

苏长龄 主编

FoxBASE+ 和 FoxPro程序设计



吉林大学出版社



前 言

为了满足初学者学习关系数据库管理系统的需求,我们组织长期从事计算机教学的教师编写了本书。在本书编写过程中,根据国家教委教高司[1997]155号文件精神,并参照国家教委专业指导委员会制定的教学大纲和计算机等级考试(二级关系数据库管理系统)大纲的要求,对内容进行了精心的选择和组织,最大限度的覆盖上述大纲的内容。

本书介绍了关系数据库的基本概念、库文件的建立、数据录入、数据库的维护和使用、函数和变量的使用、系统设置、实用编程技术、FoxPro2.5环境、启动和退出,实例相关查询RQBE的介绍和使用,屏幕生成器、菜单生成器、项目管理器的使用等。学习数据库课程必须上机实验,为满足教学需要,作者还编写了和本书内容配套的实验教材《计算机实验教程》(已出版),供读者选用。

全书严格按教材要求编写,每章有举例并上机通过,主要章节附有习题。本书是集编者多年教学经验编写而成,概念叙述层次清楚、由浅入深、循序渐进,内容通俗易懂、繁简适当,实例融汇贯通,注重编程技巧运用的介绍。本书可作为非计算专业学生和计算机专业低年级学生的数据库课程理想教材,也可作为程序设计人员的参考书。

本书由王文涛编写1~3章,阎绍铭编写4、5章,其余各章由苏长龄组织编写。由于时间仓促,且水平有限,书中难免有不当之处,敬请读者和同行指正。

编 者

1998年5月

目 录

第一章 概 述

1.1 关系数据库管理系统 FoxBASE+	1
1.1.1 数据库基本概念	1
1.1.2 关系数据库的基本特征	2
1.1.3 FoxBASE+ 主要特点	2
1.1.4 FoxBASE+ 的运行环境	3
1.2 FoxBASE+ 主要性能指标	4
1.3 FoxBASE+ 的启动和退出	5
1.3.1 FoxBASE+ 的启动	5
1.3.2 FoxBASE+ 的退出	5
1.4 FoxBASE+ 语言要素	6
1.4.1 数据类型	6
1.4.2 数据种类	7
1.4.3 运算符和表达式	9
1.4.4 命 令	13
1.4.5 文 件	15
1.5 FoxBASE+ 的磁盘文件操作命令	16
1.5.1 磁盘文件目录显示命令	16
1.5.2 文件的复制命令	16
1.5.3 文件删除命令	16
1.5.4 文件改名命令	17
1.5.5 文本文件的显示命令	17

第二章 数据库文件结构的建立和数据输入

2.1 库文件结构建立	20
2.1.1 四个结构参数	20
2.1.2 建立库文件结构	21
2.2 库文件数据输入	24
2.2.1 立即输入	24
2.2.2 添加输入	25
2.2.3 从其他库文件输入数据	26
2.3 库文件打开与关闭	26
2.4 库文件的显示	27

2.4.1 库文件结构的显示	27
2.4.2 库文件数据的显示	28
2.5 库文件的其他建立方法	30
2.5.1 库文件的复制	30
2.5.2 通过结构文件建立库文件结构	33
2.6 库文件结构修改	35
第三章 数据库文件的编辑和查询	
3.1 记录定位	37
3.1.1 记录定位的概念	37
3.1.2 记录指针的移动方法	37
3.1.3 记录指针的移动范围	39
3.2 记录的插入和删除	40
3.2.1 记录的插入	40
3.2.2 记录删除	42
3.3 记录的修改	45
3.3.1 全屏幕编辑修改命令	45
3.3.2 替换命令	49
3.4 库文件数据排序	50
3.4.1 排序命令	51
3.4.2 应用举例	51
3.5 顺序检索	52
3.5.1 检索命令	52
3.5.2 应用举例	53
3.6 索引文件	54
3.6.1 什么是索引检索?	54
3.6.2 索引文件的建立	54
3.6.3 索引文件的打开、更新和关闭	56
3.7 索引检索	59
3.7.1 索引检索命令 FIND	59
3.7.2 索引检索命令 SEEK	60
3.7.3 索引检索命令继续检索	61
3.7.4 检索命令比较	62
第四章 数据库的运算与统计	
4.1 赋值命令	64
4.1.1 “=”命令	64
4.1.2 STORE 命令	65
4.2 统计记录个数	65
4.3 数值字段求和	67
4.4 数值字段求平均值	69
4.5 数值字段分组求合计	70

