

新一代
高性能
关系数据库系统

汉字 dBASE IV

沈金发 郑甫京

清华大学出版社

新一代高性能关系数据库系统

汉字 dBASE IV

沈金发 郑南京 编著

清华大学出版社

JK

内 容 简 介

dBASE IV 是 Ashton-Tate 公司 1988 年 10 月在 dBASE III PLUS 的基础上推出的 dBASE 最新系统。它对 dBASE III PLUS 的功能进行了进一步的完善与扩充，并引进了 SQL 语言，使其适用范围更广。

本书全面地介绍了 dBASE IV 系统，并对有关重要问题进行了深入的分析与阐述。全书共分 21 章，书末有 6 个附录。

本书可作为数据库课程的教学参考书，也可作为各类数据库培训班的教材，并可供广大计算机工作者自学或参考。

为了配合本书的学习，本社软件部提供 dBASE IV 软件。

新一代高性能关系数据库系统

汉字 dBASE IV

沈金发 郑南京 编著

责任编辑 李幼哲

☆

清华大学出版社出版

北京 清华园

北京昌平环球科技印刷厂印刷

新华书店总店科技发行所发行

☆

开本：787×1092 1/16 印张：33.5 字数：794 千字

1990年7月第1版 1990年7月第1次印刷

印数：00001~20000

ISBN 7-302-00669-5/TP·230

定价：11.50 元

前 言

由于 dBASE III、dBASE III 以及 dBASE III PLUS 在我国广泛的应用, 并取得了良好的效果, 广大计算机工作者早已期待功能更加完善的 dBASE 新一代产品 dBASE IV 的推出。

dBASE IV 是 Ashton-Tate 公司 1988 年 10 月推出的最新的 dBASE 系统。dBASE IV 在总结 dBASE 多年的应用经验, 对原有的 dBASE 系统进行广泛深入的研究与分析, 并且吸收很多其它系统 (如 Foxbase, Oracle 等) 的优点的基础上, 对 dBASE 系统进行了多方面的改进, 尤其是引进了 SQL 语言, 因而功能更强, 系统更合理, 适应范围更宽广。

dBASE IV 是我国广大计算机工作者与用户等待已久的系统。了解、学习和使用 dBASE IV 是大家的愿望, 希望有一套介绍该系统的资料。本书正是应这种需要而编写的。

本书对 dBASE IV 系统作了全面详细的叙述, 特别对一些常用的和基本的功能作了深入的阐述。对读者可能遇到的一些困难问题作了细致的分析, 并提供相应的解决方法。

本书在编写过程中得到了多方面的帮助与支持。在此特向他们表示衷心的感谢。

本书由清华大学计算机软件教研组沈金发、空军后勤部自动化工作站郑甫京同志编写。

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教材, 也可作为计算机用户、计算机工作者的自学教材、参考资料与手册。

由于时间仓促, 水平有限, 难免有缺点与错误, 敬请广大读者指正。

作 者

目 录

第一章 dBASE数据库系统	1
1.1 dBASE系统的发展.....	1
1.2 dBASE命令的使用方法.....	2
1.3 dBASE命令的构成和基本规则.....	3
1.4 数据显示命令 (?/??) 和赋值命令 (STORE)	6
1.5 记录、字段和数据值.....	9
1.6 常数、变量和函数.....	10
1.7 运算符、运算规则和表达式.....	13
第二章 数据库生成	19
2.1 数据库文件.....	19
2.2 建立数据库结构.....	20
2.3 全屏幕编辑.....	25
2.4 数据输入.....	27
2.5 打开和关闭数据库文件.....	30
第三章 数据库的操作	32
3.1 数据定位.....	32
3.2 查看数据.....	36
3.3 修改数据.....	39
3.4 插入数据.....	45
3.5 删除数据.....	48
第四章 数据的应用	54
4.1 数据组织.....	54
4.2 数据检索.....	63
4.3 多重数据库文件.....	68
4.4 数据库文件连接.....	71
4.5 数据库文件间的更新.....	73
4.6 求和、求平均值和分类求和.....	75
4.7 财务、统计运算.....	77
4.8 字段、记录筛选.....	79
4.9 数据库文件间的关联.....	83
4.10 现场文件和查询文件.....	91
第五章 数据库的管理和维护	93
5.1 数据库结构的维护.....	93
5.2 索引文件、多重索引文件的管理.....	97
5.3 明细型字段的管理	103
5.4 数据库文件间的数据交换	106

