FOXBASE +
关 系 数 据 库 系 统

(第二版)

郑甫京　沈金发　编著

清 华 大 学 出 版 社

需要全本请访问 www.ertongbook.com

(京)新登字 158 号

内 容 简 介

FOXBASE + (2.00、2.10 版)是美国 FOX 软件公司推出的关系数据库管理系统,同美国 Ashton-Tate 公司的 dBASEⅢ PLUS 完全兼容。在 FOXBASE + (2.00、2.10 版)下运行 dBASEⅢ PLUS 程序无需作任何改动,而且在许多方面还进行了扩充,如增加了数组、自定义函数等。运行速度及编程环境的提高与改善,是 FOXBASE + (2.00、2.10 版)很具特色的地方,给用户带来了极大的方便。

本书全面介绍了 FOXBASE + 系统,并深入分析、阐述了系统应用中的有关问题。全书共 13 章,书末有 3 个附录。

本书第二版主要是在第一版的每章之后增加了习题,包括选择题、简答题、思考题和操作练习题。

本书可作为数据库管理系统课程的教材。对于广大计算机用户来说,本书是一本详略恰当而实用的自学用书与上机参考手册。

版权所有,翻印必究。
本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

书　名: FOXBASE + 关系数据库系统(第二版)
作　者: 郑甫京　沈金发　编著
出版者: 清华大学出版社(北京清华大学校内,邮编 100084)
http:// www .tup .tsinghua .edu .cn
印刷者: 国防工业出版社印刷厂
发行者: 新华书店总店北京科技发行所
开　本: 787×1092 1/ 16　印张: 14 .75　字数: 350 千字
版　次: 1998 年 8 月第 2 版　1998 年 8 月第 1 次印刷
书　号: ISBN 7-302-03029-4/ TP·1613
印　数: 616001～624000
定　价: 18.80 元

第 二 版 前 言

目前,微型计算机已成为人们日常生活、学习、工作的必不可少的工具之一。随着微机技术的不断发展,其应用已从单机发展到网络,实现了信息的交换和数据的共享。人们对微机的认识也由单纯的文字录入、排版、打印向信息的收集、分析和多媒体创作等方面转化。

数据库技术是计算机系统发展中比较成熟的技术。尤其是在微机系统中,数据库技术得到了更广泛的应用。

FOXBASE + 是 80 年代末由美国 FOX 公司研制开发的关系数据库管理系统,它具有运行速度快、适用机种多、支持多种操作系统、对计算机工作环境要求低、可分别在单机和网络上运行等特点,特别是 FOXPRO、Visual C + + 、Visual Basic、PowerBuilder 等最新开发的软件平台均可方便地调用和存取 FOXBASE + 的数据文件。

FOXBASE + 被定为计算机等级考试(二级)的主要内容之一。

《FOXBASE + 关系数据库管理系统》一书自 1991 年出版以来,先后 17 次重印,印数近 60 万册,深受广大读者的欢迎,促进了微机应用的普及和推广。

《FOXBASE + 关系数据库管理系统》一书被许多院校选为数据库管理系统课程的教材,并为各类计算机培训班选为指定的参考书。

此次再版除了修正已发现的错误与不当之处外,还特别在每章后面增加了选择题、简答题、思考题和操作练习题,以便读者通过做题和练习更好地掌握所学的内容。

作 者

1998 年 5 月

前　　言

随着微型计算机的普及与推广，关系数据库系统越来越受到广大用户的欢迎。FOXBASE+是美国 FOX 软件公司 80 年代推出的关系型数据库管理系统。它与目前国内流行的由 Ashton-Tate 公司开发的 dBASEⅢ PLUS 完全兼容，并且还有很多方面的扩充。

FOXBASE+是用 C 语言写成的，适用于 IBM-PC/ XT、长城系列机、VAX 11/ 780 等多种计算机，支持 IBM-DOS、MS-DOS、UNIX 等操作系统。它与 dBASEⅢ PLUS 相比，平均速度快 5.9 倍，进一步改善和扩充了 dBASEⅢ PLUS 的功能，增加了数组和自定义函数。

