

dBASE II 数据管理

福 建 电 子 计 算 机 公 司

中国计算机技术服务公司福建分公司

前 言

Ashton-Tato 软件公司推出的微型机关系数据库dBASE II以它强有力的数据处理功能和使用方便等特点占领了市场,并为大众所欢迎。可称为“大众数据库”,美国的微型机大多数都配上了dBASE II,IBM PC也不例外。

随着IBMPC在国内日益推广,dBASE II的用户愈来愈多,为了帮助用户能较快地掌握dBASE II,我们在黄新王等译本的基础上,按国内技术用语改译了这本使用说明,由于我们编译的蓝本是PA2000 II的dBASE II使用说明,难免与IBMPC dBASE II有些差异。但我们把该使用手册中的例子在IBM PC dBASE II上试运行,全部通过。只是一些命令的提示不一样而已。不容置否,这本手册完全适应于IBM PC dBASE II,可做为该用户的使用手册。

由于编者水平有限,难免有些不对的地方,望读者提出宝贵意见以资修正。

编 者

84.6.

序

电脑科技日新月异，硬件价格不断地下跌，但性能却愈来愈强。因此，目前许多人都有想买电脑的欲望与需求，同时也有买得起（微）电脑的能力。然而唯一使大家踟躇不前的因素，往往在于“不知如何使用电脑的烦恼”。面临这个难题，又倍受工作繁重的压力，可真是伤透了脑筋。dBASE II 是美国最畅销的微电脑套装软件之一，专门用来处理、管理与储藏各种数据。由于它的容易使用，加之功能上的令人满意，已使得使用电脑的烦恼问题获得了解决。该套系统采用简易的英语词句和自动问答方式，与你相交谈。一般人只要经过大约20小时的学习后，就有能力使用电脑来处理数据，改善繁重的实际工作，并立即享受电脑的好处了。这正像参加短期的汽车驾驶训练，就能享受汽车的便利一样。

dBASE II 除了本身优越的功能外，它的说明书深入浅出也是它之所以能畅销的因素之一。另外尚有一个很重要的畅销因素，那就是它能在广泛的8080，8085或Z-80微电脑上使用（像TRS-80/II，APPLE II加Z-80卡，和国内几家微电脑系统）。本人从事信息产品的开发与推广，为使我国能早日享受dBASE II管理数据库的好处，特邀黄新王、骆德廉两位信息界新秀编译dBASE II的中文说明书。期盼本书的出版，有助于读者对dBASE II的善用。文内瑕疵之处在所难免，尚祈诸位先进多予惠教。时值出版，特为之序。

郑育饒

目 录

第壹篇	dBASE I 数据库管理系统	(1)
第零章	dBASE I 系统简介	(1)
第一章	简单实用的命令	(3)
1—1	如何建立数据库 (CREATE)	(3)
1—2	新数据库的数据储存工作	(5)
1—3	用EDIT来编辑数据	(6)
1—4	全屏幕编辑的用法: MODIFY, APPEND, EDIT	(7)
1—5	dBASE I 命令简介与错误更正对答	(8)
1—6	利用表示式和关系运算符来扩充命令 (LIST)	(9)
1—7	用DISPLAY来显示数据	(11)
1—8	位置命令GO, GOTO和SKIP	(12)
1—9	交谈用的“?”命令	(14)
1—10	用APPEND和INSERT命令来扩充数据	(15)
1—11	清除数据库(DELETE, RECALL, PACK)	(16)
1—12	摘要	(18)
第二章	有扩充弹性的dBASE II 命令	(20)
2—1	利用表示式作选择与控制	(20)
2—2	常数与变量	(20)
2—3	dBASE II 的运算符	(24)
	(1) 算数运算符	(24)
	(2) 关系运算符	(25)
	(3) 逻辑运算符	(26)
	(4) \$: 子字符串逻辑运算符	(27)
	(5) 字符串运算符	(28)
2—4	改变空数据库的结构 (MODIFY)	(29)
2—5	复制数据库与结构 (COPY)	(30)
2—6	数据栏的增删	(33)
2—7	与CP/M或其他外来数据文件的交换使用 (COPY, APPEND)	(36)
2—8	用COPY和APPEND重改栏名	(37)
2—9	修改数据的捷径 (REPLACE, CHANGE)	(38)

