

计·算·机·等·级·考·试·从·书

98 大纲

谭浩强 主编

史济民 史 令 编著

FoxBASE+ 程序设计(二级)教程



特

邀计算机教育专家编写



有

配套辅导用书和样题汇编



用

于考试前系统的学习与培训



清华大学出版社

<http://www.tup.tsinghua.edu.cn>

(京)新登字 158 号

内 容 简 介

本书是根据教育部考试中心 1998 年制定的《全国计算机等级考试 考试大纲》中对二级数据库语言程序设计部分的要求编写而成的,主要介绍数据库系统及 FoxBASE+ 语言的基础知识、基本操作和程序设计。全书对考试大纲所要求的内容结合例题进行了深入浅出的剖析与说明,既考虑到数据库知识的系统性,也照顾了应试读者的需要。

本书可作为等级考试的培训教材,也可供应试人员或数据库初学者自学使用。

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+ 程序设计(二级)教程

作 者: 史济民等

出版者: 清华大学出版社(北京清华大学学研楼,邮编 100084)

[http:// www. tup. tsinghua. edu. cn](http://www.tup.tsinghua.edu.cn)

印刷者: 密云胶印厂

发行者: 新华书店总店北京发行所

开 本: 787×1092 1/16 印张: 11.5 字数: 270 千字

版 次: 2000 年 1 月第 1 版 2000 年 1 月第 1 次印刷

书 号: ISBN 7-302-02212-7/TP·2191

印 数: 0001~5000

定 价: 15.00 元

前 言

从 1999 年 4 月启用的《全国计算机等级考试 考试大纲》规定二级考试的考生可在 FoxBASE⁺、QBASIC 等五种语言中任选一种。本书就是按照新大纲中“数据库语言程序设计”部分的考试大纲编写的,可供参加上述等级考试的考生用作教材,也可供 FoxBASE⁺的初学者自学使用。

前几年由编者主编的《FoxBASE⁺ 及其应用系统开发》(清华大学出版社,1994)一书,由于内容实用,多年来有许多省市指定为全国计算机等级考试参考教程,至 1999 年 4 月已累计印刷 70 余万册。但该书覆盖面较宽,从语言基础、程序设计一直讲到系统开发。为了更好地适应等级考试考生的备考需要,我们另编了这本教程。与前一本相比,它具有以下特点:

(1) 删去了超出考试大纲要求的数据库系统开发等内容,保留了语言基础与程序设计两个部分,使篇幅更加紧凑。

(2) 各章末以小结的形式列出该章的重点,并提供适量的习题,使之与新的考试大纲有更强的针对性。

(3) 书末附录提供了 FoxBASE⁺新的考试大纲、笔试试题分析和上机考试题型示例,可供应考读者参考。

本书第 1~5 章由史令执笔,第 6 章由史济民执笔,全书由史济民统稿。不足之处,请读者不吝赐教。

编 者

1999 年 8 月于上海

目 录

第 1 章 概述	(1)
1.1 数据库的基本概念	(1)
1.1.1 数据库系统的特点	(1)
1.1.2 数据模型	(2)
1.1.3 数据库管理系统和数据库应用系统	(3)
1.2 FoxBASE+语言简介	(3)
1.2.1 引例	(4)
1.2.2 命令	(5)
1.2.3 变量和表达式	(8)
1.2.4 函数.....	(11)
1.2.5 文件.....	(15)
1.3 FoxBASE+系统概述	(16)
1.3.1 软件配置.....	(16)
1.3.2 主要技术指标.....	(17)
1.3.3 运行环境.....	(17)
1.3.4 工作方式.....	(18)
1.4 FoxBASE+的安装、启动和退出	(18)
1.5 小结.....	(19)
第 2 章 数据库的建立与维护	(22)
2.1 建立数据库文件.....	(22)
2.1.1 定义库文件结构.....	(22)
2.1.2 输入记录数据.....	(26)
2.1.3 全屏幕编辑.....	(27)
2.2 显示或修改数据库文件.....	(28)
2.2.1 显示和修改库文件结构.....	(29)
2.2.2 显示和修改记录内容.....	(30)
2.3 编辑数据库文件.....	(34)
2.3.1 记录的定位.....	(34)
2.3.2 插入数据记录.....	(36)
2.3.3 删除数据记录.....	(37)
2.3.4 成批添加库文件记录.....	(39)
2.4 复制数据库文件.....	(40)
2.4.1 同时复制结构与数据.....	(41)