由于 FOXBASE+有很多优点，广大用户迫切需要一本介绍该系统的资料，由此，我们编写了这本书。

本书从实用的角度出发，系统地对 FOXBASE+的各种功能及其使用方法作了详细的介绍。为了便于讲授、学习和使用，书中列举了大量的应用实例。

本书可作为高等院校的数据库课程参考书和各类数据库培训班教材，也可作为计算机用户、计算机工作者的参考资料。

所有关于 dBASEⅢ PLUS 的书都可作为学习 FOXBASE+的参考资料。

由于时间仓促，水平有限，书中难免有错误和不足之处，敬请读者批评指正。

作　者

1991 年 2 月

目　　录

第一章　概述 …… 1

1.1　FOXBASE + 要求的环境 …… 1

1.2　FOXBASE + 语言的成分 …… 1

1.3　赋值命令(STORE)和显示命令(?/ ??) …… 9

1.4　自学习 HELP …… 12

1.5　FOXBASE + 主要性能指标 …… 16

习题一 …… 16

第二章　建立数据库和数据录入 …… 18

2.1　定义新的数据库 …… 18

2.2　全屏幕编辑 …… 22

2.3　数据库文件打开与关闭 …… 24

2.4　数据库的显示 …… 24

2.5　建立数据库结构的其它方法 …… 28

2.6　数据库结构修改 MODIFY STRUCTURE …… 31

2.7　复制数据库 COPY TO …… 32

习题二 …… 33

第三章　数据的编辑、检索和统计 …… 36

3.1　记录定位 …… 36

3.2　插入和删除 …… 38

3.3　修改数据记录 …… 41

3.4　数据组织 …… 45

3.5　数据检索 …… 50

3.6　数据库的统计 …… 53

习题三 …… 55

第四章　多重数据库操作 …… 57

4.1　工作区选择 SELECT …… 57

4.2　数据库文件间连接 JOIN …… 58

4.3　数据库文件间更新 UPDATE …… 60

4.4　数据库文件间关联 SET RELATION …… 62

习题四 …… 64

需要全本请访问 www.ertongbook.com

第五章　FOXBASE + 函数 …… 66
5.1　数学运算函数 …… 66
5.2　字符函数 …… 69
5.3　日期函数 …… 74
5.4　转换函数 …… 77
5.5　测试函数 …… 80
5.6　标识函数 …… 87
5.7　输入函数 …… 90
习题五 …… 92
第六章　辅助功能 …… 94
6.1　数据库文件间记录追加 APPEND FROM …… 94
6.2　屏幕画面的保存与恢复 …… 95
6.3　选择系统工作状态 …… 95
6.4　磁盘文件操作命令 …… 102
6.5　其它辅助命令 …… 104
习题六 …… 106
第七章　程序设计 …… 107
7.1　程序文件的建立、修改和执行 …… 107
7.2　交互式命令 …… 108
7.3　程序的选择结构 …… 110
7.4　程序的循环结构 …… 115
7.5　程序的注释 …… 118
7.6　过程调用 …… 119
7.7　内存变量操作 …… 123
7.8　数组 …… 128
7.9　自定义函数 …… 134
7.10　错误捕获和处理 …… 135
7.11　程序调试 …… 137
习题七 …… 141
第八章　输入输出设计 …… 143
8.1　打印报表 …… 143
8.2　打印标签 …… 149
8.3　格式设计命令@ …… 153
8.4　使用@命令设计屏幕输入输出格式 …… 160
8.5　使用@命令设计报表 …… 165

习题八…… 167
第九章　与其它语言的接口 …… 168
9.1　与其它语言交换数据 …… 168
9.2　调用外部程序 RUN/ ！ …… 170
9.3　调用二进制程序文件 …… 171
习题九…… 173
第十章　并发控制 …… 174
10.1　数据的保护…… 174
10.2　死锁处理…… 179
10.3　错误的捕获与改正…… 186
习题十…… 189
第十一章　多用户命令与函数…… 190
11.1　多用户命令…… 190
11.2　网络函数…… 193
习题十一…… 196
第十二章　实用程序 …… 197
12.1　编译 FOXBASE + 程序文件 …… 197
12.2　过程文件生成器…… 198
习题十二…… 199
第十三章　FOXBASE + 2.10 …… 200
13.1　FOXBASE + 的新进展 …… 200
13.2　上拉和下拉式菜单…… 201
13.3　扩展的命令和函数…… 205
习题十三…… 208
附录 A　命令一览表 …… 209
附录 B　函数一览表 …… 218
附录 C　错误信息及其说明 …… 222