2—10	组织你的数据库 (SORT, INDEX)	(39)
2—11	找出想要的信息 (FIND, LOCATE)	(41)
2—12	从数据中获得信息 (REPORT作报表命令)	(43)
2—13	自动计数和累加 (COUNT, SUM)	(45)
2—14	数据的摘取与多余部份的清除 (TOTAL)	(46)
2—15	摘要	(47)
第三章	命令文件是什么	(48)
3—1	写出第一个命令文件	(48)
3—2	选择与决定 (IF...ELSE)	(49)
3—3	重覆一段处理 (DO WHILE...)	(51)
3—4	过程 (PROCEDURE子命令文件)	(52)
3—5	输入交谈数据 (WAIT, INPUT, ACCEPT)	(53)
3—6	屏幕上任意位置的数据信息显示 (@...SAY...GET)	(53)
3—7	综合所学的一个命令文件	(56)
3—8	多个数据库的操作 (PRIMARY, SECONDARY, SELECT) ...	(58)
3—9	常用的系统命令和函数	(59)
3—10	程序设计与命令文件规划	(60)
第四章	实用的函数与命令	(62)
4—1	以函数扩充控制	(62)
4—2	变更dBASE II的基本特性与自定值	(65)
4—3	两个数据库的数据并换 (UPDATE)	(66)
4—4	数据库的结合 (JOIN)	(67)
4—5	全屏幕编辑与定格式 (@...SAY...GET...PICTURE)	(68)
4—6	报表的输出格式 (SET FORMAT TO PRINT, @...SAY... USING)	(69)
4—7	表格的建立与印出	(69)
4—8	实用范例	(70)
第五章	dBASE II 系统组织与摘要	(72)
5—1	数据库的基本观念	(72)
5—2	数据库组织简介	(73)
5—3	dBASE II的记录、文件和数据类型	(74)
5—4	dBASE II的文件种类	(75)
5—5	dBASE II的运算符摘要	(76)
5—6	dBASE II的函数摘要	(76)
5—7	dBASE II的命令摘要	(77)
5—8	dBASE II命令的功能分类	(81)

	(1)文件结构方面.....	(81)
	(2)文件操作方面.....	(82)
	(3)编排数据库方面.....	(82)
	(4)混合数据库方面.....	(82)
	(5)编辑、更新、修改数据方面.....	(82)
	(6)变量使用方面.....	(83)
	(7)对谈输入方面.....	(83)
	(8)搜寻方面.....	(83)
	(9)输出方面.....	(83)
	(10)程序设计方面.....	(84)
第六章	代理商之会计系统范例.....	(85)
6—1	命令文件功能摘要.....	(86)
6—2	数据库的数据结构摘要.....	(90)
6—3	报表格式摘要.....	(94)
6—4	系统常数摘要.....	(96)
6—5	各程序模式之列表.....	(96)
第貳篇	如何在PA 2000电脑上使用dBASE II.....	(178)
第一章	dBASE II数据库管理系统简介.....	(178)
第二章	dBASE II磁盘片.....	(180)
第三章	PA2000系统的基本 操作 程序.....	(183)
3—1	开机操作.....	(183)
3—2	键盘.....	(184)
3—3	CRT 屏 幕.....	(184)
3—4	磁盘片.....	(185)
3—5	打印机 (或称印字机)	(186)
第四章	PDOS操作系统与dBASE II数 据 库 管 理 系 统.....	(187)
4—1	PDOS操作系统的三大部份.....	(187)
4—2	PDOS系统中常用的命令.....	(187)
4—3	PDOS (CP/M) 基本操作命令的用法.....	(188)
4—4	交谈标志 (prompt)	(189)
4—5	PDOS与dBASE II两个系统之关系.....	(190)
4—6	操作命令之格式.....	(191)
4—7	命令按错键时之更正.....	(192)
4—8	dBASE II 错误命令之 处 置.....	(193)
第五章	在PA2000之磁 盘 片 上 建 立 一 个 数 据 库.....	(194)