2.4.2	单独复制文件结构	(42)
2.4.3	单独复制数据记录	(43)
2.4.4	库文件的结构文件	(43)
2.5	文件管理	(46)
2.5.1	显示文件目录	(46)
2.5.2	磁盘文件的改名、删除与复制	(48)
2.5.3	显示文本文件	(49)
2.6	小结	(49)
第3章	数据库的查询与统计	(54)
3.1	选择显示与顺序查询	(54)
3.1.1	选择显示数据信息	(54)
3.1.2	预设过滤器与字段表	(55)
3.1.3	顺序查询	(57)
3.2	排序与索引	(58)
3.2.1	排序	(58)
3.2.2	索引	(60)
3.2.3	索引查询	(65)
3.3	统计和汇总	(67)
3.4	小结	(69)
第4章	多重数据库操作	(75)
4.1	工作区的选择与互访	(75)
4.1.1	FoxBASE+的工作区	(75)
4.1.2	工作区的选择	(76)
4.1.3	工作区的互访	(77)
4.2	多区操作的专用命令	(78)
4.2.1	数据库文件的关联	(78)
4.2.2	数据库文件的更新	(81)
4.2.3	数据库文件的连接	(82)
4.3	小结	(83)
第5章	程序设计	(88)
5.1	程序文件及其常用命令	(88)
5.1.1	程序文件的建立与运行	(88)
5.1.2	状态设置命令	(89)
5.1.3	输入输出命令	(90)
5.1.4	运行控制命令	(92)
5.1.5	其它辅助命令	(93)
5.2	程序的控制结构	(94)
5.2.1	顺序结构	(94)

5.2.2 选择结构.....	(94)
5.2.3 循环结构.....	(97)
5.3 屏幕设计	(100)
5.3.1 @格式设计命令	(100)
5.3.2 屏幕格式文件	(107)
5.3.3 光带式菜单的设计	(109)
5.4 模块化程序设计	(111)
5.4.1 子程序、函数与过程.....	(111)
5.4.2 局部变量和全局变量	(116)
5.4.3 数组	(121)
5.5 程序文件的编译与使用	(124)
5.6 系统配置文件 CONFIG.FX	(125)
5.7 小结	(126)
第 6 章 关系数据库原理	(135)
6.1 关系模型的基本概念	(135)
6.1.1 关系的数学定义	(135)
6.1.2 关系的性质	(137)
6.1.3 关系数据库的描述	(138)
6.2 关系数据操作	(139)
6.2.1 关系代数运算	(139)
6.2.2 关系完备性	(141)
附录	(143)
附录 1 全国计算机等级考试二级(基础知识和 FoxBASE+)考试大纲	(143)
附录 2 1999 年上半年全国计算机等级考试二级 (基础知识和 FoxBASE+)笔试试题	(146)
附录 3 等级考试笔试题分析和上机考试题型示例	(156)
附录 4 FoxBASE+命令一览表(2.10 版)	(158)
附录 5 FoxBASE+函数一览表(2.10 版)	(168)
附录 6 各章习题答案	(173)
参考文献	(176)

第 1 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。20 世纪 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 1.1 节简介数据库的基本概念,然后在后续各节对 FoxBASE+ 的语言成分、技术指标与启动、退出等内容进行概要的综述。

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到另一方。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,这样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 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,即层次型、网状型和关系型。其中前两类又合称为“格式化模型”。

1. 格式化模型

格式化模型适用于非关系型数据库系统。它包括层次型数据库系统与网状型数据库系统两种类型。这一代数据库系统具有以下共同特征:

(1) 采用“记录”为基本的数据结构。在不同的“记录型”(record type)之间,允许存在相互联系。图 1.1 显示了因联系方式不同而区分的两类数据模型。图(a)为“层次模型”(hierarchical model),其总体结构为“树形”,在不同记录型之间只允许存在单线联系;图(b)为“网状模型”(network model),其总体结构呈网形,在两个记录型之间允许存在两种或多于两种的联系。前者适用于管理具有家族形系统结构的数据库,后者则更适于管理在数据之间具有复杂联系的数据库。

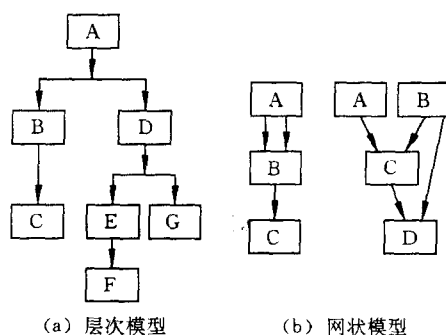


图 1.1 非关系型数据库模型示例

(2) 无论层次模型或网状模型,一次查询只能访问数据库中的一个记录,存取效率不高。对于具有复杂联系的系统,还需用户详细描述对数据的访问路径(称为存取路径),增加使用的麻烦。所以自关系数据库兴起后,非关系数据库系统已逐渐被关系数据库系统所取代,目前仅在一些大中型计算机系统中继续使用。

2. 关系模型

关系模型适用于关系型数据库系统(relational database systems,简称 RDBS)。早在 1970 年,科德(E. F. Codd)就在一篇论文中提出了“关系模型”(relational model)的概念。70 年代中期,国外已有商品化的 RDBS 问世,数据库系统进入了第二代。80 年代后,

RDBS 在包括 PC 机在内的各型计算机上纷纷实现,目前在 PC 机上使用的数据库系统主要是第二代数据库系统。

与第一代数据库系统相比,RDBS 具有下列优点:

(1) 采用人们惯常使用的表格作为基本的数据结构,通过公共的关键字段来实现不同二维表之间(或“关系”之间)的数据联系。二维表形式简单,直观明了,使用与学习都很方便。

(2) 一次查询仅用一条命令或语句,即可访问整个“关系”(或二维表),因而查询效率较高,不像第一代数据库那样每次仅能访问一个记录。通过多表联合操作(称为“多库”操作),还能对有联系的若干二维表实现“关联”查询。

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 通常由下列三部分组成,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) 其他管理和控制程序 例如安全、通信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数据库、应用程序和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。本书只讨论单用户数据库。

1.2 FoxBASE⁺语言简介

FoxBASE⁺是美国 Fox Software 公司开发的微机数据库语言,其 1.0 版于 1987 年 2

月问世, 仅隔 5 个月(1987 年 7 月)就推出了 2.0 版。与 80 年代国内外流行最广的 dBASE 相比, FoxBASE+ 2.0 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 倍~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 2.0 版快 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 2.0 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、变量、表达式、函数和文件。

1.2.1 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 是某校部分 97 级学生的情况表。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 学生

其中 CREATE 表示命令的操作, “学生”便是要建立的数据库文件的名称。

表 1.1 97 学生情况表

学 号	姓 名	性 别	出生日期	班 级	高考分数	保送	备 注
973001	丁大年	男	78/10/07	工 971	540	否	
973004	仲 杏	男	78/03/16	机 971	0	是	市三好
973005	李 芳	女	77/11/02	机 971	495	否	
973002	周 萧	男	78/08/09	机 971	510	否	
973003	刘 梅	女	79/03/22	机 971	0	是	省三好

数据库文件建好后, 如果从键盘逐条输入图 1.2 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 && 符号后的解释; 不带圆点的行为 FoxBASE+ 给出的反馈信息。圆点为 FoxBASE+ 准备接受命令的提示符, 类似 DOS 中的“>”。&& 为 FoxBASE+ 的注释符, 表明跟在它后面的都是注释。

