 恳请读者提出宝贵意见。

编著者
1994 年 6 月

目 录

第一章 FoxBASE 基本构成	(1)
1.1 数据库概念	(1)
1.2 FoxBASE 语言概况	(2)
1.3 运行 FoxBASE	(9)
1.4 FoxBASE 主要性能和编辑键	(12)
1.5 FoxBASE 的帮助功能	(14)
第二章 建立和编辑数据库	(17)
2.1 数据库结构定义规则	(17)
2.2 建立数据库	(18)
2.3 打开和关闭数据库	(20)
2.4 输入数据	(21)
2.5 查看数据库内容	(27)
2.6 数据库记录定位与删除	(30)
2.7 修改数据	(33)
第三章 数据库的操作	(39)
3.1 建立、打开和更新索引文件	(39)
3.2 数据查找	(45)
3.3 数据库记录排序	(49)
3.4 数据库结构修改和复制命令	(51)
3.5 数据库记录统计与求和	(56)
3.6 数据库分区操作	(59)
第四章 报表与标签输出	(66)
4.1 建立报表	(66)
4.2 输出报表	(67)
4.3 建立标签	(73)
4.4 输出标签	(74)
第五章 函数	(79)
5.1 数值型函数	(79)
5.2 字符型函数	(80)
5.3 文件操作函数	(85)
5.4 输出定位函数和条件函数	(87)
5.5 读键函数	(88)

第六章 命令文件	(90)
6.1 FoxBASE 命令文件的建立与执行	(90)
6.2 程序结构命令	(92)
6.3 语句注释和状态设置	(99)
6.4 输入输出命令	(104)
6.5 数组	(117)
6.6 过程	(121)
6.7 变量操作	(125)
6.8 其它命令	(129)
第七章 程序设计初步	(132)
7.1 程序设计基本步骤	(132)
7.2 面向数据结构的设计方法	(134)
7.3 程序的查错和测试	(138)
7.4 变量命名和常用命令	(142)
7.5 程序设计实践	(144)
第八章 通用数据库管理程序设计	(152)
8.1 “通用数据库管理软件”概况	(152)
8.2 菜单程序	(153)
8.3 建立数据库程序	(160)
8.4 输入数据程序	(160)
8.5 修改数据程序	(164)
8.6 查询数据程序	(165)
8.7 删除记录程序	(166)
8.8 横向计算程序	(167)
8.9 纵向计算程序	(174)
8.10 分表计算程序	(176)
8.11 表格自动生成程序	(180)
8.12 数值零不打印程序	(189)
8.13 生成空行表程序	(190)
8.14 CCED 转数据库程序	(191)
8.15 建立和编译过程文件	(194)
第九章 工具软件 DT 及主要通用程序段	(199)
9.1 数据库管理工具包 DT 概况	(199)
9.2 菜单程序	(200)
9.3 数据录入	(204)
9.4 查询或修改记录	(212)
9.5 数据库编辑	(213)
9.6 记录统计	(224)

9.7 横向计算	(225)
9.8 纵向计算	(226)
9.9 分表计算	(229)
9.10 表格编辑	(233)
第十章 FoxBASE 的配置文件	(245)
10.1 建立配置文件	(245)
10.2 配置文件中可设置的项目	(245)
附录一、《数据库管理工具包 DT》使用说明	(248)
附录二、《数据库管理工具包 DT》源程序	(265)
附录三、FoxBASE 命令	(340)
附录四、FoxBASE 函数	(348)
附录五、FoxBASE 错误信息	(352)
参考文献.....	(359)

第一章 FoxBASE 基本构成

FoxBASE 是目前微机上广泛使用的数据库管理系统，1987 年 7 月美国 Foxsoftware 公司推出 FoxBASE+2.00 版本，1988 年 7 月又推出 FoxBASE+2.10 版本。与 2.00 版相比，2.10 版主要是增加了一些开发工具，如：屏幕设计工具 FoxView，应用程序生成器 FoxCode，自动文献生成器 FoxDoc 和商业图形软件包 FoxGraph，而在命令方面扩展不多。由于 2.10 版在内存小的机器上不能运行，命令与 2.00 版差别不大，故本书的介绍以 2.00 版为准。