第三篇	dBASE II 命令使用说明	(198)
第一章	dBASE 的使用法	(198)
第二章	能够使用 dBASE 的系统所需之部件	(200)
第三章	dBASE 文件的种类	(201)
3—1	数据库文件	(201)
3—2	记忆文件	(202)
3—3	命令文件	(202)
3—4	报表格式的文件	(203)
3—5	文字输出文件	(203)
3—6	索引文件	(203)
3—7	格式文件	(203)
第四章	表达式	(204)
4—1	函数	(205)
4—2	运算	(209)
第五章	巨集替代	(212)
第六章	与非 dBASE 处理程序之间的接口	(213)
第七章	命令的分类说明	(214)
第八章	全屏幕运行	(217)
第九章	命令的定义及说明	(219)
9—1	符号定义	(219)
9—2	操作的规则	(220)
附录 A		(323)
附录 B		(330)
附录 C		(333)
附录 D		(334)

第一篇 dBASE II 数据库管理系统

第 零 章

dBASE II 系统简介

dBASE II 是一套管理数据库的工具。所谓数据库，系为许多具有相关性质的各种数据汇集在一起，并籍固定的方式予以编排存放，而形成了很有科学化的数据集合。象图书馆的藏书数据卡，企业公司的业务数据和户政事务所的户籍数据……等等均是。一般在管理这些数据时都很头痛，即费时又费工，但dBASE II提供了一些〔类似英语词句〕的简单命令，便能让你易如反掌地处理这些中小型数据库。dBASE II的功能甚强，好处不胜枚举。下面先列举几个特点来看看。

- * 能最便捷又容易地建立各类的完整数据库。

- * 用最节省的数据拷贝文件，就能在你的数据库上轻松地做编辑、显示、添加、删除和印报表等工作。

- * 享受程序与数据间彻底分开独立的好处；意即数据有了变动时，用不着再费神修改程序；反之亦然。这是其他程序语言所望尘莫及的。

- * 可以随时预先建立好那些要印的数据报表格式，并于真正印报表时，它叫自动地帮你做加减、乘除、分项和或总和，以及他种数据操作……等等想做的工作。因此你再也不必随时伺候在电脑身旁了。

- * 利用全屏幕编辑功能作屏幕输出格式的规划；所以任何复杂的数据管理工作，都可以变成“按项选做”、“填单子”和“与电脑对谈”般的容易。

- * 所有数据管理的工作，都可以写成dBASE II的命令程序来完成。因此你的例行工作或特殊情况，都可以写成程序自动控制。

- * 对杂乱的数据能做分类或索引；使得数据看起来赏心悦目，找起来方便神速！由上可见，dBASE II不愧是套神奇的系统，同时也是数据处理者的大救星，话虽如此，仍得请您花点小时间看看这本手册；学学怎么用，才能让你变成“数据处理的大王牌”。如果你不懂电脑，也没学过什么BASIC, COBOL……程序语言；没关系，等你学完这本手册之后，定会惊讶到你不仅懂电脑了，会写程序了，也会把电脑应用到你的工作或事业上了；甚至，成为同事、同学或同业间的佼佼者。

本手册的符号说明:

1. 小写字: 黑色屏幕块内的小写字, 代表我们所打进电脑的文字。
2. 大写字: 黑色屏幕块内的大写字, 代表dBASE II所反应的交谈信息或结果。而文章里的大写字则指dBASE II的命令。
3. ^……^: 此符号内的点点点代表你从终端机打进去的dBASE II命令语句。但前后的^符号请不要打入。
4. [……]: 方括号内的东西, 表示为dBASE II命令语句内的一部分, 但可任由选择要或不要。前后的方框记号也请不要打入。
5. <……>: 尖括号内的东西, 也是dBASE II命令语句内的一部分。但必须填入切实的数据: 例如: <filename>意即要填某个文件的名称。前后的尖字号也一样不要打入。
6. <enter>: 文章内凡有出现此字样的地方, 代表请你按RETURN键一下或按ENTER键一下(有些终端机没有此键)。

所要求的电脑系统:

dBASE II需要下面一些硬件和软件的工作环境:

- 8080, 8085或Z-80CPU的微电脑系统。
- 最少要有48K的内存容量(dBASE II所占的位置从5CH到A 400H)。
- CP/M(型号1.4或2点几的)、CDOS、CROMIX或全亚电脑的PDOS操作系统。
- 一个或一个以上的巨量储存装置, 如软盘驱动器或硬盘均可。
- 一台光标可任移式的终端机(以便全屏幕编辑之用)。
- 一台打印机。

dBASE II的系统规格:

- 每个数据库文件最多能放65535个记录(record)。
- 每个记录最多存1000个字符(character)。
- 每个记录最多有32个栏(field)。(信息组)。
- 每栏最多254个字符。
- 能用的数值范围: $\pm 1.8 \times 10^{63} \sim \pm 1 \times 10^{-63}$ 。
- 数字精确度为10个有效位数。
- 字串(string)最长为254个字符。
- 每行命令语句(Command Line)最长为254个字符。
- 报表的标题文字最长254个字符。
- 索引关键的名称, 最长100个字符。
- 在SUM命令里, 最多5个表达式(expression)。

其他注意事项:

在还没使用dBASE II之前, 最好先将原版的软盘复制一份, 再拿那片复制的来使用, 以免原版被破壞掉。另外, 请先在CP/M系统上执行INSTALL, COM程序, 并依据该程序所问的, 衡量你的设备而给予合适的回答, 以便选用与置妥有关dBASE II的工作环境。

第一章

简单实用的命令

首先，我们学习如何把收集来的数据建成一个数据库，并以很有系统的方式，将这些数据储存起来。另外，再由浅入深地介绍一些以后会常用来处理数据的有趣命令。在本章里，我们先学会怎么用即可。至于较复杂的命令，或较完整的定义，后面或下册都会讲到。

1-1 如何建立数据 (CREATE)

我们先从建立 (CREATE) 一个取名为“邮递格式系统”的数据库的例子作为开始吧。假设此数据库里的每一个记录 (record)，均含有下述的信息：

姓名 (NAME)：可达20个字符长。

地址 (ADDRESS)：可达25个字符长。

城市 (CITY)：可达20个字符长。

州 (STATE)：2个字符。

邮递区号 (ZIP CODE)：5个字符。

当你从CP/M操作系统进入dBASE II数据库管理系统后，先打进^CREATE^命令。随后，dBASE II便会回答你：ENTER FILENAME (讲打入数据库的文件名称)。

于是你可以按进一个到八个 (受CP/M系统的规格限制) 不具有空白或标点符号的英文字符的文件名称。即然它是个文件的名称，我们姑且命它为“名称”，而打进“NAMES”接着敲一下RETURN键后，dBASE II便马上为你建立一个叫做<NAMES.DBF>的数据库文件。而逗号后面的字符为CP/M的扩充名称。同时也是“数据库文件”的英文缩写 (DataBase File)

在数据库管理系统中，一群具有相互关系的项目汇排在一块时，我们称该群为一个记录 (record)；而每个项目均有明确的定义，我们称它为栏 (field)，即如我们的例子里，每个记录就有5个栏。dBASE II必须清楚每栏你所定义的数据，包括：每栏的名称，可放那类型的数据、数据的长度、以及小数点后面能有几位 (如果数据为数字型时) 等等信息。

```
. Create
ENTER FILENAME: names
ENTER RECORD STRUCTURE
AS FOLLOWS:
    FIELD NAME, TYPE, WIDTH, DECIMAL
    001      PLACES
```

母栏规定最多可用十个以下的大小写字符来命名：并规定要以字母起头，中间不得夹有空白，但可以有数字或单号（：）。命名时最好能符合实际意义，同时别缩写到只有电脑看得懂，而别人看不懂的地步。至于数据的类型（Type），则以一个字母来区分：C代表字符型（Character），N代表数字型（Numeric），L代表逻辑型（Logical）。本例中，每一栏均放字符型的数据。

