

上篇 语言基础

本篇共含 5 章,主要包括以下内容:

第一章在概述数据库的基本概念后,简述了 FoxBASE⁺的语言成份和主要性能。目前国内流行的 FoxBASE⁺有 2.00 和 2.10 两类版本。本书主要介绍 2.10 版,同时说明它与 2.00 版的差别。

第二至第四章依次讲述了 FoxBASE⁺用于建库、修改、查询和统计的各类命令,讨论了用于多库(或区)操作的命令和多库(或区)操作的特点。我们还结合各章内容,在每章均设专节讲一点关系数据库的基本原理,希望通过理论与应用的穿插叙述,更自然地做到“理论结合应用,用理论来指导应用”。这些节在标题前均冠以“*”号,跳过它们不影响对其它内容的学习。

程序设计是贯穿本书上、中、下三篇的一条红线。为使读者及早接触程序设计,本篇末章(第五章)讲述了程序设计的初步知识。我们还在该章介绍了一种“积木式应用程序”。根据编者的教学经验,它能帮助初学者在起步阶段就了解“功能分解”的思想和作法,较快地“进入角色”,并为学习中、下篇程序设计的内容打好基础。

第一章 概 述

计算机的出现,标志着人类开始用机器来存储数据和管理数据。随着信息处理的日益发展,计算机管理数据的方式也不断改进。50 年代末出现了文件管理系统,它把数据组织在一个个独立的数据文件中,实现了“按文件名来访问,按记录进行存取”的管理技术,大大减轻了程序员的数据管理劳动。时至今日,文件管理仍是一般高级语言普遍采用的数据管理方式。在数据量较大的系统中,在数据之间不免存在这样那样的联系,文件系统所采用的那种在文件之间缺乏联系的结构,以及一次至多存取一个记录的访问方式,已不能适应信息处理的需要,到了 60 年代末期,终于在美国诞生了第一个商品化的数据库系统——IMS 系统。

从文件管理系统到数据库系统,标志着数据管理技术的一次飞跃。但直到 80 年代在多数微型机上配置数据库管理系统后,数据库技术才真正得到广泛的应用和普及。本章将先在第一节简介数据库的基本概念,然后在后续各节对 FoxBASE⁺的语言成份、技术指标与启动退出进行概要的综述。

1.1 数据库的基本概念

1.1.1 数据库系统的特点

与文件系统相比,数据库系统具有下列特点:

1. 数据的结构化

在文件系统中,文件之间不存在联系。文件内部的数据一般是有结构的,但从数据的整体来说是没有结构的。一个数据库系统虽然也常常分成许多单独的文件,但同一数据库的文件相互联系,在整体上也服从一定的结构形式,从而更能适应大量数据管理的客观需要。

2. 数据共享

共享是数据库系统的目的,也是它的重要特点。一个数据库中的数据,不仅可以为同一企业或组织的内部各部门共享,还可以为不同组织、地区、甚至不同国家的用户所共享。而在文件系统中,数据一般都是由特定的用户专用的。

3. 数据独立性

在文件系统中,数据结构和应用程序相互依赖,一方的改变总是要影响另一方的改变。数据库系统力求减小这种相互依赖,以实现数据独立性为目标。虽然这一点现在还不能完全做到,但较之文件系统已大有改善。

4. 可控冗余度

数据专用时,每个用户拥有并使用自己的数据,难免有许多数据相互重复,这就是冗余。实现共享后,同一数据库中的数据集中存储,共同使用,因而易于避免重复,减少和控制数据的冗余。

1.1.2 三类数据模型

前面已指出,数据库中的数据从整体来看是有结构的,即所谓数据的结构化。按照实现结构化所采取的不同联系方式,数据库的整体结构可区分为三类数据模型,即(1)层次型;(2)网络型;(3)关系型。其中前两类又合称为“格式化模型”。

早期的数据库系统都采用格式化模型。前述的IMS系统层次型和1969年美国CODASYL委员会提出的DBTG系统(网络型),就是格式化模型的典型代表。1970年,美国E. F. Codd提出了关系模型的概念,首次运用数学方法来研究数据库的结构(把每个数据库文件看作一个关系)和数据操作(看作关系运算),将数据库的设计从以经验为主提高到以理论为指导。不仅如此,关系模型采用人们惯常使用的表格形式为存储结构,易学易用,使它从一开始就吸引了公众的注意,成为广大用户特别是微机用户乐于接受的数据模型。本书讨论的FoxBASE⁺数据库和其它几种常用的微机数据库如dBASE、ORACLE等,也都是关系型数据库系统。

1.1.3 数据库管理系统和数据库应用系统

1. 数据库管理系统

数据库管理系统是指帮助用户建立、使用和管理数据库的软件系统,简称为 DBMS (Data Base Management System)。DBMS 通常有下列三部分组成,即:

- (1) 数据描述语言(DDL) 用来描述数据库的结构,供用户建立数据库;
- (2) 数据操作语言(DML) 供用户对数据库进行数据的查询(包括检索与统计)和存储(包括增加、删除与修改)等操作;
- (3) 其它管理和控制程序 例如安全、通讯控制以及工作日志等公用管理程序。