1.1 数据库概念

从 70 年代起，人们就开始设计数据库管理系统，从数据之间的相互关系构造出了不同的数据模型，如：层次模型，以记录类型为结点的有向树；网络模型，以记录类型为结点的网络结构；关系模型，以二维方式表示的数据模型。关系模型具有较大的数据独立性和数据扩充能力，使用方便，因而优于前两种模型而成为较为流行的数据模型。国内 80 年代初流行的 dBASE 数据库与 FoxBASE 同属于一种数据模型，即关系型数据库，由于 FoxBASE 与 dBASE 完全兼容，在 dBASE 下编制的程序在 FoxBASE 下不用修改可直接运行，运行速度比 dBASE 快，加之 FoxBASE 的编程命令和函数又比 dBASE 丰富，故目前在微机上一一般都采用 FoxBASE 编写数据库管理程序。

什么是关系数据库？在关系数据库中，所说的“关系”可以被通俗地解释为“二维表格”，例如图 1.1 所示的表格就被称为一个“关系”。

部门	小计	耕地税	基建费		表头相当于数据库的结构
合计	10.00	4.00	6.00		表体相当于数据库的记录
a1	3.00	1.00	2.00		
a2	7.00	3.00	4.00		
		表列相当于数据库的字段			

图 1.1 二维表在数据库系统中被称为“关系”

二维表具有下列性质：

1. 二维表的表头就相当于数据库的结构，表格中的数据类型在数据库结构定义时得到确定。
2. 表列相当于数据库的字段。字段具有属性定义，其定义内容为：字段名称、字段类型（如字符型、数值型等）和字段宽度，对于数值型数据还要定义小数位。字段属性定义后，一行数据的数据类型也就得到确定。
3. 表行相当于数据库的记录，每一行数据随表列定义的不同可具有不同的数据类型。

以二维表格形式表示的数据库可以是一个部门关系，也可以是一个工资关系或其他任何关系，不同的关系可以通过“关键字”进行查找，图 1.2 表示了用“姓名”作为关键字，通过表 (a) 查找表 (b) 相同姓名人员的工资情况，这时表 (b) 可以是一个在另一个工作区打开的数据库。

