

汉字 微机网络 数据库系统

dBASE III PLUS

沈金发 郑甫京 俞盘祥

清华大学出版社

汉字微机网络数据库系统

dBASE III PLUS

沈金发 郑甫京 俞盘祥 编著

清华大学出版社

内 容 简 介

dBASE III PLUS 是 1986 年 Ashton-Tate 公司推出的 dBASE 系统的最新版本。它进一步扩充了 dBASE III 的功能, 并能在局部网络上运行, 极大地扩大了 dBASE 系统的应用范围。

本书介绍了 dBASE III PLUS 各种功能, 并对有关的输入输出、程序调试以及在应用系统开发过程中遇到的各种技术问题作了细致的介绍。它可作为数据库课的教学参考书, 也可作为各类数据库培训班的教材。可供广大的计算机应用人员、教学人员阅读参考。

汉字微机网络数据库系统

dBASE III PLUS

沈金发 郑甬京 俞 祥 编著

☆

清华大学出版社出版

北京 清华园

北京海淀昊海印刷厂印刷

新华书店北京发行所发行

☆

开本: 787×1092 1/16 印张: 25.75 字数: 658 千字

1989 年 3 月第 1 版 1989 年 3 月第 1 次印刷

印数: 00001-10000

ISBN 7-302-00394-7/TP·301

定价: 10.50 元

前 言

随着微型计算机的普及与推广,关系数据库系统 dBASE II 和 dBASE III 得到了广泛的应用,深受广大用户的欢迎,取得了良好的效益。dBASE III PLUS 是在 dBASE III 的基础上发展起来的,是 1986 年 Ashton-Tate 公司推出的 dBASE 系统的最新版本。dBASE III PLUS 进一步改善和扩充了 dBASE III 的功能,并且解决了数据共享问题。dBASE III PLUS 不但可以运行在单机上,也可以在局部网络上运行。用 dBASE III 编制的任何程序,不需作任何修改,就可以在 dBASE III PLUS 系统上运行,这为 dBASE III PLUS 的推广和应用奠定了坚实的基础。目前 dBASE III PLUS 正受着广大用户的密切注意和关切,也正在迅速普及与推广。我们使用 dBASE III PLUS 系统开发了一个大型企业的生产调度与统计管理信息系统,取得了良好的效果。从我们的经验来看, dBASE III PLUS 极大地扩充了 dBASE 系统的应用范围。

由于 dBASE III PLUS 的种种优点,广大用户迫切需要一本介绍该系统的资料,我们应这种需要编写了这本书。

本书全面系统地介绍了 dBASE III PLUS 的各种功能,并且对用户经常遇见的输入输出、程序调试等技术也作了较详细的介绍。为了使用户能更好地应用 dBASE III PLUS 来开发应用系统,本书对开发应用系统中经常碰到的技术与方法作了细致的阐述。

在本书的编写过程中得到了多方面的关心与帮助,金坚同志对部分章节作了详细的阅读,并提出了许多有益的意见,我们表示衷心地感谢。曹艳萍同志作了大量的辅助工作,我们表示深切地谢意。

本书可作为高等院校的数据库课程参考教材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与手册。

由于时间仓促,难免有缺点与错误,敬请广大读者指正。

目 录

第一章 概述	1
1.1 dBASE 系统的发展及特点	1
1.2 数据库文件	4
1.3 常数和内存变量	7
1.4 运算符、运算规则和表达式	10
1.5 命令构成和基本规则	16
1.6 键盘缓冲区	18
第二章 数据库的基本操作	20
2.1 引言	20
2.2 数据库结构的建立	21
2.3 数据的输入	32
2.4 数据定位	37
2.5 数据库的显示	41
2.6 编辑数据库	45
2.7 数据库的分类、索引和查找	61
2.8 数据库的统计、分组求和文件的生成	79
2.9 筛选文件的建立、修改和使用	85
2.10 多重数据库操作	91
2.11 目录文件的建立和使用	102
2.12 格式文件的建立、修改和使用	105
2.13 数据库文件间的关联	117
2.14 打印报表	130
2.15 打印标签	135
第三章 dBASE II PLUS 的函数和应用	142
3.1 数学运算函数	142
3.2 字符操作函数	145
3.3 日期函数	151
3.4 转换函数	153
3.5 专用测试函数	157
3.6 标记函数	167
3.7 输入函数	169