第一章　概　　述

FOXBASE + 是美国 FOX Software 公司于 1987 年 2 月推出的关系数据库系统,1987 年 7 月又进行了修改,成为 FOXBASE + 2.00 版本。目前的最高版本是 1988 年 7 月推出的 FOXBASE + 2.10 版。FOXBASE + 不仅与 Ashton-Tate 公司的 dBASEⅢ PLUS 完全兼容,而且还有很多方面的扩展。其最大优点是:速度快,适用机种广泛,支持的操作系统多。经 18 种标准测试程序测试,FOXBASE + 运行速度平均比 dBASEⅢ PLUS 快 5.9 倍。如果配置 8087/ 80287 协处理器,运行速度可得到进一步改善。本书介绍的 FOXBASE + 以 2.00 版为准。最后一章介绍了 FOXBASE + 2.10 版的新发展。

1.1　FOXBASE + 要求的环境

FOXBASE + 是用 C 语言开发的,因而可以在多种操作系统及机器环境下运行。如:IBM-PC、长城 286 等兼容机,配上 MS-DOS、CCDOS、XENIX 等操作系统都能够运行 FOXBASE + 。

在微型机上运行 FOXBASE + 至少需要 360KB(千字节)可用内存和一个硬盘。如果内存很大,FOXBASE + 则尽可能多占用内存,这样,工作更快,性能更好。

如果安装有 8087/ 80287 算术协处理器,FOXBASE + 可以充分利用它,运行速度也将达到最高。

在 80386 保护方式下运行 FOXBASE + ,至少需要 2MB(兆字节)可用内存、一个硬盘和一个 80287/ 80387 算术协处理器。

1.2　FOXBASE + 语言的成分

每一种计算机语言都有它自身的基本组成成分。FOXBASE + 含有数十种命令或语句,通过它们实现建立数据库和对数据的各种查询等操作。与其它高级语言相似,FOXBASE + 还支持多种数据文件。它能够定义和处理不同类型的数据,并可通过程序实现对数据文件中的数据的操作。同时它还拥有数十种功能丰富、使用方便的函数。

下面介绍 FOXBASE + 的主要组成成分。

1.2.1　命令

在 FOXBASE + 语言中,对数据的操作都是由命令来完成的。命令相当于一般高级语言中的语句,但比高级语言的语句更精炼,功能更强。例如,有下列一组 FOXBASE + 命令:

```
USE Student
LIST
```

```
COUNT ALL TO mid
CLOSE DATABASES
```

它们的功能是:

(1) 打开数据库文件 student,以备对该数据库文件进行操作;

(2) 将已打开的数据库文件 student 中的记录列表显示;

(3) 统计数据库文件 student 中记录的个数,结果送入名为 mid 的内存变量中;

(4) 关闭数据库文件 student,保存在磁盘上。

如果使用一般的高级语言(如 BASIC、FORTRAN 等语言),完成上述功能需要很多的语句才能完成描述"打开数据库文件"、"列表显示"、"统计"和"关闭数据库文件"等操作的详细过程。

下面列出 FOXBASE + 典型操作命令的一般形式:

命令动词 [〈范围〉] [〈表达式表〉] [FOR〈条件〉] [WHILE〈条件〉]

其中:

(1) 命令动词: 是 FOXBASE + 命令名,用来指示计算机要完成的操作,如:"USE"、"LIST"等等。

(2) 范围: 用来确定计算机执行该命令所操作的记录,可有四种选择:

ALL ——表示对数据库文件中的所有记录进行操作。

NEXT〈n〉 ——表示对从当前记录开始的 n 个记录进行操作。

RECORD〈n〉 ——表示仅对第 n 号记录进行操作。

REST ——表示对从当前记录开始到最后一个记录为止的所有记录进行操作。

(3) 表达式表: 可以是一个或多个由逗号分隔开的表达式,可用来指示计算机执行该命令所操作的结果参数。

(4) FOR〈条件〉: 通知 FOXBASE + 命令仅对满足条件的记录进行操作。如果使用 FOR 子句,而未指定范围,FOXBASE + 将记录指针重新指向数据库文件顶,并且用 FOR 条件与每个记录进行比较。

(5) WHILE〈条件〉: 在数据库文件中,从当前记录开始,按记录顺序从上向下进行比较、处理,直到不满足条件为止。

在 FOR 子句和 WHILE 子句中,〈条件〉必须是逻辑型表达式,即〈条件〉返回值必须为真(.T.)值或假(.F.)值。

为了方便用户,FOXBASE + 命令作了以下一些规定:

(1) 每条命令都以命令动词开头,且必须严格符合本书所介绍的命令语法格式。

(2) 命令中的短语(或子句)可按任意次序排放,如下边三条命令均是等价的:

```
LIST 姓名,性别 NEXT 5 FOR  教研室 = "计算机"
LIST NEXT 5 姓名,性别 FOR 教研室 = "计算机"
LIST FOR 教研室 = "计算机"姓名,性别 NEXT 5
```

(3) 一条命令的最大字符个数不得超过 254 个。

(4) 命令中的短语 (或子句)可由若干个空格隔开,每个空格也算一个字符。

(5) 如果命令太长,一行写不下,可分几行写,但除最后一行外每行末尾使用一个分行符";"。

(6) 命令动词和 FOXBASE + 保留字均可以用 4 个以上字母来简写。例如:

STORE　　可简写为 STOR

CREATE　可简写为 CREA 和 CREAT

DISPLAY　可简写为 DISP、DISPL 和 DISPLA

(7) 命令动词、FOXBASE + 保留字、文件名、字段名和内存变量名可以是任意的大小写混合形式。例如: DISPLAY, display 和 Display 均有同等效果。

(8) 尽量避免或最好不使用 FOXBASE + 保留字和命令动词作为文件名、字段名和内存变量名,因为这样将有可能带来一些不必要的混乱。

例如: 以 status 命名的字段是无法用 DISPLAY status 命令显示的,因为该命令将显示当前 FOXBASE + 的系统状态,而不是显示 status 字段的内容。

(9) 建议不要使用单个字母 A 到 J 作为数据库文件名,因为它们是 FOXBASE + 保留字,作为别名。

(10) 程序设计时, DO WHILE/ ENDDO、IF/ ENDIF、DO CASE/ ENDCASE 和 TEXT/ ENDTEXT 语句必须配对使用。

在命令语法格式中,本书对所使用的符号作如下约定:

〈 〉: 尖括号。表示必须要选的项目。

[　]: 方括号。表示可选的项目。

/　: 斜线号。表示两个项目中选择其中一个项目。

… : 省略号。表示前面项目可继续重复多次选择,项目与项目之间用逗号隔开。

1.2.2　文件

FOXBASE + 有 9 种文件,它们是:

数据库文件

数据库明细文件

索引文件

程序文件

内存变量文件

报表格式文件

标签格式文件

屏幕格式文件

文本文件

文件名遵守操作系统的文件命名规则,必须以字母开头,可含有字母、数字和下划线,由不超过 8 个字符的标识名加上 3 个字符的类型名(或扩展名)组成。

字母: 26 个英文字母,大、小写均可。

数字: 0～9 阿拉伯数字。

下划线:_

例如: PEOPLE.DBF 是一个有效的文件名。其中 PEOPLE 为文件的标识名;DBF 为类型名,也称为扩展名或文件名后缀,表示文件的类型。

在 FOXBASE + 文件中,数据库文件是最基本的文件,它存放用户关心的数据。在关

系数据库系统中,数据库文件是以表格的形式表现出来的,二维表格的每一竖栏(列)为一个“字段”,每一横行为一个“记录”。每个数据库文件最多有128个字段,最多有10亿个记录。数据库文件的扩展名为DBF。