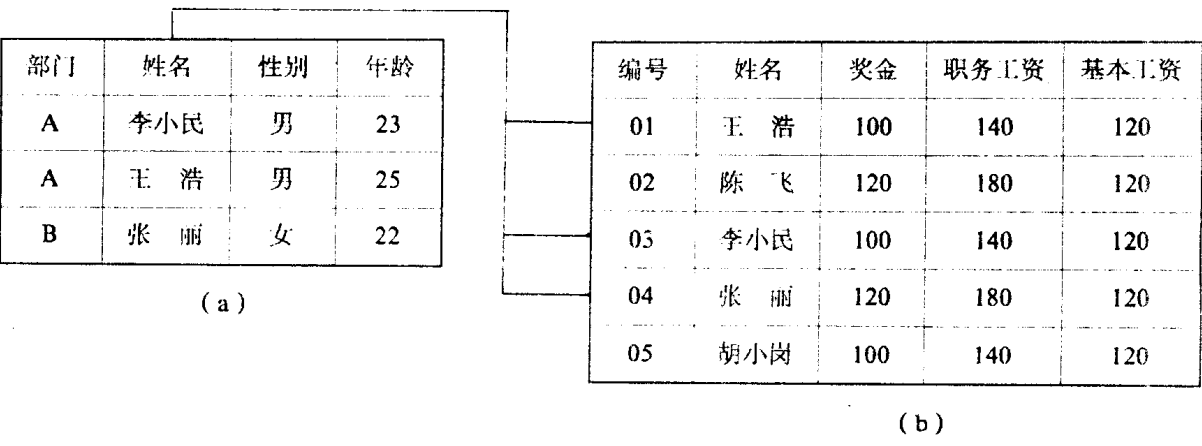


图 1.2 两个关系间的关联查找

不同的数据库除可以通过“关键字”发生联系外，还具有下面三个特点：

- 1. 连接 (JOIN)，以两个数据库中的一个相同字段为关键字，指定连接条件将第二个数据库的记录连接到第一个数据库的末尾。
- 2. 投影 (PROJECT)，从一个数据库中选取若干字段，组成一个新数据库。
- 3. 选择 (SELECT)，从一个数据库中选取若干满足指定条件的记录，组成一个新的数据库。

常说的数据库管理程序设计有两种含意，一种是系统设计，利用某种高级语言（如 C 语言）或混合语言编制的数据库管理系统，如 FoxBASE 就是用 C 语言编制；另一种是应用程序设计，也叫二次开发设计，即利用 FoxBASE（或 dBASE）提供的命令和函数编写程序，这是人们常用的设计方法，也是本书主要介绍的方法。

1.2 FoxBASE 语言概况

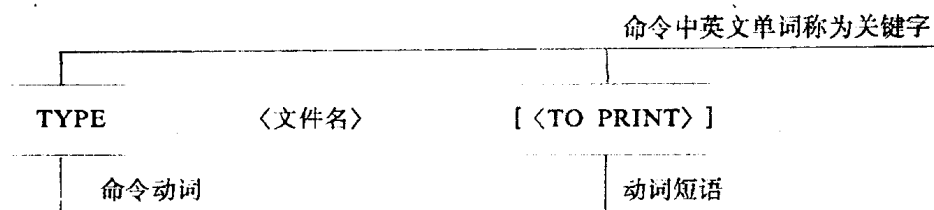
1. 运行环境

FoxBASE 用 C 语言写成，可在多种操作系统下运行，运行时内存空间至少需要 360KB（KB 表示千字节），若要较好地发挥 FoxBASE 系统性能，还应尽量留给 FoxBASE 更多的内存，如果内存较小，在进行数据库记录排序和索引等操作时将不能执行。同时最好选用高分辨率显示器，以便程序设计时易于做到美观方便，人机界面友好。

2. 命令

在 FoxBASE 中，命令用来完成系统状态设置、数据库操作等功能，可以用两种方式来执行 FoxBASE 的命令：一种是将命令写在程序中，作为程序语句，在运行程序的同时执行命令；另一种是在 FoxBASE 系统状态下键入命令直接执行。把命令写入程序在运行程序的同时执行命令被称为程序执行方式，在 FoxBASE 系统状态下直接键入命令并执行被称为命令立即执行方式。初学者一般需先在 FoxBASE 系统状态下键入不同命令进行学习，了解命令执行结果，待熟悉命令后再进行程序设计。

FoxBASE 的命令由两部分组成：一部分称为命令动词，另一部分被称为动词短语。一个典型的命令格式如下：



命令动词用来指出计算机要完成的任务，动词短语则给出完成命令所需要的参数。命令中的英文短语不仅代表操作含义，而且还属于 FoxBASE 的关键字，用户不得随意改动，命令中的非英文短语如〈文件名〉由用户指定。在该命令中，TYPE 表示输出，短语 TO PRINT 表示将指定〈文件名〉的内容送打印机输出。

本书对命令和函数格式中使用的符号做如下约定：

- 〈 〉 尖括号表示必选项。
- [] 方括号表示可选项。
- / 斜线表示在斜线上、下方的内容可任选一个。
- ... 省略号表示它前面的项目可重复，重复项目间用逗号分隔。

命令中还会包括“范围”选择，“范围”可以有下面四种选择：

- ALL 表示对数据库的所有记录操作。
- NEXT 〈n〉 表示从当前记录开始，对下面的 n 个记录进行操作。
- RECORD 〈n〉 表示仅对指定的第 n 个记录进行操作。
- REST 表示对从当前记录起，到数据库最后一个记录止的所有记录进行操作。