每一栏的长度（Size）最多为254个字符。如果该栏是放数字或十进位数时，记住！小数点也占掉一个字符。我们既然知道每栏该给的名称、数据的类型、以及长度之后，便可将这些信息打进电脑。假设你所打的像下面这样：

```
.Create
ENTER FILENAME: names
ENTER RECORD STRUCTURE
AS FOLLOWS:
      NAME, TYPE, WIDTH, DECIMAL
FIELD  PLACES
001    name,c,20
002    address,c,25
003    city,c,20
004    state,c,2
005    zipcode,c,5
BAD NAME FIELD
005    zipcode,c,5
006    <return>
```

你是否注意到第5栏里，我们犯了一个错语。那就是在栏名内有个空白，所以dBASE II告诉我们错了，并给予修改机会。另外邮递区号的数据类型，被定为“字符型”；即使如此，我们仍可将它当成数字来看待。

之所以这么定，那是为了避免dBASE II的某类命令（像^TOTAL^）在执行时。会将所有记录内同一数字栏的数据，做总和计算（不必你特地去指定它们）。所以倘若对邮递区号做总和，那是毫无意义的事；这就是为什么把它定为字符型的原因。但是仍可用「关系运算符」（relational operators），像「大于」、「小于」、「等于」或「不等于」，来运算字符型的数据；如此才不致影响到邮递区号的分类工作。

当dBASE II要求我们对第6栏下规格时，我们若是敲了RETURN键来结束规格定义的工作，dBASE II便会存起这个数据库结构（Data Structure），并问你想不想把数据储存进去。接着这个<NAMES,DBF>的数据库，便开始等候你的决择输入了。请按“Y”，让我们开始储存数据！

1-2 新数据库的数据储存工件

假使你的终端机上并没有全屏幕的编辑功能：每个记录的编号以及每栏的栏名，只能于打完一行数据时，再从下面一行显示出来。（每一栏的长度）是由前后两个冒号所钳制。而光标（Cursor）则停在可以开始储入数据的位置。当你填满数据或按 RETURN 时，下面一栏也就跟着出现。依此填完最后一栏时，又可以再继续填下一个新记录。

RECORD 00001	
NAME	:
ADDRESS	:
CITY	:
STATE	:
ZIP: CODE	:

假使 dBASE II 是在全屏幕编辑作业下：整个画面会被清除掉，然后该记录编号以及所有栏名，同时会依序排列在屏幕的左上角；并且光标会停在第一栏的头一个输入位置。（注意：假使你对这种显示的方式不满意，而想要有反白或半亮度的字时：那么请再利用 INSTALL.COM 命令，重新在它的“Y—CHANGE/MODIFY”步骤里作调整。）

每一栏的长度仍由前后两个冒号（:）的间距所表示。填满一栏或按了 RETURN 键，光标便跳到下一栏。光标也可以移回前面一栏作修改，只要你按一下 Control—E 同时按“控制键”和“E”键：简写为 Ctl—E。当填完最后一栏时，dBASE II 会自动再补一个空记录，准备再填数据。

请照下列的人名与地址将数据打入 <Names.DBF> 数据库内，我们准备用这个例子来表演 dBASE II 的命令功能。

（姓 名）	（地址）	（城市）	（州）	（邮递区号）
ALAZAR, PAT	123 Crater Rd.,	Everett, WA		98206
BROWN, JOHN	456 Minnow pl.,	Burlington, MA		01730
CLINKER, DUANE	789 Charles Dr.,	Los Angeles, CA		90036
DESTRY, RALPH	234 Mahogany St.,	Deerfield, FL		33441
EMBRY, ALBEPT	345 Sage Ave.,	Palo Alto, CA		94303
FORMAN, ED	456 Boston St.,	Dallas, Tx		75220
GREEN, TERRY	567 Doheny Dr.,	Hollywood, CA		90046
HOWSER, PETER	678 Dusty Rd.,	Chicago, IL		60631

万一你打错了，并且没法用倒退键（Backspace）或盖过（overwrite）来修改时，请先阅读本章三、四节内的编辑方法后再动，要是不幸退回到 dBASE II 的点标志（dot prompt）时，请你再打。