第四章 数据库的辅助操作	172
4.1 结构文件的生成和使用	172
4.2 数据库文件间数据转换	175
4.3 dBASE III PLUS 文件与外部文件转换	178
4.4 环境设置	183
4.5 清理、状态显示命令	191
4.6 文件的操作	194
第五章 程序设计的基本方法	198
5.1 程序设计的概念	198
5.2 程序设计方法——结构化设计方法	199
5.3 选择结构实现语句	201
5.4 循环结构实现语句	204
5.5 过程设计和过程调用	205
5.6 内存变量操作	208
5.7 TEXT 命令.....	213
5.8 接收与输入	213
5.9 命令文件的建立	214
5.10 程序实例	215
第六章 输入输出格式设计	221
6.1 使用@命令设计屏幕输入输出格式	221
6.2 使用@命令设计打印机输出格式	246
第七章 程序调试技术	249
7.1 引言	249
7.2 查看历史	252
7.3 跟踪运行程序	257
7.4 断点设置和恢复	262
第八章 与 DOS 命令的接口	263
8.1 调用 DOS 命令文件	263
8.2 调用二进制文件	264
第九章 dBASE III PLUS 的启动	267
9.1 网络的概念	267
9.2 dBASE III PLUS 的网络环境.....	268
9.3 单用户安装、启动 dBASE III PLUS 的方法	271

9.4	局部网络安装、启动 dBASE III PLUS 的方法	273
9.5	dBASE III PLUS 的配置	281
9.6	退出 dBASE III PLUS	284
第十章	数据库的安全与保密	285
10.1	dBASE 安全系统的功能	285
10.2	安全系统的建立	288
10.3	安全系统建立中需要注意的问题	298
第十一章	并发控制	300
11.1	数据的一致性	300
11.2	数据的保护	301
11.3	死锁处理	307
11.4	错误的捕获与改正	315
第十二章	网络命令与函数	318
12.1	网络命令	318
12.2	网络函数	324
第十三章	应用程序加密系统——RUNTIME+	328
13.1	程序的加密	328
13.2	程序的连接	333
第十四章	程序举例	338
附录		375
附录 A	dBASE III PLUS 命令一览表	375
附录 B	dBASE III PLUS 函数一览表	388
附录 C	出错信息及其说明	390

第一章 概 述

1.1 dBASE 系统的发展及特点

dBASE 关系数据库系统是美国 Ashton—Tate 公司推出的, 适用于微型计算机 IBM—PC/XT 以及全兼容的微型计算机。主要经历了 dBASE I、dBASE II、dBASE II PLUS 三个发展阶段。

七十年代末期, 随着计算机技术的不断发展, 微型计算机日益普及, 已经渗透到工业、农业、商业、企事业和科研管理等各个领域, 得到了广泛应用。

在整个计算机应用领域中, 数据和信息处理占据了相当大的比例。例如, 工厂的生产计划管理、仓库的器材管理、医院的药材管理及病历的管理等都需要处理大量的复杂而繁琐的数据, 以便制定出正确可行的方案, 便于领导决策。同样的, 帐目管理、人事档案的管理都需要处理数据。因而人们开始越来越关心如何使用计算机进行数据管理最为合理最为有效。

美国 Ashton—Tate 公司吸收了大中型计算机中的数据库系统的优点, 充分考虑微型计算机本身的特点, 1982 年研制出适用于微型计算机的数据库系统——dBASE II 关系数据库系统。

dBASE II 关系数据库系统的出现拓宽了微型计算机的应用领域, 推动了办公室自动化、企事业管理及财务管理等方面的现代化进程。但由于 dBASE II 系统运行速度慢, 只允许同时最多打开两个数据库文件; 每个数据库文件最多不超过 32 个字段; 数值精度低等自身条件的限制, 人们开始感到它已经远远不能满足多方面应用的要求。

dBASE III 关系数据库系统是在 dBASE II 关系数据库系统的基础上发展起来的。dBASE III 保留了 dBASE II 的全部优点和功能, 同时加强了 dBASE II 许多命令的功能, 弥补了 dBASE II 的缺点与不足, 增加了许多新的命令和函数, 扩充了系统的功能和使用范围。特别着重解决了运行速度慢、同时允许打开数据库文件少、数据库文件字段少、数据精度低等问题, 使 dBASE III 系统比 dBASE II 系统更加完善, 功能更强, 运行速度更快。但是 dBASE III 系统仍然没有完全满足用户的要求, 它仅仅还是作为单个计算机上运行的系统。