命令的书写规则为：

(1) 命令一般以动词开头，命令中关键字不能改动，但可以缩写为前四个以上的字符，比如建立程序的命令 MODIFY COMMAND 〈文件名〉可缩写为 MODI COMM 〈文件名〉。

(2) 命令中关键字使用大、小写字母均可，当一条命令有多个短语时，可以用任意个空格来分隔短语，命令中短语的先后顺序可以是任意的。

(3) 一条命令最大长度应少于 254 个字符，太长一行写不下时，可以使用分行符；在行尾分行。

表 1.1 列出了几个常用的 FoxBASE 命令。

表 1.1 FoxBASE 命令示例

命令格式	功能说明
SET TALK ON/OFF	命令执行结果发送/不发送到屏幕
SKIP [〈数值表达式〉]	根据表达式的值移动记录指针

命令格式	功能说明
SORT TO <数据库文件名> ON <字段> [A/C/D] [<字段> [A/C/D]...][<范围>][FOR <条件>] [WHILE <条件>][FIELDS <字段名表>]	按指定字段排序, 生成新的数据库文件
SUM [<范围>][<数值表达式表>][TO <内存变量表>] [FOR <条件>][WHILE <条件>]	在数据库中对满足条件的表达式求和
SUSPEND	暂停程序执行, 用 RESUME 恢复执行
USE [<数据库文件名>][INDEX <索引文件名表>][ALIAS <别名>][EXCLUSIVE]	打开数据库文件
WAIT [<提示>][TO <变量>]	暂停程序执行, 等待按键
ZAP	清除数据库所有记录

3. 函数

函数用来实现运算操作, 其意义与数学上使用的函数相近, 在 FoxBASE 中函数具有数据类型, 函数共有四种数据类型, 这四种类型分别是:

- 数值型函数 (N 型)
- 字符型函数 (C 型)
- 日期型函数 (D 型)
- 逻辑型函数 (L 型)

数值型函数返回值为数值, 如 **INT** (23.7) 返回数值 23。

字符型函数返回值为字符, 如 **REPLICATE** ("=", 5) 返回字符串 "===== "。

日期型函数返回值为日期, 如 **DATE** () 返回系统当前日期, 如 03/22/93。

逻辑型函数返回值为逻辑值, 如 **EOF** () 将测试文件是否结束, 结束返回逻辑值真, 即符号.T., 没有结束返回逻辑值假, 即符号.F.。

使用 FoxBASE 函数时注意:

- 1) 函数返回值带有相应的数据类型。
- 2) 除宏代换 & 函数外, 其它函数名后都有一对圆括号。

表 1.2 列出了几个常用的 FoxBASE 函数。

表 1.2 FoxBASE 函数示例

函数格式	功能说明
ABS (<数值表达式>)	求绝对值
ASC (<字符表达式>)	返回字符串左边第一个字符的 ASCII 码
AT (<字符表达式 1>, <字符表达式 2>)	返回表达式 1 在表达式 2 中的位置
CHR (<数值表达式>)	将表达式代表的 ASCII 码转换为字符

函数格式	功能说明
COL ()	求屏幕光标当前列位置
CTOD(〈字符表达式〉)	字符串转换为日期
DELETED ([〈数值表达式〉])	测记录是否有删除标记
EOF ([〈数值表达式〉])	测工作区中记录指针是否指向文件尾
EXP(〈数值表达式〉)	求表达式指定的 e 的幂
FCOUNT ([〈数值表达式〉])	返回指定工作区中数据库字段个数

4. 常数

常数在程序运行过程中不发生变化, 常数也具有四种类型, 分别为:

- 数值型常数 (N 型)
- 字符型常数 (C 型)
- 日期型常数 (D 型)
- 逻辑型常数 (L 型)