在微机 DBMS 中,DDL 和 DML 常合二为一,成为一体化的语言。本书介绍的 FoxBASE⁺语言就是这类一体化语言。两种语言都带有自己的翻译程序。与普通高级语言一样,翻译程序也可以区分为编译执行方式和解释执行方式两类。FoxBASE⁺的 DBMS 既有解释型,也有编译型。

2. 数据库应用系统

数据库应用系统是在 DBMS 支持下运行的一类计算机应用系统,简称为 DBAS。在前面加上“数据库”三字,是为了区别于使用普通文件和由文件管理系统支持的应用系统。一个数据库应用系统通常由(1) 数据库;(2) 应用程序;(3) 支持它们的专用或通用的 DBMS 构成。在微机上运行的 DBAS 一般都使用通用的 DBMS(例如 FoxBASE⁺ DBMS),只有数据库和应用程序需要由用户开发。

随着计算机应用从单机到网络的发展,数据库技术也朝网络应用的方向有了新的发展。其中主要的有:(1) 从单用户应用扩展到多用户应用;(2) 从集中式应用扩展到分布式应用。早期的微机 DBMS 都是单用户系统,80 年代中期后陆续扩展了多用户的功能。与此同时,分布式数据库也在实际应用中有了一定程度的发展,并出现了一些用于微机的网络数据库。本书主要讨论单用户数据库,对多用户数据库将在第十三章作简单介绍。

1.2 FoxBASE⁺的语言成份

FoxBASE⁺是美国 Fox Software 公司开发的微机数据库语言,其 1.00 版于 1987 年 2 月问世,仅隔 5 个月(1987 年 7 月)就推出了 2.00 版。与 80 年代国内外流行最广的 dBASE 相比,FoxBASE⁺ 2.00 版一方面与 dBASE Ⅲ 保持兼容,使后者编写的程序不需改动就可在 FoxBASE⁺上运行;同时在功能和性能上又较它有重大的改进,其运行速度高达 dBASE Ⅲ 的 6~7 倍,比编译 dBASE Ⅲ 也快 2 倍左右。1988 年 7 月推出的 2.10 版,其速度更比 2.00 版加快 20%,平均比 dBASE Ⅲ 要快 8 倍。

FoxBASE⁺有单用户和多用户两类版本。现在国内使用的中西文 FoxBASE⁺,是根据西文的多用户版本汉化开发而成的,也有 2.00 和 2.10 两种版本。本节将对中西文 FoxBASE⁺语言的各种成分作概要的综述,包括它的命令、文件、变量,表达式和函数。

1.2.1 认识 FoxBASE⁺——一个引例

为使初学者对 FoxBASE⁺有一感性认识,下面先讲一个简单的例子。

[例 1-1] 表 1.1 是载有某厂设备情况的一张表格。在 FoxBASE⁺的支持下,只用一条命令就能将该表的结构存入计算机,然后输入数据,构成 FoxBASE⁺的一个数据库文

件。该命令的形式是：

CREATE 设备

其中 CREATE 表示命令的操作，“设备”便是要建立的数据库文件的名称。

表 1.1 设备情况表

编号	名称	启用日期	价格	是否主要设备	备注
010-1	仪表车床	06/01/78	3535.10	否	从光华仪表厂调入 91 年 12 月封存
016-1	普通车床	03/05/87	27132.73	是	
016-2	普通车床	12/11/84	6700.30	否	
037-1	平面磨床	01/15/89	241292.12	是	
037-2	外圆磨床	07/21/87	62044.61	是	
038-1	抛光机	10/12/90	15275.00	否	
210-1	马达	05/08/92	2350.25	否	

数据库文件建好后,如果从键盘逐条输入图 1.1 所示的一组命令,屏幕上将依次显示每条命令及其执行结果。图中以圆点开始的行为命令行,其作用见()中的简释;无圆点的行为 FoxBASE+ 给出的反馈信息。圆点为 FoxBASE+ 可以接受命令的提示符,类似 DOS 的“>”。

```
.use 设备                                (打开名为“设备”的数据库文件)
.list                                    (列表显示当前数据库文件内容)
Record#  编号  名称  启用日期  价格  主要设备  备注
1 010-1  仪表车床  06/01/78  3535.10  .F.      Memo
2 016-1  普通车床  03/05/87  27132.73  .T.      Memo
3 016-2  普通车床  12/11/84  6700.30  .F.      Memo
4 037-1  平面磨床  01/15/89  241292.12  .T.      Memo
5 037-2  外圆磨床  07/21/87  62044.61  .T.      Memo
6 038-1  抛光机    10/12/90  15275.00  .F.      Memo
7 210-1  马达      05/08/92  2350.25  .F.      Memo
.copy to 主设备 fields 名称,启用日期,价格 for 主要设备  (将主要设备复制到“主设备”文件)
3 records copied

.use 主设备                                (打开“主设备”文件)
.list
Record#  名称  启用日期  价格  (显示其内容)
1 普通车床  03/05/87  27132.73
2 平面磨床  01/15/89  241292.12
3 外圆磨床  07/21/87  62044.61
.replace all 价格 with 1.2 * 价格          (将所有价格上调 20%)
3 replacements

.list
Record#  名称  启用日期  价格  (显示其内容)
1 普通车床  03/05/87  32559.28
```