第五章 多重数据库操作

5.1 工作区选择	75
5.1.1 多区操作的特点与种类	75
5.1.2 工作区的选择	76
5.1.3 工作区的访问	78
5.2 库文件之间的数据更新	80
5.3 库文件之间的关联	83
5.3.1 关联的方法	83
5.3.2 关联库文件之间的数据更新	85
5.4 库文件之间的连接	86

第六章 内存变量及数组

6.1 内存变量操作	90
6.1.1 内存变量的定义和赋值	90
6.1.2 内存变量的显示	90
6.1.3 内存变量的释放	92
6.1.4 内存变量文件	93
6.2 数 组	94
6.2.1 数组的定义	94
6.2.2 数组的使用	95
6.2.3 数组与库文件之间的数据交换	96

第七章 函 数

7.1 数学运算函数	101
7.2 字符函数	103
7.3 日期与时间函数	108
7.4 转换函数	110
7.5 测试函数	111
7.6 标识函数	116
7.7 输入函数	117

第八章 系统环境参数与配置文件

8.1 配置文件	123
8.1.1 DOS 系统配置文件	123
8.1.2 FoxBASE+ 系统配置文件	124
8.2 SET 命令和 SET 命令组	125
8.2.1 输出环境设置	126
8.2.2 状态设置	128
8.2.3 文件记录环境设置	131
8.3 利用菜单选择法设置参数	132
8.4 系统状态显示	135

第九章 程序设计

9.1 程序文件的建立、修改和执行	138
-------------------	-----

9.1.1 程序文件的建立和修改	138
9.1.2 程序文件的运行	139
9.2 交互式命令	139
9.2.1 输入字符串命令	139
9.2.2 输入单字符命令	141
9.2.3 输入任意类型数据命令	142
9.2.4 TEXT-ENDTEXT 命令	143
9.2.5 屏幕编辑修改变量语句	143
9.3 程序设计	145
9.3.1 分支程序	145
9.3.2 循环程序	149
9.4 子程序与过程文件	153
9.4.1 子程序	153
9.4.2 过程文件	154
9.4.3 自定义函数	158
9.4.4 程序的注释	159
9.5 全局变量和局部变量	160
9.6 程序调试	162
9.6.1 保留和查看历史	162
9.6.2 跟踪程序执行	164
9.6.3 断点设置和恢复	165
9.6.4 错误捕获和处理	166
第十章 输入输出格式设计及应用程序举例	
10.1 屏幕格式语句	168
10.1.1 屏幕的坐标	168
10.1.2 清屏和画框语句	168
10.1.3 屏幕显示输入输出语句	169
10.2 菜单设计命令	171
10.2.1 光条菜单命令	171
10.2.2 下拉式菜单命令	173
10.2.3 上托式菜单命令	176
10.3 输入屏幕格式设计	177
10.3.1 用屏幕格式语句设计输入格式	178
10.3.2 屏幕格式文件	179
10.4 输出格式设计	181
10.4.1 报表格式文件	181
10.4.2 标签文件	185
10.4.3 带横竖线表格的输出设计	188
10.4.4 报表格式的改进	189
10.5 应用程序举例	191

第十一章 FoxPro2.5 环境

11.1 启动和退出 FoxPro	207
11.2 菜 单	209
11.3 菜单笺含义	209
11.4 窗 口	213
11.5 对话框	214
11.6 Command 窗口	215
11.7 获取帮助	215

第十二章 建立 FoxPro 数据库

12.1 建立新的数据库	216
12.2 定义数据库结构	216
12.3 打开和关闭数据库	219
12.4 拷贝数据文件及数据结构	220
12.5 修改数据库结构	222

第十三章 数据的编辑、索引与排序

13.1 增加数据	224
13.2 浏览数据	225
13.3 移动指针	226
13.4 编辑数据	227
13.4.1 修改数据	227
13.4.2 删除和恢复删除记录	227
13.5 索引与排序	228
13.5.1 索引分类	228
13.5.2 建立索引	229
13.5.3 使用索引	231
13.5.4 排 序	233