^USE Names^

^APPEND^

此时你又可以恢复储存数据的工作了(这两道命令在后面作说明)。

当填完最后一个邮递区号时,光标便跑到下一个记录第一栏的起头位置。此时只要按一下RETURN键,便可以结束储存数据的工作。假使你又填进了一些数据,那就得按Control-Q键(一起按“控制键”和“Q”键),方能结束。此时dBASE II便离开储存数据状态,而出现点标志(.),再等待你下达新命令给它。储完数据后,如果不想再做,打个^QUIT^命令,便可以退出dBASE II的掌握了。在执行^QUIT^命令时,dBASE II会自动关闭所有的文件,并清除暂用的数据,以免意外的因素破坏了你所建的数据库。

1—3 用EDIT来编辑资料

假使在储存时打错了,现在可以利用全屏编辑状态(Full Screen Edit mode)来做修改工作。请打进:

^USE Names^

^EDIT<number>^

└ 库中记录的编号。

这儿的“number”系指本数据库中某一个记录的编号。dBASE II会替你找出那个记录;此时运用全屏编辑的操作方法,你就能轻松地修改它的任何数据。若想继续改下一个记录,请按ctl-C键一次;同样地,按ctl-R键,便能倒退一个位置。下面让我们来试试看,先打“EDIT 3”。

RECORD 00003	DELETED
NAME : CLINKER, DUANE :	
ADDRESS : 789 Charles Dr.	
CITY : Los Angeles	
STATE : CA;	
ZIP; CODE : 90036;	

另者,想删掉该个记录就按ctl-U键。屏幕上头瞬时出现了“DELETED”(被删除)的字样。若再按一下ctl-U键,又可复原该个记录,同时“DELETED”字样也消失了。现在可用^LIST^(列出)或< DISPLAY^(显示)命令,来看看你所存的数据。你定会注意到所有被关掉的记录前,都出现了一个星号(*)。该星号只是用来标明该个记录已被删除(但没消失,请注意此点)。

如果你只是打着玩的,并不是真地想改变数据库里原有的内容,那么按ctl-Q键便能脱离全屏幕编辑操作。不过按了ctl-W键(若用Superbrain的终端机,请按ctl-O),便承认刚才所改的并将它们完全储存起来,而退回到dBASE II的受命状态(Command mode)。

1-4 全屏幕编辑的用法 MODIFY, APPEND, EDIT

下面列出全屏幕编辑操作时，所有可用的控制键之按法及其作用：

- ctl—X：把光标移到下一栏（或按ctl—F键）。
- ctl—E：把光标移到上一栏（或按ctl—A键）。
- ctl—D：光标向右方移一位。
- ctl—S：光标向左移一位。
- ctl—V：「插字与可盖过」的交互选择。
- ctl—G：删掉光标所指的一字。
- <Pubout>：删掉光标左边一字。
- ctl—P：打印机「用与不用」的交互选择。
- ctl—Q：不作任何更改而迳退返dBASE II的正常操作情况：即使在修改状态下(MODIFY mode)也一样。

MODIFY (修改) 命令下的控制键功能：

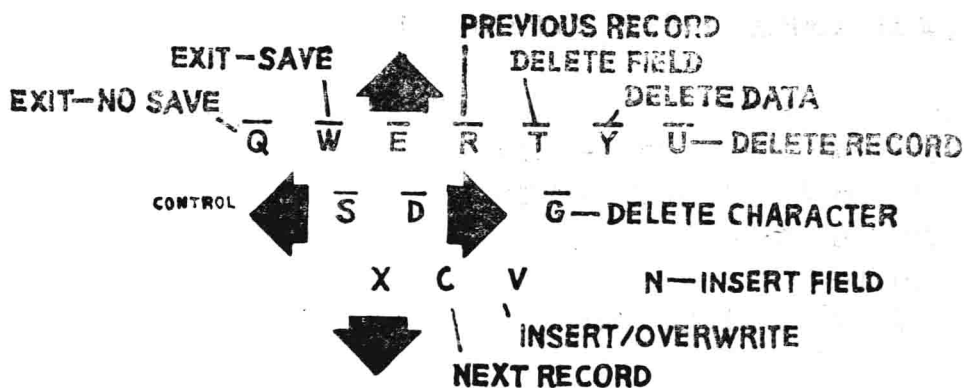
- ctl—T：删掉光标所指一栏，再将下面各栏往上挪。
- ctl—Y：清除该栏内的数据，并不影响他栏。
- ctl—N：将光标所在一栏前，插入一列空白，以便增建新栏。