5.5	其它维护命令	109
5.6	环境设置和查看	112
5.7	目录文件的使用	122
第六章	dBASE IV 函数及其应用	126
6.1	三角函数及其应用	126
6.2	算术函数及其应用	129
6.3	财务函数及其应用	134
6.4	字符操作函数及其应用	135
6.5	日期函数及其应用	141
6.6	检测函数及其应用	144
6.7	转换函数及其应用	152
6.8	标识函数及其应用	156
6.9	环境函数及其应用	157
6.10	输入函数及其应用	158
第七章	程序设计	163
7.1	程序设计方法——结构化程序设计	163
7.2	交互式命令	164
7.3	选择	166
7.4	循环	173
7.5	数据库的循环操作	178
7.6	程序调用和参数传递	179
7.7	注释与文本语句	185
7.8	内存变量操作	186
7.9	错误捕获和处理	192
7.10	自定义函数	195
7.11	事务管理	198
7.12	程序文件的建立和执行	200
第八章	数组	203
8.1	下标变量	203
8.2	数组和数组定义语句 DECLARE	203
8.3	数组赋值、显示和删除	208
8.4	数据库文件与数组之间的数据转换	214
8.5	数组在命令和函数中的应用	223
第九章	屏幕输入输出设计	225
9.1	显示器	225
9.2	屏幕输入输出设计命令 @	225
9.3	屏幕画面美化	231
9.4	功能符和模式符的应用	233
9.5	屏幕格式文件	248
9.6	窗口	250
9.7	ON READERROR 命令	254
第十章	打印机输出设计	255

10.1 打印报表	255
10.2 打印标签	256
10.3 利用@命令设计打印机输出格式	258
10.4 使用? /? ? 和? ? ? 命令设计打印机输出格式	260
10.5 ON PAGE、EJECT和EJECT PAGE 命令	263
10.6 系统内存变量与打印机管理	265
第十一章 调试技术	277
11.1 问题的提出	277
11.2 单命令方式下的查错与改错	279
11.3 跟踪执行程序	281
11.4 断点设置和恢复	288
11.5 全屏幕程序调试编辑命令	289
第十二章 dBASE IV 与外部接口	292
12.1 与外部文件交换数据	292
12.2 调用 DOS 命令和外部程序	297
12.3 调用二进制程序文件	298
第十三章 用户可定义菜单	302
13.1 菜单定义	302
13.2 接收与处理	305
13.3 激活和关闭菜单	308
13.4 显示菜单	313
13.5 应用举例	313
第十四章 SQL基础	317
14.1 SQL 语言的发展与特点	317
14.2 SQL 命令	318
14.3 数据结构 TABLE (表格) 的定义	319
14.4 表格数据的插入、修改和删除	322
14.5 窗口	325
14.6 索引	326
14.7 数据库与目录	327
第十五章 基本查询	329
15.1 简单查询	329
15.2 SQL 的内部函数	334
15.3 谓词的应用	334
15.4 排序显示	337
15.5 记录的分组查询	338
15.6 并操作	340
第十六章 连接与嵌套查询	341
16.1 二元连接查询	342
16.2 多元连接查询	343
16.3 单层嵌套查询	343
16.4 多层嵌套与相关嵌套查询	345