第十四章 数据库查询与报表

14.1 关系运算和逻辑运算	235
14.2 WHILE 和 FOR 子句的区别	236
14.2.1 使用 WHILE 子句	236
14.2.2 使用 FOR 子句	239
14.3 单个和多重记录查询	239
14.3.1 用 Locate 选项查询	240
14.3.2 用 Seek 选项查询	241
14.4 过滤器	242
14.5 限制查询	244
14.6 建立报表	246
14.6.1 建立报表的方法	246
14.6.2 报表生器介绍	247

第十五章 关系数据库文件和视图

15.1 关系数据库文件.....	250
15.2 工作区.....	251
15.3 视图简介.....	252
15.4 视图使用.....	253
15.4.1 打开数据库建立索引文件.....	254
15.4.2 建立关联.....	254
15.4.3 数据库文件的关联查询.....	257

第十六章 实例相关查询 RQBE

16.1 RQBE 介绍	259
16.2 RQBE 使用	262

第十七章 屏幕生成器

17.1 建立屏幕.....	269
17.2 有关屏幕的系统菜单.....	269
17.3 屏幕设计工具.....	269
17.4 建立第一个屏幕——快速屏幕.....	270
17.4.1 建立快速屏幕.....	271
17.4.2 屏幕调整.....	274
17.4.3 生成屏幕程序代码.....	276

第十八章 菜单生成器

18.1 菜单条、菜单笺和子菜单	277
18.2 菜单命令.....	277
18.2.1 DEFINE 命令	278
18.2.2 ON 命令	282
18.2.3 ACTIVATE 和 DEACTIVATE 命令	283
18.3 使用菜单生成器.....	283
18.3.1 启动菜单生成器.....	284
18.3.2 Menu 菜单	284
18.3.3 菜单结构.....	287
18.3.4 菜单实例.....	289
18.3.5 生成菜单程序代码.....	290

第十九章 项目管理器

19.1 进入项目管理器.....	294
19.2 项目管理菜单.....	295
19.3 使用项目管理器.....	298
附录一 FoxBASE+2.1 命令一览表	302
附录二 FoxBASE+2.1 函数一览表	310
附录三 FoxBASE+2.1 错误信息表	314
附录四 ASCII 码表	325

第一章 概述

1.1 关系数据库管理系统 FoxBASE⁺

1.1.1 数据库基本概念

1. 数据和数据处理

人类活动的整个历史都离不开对现象、消息、情报、指令及信号等的收集、保存、利用和处理。为了记载这些现象、消息、情报等,人们用各种各样的物理符号及组合来表示它们,这些符号及其组合就是数据。

计算机中所说的数据是指数字、字符及凡能通过计算机进行处理的一切符号的集合。

数据与信息在概念上也是有区别的。数据是信息的载体,是信息的具体表示形式。信息是经过加工处理后得到的有价值的信息,信息为人们所利用可转化为社会财富。

数据处理是指对数据的收集、记载、分类、排序、计算或加工、传输制表、递交等过程。数据处理通常也称为信息处理。

2. 结构化数据

按一定方式组织起来的数据称为结构化数据。数据间的关系采用二维表格的形式联系起来,这样的数据模型称为关系模型。FoxBASE⁺主要研究具有关系模型结构的数据。

3. 数据库

通俗地说,数据库(Data Base)就是数据的仓库,但并不仅仅是指一些数据的简单的集合。按一定方式组织起来,独立于应用程序并能得多用户共享的数据的集合称为数据库。关系数据库实际上就是二维表。

4. 数据库管理系统

数据库管理系统(Data Base Management System)简称 DBMS,是对数据库进行管理的各种软件的总称。其功能是接受用户提出的请求,执行用户的命令和程序,对数据库进行各种操作和访问,以达到维护数据库和管理应用数据库的目的。FoxBASE⁺就是这样的数据库管理系统。

FoxBASE⁺是美国 Fox Software 公司于 1986 年 6 月推出的关系数据库管理系统,1987 年进行了修改,成为 FoxBASE⁺2.00 版,1988 年 7 月又推出 FoxBASE⁺2.10 版本,是目前最为广泛流行的版本。其最大的优点是速度快,适用多种操作系统,用户接口好,现在已广泛应用于国民经济的各个领域,特别适合各种信息管理,如财产和工资管理、仓库管理、档案管理、图书资料的查询和管理、户籍管理、教务管理等等。