2 平面磨床	01/15/89	289550.54
3 外圆磨床	07/21/87	74453.53

图 1.1 FoxBASE+的若干命令和执行情况

在图 1.1 中,执行 LIST 命令后所显示的便是数据库文件的内容。它除了没有表格线;以及第 5 栏中的“是”、“否”改用 .T. 和 .F.,第 6 栏的备注内容均用 Memo 代替外,和表 1.1 完全相同。还需指出,在诸如 FORTRAN、PASCAL 等高级语言中,要完成该图所示的列表显示文件内容、复制文件的部分字段和记录、或更改所有记录中某个字段的数据等操作,常常要执行一段程序。而 FoxBASE+ 每次仅用一条命令(如 LIST、COPY 和 REPLACE)就可完成这些较复杂的操作。数据库语言所以受到用户欢迎,采用这种命令式的语言是一个主要的原因。

1.2.2 FoxBASE+的命令

在上述引例中,已经用到若干种 FoxBASE+ 的命令。其中有:

LIST

USE <文件名>

REPLACE <范围><字段名> WITH <表达式>

COPY TO <文件名> FIELDS <字段名> FOR <条件>

等等。一般地说,FoxBASE+ 的命令总是由一个称为命令字的动词开头,后随若干功能子句或短语,用来说明命令的操作对象、操作结果与操作条件。这些命令形式规范,言简意赅,一句可顶一般高级语言许多句。由于命令中只提对操作的要求,不描述具体的操作过程,所以又称为“非过程化”语言,以区别于一般高级语言的“过程化”语言。下面再说明几点:

1. 命令的功能子句

大多数命令中含有功能子句,其中有些是必有的,有些是供选用的,后者又称为可选项。在本书后文介绍命令格式时,可选项将一律用[]括住,以便识别。例如在上述的 REPLACE 和 COPY 命令中,共有 5 个功能子句,即:<范围>、FIELDS<字段名表>、FOR<条件>、TO<文件名>和<字段名> WITH <表达式>。其中前三项是可选的,后二项是必有的。所以它们的命令格式可分别写成:

REPLACE [

COPY TO <文件名> [[FIELDS]<字段名表>] [FOR <条件>]

注意 FIELDS 子句中的[FIELDS]也是可选的,在书写时可以省去,只写出需要复制的字段名即可。

功能子句的种类很多,有些是在某些命令中专用的,例如“<字段名> WITH <表达式>”仅在 REPLACE 等个别命令中使用。有些子句则使用很广,在许多命令中都被用作可选项。以下对部分常用的可选项作一些说明:

(1) 范围子句 用于指定命令可处理的记录范围,它共有 4 种形式,其写法和含义是:

ALL 代表数据库文件的全部记录

RECORD N 表示库文件中第 N 个记录(N 可为表达式,下同)

NEXT N 表示从当前记录开始的 N 个记录

REST 表示从当前记录开始直到库文件尾的所有记录

(2) FIELDS 子句 用于规定命令可处理的字段。如果说〈范围〉子句是在水平方向对库文件数据所进行的限制,则本子句是在垂直方向对库文件数据的限制。它也有两种形式:

[FIELDS] 〈字段名表〉

[FIELDS] 〈表达式表〉

后一形式允许把字段名扩充为表达式。例如:

LIST 名称,1.2* 价格

就表示只列表显示数据库文件中的名称和价格两个字段,且其中价格字段需把库文件中的数据乘上 1.2 后再显示。

(3) FOR 子句和 WHILE 子句 用于把命令的操作限制于符合子句中〈条件〉的记录。这两种子句的差别是,FOR 子句能在整个库文件中筛选出符合条件的记录;而使用 WHILE 子句时,先顺序寻找出第一个满足条件的记录,再继续找出后随的也满足条件的记录。一旦发现有一个记录不满足条件,就不再往下寻找。例如,若将图 1.1 中 COPY 命令使用的 FOR 子句改成 WHILE 子句,写作:

COPY TO 主设备 FIELDS 名称,启用日期,价格 WHILE 主要设备

则复制到主设备文件中的记录一个也没有。这是因为,库文件中的第一个记录就因不满足条件而被舍弃,虽然其后的第二、四、五等记录都满足主要设备的条件,但因查找已经停止,当然也不再复制了。

FOR 子句和 WHILE 子句同时使用时,WHILE 子句将优先处理。

2. 命令的书写规则

为了简化键盘输入,FoxBASE+ 允许命令字和功能子句中的功能字使用缩写的形式,只要求写出这些字的至少前 4 个字母。例如:

REPLACE 可写作 REPL,REPLA,REPLAC 或 REPLACE

WHILE 可写作 WHIL 或 WHILE

FIELDS 可写作 FIEL,FIELD 或 FIELDS

等等。此外,一条命令中含有多个功能子句时,子句的书写次序无关紧要。例如以下两种写法:

COPY TO 主设备 FIEL 名称,日期,价格 WHIL 主要设备

COPY FIEL 名称,日期,价格 TO 主设备 WHIL 主要设备

其功能是等效的。

一条命令最多可包含 256 个字符,一行写不下可分行书写,并在分行处加一分号“;”。

例如:

COPY FIEL 名称,日期,价格;