16.5 关键字ANY和ALL, 谓词EXISTS	347
第十七章 SQL与dBASE	349
17.1 SQL方式中使用dBASE 命令与函数	349
17.2 访问dBASE数据库文件	352
17.3 数据的传送	353
17.4 保存 SQL 的查询结果	354
17.5 使用权的授予和收回	355
第十八章 数据共享	358
18.1 网络	358
18.2 数据的一致性	358
18.3 数据的保护	360
18.4 死锁处理	363
18.5 出错处理	367
18.6 网络命令与函数	367
第十九章 数据库保护	375
19.1 dBASE 安全系统	375
19.2 安全系统的建立	377
第二十章 菜单系统	383
20.1 菜单系统简介	383
20.2 控制中心	386
20.3 BROWSE和EDIT	388
20.4 编辑	394
20.5 数据库设计	399
20.6 现场查询和更新查询设计	404
20.7 格式设计	422
20.8 报表设计	429
20.9 标签设计	437
20.10 打印	440
20.11 应用程序	443
20.12 目录 Catalog	444
20.13 工具 Tools	446
20.14 dBASE IV 图形功能	455
第二十一章 dBASE IV 安装和启动	458
21.1 dBASE IV 的工作要求	458
21.2 单用户dBASE IV 安装和启动	458
21.3 多用户dBASE IV 安装和启动	464
21.4 存取控制——用户增加和删除	471
21.5 dBASE IV 配置	472
21.6 退出dBASE IV 系统	478
附录A 命令一览表	479
附录B SET 命令索引表	492
附录C 函数索引表	498

附录D 出错信息	501
附录E 网络出错信息	523
附录F 数据库文件结构	526
参考书目	527

第一章 dBASE 数据库系统

1.1 dBASE 系统的发展

60年代后期,随着计算机技术的发展与更新,随着计算机应用领域的扩大,人们对计算机管理提出了新的要求:要求管理的范围不仅限于某个局部,而是整体管理,如管理一个部门或一个单位的全部数据;管理的数据量也急剧增加;管理的限制也增多,例如要求数据的保密,数据的一致,数据使用的权限等等。因此,计算机技术的一个新的分支——数据库技术随之产生和发展。70年代初,关系数据库的产生与发展使数据库技术发展到一个新的阶段。关系数据库简单易学,为数据库的广泛应用奠定了技术基础。70年代后期,微型计算机日益普及,几乎渗透到了各个领域。如果使微型计算机配有关系数据库,那么它对计算机应用与普及的影响是极为可观的。1982年美国Ashton-Tate公司研制并推出了适用于微型计算机的关系数据库系统dBASE II。尽管dBASE II还相当粗糙,功能上有很多限制,有许多不尽人意的地方,但它却大大地拓宽了微型计算机的应用领域。尤其在我国,有不少单位与部门使用了dBASE II,进行了某些管理与应用。但是,由于dBASE II本身的一些固有的缺点,例如运行速度慢,数值精度低,文件字段数的限制,可同时使用的数据库文件数仅限于2个,这些都远远不能满足多方面使用的要求,限制了dBASE II进一步广泛的应用与普及。

dBASE III是在dBASE II应用的基础上,充分了解与发现dBASE II在应用中的不足,对dBASE II作了较大改进而开发出来的dBASE第二代产品。它充分地保留并发展了dBASE II简单易学等优点,同时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,弥补了dBASE II功能上的不足,并着重解决了运行速度慢、同时允许打开的数据库文件少、数据库文件字段少、数据精度低等问题。从而使dBASE III比dBASE II更完善,功能更强,运行速度更快。

尽管dBASE III解决了dBASE II的许多问题,但是,由于它还是作为在单个计算机上运行的系统,在使用上仍然受着相当的限制,从而限制了dBASE系统更广泛地普及与应用。因此,1986年Ashton-Tate公司推出了dBASE III PLUS系统。dBASE III PLUS在dBASE III的基础上又增加了许多新的命令与函数,扩充了dBASE III的功能,并着重解决了数据共享问题。dBASE III PLUS可以运行在局部网络系统上,使多个用户能同时访问服务器共享盘中的数据,并提供相应的安全保密措施。此外,dBASE III PLUS还可对程序进行伪编译加密,加快程序运行速度,防止程序遭到破坏与窃取。因此,dBASE III PLUS比dBASE III更完善,功能更强,适应性更广,实用性更强。

1988年10月Ashton-Tate公司推出了dBASE IV系统。dBASE IV是dBASE系统的最近版本,是dBASE系统的第四代产品。dBASE IV总结了dBASE系统多年应用的经验,对dBASE系统进行了全面分析,广泛地引进了其它关系数据库系统的优点。它是在dBASE III PLUS的基础上进行研制的。dBASE IV不但对dBASE III PLUS已有的命令进行了扩充,以增强其功