数据库明细文件是数据库文件的辅助文件。当数据库文件含有明细型字段时,系统自动为此数据库文件生成一个明细文件,用于保存明细型字段中大块的数据。数据库明细文件的扩展名为DBT。

表1.1列出了FOXBASE+支持的各种文件扩展名和它们的简单用途,这些文件的建立方法和详细功能,将在以后章节中介绍。

表 1.1

文件类型	扩展名	用途
数据库文件	.DBF	以二维表格的形式存放用户关心的数据,是数据库系统的最基本文件。
数据库明细文件	.DBT	存放数据库文件中明细型字段的数据,是数据库文件的辅助文件。
索引文件	.NDX .IDX	保存对数据库文件中各记录的索引,以便实现对记录的快速查找。
程序文件	.PRG .FOX	存放用FOXBASE+命令编写的应用程序。经编译后,扩展名改为FOX。
内存变量文件	.MEM	保存定义和使用的内存变量。
报表格式文件	.FRM	存放用REPORT命令输出的报表格式。
标签格式文件	.LBL	存放用LABEL命令打印标签的格式。
屏幕格式文件	.FMT	存放用户定义的屏幕输入/输出格式
文本文件	.TXT	按照SDF(标准数据格式)或定界格式保存数据,以便与高级语言等其它软件交换数据。

1.2.3 变量

FOXBASE+使用两种变量:字段变量和内存变量。字段变量是存在数据库文件(.DBF)中的,每个数据库文件都包含有若干字段变量。

内存变量不同于字段变量,它存在内存之中,独立于数据库文件。它的设置为用户使用数据库系统带来极大的方便。内存变量是一种临时工作单元,通常用来保存中间结果或保存对数据库进行某种分析处理后得到的数据结果,或者用于控制流程,要用时可以随时定义;不用时又可以释放。

FOXBASE+变量由变量名、变量类型和变量宽度三部分组成。

变量名由汉字、字母、数字和下划线组成,最多不超过10个字符。必须以字母或汉字开头。

例,下面是合法的变量名:

CLIEN_ID　姓名　CITY　A1

下面是非法的变量名:

1A　姓　名　ZIP:CODE

需要全本请访问 www.ertongbook.com

FOXBASE + 的字段变量有五种类型,它们分别是:

字符型字段

数字型字段

日期型字段

逻辑型字段

明细型字段

内存变量有四种类型,它们分别是:

字段型内存变量

数字型内存变量

日期型内存变量

逻辑型内存变量

① 字符型 (C 型)

字符型变量的数据值是由可显示的汉字和字符(包括空格)构成的。用字母 C 表示。系统允许一个变量最大宽度为 254 个字符。

字符型变量宽度为该变量最大允许的字符数。

② 数字型 (N 型)

数字型变量的数据值是由正负号、小数点和 0 至 9 的数字构成的,可进行算术运算的数据,用字母 N 表示。

数字型变量宽度为该变量最大整数位、最大小数位之和,小数点和正负号分别算作一个数位。小数宽度为最大小数位。

③ 日期型 (D 型)

日期型变量是用于存放日期的数据。用字母 D 表示。通常采用美国格式: 月/ 日/ 年,其中: 月、日、年均为两位数字。

日期型变量宽度固定为 8 个字节。

④ 逻辑型 (L 型)

逻辑型变量的数值是由逻辑真(.T.)和假(.F.)值所构成的。用字母 L 表示。接受 T、t、Y、y 为逻辑真(.T.)值;接受 F、f、N、n 为逻辑假(.F.)值。

逻辑型变量宽度固定为 1 个字节。

⑤ 明细型 (M 型)

内存变量没有明细型,只有字段变量有明细型。明细型字段变量是用于存放大块的 ASCII 码数据的,数据被存放在数据库文件的辅助文件(.DBT)中。它在数据库文件中用 memo 一词指明,用字母 M 表示。明细型字段变量在数据库文件中的宽度固定为 10 个字节。

明细型字段存放数据的宽度是可变的,最多可达 65536 个字节。如果另一个字处理程序或文本编辑程序被指定 CONFIG.FX 或 CONFIG.DB 文件中的 WP,则明细型字段将具有该字处理程序同样的限制。