TO 主设备 WHILE 主要设备

3. 命令的执行方式

FoxBASE+的命令具有(1)交互执行和(2)程序执行两种执行方式。前者当用户从键盘送入命令后,按下“回车”键命令随即执行;后者则将需要执行的命令编成程序,存入一个程序文件中,然后发有关命令执行程序文件。FoxBASE+共有 170 余种命令,其中除少数几种只能用于程序执行方式外(参阅第五章),大多数命令均适用于两种执行方式。

顺便指出,FoxBASE+的程序也有两种执行方式,即(1)解释执行和(2)编译执行。FoxBASE+程序的编译是由一个称为 FOXPCOMP.EXE 的准编译(或伪编译)程序来完成的,但编译后所产生的是一种特殊代码文件,仍须由 FoxBASE+解释执行,不能够独立运行。尽管如此,编译执行仍比解释执行具有下列的优点:

(1) 缩短了程序长度。程序中的注释和为保持程序清晰而插入的空格,在编译中将全被删除;程序缩短了,占用的磁盘空间也可以节省;

(2) 提高了程序的装入速度和执行速度。这是因为解释执行时,程序一边从磁盘装入内存,一边要翻译为内部代码,编译执行时这项工作已预先在编译阶段完成,装入时的速度就快多了。加上编译时要对程序进行优化,所以程序执行速度也有所提高;

(3) 有利于程序的保密。源程序在编译后变成特殊代码,既不能读懂,也不能修改,从而达到了程序的保密和安全。

1.2.3 FoxBASE+ 的文件

在 FoxBASE+中,所有的数据和程序都是用文件的形式在磁盘上存储的。按照不同的用途,FoxBASE+的文件可划分为 9 种类型。表 1.2 列出了它们的名称,用途和默认扩展名。

表 1.2 FoxBASE+ 的文件类型

序号	名 称	默认扩展名	用 途
1	数据文件	.DBF	存放数据库的结构和数据,是 FoxBASE+ 中最重要的文件
2	备注文件	.DBT	存放 DBF 文件中备注型字段的内容,是 DBF 文件的辅助文件
3	索引文件	.IDX	存放 DBF 文件记录的索引,以实现记录的快速查询
4	文本文件	.TXT	不存结构只存数据的文件,用于同其它语言的数据交换
5	内存变量文件	.MEM	用于保存内存变量,以备需要时使用
6	程序文件	.PRG(未编译) .FOX(编译后)	又称命令文件,用于存放用 FoxBASE+ 语言编写的程序
7	屏幕格式文件	.FMT(未编译) .FMX(编译后)	用于记录用户定义的屏幕输入格式,也可看作一种特殊的程序文件
8	报表格式文件	.FRM	用于规定 REPORT 命令输出的数据报表格式
9	标签格式文件	.LBL	用于规定 LABEL 命令输出的标签项目与格式

在 9 种文件中,最主要的是数据库文件和程序文件。前者用于存放数据库应用程序的数据,后者用于存放数据库应用系统的程序。表中序号 2~4 的 3 种文件可视为数据库文件的派生文件,序号 7~9 的 3 种文件可视为程序文件的派生文件。

所有文件都是按名访问的。文件名由不超过 8 位且不含空格的字符串构成,扩展名则由不超过 3 位且不含空格的字符串构成。扩展名主要用来区分文件的类型,除非用户另有指定,在文件建立时系统将自动给它们加上默认的扩展名。程序文件或屏幕格式文件各有 2 种默认扩展名,一种用于没有编译过的文件,另一种用于经过准编译程序编译后所产生的文件。

1.2.4 FoxBASE⁺的变量

与其它高级语言相似,FoxBASE⁺的数据也可区分为常量和变量。变量是指内容可变的数据。FoxBASE⁺规定,变量的名称最长为 10 个字符,其中可包括字母、汉字、数字和下划线,但必须以字母或汉字开头。

FoxBASE⁺有两类变量,即字段变量和内存变量。

1. 字段变量

这是随数据库文件的建立而定义的一类变量。数据库文件的每个字段对应于一个字段变量,例如图 1.1 中的“设备”数据库就含有 6 个字段变量。

FoxBASE⁺的字段变量有字符(C)、数值(N)、日期(D)、逻辑(L)和备注(M)等 5 种类型,详见第 2 章第 2.1.1 节。在上述数据库文件的字段变量中,编号与名称为 C 型,价格为 N 型,启用日期为 D 型,主要设备为 L 型,备注为 M 型。

字段变量属多值变量。一个数据库文件有多少条记录,其字段变量就具有多少个值。

2. 内存变量

内存变量用于存储在命令或程序执行中临时用到的输入、输出或中间数据,它独立于数据库文件而存在。内存变量也有 5 种类型。其中有 4 种(C、N、D、L)与字段变量相同,另一种称为屏幕(S)型,专用于存储当前屏幕的整幅信息。

FoxBASE⁺也是第一个把数组引入内存变量的微机数据库语言,并因此提高了它的数值运算和数据处理能力。有关数组的详细内容,请参阅第 7.2 节。

内存变量通常用赋值命令定义后才能使用,但也可直接通过带有“TO<内存变量>”子句的命令来建立。现举例说明如下:

[例 1-2] 用赋值命令建立内存变量

```
. 设备价=10000
10000
.STORE 9000 TO 优惠价,批发价
9000
.? 设备价,优惠价,批发价
      10000      9000      9000
. 折扣=设备价-优惠价
      1000
.? 折扣
      1000
.STORE “车床” TO 设备名
```

```

      车床
. ? LEN (设备名)
      4
. 设备名="外圆磨床"
      外圆磨床
. ? LEN(设备名)
      8