能, 还增加了许多新的命令、函数以及数组、自定义函数等。它还引进了控制中心 (Control Center), 以菜单驱动方式操作数据, 菜单形式灵活多样, 非常适用于 dBASE 的初学者。尤其是 dBASE IV 引进了 SQL 语言。SQL 语言在关系数据库的发展中具有举足轻重的地位, 它有许多优点, 它已经广泛地被计算机系统作为关系数据库系统的用户接口。SQL 语言的引进无疑对增强其系统功能大有益处。

1.2 dBASE 命令的使用方法

dBASE IV 的命令具有两种工作方式, 一种是单命令工作方式, 也称立即执行方式; 另一种是批命令工作方式, 也称程序执行方式。

1.2.1 立即执行方式

立即执行方式是在 dBASE IV 圆点提示符下输入 dBASE 命令, 然后按回车 (\leftarrow) 键, 通知计算机, 命令立即被执行。例如, 在当前磁盘目录下, 显示数据库文件清单, 在 dBASE IV 圆点提示符下输入

```
.DIR $\leftarrow$ 
```

执行的结果, dBASE IV 显示数据库文件的清单; 如果在当前磁盘目录中没有数据库文件, 则显示信息 “None”。

按回车键之前, 可以编辑已输入的命令。可以使用向左 (\leftarrow) 和向右 (\rightarrow) 键移动光标, 使用插入 (Ins) 键和删除 (Del) 键插入和删除字符, 还可以按 Esc 键撤消或删除整个命令。

在圆点提示符下, 命令行最多可输入 254 个字符, 如果命令行超过 254 个字符, 可按 Ctrl-Home 键打开编辑窗口, 在编辑窗口中输入命令行。按回车 (\leftarrow) 键, 结束命令行的输入, 关闭编辑窗口, 命令立即被执行。在编辑窗口中, 命令行最大可含有 1024 个字符。

如果输入的命令语法有错或者丢掉了必要的参数项, dBASE IV 将出现提示信息。此提示信息提供三种选择:

1. Cancel: 取消执行, 并消除命令, 返回到圆点提示状态。
2. Edit: 返回到圆点提示状态, 重新编辑, 编辑后按回车 (\leftarrow) 键, 再执行该命令。
3. Help: 需要 dBASE IV 提供帮助信息。

dBASE IV 在圆点提示符下有一个内存缓冲区, 它自动保存所输入的命令, 这个内存缓冲区称为历史缓冲区。当需要重复执行一条命令时, 不须重新输入该命令, 而是通过按上移 (\uparrow) 键和下移 (\downarrow) 键, 显示以前所输入的命令 (以相反的次序, 每次只显示一条命令。), 寻找所需要的命令, 按回车键进行执行。在按回车键执行之前, 可以利用标准编辑键对历史缓冲区命令进行编辑。

启动 dBASE IV 时, 历史缓冲区默认最多保存 20 条输入命令。但可以通过环境设置命令 SET HISTORY TO 随时改变这个默认值, 最大可设置保存 16000 条输入命令。有关历史缓冲区的设置及管理将在第十一章 “调试技术” 中详细讨论。

1.2.2 程序执行方式

把要执行的命令组织在一起，命名一个名字，以文件的形式存入磁盘，该磁盘文件称为程序文件（或命令文件）。这个程序文件是用 DO 命令调用执行的。

例如，执行下列一组环境设置命令：

```
. SET TALK OFF
. SET ESCAPE OFF
. SET DEVICE TO PRINTER
. SET AUTOSAVE ON
```

以上每条命令均在圆点提示符下，是采用单命令执行方式进行的，即每键入完一条命令，dBASE IV 就立即执行该命令。若采用程序执行方式，将上述四条命令组织在一起，以文件 ENVI.PRG 的形式保存到磁盘上，则键入程序调用 DO 命令，会得到同样的结果：

```
. TYPE ENVI.PRG
ENVI.PRG 10/17/89
SET TALK OFF
SET ESCAPE ON
SET DEVICE TO PRINTER
SET AUTOSAVE ON
. DO ENVI
```