1.2.4 表达式

FOXBASE + 使用四种类型的运算符：算术、比较、逻辑和字符串运算符。

1. 算术运算

算术运算符对表达式进行算术运算，产生数值结果。它包括 6 种运算符：

加号 +　　除号/
减号 -　　乘方 * *
乘号 *　　括号( )

运算规则：

先乘除，后加减；乘方优先于乘除；函数优先于乘方，括号最优先；同级运算符自左至右顺序运算。

括号无大、中、小之分，一律用圆括号。圆括号( )内可再套用圆括号( )。

2. 比较运算

比较运算符对两个表达式进行比较运算，产生逻辑型(真、假值)结果。它包括 8 种运算符：

小于 <　　小于或等于 < =
大于 >　　大于或等于 > =
等于 =　　子字符串比较 $
不等于 < > 或 #　　字符串比较 = =

使用比较运算符时，应注意以下几点：

(1) 比较运算符可用在字符、数字和日期型表达式中，但用于比较的两个表达式数据类型必须相同。

(2) 数字型数据是按其数值大小进行比较；字符型数据是按其 ASCII 码值的顺序进行比较；日期型数据是按年、月、日的先后进行比较。

例：

```
. ?123 < =456                                    & & 数字型比较
.T.
. ? 'AB' < 'BB'                                  & & 字符型比较
.T.
. ?CTOD( '01/ 01/ 60') < = CTOD( '01/ 07/ 86')  & & 日期型比较
.T.
```

这里用到显示命令 ?，它的作用是显示表达式的值。显示命令 ? 前的圆点是 FOXBASE + 提示符，表示此行为键入的命令。

(3) 子字符串比较 $。如果 A 和 B 都是字符串，而且 A 与 B 相同或 A 是 B 的子串，则 A $ B 的结果是真值。

例：

```
. ? "this" $ "this is a string"
```

```
.T.
. ?"this"$ "THIS IS A STRING"
.F.
. ?"this is a string"$ "this is a string"
.T.
```

(4) 字符串比较符“==”与比较符“=”不同,它与 EXACT 的状态无关,只有当比较符“==”两边字符串完全相等时,返回结果才为真值。

例:

```
.SET EXACT OFF
. ?"abcd "=="abcd"
.F.
. ?"abcd"=="abcd "
.F.
. ?"abcd"=="abcd "
.T.
.SET EXACT ON
. ?"abcd "=="abcd"
.F.
. ?"abcd"=="abcd "
.F.
. ?"abcd"=="abcd"
.T.
.
```

3. 逻辑运算

逻辑运算符对一个或两个逻辑型表达式进行逻辑运算,产生逻辑型(真、假值)结果。它包括 4 种运算符:

逻辑与　.AND.

逻辑非　.NOT.(或 !)

逻辑或　.OR.

括号　　( )

运算规则:

先逻辑与后逻辑或;逻辑非优先于逻辑与;括号最优先。括号无大、中、小之分,一律用圆括号( ),可以在圆括号内再套用圆括号( )。

各种逻辑运算符运算结果归纳如下:

逻辑与 .AND.

逻辑型表达式 A	逻辑型表达式 B	A.AND.B
T	T	T
T	F	F
F	T	F
T	F	F

逻辑或 .OR.

逻辑型表达式 A	逻辑型表达式 B	A.OR.B
T	T	T
T	F	T
F	T	T
F	F	F

逻辑非 .NOT. (!)