dBASE III PLUS 关系数据库系统是在 dBASE III 关系数据库系统的基础上发展起来的, 是 1986 年推出的 dBASE 系统最新版本。dBASE III PLUS 关系数据库系统不但保留了 dBASE II、dBASE III 系统的全部优点和功能, 扩充了许多 dBASE II、dBASE III 命令的功能, 增加了许多新命令和函数, 而且特别着重解决了广大用户迫切希望解决的数据共享问题。dBASE III PLUS 可以运行在局部网络系统上, 使多个用户能同时存取共享盘中的数据, 提供了相应的安全和保密措施, 保证了数据的正确性与可靠性。同时, dBASE III PLUS 可对程序进行伪编译加密, 加快程序的运行速度, 防止程序遭到破坏。

dBASE III PLUS 关系数据库系统比 dBASE II、dBASE III 更加完善、功能更强、适应性更广。另一方面 dBASE III PLUS 系统与 dBASE III 系统是兼容的,即用 dBASE III 编制的程序不必进行任何的修改就可以在 dBASE III PLUS 系统上运行,这就大大增强了 dBASE III PLUS 的生命力。

目前, dBASE II 和 dBASE III 关系数据库系统已经深受用户的广泛欢迎,在国内外得到了很好的应用,取得了良好的社会效益和经济效益,被誉为“大众化数据库”。可以肯定,不久的将来, dBASE III PLUS 关系数据库系统将会在我国工农业、企事业、大专院校以及各级管理部门得到广泛的普及与应用,成为微型计算机应用软件的主要开发工具。

dBASE III PLUS 关系数据库系统由丰富的功能比较强的数据说明语言、数据操作语言、程序控制语言、系统工程环境设置命令、网络控制和操作命令以及全屏幕编辑修改命令等组成。具有如下几个特点:

一、采用人机会话的操作方式,具有友好的用户界面。

例,下面的命令企图将字符串“李静”保存在内存变量 `mname` 中。但命令中有错,通过会话的方式,使命令逐步修改成正确。

```
. STORE "李静" TP mname          &&命令语法错
Unrecognized phrase/key word in command.
```

?

```
STORE "李静" TP mname
```

```
Do you want some help? (Y/N) No    &&提示需要帮助,若需要则系统将显示
                                     正确的语法格式
```

```
. STORE "李静" TO mname            &&重新输入正确的命令
李静                                &&结果
```

当完成一条命令输入后,系统自动进行分析、检查语法,如果有错误,系统指出出错的位置,根据用户的需要可以给出标准的语法格式供用户参考。运行的结果分别可在打印机和显示器上输出。

二、具有单命令和程序两种工作方式。

例,执行下列一组命令:

```
. SET TALK OFF          &&把命令的执行结果不发送到屏幕上
. SET MENU OFF          &&在执行全屏幕命令期间不显示菜单
. SET ESCAPE ON         &&在按 ESC 键时停止命令文件的执行
. SET DEVICE TO PRINT   &&把@...SAY 命令的结果送往打印机
```

上面每条命令执行是采用单命令工作方式进行的,即每键入完一条命令,系统就执行该命令。我们也可以采用程序的工作方式,将上述一组命令,组织在一起以文件 `ENVI.PRG` 的形式保存。键入程序调用 `DO` 命令完成。

```
. TYPE ENVI.PRG
SET TALK OFF
SET MENU OFF
```

SET ESCAPE ON
SET DEVICE TO PRINT

. DO ENVI

三、dBASE III PLUS 命令直观、易于理解和记忆。dBASE III PLUS 命令和函数所使用的词以及运算符与英语中使用的词和数学中使用的符号基本相同。

例如：“LIST”词的英语意思是“列表显示”，在 dBASE III PLUS 中它也代表“列表显示”，它使当前数据库文件中的数据顺序显示出来。“ACCEPT”的意思是“接受”，在 dBASE III PLUS 中它也代表“接受”，它从键盘上接收用户所输入的数据。