1.3 dBASE 命令的构成和基本规则

1.3.1 dBASE 命令的构成

先举例说明典型的 dBASE 命令的形式。

假定现有一个学生数据库文件 student，它含有每个学生的姓名和数学成绩的数据，求数学不及格的人数，并打印出他们的姓名和成绩。

解决此问题可用下列一组 dBASE 命令实现：

```
. USE student
. COUNT ALL FOR 数学成绩<60
. LIST ALL姓名, 数学成绩 FOR 数学成绩<60 TO PRINTER
. CLOSE ALL
```

这四条命令的功能分别是：

- (1) 打开名字为 student 的数据库文件，以备对该数据库文件进行操作。
- (2) 在已打开的数据库文件 student 中，统计数学成绩在60分以下的人数。
- (3) 由打印机输出60分以下同学的姓名和数学成绩。
- (4) 关闭数据库文件 student，保存在磁盘上。

在 dBASE IV 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，但往往又比语句更精炼，功能更强。例如，如果使用一般的高级语言（如 BASIC、FORTRAN 等语言），则解决上例所提的问题就需要很多的语句，以便用来描述“打开数据

库文件”、“统计”、“打印”和“关闭数据库文件”等操作的详细过程。

从上例可以看出，dBASE IV 的数据库操作命令都是以命令动词开头，后面可以跟一个或多个子句。例如，第二条命令的命令动词为 COUNT，子句为 ALL 和 FOR。

下面列出 dBASE IV 数据操作命令的一般语法形式：

〈命令动词〉 [〈表达式表〉] [〈范围〉] [FOR〈条件〉] [WHILE〈条件〉] [TO FILE 〈文件名〉/TO PRINTER/TO ARRAY 〈数组表〉/TO 〈内存变量〉] [ALL [LIKE/EXCEPT 〈通配符〉]] [IN 〈别名〉]

可以看出，它可分成八个部分。下面分别进行说明：

1. 命令动词。它是 dBASE 的命令名，用来指示计算机要完成的操作，例如上面例子中的“USE”、“COUNT”等。

2. 表达式表。它是一个或多个由逗号分隔开的表达式，用来指示计算机执行该命令所操作的结果参数。例如，在上面例子的第三条命令中，表达式表为姓名、数学成绩。

3. 范围。它表明命令可以操作的记录。范围可有下列四种选择：

RECORD〈n〉——仅对指定的n号记录进行操作。

NEXT〈n〉——对从当前记录开始的n条记录进行操作。

ALL——对数据库文件中所有记录进行操作。

REST——对从当前记录开始到数据库文件结尾的记录进行操作。

· 例如，上面例子中第二、三条命令的范围均为 ALL。

4. FOR〈条件〉。它告诉 dBASE IV，命令仅对满足条件的记录进行操作。如果使用 FOR 子句，dBASE IV 将记录指针重新指向数据库文件顶，并且用 FOR 条件与每条记录进行比较。

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，也就是说，〈条件〉返回值必须是真的 (.T.) 值或假的 (.F.) 值。例如上面例子中的第二、三条命令，〈条件〉为“数学成绩 < 60”。当数学成绩低于 60 分时，条件“数学成绩 < 60”返回真 (.T.) 值；当数学成绩高于或等于 60 分时，条件“数学成绩 < 60”返回假 (.F.) 值。

6. [TO...]。它控制命令操作结果的输出。有些命令允许操作结果向文件输出，有些命令允许操作结果向打印机输出，有些命令允许操作结果向内存变量输出。但是，应该注意，允许操作结果向内存变量输出的命令必也允许操作结果向一个数组元素输出。

7. ALL [LIKE/EXCEPT〈通配符〉]。它告诉 dBASE IV，包括或不包括与通配符相匹配的文件、字段或内存变量。通配符是与文件名、字段名或内存变量名相符合的一般形式。在通配符中可使用符号 ? 和 *。? 代表任何单一的字符，* 代表一组任意字符。

8. [IN...]。它允许在当前工作区下操作其它工作区中的数据库文件。IN 子句可含有别名、字母、数字或赋给别名、字母、数字的表达式。

下面举两个例子，并详细说明每条命令中子句的作用。

例 1

DISPLAY	姓名, 职称, 工资	NEXT 6	FOR 工资 > 200.00	TO PRINTER
命令动词	表达式表	范围	FOR〈条件〉	[TO...]