利用 FoxBASE⁺的有关命令能准确迅速地进行查找、检索、统计、增删、插入、分类、汇总、打印等。另外 FoxBASE⁺还提供了一套自成体系的数据库语言,用户可以根据自己的需要编

制一些应用程序去完成特定的任务,从而提高工作和办事效率。

1.1.2 关系数据库的基本特征

关系数据库的基本特征是将数据之间的关系组织成一张二维表格。例如:下表是一张关于职工的二维表格,我们可以把它看成是一个关系数据库(见表 1.1)。

表 1.1 教工简历表

姓 名	性 别	工作日期	婚 否	教研室	职 称	月收入	简 历
宋常林	男	10/04/68	.T.	数 学	教 授	979.00	
杨永贵	男	02/07/93	.F.	计算机	助 教	410.00	
王 红	女	03/29/84	.T.	计算机	副教授	746.00	
马 伟	男	10/17/88	.T.	计算机	讲 师	587.00	
于 新	女	04/21/90	.T.	数 学	讲 师	574.00	
刘建设	男	02/23/92	.T.	计算机	讲 师	551.00	

表格的名称称为数据库文件名。表格第一行的每一栏称为一个字段,各栏均可起一个名称,称为字段名(姓名、性别、工作日期、婚否、教研室、职称、月收入、简历)。第二行开始的每一行称为一个记录。一般来说这种表应具备以下特点:

- (1) 各字段名不能相同,且字段是数据的最小单位;
- (2) 表中每一列数据类型必须相同;
- (3) 各条记录不能完全相同;
- (4) 表中行的次序与列的次序可以分别任意排列。

通过命令可以建立满足以上条件的关系数据库,并将数据库放到磁盘上保存。一个数据库由库名(表格名称)、库结构(各字段信息)、记录(具体数据)三部分组成,称为数据库的三要素。

关系型数据库基本的关系运算有三种:

(1) 选择:若我们从全部记录中挑选出满足一定条件的记录,这种对记录行的操作运算叫做选择。选择操作是对关系的横向操作并不改变关系的框架结构;

(2) 投影:若我们从关系框架全部字段中选取一部分我们关心的字段及字段值,那么这种操作称为投影。投影操作是纵向操作即对字段操作,这一操作将改变关系框架,等于从原关系框架的字段中选出一部分而构成一个新的关系;

(3) 连接:将两个关系通过比较投影运算连接生成一个新的关系,这种操作称为连接。连接后生成的新关系,反映出原来两个关系之间的联系。

1.1.3 FoxBASE⁺主要特点

1. 功能强

FoxBASE⁺有几百条命令和函数,提供了多用户命令 10 条及函数 5 个。FoxBASE⁺的数据类型有 5 种,文件类型有 10 种。还提供了 DIMENSION 语句,可以建立一维和二维的内存变量数组。另外 FoxBASE⁺还有用户自定义函数、新的菜单驱动接口、交互式调试功能等。过

程文件允许包含 128 个过程。

2. 操作方便灵活

(1) FoxBASE+ 为用户提供了二种工作方式,即单命令工作方式和批命令(程序)方式。单命令工作方式也称为立即方式,是在 FoxBASE+ 的圆点提示符“.”下直接输入命令,每输入一条命令就执行一条。而批命令方式是将一批命令集中在一个命令文件中一起执行。批命令方式也称程序方式,因为它可利用系统提供的语言系统编制程序。像其他高级语言那样,程序以文件形式存储在磁盘上;

(2) FoxBASE+ 提供全屏编辑功能,可以利用光标控制键将光标定位到屏幕所需要位置进行增、删、修改等操作;

(3) FoxBASE+ 提供了一个专用于存放历史命令的缓冲区。该缓冲区默认可以存放 20 条最近执行的命令,并可以用 ↑、↓ 键查看和修改。这样,一方面用户可以随时察看自己刚输入的命令,另一方面对比较长的命令,可以不必重复输入,只要调出修改即可。

3. FoxBASE+ 系统软件组成丰富

FoxBASE+ 的系统软件一般包括以下几个主要文件:

执行文件	FOXPLUS. EXE
覆盖文件	FOXPLUS. OVL
文字原始数据文件	FOXPLUS. RSC
帮助文件	FOXPHelp. HLP
过程组合文件	FOXBIND. EXE
准编译文件	FOXPCOMP. EXE

其中 FOXPLUS. EXE 和 FOXPLUS. OVL 是两个最基本、最重要的文件,也是运行 FoxBASE+ 的必备文件。FOXBIND. EXE 文件可自动地收集命令文件进入过程文件,并在每个过程前冠以注释,提高编制过程文件的速度。

除以上几个主要文件以外,FoxBASE+ 2.1 版本还增加了一组强有力的开发工具软件,包括控制中心 FoxCenter、屏幕设计工具 FoxView、应用程序生成器 FoxCode 等等。

1.1.4 FoxBASE+ 的运行环境

1. 硬件环境

FoxBASE+ 可在多种机型下运行,在微机上运行标准版 FoxBASE+ 至少需要 512KB 可用内存及一个硬盘和一个软盘。

运行汉字 FoxBASE+ 则要求更多的内存空间,通常需要 1 兆以上。

2. 软件环境

FoxBASE+ 可在多种操作系统下运行,如 MS-DOS、PC-DOS 或 XENIX 等,运行汉字 FoxBASE+ 还必须要有汉字操作系统支持。目前较流行的汉字操作系统 CCDOS、UCDOS、SP-DOS 等,均支持 FoxBASE+。

值得注意的是在 DOS 启动盘根目录中必须有 CONFIG. SYS 文件,这个文件不是 FoxBASE+ 的文件,它是 DOS 操作系统用来设置系统参数的文件。由于 FoxBASE+ 也是借助于此文件作为它的某些参数的系统设置,故必须在这个文件中进行适当的设置。否则在进入

FoxBASE+ 后, 执行命令时有可能出现“不能打开文件”或“打开文件太多”的警告, CONFIG. SYS 文件中有二条语句对 FoxBASE+ 是至关重要的, 即

FILES=40

BUFFERS=35

其中 FILES 语句用来设置 DOS 操作系统在同一时间可打开和访问的文件数, 这个数值直接关系到 FoxBASE+ 能打开的文件数。由于系统需要占去一些文件数, 故建议设置为 40。

其中 BUFFERS 语句用来设置计算机启动时 DOS 操作系统建立磁盘缓冲区的数量。设置太多或太少都会影响存取文件的速度。为了使 FoxBASE+ 具有较好的性能, BUFFERS 可设置在 20~40 之间。

如果根目录没有 CONFIG. SYS 文件或该文件中没有以上两条语句, 可以用任何一个文件编辑器建立或修改, 也可以用 COPY 命令从键盘输入建立:

C:\>COPY CON CONFIG. SYS

输入 FILES=40 和 BUFFERS=35 等后按 CTRL+Z(或 F6 键)存盘退出。

最后还要提醒读者注意: 当建立或修改 CONFIG. SYS 文件后, 必须要重新启动 DOS 操作系统, 因为只有重新启动 DOS 操作系统, DOS 操作系统才能重新检查 CONFIG. SYS 文件, 并将新的配置取代旧的配置。

1.2 FoxBASE+ 主要性能指标

1. 有关记录和字段

每个库文件最大记录数	10 亿个
每个记录的最多字符数	4000
每个记录的最多字段数	128
每个字段的最多字符数	254
其中: 字符型字段	≤ 254
数字型字段	≤ 19
日期型字段	= 8
逻辑型字段	= 1
备注型字段	= 10(备注文件 ≤ 64K)

2. 有关文件

同时可打开的库文件个数	10
每个库同时可打开的索引文件数	7
同时可打开的索引文件数, 最多	21
同时可打开的所有文件数	48

3. 有关命令、数组和内存变量

字符串最多字符数	254
命令行最多字符数	254
索引关键字最多字符数	100

内存变量最多可用字节数	6000
最多内存变量个数	3600(默认为 256)
最多数组数	3600
每个数组中最多数组元素个数	3600

1.3 FoxBASE⁺的启动和退出

1.3.1 FoxBASE⁺的启动

1. FoxBASE⁺的安装

可以先在硬盘上建立一个 FoxBASE⁺的子目录,把所有 FoxBASE⁺的文件都复制到该目录下。子目录名常用 FOX,以区别于其他子目录。

例如:在根目录下,键入以下命令

```
C:\>MD FOX
```

就建立了一个名为 FOX 的子目录,然后用以下命令将 A 盘的 FoxBASE⁺的系统文件拷贝到 FOX 子目录下。

```
C:\>CD FOX
```

```
C:\FOX>COPY A:*. *
```