四、dBASE III PLUS 命令具有英语自然语法结构的特点，结构灵活。

例：从当前数据库文件中输出所有男同学的姓名：

. LIST ALL 姓名 FOR 性别="男"

如果将 FOR 子句与姓名互换，也得到相同的结果。

. LIST ALL FOR 性别="男" 姓名

如果将 ALL 子句移到后面，也得到相同的结果。

. LIST FOR 性别="男" ALL 姓名

五、具有灵活的屏幕管理。系统规定一种标准的屏幕输入/输出格式。同时允许用户根据自己习惯定义不同的屏幕输入/输出格式。

六、数据高度独立。dBASE III PLUS 系统的数据存放形式与应用程序无关。当应用程序发生了变化，数据存放形式并不需要随之改变；同样，数据存放形式发生了改变，应用程序不需要随之变化。

例：在数据库文件中，数据是按姓名、性别、出生年月、婚否、民族、工资、个人简历的顺序存放。下面是显示输出数据库文件中数据的程序。

DO WHILE .NOT.EOF()

 @ 1,4 SAY 姓名

 @ 1,24 SAY 性别

 @ 2,4 SAY 出生年月

 @ 2,24 SAY 婚否

 @ 3,4 SAY 民族

 @ 3,24 SAY 工资

 WAIT

 SKIP

ENDDO

RETURN

若上述应用程序发生改变，将第三句与第四句互换，先取出生年月，后取性别显示。

DO WHILE .NOT.EOF()

 @ 1,4 SAY 姓名

 @ 1,24 SAY 出生年月

 @ 2,4 SAY 性别

```

@ 2,24 SAY 婚否
@ 3,4 SAY 民族
@ 3,24 SAY 工资
WAIT
SKIP
ENDDO
RETURN

```

这时，数据存放形式不改变，同样可以满足要求。反之，若数据存放顺序改变为姓名、出生年月、性别、婚否、民族、工资、个人简历，应用程序可以不加改变。

七、可与其它语言互通数据。dBASE III PLUS 关系数据库系统不但可以从 BASIC、MULTPLAN 等高级语言调用数据，而且也可以将自身的数据传送给 BASIC、MULTPLAN 等高级语言使用。

八、实现数据的共享。dBASE III PLUS 系统支持局部网络，使多个用户共同存取共享盘中的数据 and 文件。并提供相应的安全和保密措施。

例如：对局部网络共享盘，用户 1 所生成的数据，在一定限制条件下，用户 2 也允许存取。同样，用户 2 所生成的数据，在一定限制条件下，用户 1 也允许存取。

1.2 数据库文件

2.1 数据库文件的组织

先看一个数据库文件在计算机中的组织形式和存放形式。

例如，现有一张人员情况登记，如表 1-1。

表 1-1

姓 名	性别	出 生 年 月	婚否	民族	工 资	个 人 简 历
邹 平	男	1981.10.17	否	汉	65.00	
李 静	女	1958.4.21	婚	满	127.00	
刘建军	男	1956.2.23	婚	汉	145.00	
王小力	男	1963.2.7	否	汉	54.00	
苏 伟	男	1945.10.4	婚	回	179.50	
赵燕华	女	1954.3.29	婚	汉	137.50	

在计算机中，表 1-1 是以文件的组织形式被存放在磁盘上的。假定文件名为 people.dbf，其组织如下：

Record # 姓 名 性别 出生年月 婚否 民族 工资 个人简历

1	邹平	男	10/17/61	.F.	汉	87.50	Memo
2	李静	女	04/21/58	.T.	满	127.00	Memo
3	刘建军	男	02/23/56	.T.	汉	145.00	Memo
4	王小力	男	02/07/63	.F.	汉	54.00	Memo
5	苏伟	男	10/04/45	.T.	回	179.50	Memo
6	赵燕华	女	03/29/54	.T.	汉	137.50	Memo

从上面可以看出：

(一) 一张二维表，称为一个关系。在计算机中是以文件的组织形式存放在磁盘。该文件称为一个数据库文件。

(二) 垂直方向。二维表中每个栏，在数据库文件中都有一个字段与其相互对应。例如：二维表中“姓名”栏，与数据库文件中“姓名”字段相互对应；二维表中“工资”栏，与数据库文件中“工资”字段相互对应等等。