```

说明:

(1) 用赋值语句“<变量名>=<表达式>”和 STORE 命令均可为内存变量赋值。前者一次只能为一个变量赋值,后者一次可以为多个变量赋值。

(2) “?<表达式表>”用于显示命令中各个表达式的值。“?”可理解为“*What is/are*……”,故“? 名称”意为显示变量“名称”的内容。

(3) LEN () 为 FoxBASE+ 的一种函数,用于计算 C 型变量的字符串长度。

本例表明,内存变量的类型和长度随所赋数据的类型和长度而定,例如变量“设备名”首次赋值后长度为 4,再次赋值后长度就变为 8 了(一个汉字要占 2 个字符的位置)。

通过把内存变量作为临时存储单元的有关命令,可以直接定义内存变量。这类命令也有许多种,后文将陆续介绍。下例将用到其中的两种。

[例 1-3] 试求表 1.1 中主要设备的平均价格。

有一种计算的步骤是:(1) 打开“主设备”数据库文件;(2) 计算价格总额(TOTAL);(3) 统计设备台数(NUM);(4) 平均价格=价格总额/台数。用 FoxBASE+ 来实现,可使用以下的命令序列:

```

. USE 主设备           &&. 打开“主设备”库文件
. SUM 价格 TO TOTAL    &&. 对“价格”字段求和,存入变量 TOTAL
. COUNT TO NUM         &&. 将“主设备”文件的记录数存入变量 NUM
. AV_PRICE=TOTAL/NUM   &&. 计算平均价格并存入 AV_PRICE

```

本例共建立 3 种内存变量,其中只有 AV_PRICE 是通过赋值语句建立的,TOTAL 和 NUM 则是在执行 SUM 和 COUNT 命令时建立的,并分别用于存储这两条命令的统计结果。两条命令都带有 TO <内存变量名>形式的功能子句,能直接为内存变量赋值,不必再用例如 TOTAL=0,NUM=0 一类的赋值命令专为变量定义和赋初值。

正如 LIST 命令能够显示数据库文件的内容一样,使用 LIST MEMORY 命令可以显示内存变量的全部内容。本例建立的三种内存变量,如果列表显示,则在屏幕上可见:

```

. list memory
TOTAL      Pub   N      39656 3.35  ( 396563.35000000)
NUM        Pub   N          3 (      3.00000000)
AV_PRICE    Pub   N     132187.78 ( 132187.78333333)

  3 variables defined,      0      bytes used
253 variables available,  6000    bytes available

```

其中 Pub 是 Public 的缩写,表示这些内存变量为“全局”变量,N 表示变量为数值型,末 2 行分别指出内存中已有的变量数和占用字节数,以及尚可定义的变量和字节数。

3. 两类变量的差异

表 1.3 中列出了字段变量和内存变量的主要差异,作为本节的小结。

表 1.3 字段变量和内存变量的比较

字 段 变 量	内 存 变 量
1. 是数据库文件的组成部分	1. 独立于数据库文件而存在
2. 随数据库文件的定义而建立	2. 使用时随时建立,可用赋值命令或带有 TO(内存变量)子句的命令
3. 有 C,N,D,L,M 5 种类型	3. 有 C,N,D,L,S 5 种类型
4. 多值变量	4. 单值变量
5. 关机后保存在数据库文件中	5. 除非预先存入内存变量文件,关机后不保存

前已说明,两类变量的命名规则是一样的。但如果内存变量与数据库文件的字段同名,系统将不加区别地都按字段变量对待。为此,如果两类变量中出现同名变量,通常在内存变量名前冠以字母“M”,以便区分。

1.2.5 表达式和函数

前文已数次提到过表达式和函数,它们都是 FoxBASE⁺语言的重要成分,现分述如下:

1. 表达式

表达式由常量、变量或函数用运算符与圆括号连接而成。例如例 1-3 中的 TOTAL/NUM 就是表达式的一个例子。

大多数表达式含有一至多个运算符。FoxBASE⁺的运算符有算术运算符、字串运算符,关系运算符和逻辑运算符 4 类,详见表 1.4。

表 1.4 FoxBASE⁺的运算符

优先次序	分类	运算符种类与优先次序
先 ↓ 后	算术运算符	(), → *, / → * * 或 ^ → +, - 括号, 乘, 除 乘方 加, 减
	字串运算符	+, - (注) 二串相并, 二串相并, 前串尾端空格移到新串末端
	关系运算符	<, <=, =, >=, >, (注) <> (不等于), \$ (前串包含在后串内)
	逻辑运算符	() → .NOT. 或 ! → .AND. → .OR. 括号 非 与 或

注: 字串运算符和关系运算符无优先次序, 先出现先执行。

相应于上述 4 类运算符,可组成 4 种类型的表达式。其中算术表达式和字串表达式用于进行运算,关系表达式用于构成 FOR/WHILE 功能子句中的条件,而逻辑表达式则可用于把关系表达式组成复合条件。例如:

[例 1-4] FoxBASE⁺表达式示例

```
.B=20
20
.C=25
25
.D=C/B
```

```

.L1=C=B
.F.
.AA="车床"
车床
.BB="六角"
六角
.CC=BB+AA
六角车床
.L2=AA$CC
.T.
.? L2.AND.L1
.F.
.? .NOT. (L2.AND.L1)
.T.
.? .NOT.L2.AND.L1
.F.
2. 函数

```

函数是 FoxBASE+ 语言的又一重要组成部分,通常取〈函数名〉()的形式。FoxBASE+ 有上百种标准函数;用来支持各种运算,检测系统状态,或作出某种判断。适当地运用这些函数,能增强和扩展命令与程序的功能,减少编写程序的工作量。

每种函数都具有某种操作功能,所以它也有操作数和结果数,称为自变量和返回值。必须强调,函数总是只返回一个值。因此函数也可看作具有一定类型的数据,可以与常量或变量一样构成表达式的组成部分。

从实质上说,函数其实就是预先编好的子程序。调用一个函数实际上就是执行一个函数子程序。FoxBASE+ 还允许用户编写自己的函数子程序,称为“用户自定义函数”。标准函数名也可缩写为前 4 个字母,自定义函数须写全名。

为便于叙述,以下把 FoxBASE+ 的函数按功能划分为 5 类,按类作简要介绍并举例说明。对常用的函数,下文将结合各章内容陆续介绍。表 1.5 列出了各类函数的名称。附录二列出了每种函数的格式和简要功能。前者按功能类排列,后者按函数名的字母顺序排列。

(1) 数值运算函数

包括数值计算,数值比较,以及将非数值型数据转换为数值型数据的各种函数,如 ABS()、INT()、SQRT()、MAX()、MIN()、VAL() 等,共有 11 种。它们的功能与其它计算机语言的同类函数相似,就不举例说明了。

(2) 日期和时间函数

包括查年月日、查时间、查星期几,以及日期转换与比较等函数,共 16 种,其中 MAX()、MIN() 两种函数也可用于数值型,如不算它们便是 14 种。

[例 1-5] 日期/时间函数举例

```

.串1="04/20/93"
04/20/93
.日期1=CTOD(串1)
04/20/93
. ? YEAR(日期1)

```

1993
 . ? CMONTH(日期 1)
 April
 . ? DAY(日期 1)
 20
 . ? CDOW(日期 1)
 Tuesday
 . 日期 2=CTOD("05/01/92")
 05/01/92
 . ? MAX(日期 1,日期 2)
 04/20/93
 . ? MIN(日期 1,日期 2)
 05/01/92

表 1.5 FoxBASE+ 2.00 版函数一览表

功能分类	函 数 名 称
数值运算	数值计算 ABS,EXP,INT,LOG,MOD,ROUND,SQRT
	数值转换 ASC,VAL
	数值比较 MAX,MIN
日期时间	年 月 日 DATE,DAY,MONTH,CMONTH,YEAR,SYS(1)
	星期 DOW,CDOW
	时间 TIME,SYS(2)
	日期转换 CTOD,DTOC,SYS(10),SYS(11)
字符串操作	字符串检测 LEN
	字符串转换 &.,CHR,STR,LOWER,UPPER,TRANSFORM,SYS(15)
	字符串生成 REPLICATE,SPACE
	子串操作 AT,LEFT,RIGHT,TRIM,LTRIM,RTRIM,STUFF,SUBSTR
状态判断	文件状态 FILE,FLOCK
	记录状态 BOF,EOF,DELETED,FOUND,LOCK,RLOCK
	字符串状态 ISALPHA,ISLOWER,ISUPPER
	显示器状态 ISCOLOR
	READ 状态 UPDATED
状态参数测试	立即 IF 函数 IIF
	数据库文件 DBF,ALIAS,FIELD,FCOUNT,RECCOUNT,RECNO,RECSIZE,LUPDATE
	其它文件 NDX,SYS(3),SYS(7),SYS(14),SYS(16)
	输入键测试 INKEY,READKEY
	功能键测试 FKLABEL,FKMAX
	显示器测试 COL,ROW,SYS(100),SYS(101),SYS(103)
	打印机测试 PCOL,PROW,SYS(6),SYS(13),SYS(102)
	磁盘状态 DISKSPACE,SYS(5)
	内存/工作区状态 SYS(12),SELECT
	出错状态 ERROR,MESSAGE
	系统状态 OS,VERSION,GETENV,SYS(0),SYS(9),SYS(17)
	表达式类型 TYPE

也可用于比较日期。

(3) 字符串操作函数

包括字符串的转换、生成、测试和子字符串操作等函数,共 18 种。例如

[例 1-6] 字符串操作函数举例

```
. WA="mastership"
mastership
. ? LEFT(WA,6)
master
. ? RIGHT(WA,4)
ship
. WB=SUBSTR(WA,2,1)
a
. WC=UPPER(WB)+SPACE(1);
      +RIGHT(WA,4)+SPACE(1);
      +LEFT(WA,6)
A ship master
. ? LEN(WC)
13
```

(4) 状态判断函数

包括对文件状态、记录状态、以及字符串首字符状态作判断的 13 种函数,如 FILE()、EOF()、FOUND()、DELETED()、RLOCK()、ISALPHA()等。这类函数主要用在程序中,其返回值都是逻辑型,非“真”即“假”。例如