安装完毕。

2. FoxBASE⁺的启动步骤

运行汉字 FoxBASE⁺时,首先应启动汉字操作系统;

进入硬盘存放 FoxBASE⁺系统的子目录;

再键入命令 FOXPLUS(若多用户系统,应输入 MFOXPLUS)。例如:

```
C:\FOX>FOXPLUS
```

回车后稍等片刻,屏幕上出现用虚点线连结的几个大字 Foxbase,说明已进入 FoxBASE⁺系统。另外,还可以看到 FoxBASE⁺的提示符小圆点“.”。

至此 FoxBASE⁺引导完毕。

有时为了方便使用,常采用批处理文件或设置路径的方法,使 FoxBASE⁺的启动更为迅速与简捷。

1.3.2 FoxBASE⁺的退出

在圆点状态下可用 QUIT 命令退出 FoxBASE⁺系统。

```
.QUIT
```

回车后显示:FoxBASE⁺正常退出,并出现 C:\FOX>,说明已回到 DOS 下。

QUIT 命令的另外作用是关闭一切 FoxBASE⁺的文件,并将所有内存中的数据或文件保存到硬盘上,所以它也有避免数据丢失的功效。如果用户没有用 QUIT 命令而热启动或直接切断电源,就有可能造成数据永久性的丢失,所以建议读者养成在关机前先用 QUIT 命令退出 FoxBASE⁺的习惯。

1.4 FoxBASE⁺语言要素

FoxBASE⁺有单用户和多用户两类版本。现在国内使用的中西文 FoxBASE⁺是根据英文的多用户版本汉化而成的。本节将对中西文 FoxBASE⁺2.1 语言的各种成分作概要综述,它们包括:常量、变量、函数、表达式、命令和文件等。

1.4.1 数据类型

数据库操作的对象是数据,数据按其结构、处理方式、用途等属性的不同,可分为若干不同的类型,只有相同类型的数据之间才能进行数据操作,这是数据库操作的基本原则,因此,数据类型在数据库中是一个非常重要的概念。FoxBASE⁺的数据类型共有六种,使用 FoxBASE⁺必须掌握它们的构成和使用规则。

1. 数值型

数值型(Numeric)类型代号为 N,数值型数据只能由阿拉伯数字、小数点和正负号构成。数值型数据之间可以进行算术运算,其长度是指组成数值型数据的字符个数,其大小必须 ≤ 19 ,数值精度为 16 位有效数字。例如: +12345 的宽度为 6; -0.1234567 的宽度为 10; 12345678 的宽度为 8。

2. 字符型

字符型(Character)类型代号是 C。字符型数据可以由一切可打印的汉字、ASCII 字符等组成,在 FoxBASE⁺中,字符型数据之间不能进行算术运算,但可以把几个字符型数据连接起来,也可比较它们中字符所对应的 ASCII 码的大小。一个字符型数据所含字符的个数称为字符型数据的长度,一个字符型数据的最大长度为 254 个字节。例如:“12ABab”的长度为 6;“计算机系”的长度为 8;“123,456”的长度为 7。

3. 日期型

日期型(Date)类型代号是 D。日期型数据用来存放表示日期的数据。系统默认的日期型数据的格式:“月/日/年”(MM/DD/YY)。长度固定为 8 个字符。两个日期型数据可以进行大小比较,日期在前的为小,日期在后的为大。两个日期型数据之间可以进行减法运算,结果为数值型数据。日期型数据可以加或减一个数值型数据,结果依然是日期型数据。这些运算中的数值型数据,均表示两个日期间相差的天数。

4. 逻辑型

逻辑型(Logical)类型的代号为 L。逻辑型数据只有两个值,即逻辑真和逻辑假。它的长度固定为 1。逻辑真可用 .T. (或 .t.、.Y.、.y.) 表示,逻辑假可用 .F. (或 .f.、.N.、.n.) 表示。逻辑型数据之间只能进行逻辑运算。