(三) 水平方向。二维表中每一行填写一个人的具体数据，而在数据库文件中就与一个记录相互对应。如：二维表中（李静，女，1958.4.21，婚，满，127.00，个人简历）一行，在数据库文件中就与（2，李静，女，04/21/58，.T.，满，127.00，memo）记录相互对应；二维表中（王小力，男，1963.2.7，否，汉，54.00，个人简历）一行，在数据库文件中就与（4，王小力，男，02/07/63，.F.，汉，54.00，memo）记录相互对应。

(四) “Record #”意为记录号，表示该记录在数据库文件中的位置号。如：“3，刘建军，男，02/03/56，.T.，汉，145.00，memo”表明在第3个记录位置；“5，苏伟，男，01/04/45，.T.，回，179.50，memo”表明在第5个记录位置。

(五) 在数据库文件中，每一列，即每一个字段必须有一个名字，该名字称为此列的字段名，同一数据库文件中的不同的字段必须具有不同的字段名，不允许重名。

(六) 在数据库文件中，每一字段中所有的数据值的类型都相同，数据值的类型，称为数据类型。

1.2.2 文件名

数据库文件名必须用字母打头，可含有字母、数字和下划线，最多不超过8个字符组成。

后缀部分为 .DBF。文件名内不允许嵌入空格符，也不允许用单个 A、B、C、D、E、F、G、H、I、J 这 10 个字母（包括小写的）作为文件名。

字母：大小写 26 个英文字母

数字：0、1、2、3、4、5、6、7、8、9 等数字

下划线：_

例：下面都是合法的文件名

A123 BASE

PEOPLE B_4

qwe

例：下面是非法的文件名

123A 不能以数字开头

A/B */* 是非法的
TELE_NUM 中间含有空格

1.2.3 记录

数据库文件是由记录组成的，记录是由字段组成的。dBASE III PLUS 允许一个数据库文件中最多可有 10 亿个记录，最多可有 128 个字段。每个记录各个字段内容总和最多不超过 4000 个字节。但是，若数据库文件中含有一种明细 memo 字段。如上一例中的“个人简历”字段便是此类型。它可以使每个记录最多达到 512K 个字节。1K 等于 1024。记录号是按记录输入计算机的顺序编排的。

1.2.4 字段

在 dBASE III PLUS 中，字段由字段名、字段类型和字段宽度三部分组成。

字段名必须由汉字或字母开头，可由汉字、字母、数字和下划线组成，最多不超过 10 个字符。字段名中间不能有空格。

例：下面都是合法的字段名

a	姓名	性别
abc	成绩	工作单位
zip_code		

下面列出的是无效的字段名

A₁₂₃, B₄
Reading,
姓_ _名

在 dBASE III PLUS 中，字段共有五种类型：字符型（C 型）、数字型（N 型）、逻辑型（L 型）、日期型（D 型）和明细型（M 型）。

一、字符型字段及宽度

字符型字段的数据值是由可打印的汉字和字符（包括空格）构成的。用字母 C 表示。宽度为该字段中最大的字符数。系统所允许最大宽度为 254 个字符。例如：“姓名”字段便是一个字符型字段，定义字段宽度为 6 个字节，对应的 6 个数据值是邹平、李静、刘建军、王小力、苏伟和赵燕华。

二、数字型字段及宽度

数字型字段的数据值是由正负号、小数点、0 至 9 数字所构成的、可进行算术运算的数据。用字母 N 表示。数字的宽度为该字段最大整数位、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。例如：字段“工资”是数字型的，最大小数位为 2 位，通常工资是不超过 1 千元的，因此最大整数位为 3 位，小数点一位，所以“工资”字段宽度为 6 个字节，小数位为 2 个字节。

三、逻辑型字段及宽度

逻辑型字段的数据值是由真（.T.）或假（.F.）所构成的、用字母 L 表示。在输入时，逻辑真值（.T.）也可用 t、Y 或 y 来表示；逻辑假值（.F.）也可用 f、N 或 n 来表示。字段的宽度固定为 1 个字节，用户不能选择。例如：字段“婚否”便属于逻辑型，已