逻辑型表达式 A	! A (.NOT.A)
T	F
F	T

4. 字符串运算

字符串运算符对两个字符型数据进行连接运算,产生一个新的字符型数据。它包括两个运算符:

连接运算符 +

压缩空格运算符 -

字符串运算符 + ,其作用是将两个字符串原样连接成一个新的字符串。

例如:

若: 变量 A 的内容为"this] "

变量 B 的内容为"is] a] string"

则: A + B = "this] is] a] string"

字符串运算符 - ,其作用是将两个字符串连成一个字符串。与" + "不同的是,它将运算符前字符串尾部的空格移加到运算符后字符串的尾部。

例:

若: 变量 A 的内容为: "this] "

变量 B 的内容为: "is] a] string"

则: A - B = "this is] a] string] "

5. 表达式

在 FOXBASE + 中,由运算符和括号将常数、变量和函数连接起来的有意义的式子称为表达式。当不同类型的运算符在同一表达式中出现时,运算规则为:

比较运算符优先于逻辑运算符;算术、字符串运算符优先于比较运算符;括号最优先;同级自左至右顺序运算。括号无大、中、小之分,一律用圆括号( ),可以在圆括号内再套用圆括号。

同变量一样,表达式按运算结果可分为:字符型表达式(C 型)、数字型表达式(N 型)、逻辑型表达式(L 型)和日期型表达式(D 型)。

例:

字符型表达式:

"姓名"+ 姓名

数字型表达式:

88 - 52×63,　　2 * SQRT (4)

逻辑型表达式:

9 < 2, mselect $ "Aal"

日期型表达式:

CTOD ('01/ 01/ 80')

1.2.5 函数

函数用来实现某些特定的运算。FOXBASE + 提供了 77 种函数,从概念来讲,FOXBASE + 函数与数学中的函数没有什么根本区别。注意:

(1) 所有函数必须跟随有圆括号,无论函数是否需要参数(除宏代换函数 & 外)。

(2) 每个函数必须有一个返回值。

(3) 返回值具有一定数据类型。依其类型可以将函数划分为:字符型函数、数字型函数、日期型函数和逻辑型函数。

(4) 传送给函数的参数也有一定的数据类型,必须按要求的数据类型传送参数值。

表 1.2 仅列出返回值为数字型的函数。

表 1.2　FOXBASE + 的数字型函数

函 数 名	形　　式	功　　能
ASC	ASC (〈字符型表达式〉)	求字符串首字符的十进制 ASCII 码值
AT	AT (〈字符型表达式 1〉, 〈字符型表达式 2〉)	子字符串查找
ABS	ABS (〈数字型表达式〉)	求绝对值
EXP	EXP (〈数字型表达式〉)	求指数
INT	INT (〈数字型表达式〉)	求整
LOG	LOG (〈数字型表达式〉)	求自然对数
SQRT	SQRT (〈数字型表达式〉)	求平方根
MAX	MAX (〈数字型表达式 1〉, 〈数字型表达式 2〉)	求最大值
MIN	MIN (〈数字型表达式 1〉, 〈数字型表达式 2〉)	求最小值
ROUND	ROUND (〈数字型表达式 1〉, 〈数字型表达式 2〉)	四舍五入

MOD	MOD(〈数字型表达式1〉,〈数字型表达式2〉)	求模

1.3 赋值命令(STORE)和显示命令(? ??)

在FOXBASE+中,显示命令? ??和赋值命令STORE几乎是使用最多的命令,下面分别介绍。

1.3.1 赋值命令

赋值命令STORE的主要功能是给内存变量提供数据,它可完成以下操作:

(1) 可用于建立内存变量,并赋予内存变量初值。

例:

```
.STORE    100    TO    mnum
100
```

它的作用是:建立内存变量mnum,赋予初值100。

(2) 可用于为已经建立的内存变量重新赋值。

例:

```
.STORE    2*mnum    TO    mnum
200
```

其作用是:为已经建立的内存变量mnum重新赋值。假若内存变量mnum原值为100,则新值将变为200。

(3) 一个STORE命令可以同时建立若干个内存变量或同时为若干个内存变量重新赋值。

例: 建立内存变量ma、mb、mc、md,初始化为0。

```
.STORE    0    TO    ma,  mb,  mc,  md
0
.
```

(4) STORE命令给单个内存变量提供数据时,可以简写成:

〈内存变量〉=〈表达式〉

例:

```
.munm = 10   *   munm
```

完全等价于

```
.STORE  10  *  munm  TO  munm
```

除此之外,STORE还用于为数组赋值。

1.3.2 显示命令? ??

显示命令? ??可以完成以下操作:

需要全本请访问 www.ertongbook.com