命令动词“DISPLAY”指示计算机要完成“显示”数据操作。表达式表“姓名、职称、工资”指示计算机对每条记录“显示”出它的姓名、职称、工资字段数据。范围“NEXT 6”表明从当前记录开始连续对6条记录进行“显示”操作。FOR〈条件〉“FOR工资>200.00”表示对工资字段大于200.00的记录进行“显示”操作。[TO...]“TO PRINTER”控制“显示”的结果在打印机输出。由此可知,上述 DISPLAY 命令的功能是: 打印输出从当前记录开始连续6条记录中工资高于200.00元人的姓名、职称、工资。

例 2

SUM
工资
ALL
FOR 教研室 = '计算机'
TO mgz

命令动词 | 表达式表 | 范围 | FOR 〈条件〉 | [TO...]

命令动词“SUM”指示计算机要完成“求和”数据操作。表达式表“工资”指示计算机对工资字段进行“求和”操作。FOR〈条件〉“FOR 教研室 = ‘计算机’”表示对教研室字段为“计算机”的记录进行“求和”操作。[TO...]“TO mgz”控制“求和”结果存放在内存变量 mgz 中。由此可知,上述 SUM 命令的功能是: 计算“计算机”教研室所有人的工资总和, 将结果存入内存变量 mgz 中保存。

1.3.2 基本规则

dBASE IV 命令有以下规则:

1. 每条命令以命令动词开头, 必须严格符合本书所介绍的命令语法格式。

2. 命令中的子句可按任意次序排放。如下边三条命令是等价的:

LIST 姓名, 性别, 职称 NEXT 5 FOR 教研室 = “计算机”

LIST NEXT 5 姓名, 性别, 职称 FOR 教研室 = “计算机”

LIST FOR 教研室 = “计算机” NEXT 5 姓名, 性别, 职称

3. 命令中的子句可由若干个空格隔开, 每个空格也算一个字符。

4. 在单命令工作方式下, 一条命令的最大字符个数可达254个。如果一条命令的长度超过254个字符, 可通过按 Ctrl-Home 键打开编辑窗口。在编辑窗口中, 命令行的总长度最大可达1024个字符。

5. 命令动词和 dBASE IV 保留字均可用4个以上字母简写。例如, 下边三条命令是等价的:

DISPLAY MEMORY

DISPLA MEMO

DISP MEMO

6. 可以以小写、大写和大小写混合的形式输入命令行。

7. dBASE IV 中没有绝对不能使用的单词, 但最好不使用 dBASE IV 命令动词和保留字作为字段名、内存变量名、数组名以及文件名, 否则将有可能带来不必要的混乱。

例如, 以 status 命名的字段是无法由 LIST status 命令显示的。因为该命令将显示当前 dBASE IV 的系统状态, 而不是显示 status 字段的内容。

8. 不能使用单个字母 A 到 J 作为数据库文件名, 因为它们都是 dBASE IV 的保留字, 作为别名 (ALIAS)。

9. 程序设计时, DO WHILE/ENDDO, DO CASE/ENDCASE, IF/ENDIF, SCAN/ENDSCAN, PRINTJOB/ENDPRINTJOB 以及 TEXT/ENDTEXT 命令必须配对使用。

10. 程序设计时, 如果命令太长, 一行写不下, 可分几行写, 但每行 (最后一行除外) 末尾要使用一个分行符 “;”。

1.3.3 符号约定

在命令语法格式中, 本书对所使用的符号作如下约定:

[]: 方括号, 表示是可选的项目。若选择该项目, 不要写方括号本身。

< >: 角括号, 表示括号内的项目是必须要选的, 但不要写角括号本身。

/: 斜线号, 表示两个项目中选择其中一个项目, 但不要写斜线号本身。

…: 省略号, 表示前项可继续重复多次选择。项与项之间用逗号隔开。

1.4 数据显示命令(???)和赋值命令(STORE)

在 dBASE IV 中, 显示命令???)和赋值命令 STORE 是用的最多的命令。特别是显示命令?/??, 在程序设计中几乎每个程序中都要用到, 应很好掌握。

1.4.1 数据显示命令 ?/? ?