婚用逻辑真 (.T.) 来表示, 未婚用逻辑假 (.F.) 来表示。

四、日期型字段及宽度

日期型字段是用于存放日期的数据。用字母 D 表示。一般格式是“月/日/年”。其中: 月、日、年均两位数字。例如: “03/25/86”表示 1986 年 3 月 25 日。“05/07/87”表示 1987 年 5 月 7 日。这种字段宽度固定为 8 个字节, 用户不能选择。日期型数据可以和数字相加减, 两个日期型数据可以相减。例如: 字段“出生年月”就可以定成日期字段, 宽度为 8 个字节。另外, 系统允许用户定义日期型格式为“日/月/年”。

五、明细型字段及宽度

明细型字段是用于存放大块的 ASCII 码数据, 数据被存储在数据库文件的辅助文件 (.dbt) 中。它在数据库文件中是用 memo 一词指明的。用字母 M 表示。明细型字段在数据库文件中的宽度是固定为 10 个字节, 用户不能选择。

明细型字段存放数据的宽度是可变的, 最多可达 5000 个字节。如果另一个字处理程序或文本编辑程序被指定为 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。例如: 字段“个人简历”是一个明细型字段。

1.3 常数和内存变量

1.3.1 常数

常数是在命令或程序执行过程中不改变的数据, 共有四种类型: 字符型、数字型、逻辑型、日期型。

一、字符型常数

字符型常数是由可打印的汉字和字符 (包括空格) 构成的, 必须被包含在定界符内。有三种定界符: 单引号 ' '、双引号 " "、方括号 []。

例: 下面是合法的字符型常数

'12.35', 'ABC', 'Y', "姓名", [红的], [王力]

而下面是非法的字符型常数

'24.50", [姓名", "工厂', '干部]

当字符型常数本身就包括定界符时, 那就应该选择另外一种异于该定界符的定界符作为真正的定界符。例如: 'abc "def" ghi', [abc 'def' ghi] 都是合法的字符型常数, 前者常数为 abc "def" ghi, 后者常数为 abc 'def' ghi。而 "abc "def" ghi" 和 'abc 'def' ghi' 则是非法的字符型常数。

二、数字型常数

数字型常数是由正负号、小数点和 0 至 9 数字构成的。例如: 123.55, 147, +200, -150。

三、逻辑型常数

逻辑型常数是由真假值构成的。

.T., .t., .Y., .y. 代表“真”

.F., .f., .N., .n. 代表“假”

四、日期型常数

日期型常数一般格式为：月/日/年。月、日、年分别为二位数字。例如：09/25/87 表示 1987 年 9 月 25 日；11/07/87 表示 1987 年 11 月 7 日。

1.3.2 内存变量

例如：. SUM ALL 工资 TO tot

. ? tot

第一条命令是求出应发放工资的总数，第二条命令是显示应发放工资的总数。我们称 tot 为内存变量。

dBASE III PLUS 变量共有两种，内存变量和字段。内存变量是存在内存之中，而不在于数据库文件中。它的设置给用户使用数据库系统带来更多的方便。用户可以随意定义自己的内存变量，作为中间结果保留，而后再作进一步的计算和作数据处理使用；或用来表示对数据库进行某种分析处理后得的数据结果。如上面例子，字段工资的内容求和，其结果可暂存放在内存变量中，然后可以直接引用内存变量的内容作计算显示。内存变量也经常用来控制程序流程。

一、内存变量名

内存变量名必须以一个字母或汉字开头，可由字母、汉字、数字和下划线组成，中间不允许有空格。最长不超过 10 个字符。

二、内存变量类型

例：先给下述几个内存变量赋值。

- . 姓名 = “李静”
- . 年龄 = 27
- . 日期 = ‘10/24/60’
- . 出生年月 = CTOD(日期)
- . 婚否 = .Y.

上述各个内存变量的类型

. DISPLAY MEMORY

姓名	pub	C	“李静”
年龄	pub	N	27(27.00000000)
日期	pub	C	“10/24/60”
出生年月	pub	D	10/24/60
婚否	pub	L	.T.