在图 1.2 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 7 栏中的“是”、“否”改用 .T. 和 .F. 外, 和表 1.1 完全相同。还需指出, 在诸如 FORTRAN、PASCAL 等高级语言中, 要完成该图所示的操作例如列表显示文件内容、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, 常常要执行一整段程序。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, 采用这种命令式语言是一个主要的原因。

```
.use 学生                                && 打开名为“学生”的数据库文件
.list                                    && 列表显示当前数据库文件内容
```

Record #	学号	姓名	性别	出生日期	班级	高考分数	保送	备注
1	973001	丁大年	男	10/07/78	工 971	540	.F.	memo
2	973004	仲 杏	男	03/16/78	机 971	0	.T.	memo
3	973005	李 芳	女	11/02/77	机 971	495	.F.	memo
4	973002	周 萧	男	08/09/78	机 971	510	.F.	memo
5	973003	刘 梅	女	03/22/79	机 971	0	.T.	memo

```
.copy to 保送生 fields 姓名,性别,高考分数 for 保送  && 将保送生复制到“保送生”文件
2 records copied

.use 保送生                                && 打开“保送生”文件
.list                                    && 显示其内容
```

Record #	姓名	性别	高考分数
1	仲 杏	男	0
2	刘 梅	女	0

```
.replace all 高考分数 with 600            && 将所有高考分数改为 600
2 replacements

.list                                    && 显示其内容
```

Record #	姓名	性别	高考分数
1	仲 杏	男	600
2	刘 梅	女	600

图 1.2 FoxBASE+的若干命令和执行情况

1.2.2 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 <字段名表> FOR <条件>

等。一般地说,FoxBASE+的命令总是由一个称为命令字的动词开头,后随若干命令子句(亦称功能子句)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. 命令子句

大多数命令中含有命令子句,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 个命令子句,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,后两项是必有的。所以它们的命令格式可分别写成:

REPLACE [**<范围>**] **<字段名>** **WITH** **<表达式>**

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

注意 FIELDS 子句中[FIELDS]也是可选的,即在书写时可以省去,只写出需要复制的字段名即可。命令子句的种类很多,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 姓名,高考分数+600

就表示只列表显示数据库文件中的姓名和高考分数两个字段,且其中高考分数字段需把库文件中的数据加上 600 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,如将图 1.2 中 COPY 命令使用的 FOR 子句改成 WHILE 子句,写作:

COPY TO 保送生 FIELDS 姓名,性别,高考分数 WHILE 保送

则复制到保送生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、第五等记录都满足保送的条件,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。例如若执行命令:

COPY TO 保送生 FOR 保送 WHILE 性别="男"

则在复制生成的“保送生”数据库文件中,将只有姓名为“仲 杏”的一个记录,即:

2. 命令的书写规则

为了简化键盘输入, FoxBASE⁺允许命令字和命令子句中的功能字使用缩写形式, 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命令子句时, 子句的书写次序无关紧要。例如以下两种写法:

```
COPY TO 保送生 FIELDS 姓名, 性别, 高考分数 WHILE 保送
COPY FIELDS 姓名, 性别, 高考分数 TO 保送生 WHILE 保送
```

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 FIELDS 姓名, 性别, 高考分数;
TO 保送生 WHILE 保送
```

3. 命令的执行方式

FoxBASE⁺的命令具有交互执行和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 180 余种命令(详见附录 4), 其中除少数几种只能用于程序执行方式外(参阅第 5 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, 即解释执行和编译执行。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 变量和表达式

与其它高级语言相似,FoxBASE⁺的数据也可区分为常量和变量。在 FoxBASE⁺中,字符型常量可用" "(双引号)或' '(单引号)或[]为定界符,例如"Tea"或[Tea];日期型变量可写作 MM/DD/YY,例如 03/31/99;逻辑型变量可以前、后圆点为定界符,例如. T.(真)或. F.(假)。

注意:上述所说的定界符必须为西文符,即必须在西文状态下输入。

1. 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 1.2 中的“学生”数据库就含有 8 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 2.1.1 节。在上述数据库文件的字段变量中,学号、姓名、性别与班级为 C 型,高考分数为 N 型,出生日期为 D 型,保送为 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 和 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。第 5 章将介绍数组的部分应用。

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示例。