数值型常数为: 234 0.5 47.23 等, 与人们常用数据相同。

字符型常数为: “234” “部门” “abcd123” “* * * * *” 等, 这些数据必需包含在规定的界限符内, 有三种界限符, 分别是“ ”、‘ ’ 和 [], 一般常用双引号作界限符。字符型常数又被称为字符串常数, 当字符串常数内还需含有界限符时, 必选用不同的界限符, 如“部门=[a1], 单位=[a2]”, 又如: “abc ’ ### ’ def”, 如果写成 “abc “###” def” 则为非法字符串。注意: 界限符内一行最多可写 254 个字符, 当两个界限符紧密连在一起时, 这个字符串就被称为空字符串, 形如 “ ”。

日期型常数为: 12/29/86 (月/日/年) january (一月) thursday (星期四) 等, 日期取值范围在 01/01/01 到 12/31/99 之间, 月份取值在 1 到 12 月之间, 星期取值在星期日到星期六之间。

逻辑型常数为: 逻辑真 .T. 与逻辑假 .F.。

5. 变量

变量在程序运行过程中会发生变化, 变量可分为下面两种:

(1) 字段名变量。字段名变量即数据库的字段名, 具有五种类型, 分别为:

- 数值型变量 (N 型)
- 字符型变量 (C 型)
- 日期型变量 (D 型)
- 逻辑型变量 (L 型)
- 明细型变量 (M 型)

(2) 普通变量。普通变量由用户自定变量名, 具有四种类型, 分别为:

- 数值型变量 (N 型)
- 字符型变量 (C 型)

- 日期型变量 (D 型)
- 逻辑型变量 (L 型)

用户定义了一个变量就相当于在内存空间开辟了一个用于存放数据的存储单元, FoxBASE 的变量类型在使用前不需要先定义, 变量类型随存放的数据类型而定。对变量的使用可归纳出以下几点:

(1) 给变量命名。变量名由汉字、字母、数字和下划线“_”组成, 必须以汉字和字母开头。变量名中, 字母为 26 个英文字母, 大小写均可; 数字为 0 至 9。变量名最多不能超过 10 个字符, 超过 10 个字符只认前 10 个字符, 同时变量名不允许用 FoxBASE 命令和函数中的关键字。

合法变量名如:

aa a1 bb12 部门 w ls_1

非法变量名如:

bb: type 2b

(2) 给变量赋值。变量的作用是存放数据, 可以用下面三种方法给变量赋值:

① 用赋值号“=”给变量赋值。

格式为:

〈变量名〉= 〈表达式〉

例: 给变量 a1 赋值 5。

. a1=5

圆点·为 FoxBASE 提示符, 其后为用户输入内容。按回车键执行后变量 a1 存储内容为 5。

② 用 STORE 命令给变量赋值。

格式为:

STORE 〈表达式〉 TO 〈变量名表〉

〈变量名表〉中可以由逗号分隔的多个变量, 用 STORE 命令给变量赋值的好处是一次可给多个变量赋值一个相同内容。

例: 给变量 aa、a1 和 bb12 赋值 5。

. store 5 to aa, a1, bb12

③ 用输入命令给变量赋值。

输入命令如: ACCEPT 和 INPUT 等, 这些命令主要用于程序中作为输入语句, 如:

input to aa

程序执行到 INPUT 语句时, 等待用户从键盘输入一个数据给变量 aa。

(3) 变量的类型和结果以最后存放内容为准, 如:

```
. aa=0.5  
. aa="张三"
```

由于字符串“张三”执行在后, 故变量 aa 最后为字符型, 并取值为字符“张三”, 实际显示时不包括界限符“ ”。

(4) 在输入命令中, 输入命令本身就决定了变量类型, 设有一名为 aa 的变量, 则:

ACCEPT“输入姓名”TO aa

ACCEPT 要求用户输入一个字符数据, 故变量 aa 为字符变量。

INPUT“输入编号”TO aa

INPUT 要求用户输入一个数据, 如果输入 5, 则变量 aa 为数值变量, 如果带双引号输入“5”, 则变量 aa 为字符变量。

(5) 在字符型变量执行时, 界限符内是什么内容, 执行结果也就是什么内容。如:

```
. aa="abcd123"  
. ? aa  
abcd123  
. cc="部门=[a1], 单位=[a2]"  
. ? cc  
部门=[a1], 单位=[a2]  
这里?为输出命令, 下行为输出结果。
```

(6) 在命令和函数中, 字段名变量可以向普通变量一样进行操作。

建议不要用单个字母 A 到 J 作变量名 (包括字段名变量), 因为 FoxBASE 已将其用作工作区代号, 也不要 FoxBASE 关键字作变量名。

6. 运算符

FoxBASE 有四种运算符, 分别是:

(1) 算术运算符

算术运算符对数值型数据进行运算, 产生数值型数据, 算术运算符分别为:

()	优先运算符, 需要用到多个括号时一律用圆括号, 最内层优先。
+	加法运算符
-	减法运算符
*	乘法运算符
/	除法运算符
** 或 ^	指数运算符

(2) 比较运算符

比较运算符用来对两个数据进行比较运算, 产生逻辑值真或假, 比较运算符分别为:

< 小于	> 大于	= 等于	< > 或 # 不等于
< = 小于等于	> = 大于等于	= = 字符串全等	\$ 字符串包含

用比较运算符时注意:

① 比较运算符两边的内容必须是同一类型，如：

35 > 21 "abc" < "c" a2 >= a1 (a2, a1 是相同类型的变量)

在比较运算中，数值型数据按其值的大小进行比较。字符型数据按 ASCII 码取值大小进行比较，比较顺序为从左至右逐字符比较，比较规则为：

字符相同，谁长谁大，如 "abcd" 大于 "abc"。

字符长度相同，从左至右逐字符进行比较，谁左边 ASCII 码大谁就大，如 "bbc" 大于 "abc"，因为左边第一个字符 b 的 ASCII 码为 98，a 的 ASCII 码为 97。

如果汉字参与比较，则按汉字汉语拼音首字母（相当于英文字母）的 ASCII 码比较。

日期型数据则按年、月、日的先后顺序进行比较。

② \$ 是字符串包含运算符，它被用来表示两个字符串之间的包含关系，如："张" \$ "张明"，"c" \$ "abc"，都为逻辑真.T.，因为字符"c"包含在字符"abc"中，而"t" \$ "abc"为逻辑假.F.，因为字符"t"不包含在字符"abc"中。

③ == 是字符串全等运算符，它被用来表示两个字符串之间的全等关系，如："C" == "C"，"张" == "张"都为逻辑真.T.，因为字符"C"全等于字符"C"，而"C" == "C1"为逻辑假.F.，因为字符"C"不等于字符"C1"。

(3) 逻辑运算符

逻辑运算符用来对逻辑型数据进行运算，逻辑运算符分别为：

() 优先运算符

.NOT. 逻辑非

.AND. 逻辑与

.OR. 逻辑或

逻辑运算优先顺序为：()、.NOT.、.AND.、.OR.，下面是几个逻辑运算例子：

工资 > 180 .OR. 住房面积 > 70

选择 = "Y" .AND. kk = 10

.NOT.EOF ()

在逻辑运算中，逻辑非表示条件不成立，逻辑与表示两端条件都成立，逻辑或表示两端只需要有一个条件成立，满足这些条件就返回逻辑值.T.，否则返回逻辑值.F.。

(4) 字符串运算符

字符串运算符用来对两个字符型数据进行连接计算，并将其合并为一个新字符串，字符串运算符为：

+ 字符串连接

- 字符串压缩尾部空格连接

字符串+连接时，前一字符串尾部有空格时空格参与连接；字符串-连接时，前一字符串尾部有空格时空格被转移，前一字符串尾部空格被转移到连接后的字符串尾部，例如：

"abc" + "ab c"

结果为：

"abc ab c"