[例 1-7] 状态判断函数举例

```
. FILE(设备)
.T.
. USE 设备
. ? BOF(设备)
.F.
. ? WC
A ship master
. ISALPHA(WC)
.T.
. ISUPPER(WC)
.T.
```

(5) 状态参数测试函数

包括对系统的文件、输入键、功能键、显示器、打印机、内外存储器、以及系统软件 and 状态设置等各类参数进行测试的函数。这类函数的数量最多,功能多样,FoxBASE+ 2.00 版有 40 余种,2.10 版又加 6 种。因为它们大都在程序中使用,在本书中篇将结合程序设计摘要介绍,这里就暂时不讲了。

1.3 FoxBASE+ 系统概况

上节巡视了 FoxBASE+ 的语言成分,通过对每种成分的初步解剖,使读者对它的内部构成有一大概的了解。本节将从整个系统出发,继续介绍 FoxBASE+ 的软件配置、主要技术指标和运行环境。换言之,上节是从内部构成对它作一番浏览,而本节则将从外特性的角度对这一系统进行整体的考察。

1.3.1 FoxBASE+ 的软件配置

FoxBASE+ 2.00 有单用户和多用户两种版本,它们各包含 5 个程序和 1 个文件,如表 1.6 所示。表中前两项为主要成分:其中执行程序是经常使用的,在 FoxBASE+ 启动后全部驻留内存。而覆盖程序在 FoxBASE+ 启动时仅有一部分被装入内存,其余的要到需用时才临时调入;而且当调入新内容时,原在内存暂不使用的那部分程序被覆盖。这就是取名覆盖程序的由来。

表 1.6 FoxBASE+ 的软件配置

文 件 种 类		文 件 名 称	
		单用户版本	多用户版本
主要成分	① 执行程序	FOXPLUS.EXE	MFOXPLUS.EXE
	② 覆盖程序	FOXPLUS.OVL	MFOXPLUS.OVL
辅助成分	③ 准编译程序	FOXPCOMP.EXE	
	④ 过程组合程序	FOXBIND.EXE	
	⑤ 帮助程序	FOXPHelp.HLP	
	⑥ 欧洲文字基准文件	EUROPEAN.MEM	

在 4 种辅助成分中,准编译程序用于编译 FoxBASE+ 的程序文件。如前所述,该程序只能产生一种中间代码文件,这种代码不能独立运行,所以称为“准”编译或“伪”编译程序。过程组合程序用于将若干较小的、分散的程序文件合并为一个较大的过程文件,借以提高运行速度和方便管理(以上均见第 7.4 节)。帮助程序则用来提供联机帮助信息,以帮助用户正确使用 FoxBASE+。最后一个文件——欧洲文字基准文件是专为法、德、意等欧洲文字的用户设置的一个专用内存变量文件,不使用这些文字的用户不需要用它。

在 FoxBASE+ 2.10 版中,帮助程序被扩展为一个称为 Foxcentral (FOX 控制中心)的用户接口。通过这一接口,还可以进一步调用一组新的附加程序,如 FoxView, FoxCode, FoxGen, FoxDoc, FoxGraph 等。这些附加程序的作用主要是改善 FoxBASE+ 的程序设计环境,使用户不需要编程就能处理大部分数据操作,甚至自动生成完整的 FoxBASE+ 应用程序。因此,有时也把这些附加程序统称为非编程环境工具。

2.00 版的软件通常安装在 2 张系统盘(5 英寸软盘)上,其中一张存储表 1.5 中①、③、④三种软件,另一张存储其余的三种软件(②、⑤、⑥)。2.10 版的系统盘扩充为 7~10 张软盘。

1.3.2 FoxBASE+ 的主要技术指标

本节将说明系统各类主要指标所允许的最大参数值和默认参数值。

1. 关于数据库文件

记录数 ≤10 亿 单个记录长度 ≤4000 字节

字段数	≤128	单个文件长度	≤20 亿字节
最大字段宽度:		定长字段宽度:	
N 型	19 字节	D 型	8 字节
C 型	254 字节	L 型	1 字节
M 型	64K 字节		

2. 关于其它文件

过程文件中的过程数	≤128
程序文件中的变量数	≤2048
屏幕格式文件中的屏幕页数	≤128

3. 可同时开打的文件数

总数: 最多 48 个 默认 16 个

其中:

数据库文件数	最多 10 个
索引文件数	最多 21 个

其中同一数据库文件的索引文件 最多 7 个

4. 关于内存变量

总项数:	C 型变量长度
最多 3600	最长 64K 字节
默认 256	默认 6K 字节
单个数组项数: ≤3600	N 型变量精度 不超过 16 位

1. 3. 3 FoxBASE+的运行环境

FoxBASE+能在多种中西文计算机环境中运行。从 IBM-PC/XT,AT,286,386 等微型计算机,到国产的长城、东海、浪潮等系列微型机,以及它们的兼容机种,都可以运行 FoxBASE+。

充足的内存空间,是 FoxBASE+发挥最佳性能、提高运行速度的重要因素。西文 FoxBASE+2.00 版至少要求有 360kB 的内存空间,中西文 FoxBASE+至少要求 640kB 的内存空间。实际上在这样紧缩的容量下,FoxBASE+的许多功能都受到制约。当机器中带有 8087、80287 等协处理器芯片,或换用强大的 80386 为主处理器芯片后,可以明显提高数据处理的速度。但如果没有充足的内存,就不能发挥这些芯片的潜力。举例来说,如果在 386 微型机上运行 FoxBASE+/386,内存空间至少要在 2MB 以上,而且多多益善。