5. 备注型

备注型(Memo)类型的代号为 M。如上所述,由于一个字符型数据长度最大为 254,但在实际应用中往往需要在一个数据中保存更多的字符信息。于是,FoxBASE⁺采用一个附加文件(称为备注文件,文件扩展名固定为 .DBT)来存放这些长度较大且又不等长的字符信息块,而在该数据中只存放这些信息块的地址指针,该数据的类型便称为备注型,其长度固定为 10。备

注型数据可以被编辑、显示和打印,但不能进行任何类型的运算。

6. 屏幕型

屏幕型类型的代号是S。屏幕型数据用以存放屏幕信息。在FoxBASE⁺中,屏幕型数据只存放整幅屏幕的信息。一幅屏幕的信息包括25×80个字符,每个字符还带1个字节的颜色码。因此,一个屏幕型数据需要占内存4KB,屏幕型数据除了存放或显示屏幕信息,不能进行任何其它类型的操作运算。

除上述6种数据类型外,还有一种A型(Array),用来标识数组。但是,A型并不是一种独立的数据类型,因为数组中的各个元素均有各自的类型,而且可以各不相同。

在FoxBASE⁺中,数据类型是可以转换的,不同类型的数据之间,通过类型转换之后便可进行运算或操作。FoxBASE⁺中主要是通过一些专门的转换函数来实现数据类型转换的,在实际应用中,以字符型与数值型、日期型之间的互相转换最为常见。

1.4.2 数据种类

FoxBASE⁺的数据有常量、变量、函数3种。

常量是固定不变的数据。变量是其内容可以变化的数据。函数是一种由其它数据根据某种规则运算或操作后得出的数据。

1. 常量

FoxBASE⁺的常量按其数据类型可以分为数值型、字符型、日期型或逻辑型4种。

(1) 数值型常量:

数值型常量可以是整数或者小数,但不可是分数。可以用通常记数法表示,也可用科学记数法表示。如:-9、123.45、1.23407E3等。其中1.23407E3表示1.23407乘以10的3次方,即用科学记数法表示的1234.07。

(2) 字符型常量

字符型常量是用单引号、双引号或方括号作为定界符括起来的数据,又称为字符串或简称为串。定界符必须成对匹配使用,如'数据库'、"数据库"、[FoxBASE]、等等。如果某种定界符本身也是字符串的内容,则需用另一种定界符括起来。应当指出" "也是字符型常量,它表示长度为零的空字符串。

(3) 日期型常量

日期型常量是用于表示日期的常量。在FoxBASE⁺中,日期型常量无法直接表示,一般是用字符串通过转换函数CTOD()来表示,该函数的功能是将字符型数据转换成日期型数据,如:CTOD("11/07/96")表示1996年11月7日,其中字符串的格式为MM/DD/YY,即为月/日/年,此表示格式为美国表示格式,FoxBASE⁺向用户提供了SET DATE命令,可根据需要改变使用格式。注意:"11/12/96"不是日期常量,是字符串常量,11/12/96也不是日期常量,而是数值表达式,正斜线/表示除法运算符。

(4) 逻辑型常量:

逻辑常量只能取两个值,即逻辑真或逻辑假。逻辑真用.T.(或.t.、Y.、y.)表示,逻辑假用.F.(或.f.、N.、n.)表示。

2. 变量

FoxBASE⁺中有两种变量,即字段变量和内存变量。FoxBASE⁺规定,变量的要素是变量名称、变量值类型、长度、作用域和保存方式等。变量名称必须以汉字或字母开头,由汉字、字母、数字和下划线组成,其宽度不得超过 10 个字符。

(1) 字段变量

字段变量是指库文件中已定义的任一字段,由字段名标识,对应于二维表中的一个属性,字段名即对应属性名。字段变量是在建立库文件时定义,又是组成库文件的基本单元,它是一个多值变量,变量值的多少取决于库文件中记录数的多少。字段的值对应属性的值,字段可取值的范围,即对应属性值的域。

字段变量有数值(N)、字符(C)、日期(D)、逻辑(L)和备注(M)五种数据类型。

数值型:表示整数或小数,宽度为 1~19 位。若是含有正负号和小数,正负号和小数点均占一位,小数位一般为 0~15,即至少应比字段宽度少两位。数值字段的初始值为 0。