数据显示命令 ?/? ? 可以完成如下操作:

1. 显示变量、表达式和常数的值。

(1) 如某一变量已经赋值, 则可用单问号? 显示该变量的值。

例

```
. STORE 34 TO Mvar
```

```
34
```

```
. ? Mvar
```

```
34
```

(2) 用单问号? 可以显示表达式的值。

例 1

```
. STORE 4 TO Mvar 1
```

```
4
```

```
. STORE 6 TO Mvar 2
```

```
6
```

```
. ? (Mvar 1 + Mvar 2) / 2
```

```
5
```

例 2

```
. STORE 'ABC' TO Ma
```

```
ABC
```

```
. STORE 'DEF' TO Mb
```

```
DEF
```

```
. ? Ma + Mb
ABCDEF
```

可以看出，? 命令具有计算和输出显示的双重功能，在遇到表达式时，它先计算后显示输出计算的结果。

(3) 用 ? 命令可以原样显示输出常数。

例 1 显示输出数字常数

```
. ? 4
```

4

例 2 显示输出字符串常数

```
. ? "THIS IS A STRING"
```

THIS IS A STRING

应当注意，这里的引号是用来确定字符串起止和终止界限的，它本身不作为字符串中的字符。

2. 显示若干个变量、表达式和常数的值。

(1) 用一条? 命令可以显示若干个变量的值。用一条命令可以显示若干个表达式的值，也可用一条命令显示若干个常数。变量之间、表达式之间、常数之间需用一个逗号隔开。

例 1

```
. STORE 'AB' TO Mvar 1
```

AB

```
. STORE 'CD' TO Mvar 2
```

CD

```
. STORE 'EF' TO Mvar 3
```

EF

```
. ? Mvar 1 , Mvar 2 , Mvar 3
```

AB CD EF

例 2

```
. STORE 4 TO Mnum 1
```

4

```
. STORE 3 TO Mnum 2
```

3

```
. ? Mnum 1 + Mnum 2 , Mnum 1 - Mnum 2
```

7

1

(2) 用一条 ? 命令可以同时显示若干变量、若干表达式、若干常数的值。

例

```
. STORE 32 TO Mnum 1
```

32

```
. STORE 12 TO Mnum 2
```

12

```
. ? 'Mnum 1 + Mnum 2 = ' , Mnum 1 + Mnum 2
```

Mnum 1 + Mnum 2 =

44

3. 用 ? 命令可以输出一个空行。

若 ? 命令后面没有任何表达式,则表示本行什么也不显示输出,相当于输出一个空行。

例

```
. ? '3 * 4 = ', 3 * 4  
. ?  
. ? '3 * 5 = ', 3 * 5
```

结果

```
3 * 4 = 12  
  
3 * 5 = 15
```

4. 双问号 ?? 命令的使用等同于单问号 ? 命令的使用。它们的区别是:单问号 ? 命令从下一行开始显示输出,双问号 ?? 命令在当前光标位置开始显示输出。

例

```
. ? 'A', 'B', 'C'  
. ?? 'D', 'E', 'F'  
. ? 'G', 'H', 'I'  
. ?? 'J', 'K', 'L'  
. ? 'M', 'N', 'Q'  
. ?? 'P', 'Q', 'R'
```

结果

```
A B C D E F  
G H I J K L  
M N Q P Q R  
.
```

1.4.2 赋值命令 STORE

赋值命令 STORE 的主要功能是给内存变量提供数据。下面对它作一些说明。

(1) STORE 命令可以用于建立内存变量,并赋予内存变量初值。例如

```
. STORE 100 TO Mnum
```

它的作用是:建立内存变量 Mnum,赋予初值100。

(2) STORE 命令也可用于为已经建立的内存变量重新赋值。例如

```
. STORE 2 * Mnum TO Mnum
```

它的作用是: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。星号表示乘号。

(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。例如,建立内存变量 Ma、Mb、Mc、Md,初始化为 0:

```
. STORE 0 TO Ma, Mb, Mc, Md
```

(4) 当 STORE 命令给单个内存变量提供数据时,可以写成<内存变量>=<表达式>形式。例如

```
. Mnum = 10 * Mnum
```

完全等价于