5 variables defined, 38 bytes used

251 variables available, 6034 bytes available

从第三列可以看出，内存变量姓名、日期是字符(C)型，年龄是数字(N)型，出生年月是日期(D)型，婚否是逻辑(L)型。

内存变量有四种类型：字符型、数字型、逻辑型和日期型。

字符型 通常用于存储字符串，但也可以用 CHR 函数把二进制序列存储为字符串。字符

型变量最多可包含 254 个字符。字符型内存变量占有的内存总空间是变量的实际长度加两个字节。

数字型 用于存放在计算机中使用到的数值,数字型变量的精度达到 15.9 位(不计小数点)。而对于非零数字的比较,最多可有 13 个数字的精确度。

逻辑型 逻辑型变量用以存放真(.T.)假(.F.)值。逻辑型变量总是占用一个字节,但在内存中需要占用两个字节。接受 T、t、Y、y 为真, F、f、N、n 为假。当由键盘生成逻辑型变量时,逻辑值两边必须有圆点。

例: . STORE .T. TO Tlogic
. STORE .F. TO Flogic

日期型 用于存放日期。日期型变量的长度总是占用 8 个字节,但在内存中需要占用 9 个字节。当从键盘生成日期变量时,必须使用 CTOD 函数。通常采用美国格式,即:月/日/年。

三、内存变量的限制

内存变量同时最多只能有 256 个,且共占内存空间不超过 6,000 个字节。

四、内存变量的使用

内存变量可由 STORE 〈表达式〉 TO 〈内存变量〉和 〈内存变量〉= 〈表达式〉两种方式初始化。

例: . STORE '王力' TO name
. name='王力'

第一条命令是初始化内存变量 name 为 '王力'。第二条命令等价于第一条命令。

内存变量名可以随意定义,但唯一限制是以 〈内存变量〉= 〈表达式〉方式初始化内存变量时,内存变量名不能与 dBASE III PLUS 命令动词同名。

例:下面是合法的初始化内存变量

. Mtotal=0

建立一个内存变量 Mtotal,初始值为 0 的数字型变量。

下面是非法的初始化内存变量

. total=0

total 是 dBASE III PLUS 的命令动词,系统认为是语法错。

内存变量与当前数据库文件的字段名同名时,取字段而不取内存变量。如果要取内存变量,可在内存变量名前加前缀 M→。

例如: . USE people
. STORE '王 红' TO 姓名
. ? 姓名

第一条命令是打开数据库文件 people;第二条命令是初始化内存变量 '姓名' 为 '王红';第三条命令执行结果是根据字段优先原则,显示出 '邹平',而不是内存变量的内容 '王红'。如果要显示内存变量的内容,则应改为:

. ? M→姓名

王 红

内存变量通常用作存放临时数据,可以随时用 RELEASE 命令直接清除。

例：上例内存变量 Mtotal 不需要时，可用下面命令清除。

. RELEASE Mtotal

可用 SAVE 命令以内存文件形式将变量存储到磁盘上保留，用 RESTORE 命令随时将磁盘上的内存文件调回到内存中。

例：. SAVE TO SMMM

. RESTORE FROM SMMM

第一条命令是将当前内存变量全部保存在磁盘 SMMM 内存文件中。第二条命令是将磁盘 SMMM 内存文件调回到内存中。

1.4 运算符、运算规则和表达式

dBASE III PLUS 定义了四种类型的运算符：算术、比较、逻辑和字符串运算符。

1.4.1 算术运算符

算术运算符可以对表达式进行算术运算，并产生数值运算结果，除两个日期型数据相减外，共包括六种运算符：

加号 + 减号 - 乘号 * 除号 / 乘方 **
或 ^ 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。括号无大、中、小之分，一律用 ()，可以在 () 内再套用 ()。

同级运算符则自左至右顺序运算。

用图 1.1 表示优先顺序为：

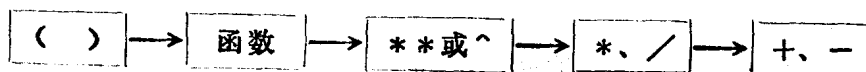


图 1.1

注意：

1. 在表达式中 * 号不能省略，也不能用 “.” 代替，例如：不能把 ma*3 写成 ma3，4*5 写成 4.5，否则就会和变量 ma3 和小数 4.5 相混淆。

2. 乘方 ^ 表示乘幂，如：A^B 表示 A^B，B 可以是算术表达式和函数，但它应该用括弧括起来。

例如：. ? 3^3.1

30.14

. ? 4^(INT(2.4)+1)

64.00

函数 INT 表示取整，如 INT(2.4) 得为 2。

3. 在表达式中，变量或常数必须都是数字型的。

例：. STORE 13.4 TO ma

. ? ma*2