字符型:宽度可为 1~254 个字符。字符型字段变量的初始值为该字段宽度的空格串。

日期型:宽度固定为 8。日期型字段变量的初始值不表示任何日期,其显示形式:

“ / / ”。(注:本书中符号“ ”表示空格)

逻辑型:宽度固定为 1。逻辑型字段的初始值为 .F.。

备注型:宽度固定为 10。备注型字段(或称记忆型字段)的实际内容存放在备注文件中(.DBT),在字段中显示时仅以“Memo”表示。

一个库文件中字段变量的个数最多不得超过 128 个,而且它们的总宽度不得超过 4000 个字节。

(2) 内存变量

内存变量是一种独立于库文件而存在的变量,用来存储在命令或程序执行中各种临时用到的输入、输出数据或中间结果。因为它是在内存中开辟的临时工作单元,所以,需要时可以临时定义,不需要时可随时释放。常用的内存变量有 4 种数据类型(数值型、字符型、日期型和逻辑型),有关规定与同类型字段变量相同。

FoxBASE⁺把数组作为内存变量,因此,提高了它的数值运算和数据处理能力。有关内存变量更详细的介绍,请见本书第六章内容。

字段变量与内存变量进行比较如表 1.2 所示:

表 1.2 字段变量与内存变量比较

字段变量	内存变量
是库文件的组成部分	独立于库文件而存在
数据类型和宽度相对固定	类型和宽度随赋值的类型和宽度而定
随库文件的打开而建立	使用时随时定义
有 N、C、D、L、M5 种类型	有 N、C、D、L4 种类型
多值变量	单值变量
关机后保存在库文件中	若预先未存入内存变量文件中,关机后自动消失

前已说明,两类变量的命名规则是一样的。当访问变量时,若字段变量与内存变量同名,系统则不加区别的访问字段变量;若想访问同名的内存变量,则需要在变量名前冠以字母 M 和

—>,即“M—>变量名”,以示区别。注意:—>是由减号“—”和大于号“>”组成,而不是右箭头。

3. 函数

函数是FoxBASE+数据的一个重要组成部分,通常取<函数名>()的形式。FoxBASE+2.1向用户提供了上百个标准函数,用来支持各种运算,检测系统状态,作出某种判断。适当地使用这些函数,能增强和扩展命令和程序的功能,减少编程的工作量,提高编程效率。

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,称为自变量和函数返回值。必须强调,函数只有一个返回值。因此,它也与常量、变量相似,具有一定的数据类型。函数按返回值的数据类型不同可分为C型,D型,N型和L型四种。

从实质上看,函数其实就是预先编好的子程序,调用一次函数,实际上就是执行一次用该函数名标识的子程序。关于函数的进一步介绍请读者参阅后面的章节。

1.4.3 运算符和表达式

1. 运算符

运算符是对数据进行简单加工处理的一种符号。由于FoxBASE+中的数据有不同的数据类型,并且只有相同类型的数据之间才能进行运算,所以,了解运算符的分类和功能尤为重要。运算符可分为4种:算术运算符、字符串运算符、关系运算符和逻辑运算符。

运算符按加工对象的数目不同可分为单目运算符和双目运算符两种,单目运算符的加工对象是一个数据,双目运算符的加工对象是两个数据。

(1) 算术运算符

算术运算符主要用于对数值型数据进行加工,加工结果仍是数值型数据。但其中的加、减运算符对日期型数据也可加工,加工结果可能是一个新的日期,也可能是一个整数。算术运算按其运算的优先级别从高到低排列如下:

算术运算符	运算规则	说 明
()	圆括号	
+, -	正, 负	单目运算符
*, ^	乘幂	双目运算符
*, /	乘, 除	双目运算符
+, -	加, 减	双目运算符

当加法和减法运算符用于日期型数据时,其运算规则如下:

日期型数据+整数数据=日期型数据加上整数天后的一个新日期

日期型数据-整数数据=日期型数据减去整数天后的一个新日期

日期型数据1-日期型数据2=两个日期之间相差的天数

以下例子中,圆点提示符“.”为FoxBASE+系统状态提示符,当启动FoxBASE+屏幕上出现“.”时,表示等待接收FoxBASE+命令。

“?”为FoxBASE+的一条具有运算和显示功能的命令,当其后为一表达式或用“,”分开的若干个表达式时,先完成表达式的计算后显示其结果。