```
. tel='64253136'
64253136
. m=789
789
. ? tel,m
64253136      789
. stor '上海' to add1,add2
上海
? len(add1)
```

4

```
add1='黑龙江'
? len(add1)
6
. date=ctod('05/17/99')    && 赋值或输入日期型变量需使用函数 CTOD
05/17/99
```

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“? 变量名”意为显示变量中的内容。FoxBASE+ 还有一种 ?? 命令,其作用与 ? 命令相同。二者的区别仅在于 ? 命令的结果须从下一行开始显示输出,而 ?? 命令的执行结果则从当前光标位置开始显示。

③ LEN()与 CTOD()为 FoxBASE+ 的函数。LEN()用于计算 C 型变量的字符串长度,CTOD()用于将字符型数据转换为日期型数据。

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 ADD1 首次赋值后长度为 4,再次赋值后长度就变为 6 了(一个汉字占 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。下例将用到其中的一种。

[例 1-3] 试求表 1.1 中学生的总人数。

```
.use 学生          && 打开“学生”库文件
.count to rs       && 统计记录个数存入 RS 变量
```

本例中的 COUNT 命令带有 TO <内存变量>形式的功能子句,能直接为内存变量赋值,不必事先用如 RS=0 一类的赋值命令专为变量定义和赋初值。

(3) 两类变量的差异

表 1.2 中列出了字段变量和内存变量的主要差异。

表 1.2 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>子句的命令建立
3. 有 C、N、D、L、M 等 5 种类型	3. 有 C、N、D、L、S 等 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,否则关机后不保存

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,系统将优先访问字段变量。在一般情况下,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。

2. 表达式

表达式是 FoxBASE+ 语言的重要成分,它通常由常量、变量用运算符连接而成。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、字符串运算符、关系运算符和逻辑运算符 4 类,详见表 1.3。

表 1.3 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
	字符串运算符	+ - (注) 二串相并 二串相并, 前串尾部空格移到新串末端
	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: 字符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符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。下面举几个表达式运算的例子。

[例 1-4] 字符串表达式运算示例。

```
.S1="江苏□"      &&□表示空格
.S2="南京□□"
.? S1+S2
江苏□南京□□
.? S1-S2
江苏南京□□□      && 空格集中至尾部
```

[例 1-5] 逻辑表达式运算示例。

```
(1) .T=.F.      && 变量 T 赋值为“假”
    .F=.T.      && 变量 F 赋值为“真”
    .N=T        && 变量 T 的值赋给变量 N,故 N 也为“假”
    .Y=F        && 变量 F 的值赋给变量 Y,故 Y 也为“真”
    .? Y.and..not.N  && Y 为“真”,.not.N 也是“真”,相“与”后仍是“真”
    .T.
(2) .Y=5        && 变量 Y 赋值为 5
    .X=Y=3      && FoxBASE+ 不支持连续赋值,故此式应理解为将逻辑表达式 Y=3 的值(此时为“假”)赋值给变量 X
    .? X        && X 为逻辑变量,其值为“假”
    .F.
```

[例 1-6] 日期表达式运算示例。

```
. date=ctod("05/17/99")      && 将日期型数据 05/17/99 赋给变量 date
05/17/99
. ? date - 30                  && 5 月 17 日减去 30 天为何日?
04/17/99                      && FoxBASE+ 允许对日期型数据进行加减天数的运算
. ? date + 31                  && 5 月有 31 天,故加 31 天才是 6 月 17 日
06/17/99
```

1.2.4 函数

函数是 FoxBASE+ 语言的又一重要组成部分,基本形式为

<函数名>()

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”,详见第 5 章。标准函数名也可缩写为前 4 个字母,但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 1.4 列出了各类函数的名称。附录 5 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。

表 1.4 FoxBASE+ 2.10 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