前一字符串尾部空格保留，又如：

`"abc" + "ab c"`

结果为：

`"abcab c"`

前一字符串尾部空格被移到连接后的字符串尾部。

7. 表达式

由各种变量、函数、常数和运算符按一定规则组成的式子叫表达式。最简单的表达式就是一个变量或者是一个常数，形如：aa、“234”或 12.5。表达式类似数学上的算式，运算后得到一个与运算数据类型相同的结果。根据运算符的不同，表达式可分为下面四种：

(1) 算术表达式

参与运算的数据或变量必须是数值型，运算结果也是数值型，如：

`a1+a2*2` `(2+53.2)/2`

(2) 字符表达式

参与运算的数据或变量必须是字符型，运算结果也是字符型，如：

`"ab"` `"cd" $ "abcdef"` `"姓名"+"住址"` `a1+a2`

其中字符型常数要用界限符括起，字符型变量则不需要界限符。

(3) 关系表达式

参与运算的常数或变量必须是相同类型，运算结果是逻辑值真或假，如：

`5>3` `"ab"<"w"` `性别="男"`

其中未加引号的性别是字段名变量。

(4) 逻辑表达式

逻辑表达式运算结果是逻辑值真或假，如：

`性别="男".AND.职称="工程师"` `.NOT.EOF( )`

1.3 运行 FoxBASE

1. 进入 FoxBASE

多用户系统进入 FoxBASE 的命令为 MFOXPLUS，单用户的命令为 FOXPLUS，本书假设读者使用的是多用户系统。有下面两种进入 FoxBASE 的方法：

(1) 交互方式

在 DOS 下，设文件 MFOXPLUS.EXE 在 C 盘根目录，键入：

`C> MFOXPLUS` 按回车 (Enter) 键

此时可进入 FoxBASE，其系统提示符为圆点“.”。

(2) 批命令方式

批命令方式需要建立一个批命令文件, 可用 DOS 的 EDLIN、COPY 和外部编辑软件 WPS 等来建立批命令文件。最简单的方法是用 DOS 的 COPY 命令来建立批命令文件。设文件 MFOXPLUS.EXE 在 C 盘的 FOX 子目录下, 键入:

```
C> COPY CON: FOX.BAT
CLS                清屏幕
CD \FOX            转FOX子目录
MFOXPLUS           进入FoxBASE
CD \               当FoxBASE运行结束后回根目录
·Z
```

上面每输入一条命令后都要按回车键。其中 ·Z 表示按下 Ctrl 键不放, 再按一下字母 Z, 用途是在当前磁盘上存储 FOX.BAT 文件。批命令文件 FOX.BAT 建立后在 DOS 根目录下运行 FoxBASE 只需键入:

```
C> FOX
```

2. 退出 FoxBASE

退出 FoxBASE 只需键入:

```
.QUIT 回车
```

QUIT 为 FoxBASE 的退出命令, 前面的圆点表示在 FoxBASE 系统下。

3. 执行 FoxBASE 命令

进入 FoxBASE 后, 屏幕下方会显示一个圆点“.”, 这个圆点为 FoxBASE 系统提示符, 在圆点后就可以输入 FoxBASE 的命令。有下面两种执行 FoxBASE 命令的方法:

(1) 交互方式

交互方式也称为命令立即执行方式, 在交互方式下键入一个命令并回车后, 屏幕上将立即显示命令执行结果, 如果输入有错, 屏幕将显示错误信息提示。

注意: 下面每键入一条命令都要按回车键, 书中将不再注明!

交互方式下命令执行情况为:

```
. aa = "bbb"
bbb
```

赋给变量 aa 一个字符串“bbb”并按回车后, 屏幕下一行立即显示 aa 变量的值 bbb。

```
. cc = 2.3 * 2
4.6
```

赋给变量 cc 一个数值算式并按回车后, 屏幕下一行立即显示 cc 变量的值 4.6。

```
. a1 = aa + cc
Operator / operand type mismatch (操作符或操作数类型不匹配)
```