运行 FoxBASE+的计算机最好配置一台硬盘驱动器。太小的磁盘空间也会影响运行速度,例如随着磁盘空间的减少,给程序分配新的空间会被迫花费越来越长的查找时间。

MS-DOS、CCDOS、OS-2、UNIX、XENIX 等操作系统都可支持 FoxBASE+的运行。如果使用 DOS 操作系统,单用户 FoxBASE+要求 DOS 的版本在 2.0 版以上,多用户 MFoxBASE+要求 DOS 的版本不低于 3.10 版。

1.4 FoxBASE+的安装、启动和退出

如前所述,FoxBASE+ 2.00 版有 2 张系统盘,FoxBASE+ 2.10 版有 10 张系统盘。运行 FoxBASE+ 以前,通常都先把系统盘上的文件全部复制到硬盘上,以后直接由硬盘启动。安装 2.10 版的 10 片系统盘上的全部软件,大约要占用 2.2MB 的硬盘空间。

启动 FoxBASE+ 有 2 种方式。以中西文 FoxBASE+ 2.00 版为例。

第一种方式是在操作系统提示符之后键入:

MFoxPLUS

启动后,屏幕上显示出 FoxBASE+ 的圆点提示符,表示系统进入交互执行状态,等待用户键入要执行的命令。

第二种方式是键入:

MFoxPLUS <程序文件名> [WITH <参数表>]

启动后系统立即执行<程序文件名>所指定的 FoxBASE+ 程序。

如果使用的是单用户版本,则上述两处的 MFoxPLUS 均改为 FoxPLUS。

如果 FoxBASE+ 没有安装在硬盘上,可以将装有 FoxPLUS 或 MFoxPLUS 的系统盘插入驱动器 A,并改由软盘启动。此时用户可键入:

A: MFOXPLUS 或 A: FOXPLUS

当系统在屏幕上提示:

Can't find file MFOXPLUS.OVL(或 FOXPLUS.OVL)时,再将第二次插入的盘取出,换上装有 .OVL 程序的系统盘,重新键入 A:,启动即可成功。

FoxBASE+ 的退出命令是 QUIT,执行 QUIT 时,系统将关闭所有文件,然后退出 FoxBASE+,屏幕上重新出现操作系统的提示符。

习 题

- 一、与普通文件系统相比,数据库系统有哪些特点?
- 二、数据库有哪几种常用的数据模型? FoxBASE+ 属于哪一类?
- 三、什么是数据库管理系统? 简要说明它的组成。
- 四、与 PASCAL 等一般高级语言相比,FoxBASE+ 的命令有何特点?
- 五、FoxBASE+ 的命令有几种执行方式?
- 六、什么是数据库文件? FoxBASE+ 的数据库文件可容纳多少个记录? 每个记录可包括多少字段?
- 七、FoxBASE+ 的文件有哪些类型?
- 八、说明记录变量和内存变量的意义和区别。
- 九、FoxBASE+ 有多少标准函数? 按照函数的功能,可划分为哪些种类?
- 十、与 dBASE 相比,FoxBASE+ 的主要优点是什么?

第二章 数据库文件的建立与维护

开发一个数据库应用系统,首先需要建立数据库。数据库由一个或多个数据库文件组成。本章将围绕数据库文件,着重讨论数据库文件的建立以及修改和复制数据库文件的各种方法。章末简介了关系数据库的基本概念,供有兴趣的读者阅读。

2.1 数据库文件的建立

表 2.1 是描述工厂设备基本情况的二维表,这是在日常管理中广泛使用的一种形式。其中每一列指明设备的一种属性,属性的名称如编号、名称、价格等在每列的顶端给出,每一行则给出某一种设备的具体属性值。

表 2.1 设备基本情况表

编号	名称	启用日期	价格	使用年限	部门	主要设备	在役状况	备注
010-1	仪表车床	06/01/78	3535.10	20	一车间	否	租入	从光华仪表厂租入
016-1	普通车床	03/05/87	27132.73	15	二车间	是	在役	
016-2	普通车床	12/11/84	6700.30	18	一车间	否	在役	
037-1	平面磨床	01/15/89	241292.12	18	二车间	是	封存	1991年12月封存
037-2	外圆磨床	07/21/87	62044.61	18	三车间	是	在役	
038-1	抛光机	10/12/90	15275.00	20	三车间	否	在役	
210-1	马达	05/08/92	2350.25	8	动力科	否	在役	

数据库文件(简称库文件,下同)是 FoxBASE+ 用于储存数据的主要形式,一个库文件对应一个二维表,表的一行(除顶端行)对应库文件的一个记录,表的一列对应库文件的一个字段。

一个库文件由“文件结构”与“记录数据”两部分组成,要生成新的库文件必须从定义库文件的结构开始。

2.1.1 定义文件的结构

定义库文件的结构就是要指明该文件包含多少个字段,并对每个字段指明字段名、字段类型、字段宽度与小数位数等四个结构参数。

1. 四个结构参数