APPEND (增补) 命令下的控制键功能。

- ctl—R：将该个记录先存进磁盘，并把光标移到下一个记录。
- <Return>：当光标停在一个新记录的起填位置时，若按此键便能重返dBASE II的正常操作。
- ctl—Q：擦除刚填的记录，迳退返dBASE II的正常操作。

EDIT (编辑) 命令下的控制键功能（不能在APPEND状态下使用）：

- ctl—C：将该记录写进磁盘，并前进一个记录。
- ctl—R：将该记录写进磁盘，并倒退一个记录。
- ctl—U：记录的「删除与复原」之交互选择。
- ctl—W：存入所有修改过的数据，再重返dBASE II的正常操作状态。（若用Superbrain的终端机，请改按ctl—O键）。
- ctl—Q：放弃任何记录内刚更改过的数据。并出现交谈标志(·)。按RETURN键后便重返dBASE II的正常操作。



1—5 dBASE II 命令简介与错误更正对答

dBASE II 的命令通常是动词，并且要等系统的点标志 (·) 出现后，才会进入受命状态(Command mode)。刚开始时先要把准备操作的数据库打开，即下 ^USE>filename > ^命令 (指明使用<某一文件名称>)。在数据库使用中，如果想看眼前的这个记录 (the current record) 请用 ^DISPLAY ^命令，若想看文件库里所有的文件，便打 ^LIST ^命令。dBASE II 的任何命令字，均可以简写到四个字母；但更多的字母也对。例如 ^DISPLAY ^， ^DISP ^或 ^DISPL ^都正确，但 ^DISPRAY ^就错了。

在dBASE II 正式启用前的选用与置妥中 (执行INSTALL COM命令)，若选择了“错误更正对答”。则当dBASE II 扫描所打进去的命令，侦测到有错误时，它即刻会发出错误信息：“UNKNOWN COMMAND” (： 看不懂的命令)。接着，会再给你第二次的修正机会，省得整行重打。

例如：打了 ^EDUT3 ^， dBASE II 会告诉你它看不懂，并出现☆☆☆UNKNOWN COMMAND等字样：

```

Edut3
☆☆☆UNKNOWN COMMAND
Edut3
CORRECT AND RRETY (Y/N)? y
CHANGE FROM : u
CHANGE TO   : j
Edit 3
MORE CORRECTIONS (Y/N)? n

```

接着dBASE II 会问你是否要再试改？如果你决定要，就打“Y”；因此可以省掉重打的麻烦。dBASE II 所显示的“CHANG FROM”，是问你要改那一部分？你只要把错误的部份重打一遍后，并按RETURN即可。它接着会问你：想改成什么？“CHANGE

TO:”。等你改完后,再按RETUTN键。dBASE II便自动帮你更正,同时问你要不要再多改点?“MORE CORRECTION(Y/N)?”:如果没问题便打“N”这种对答式的错误更正法,对很长的命令修改时特别方便。

〔提示〕: ^ERASE^命令可以将屏幕彻底清除,并在屏幕左上角示出点标志,以便接受新命令。
erase

1—6 以表达式和关系运算符来扩充命令 (LIST)

dBASE II有个极强的功能,那就是能扩充和缩剪命令语句。意即,你可以在大部份的命令字后面,添加叙述和表达式(expressions),以控制命令的执行情况。这些命令行(command lines)可用大写或小写字体,但最长不得超过254个字符。当命令行长过屏幕宽度时,只要在该行的最末一字,打个分号(;) (后面不可再打空白), dBASE II便将下一行视为此命令行的一部分。

既然dBASE II是一个关系式数据管理系统(Relational DBMS);因此有很多关系运算符(relational operators)是很有用的:

<: 代表小于

>: 代表大于

=: 代表等于

<=: 代表不大于(即小于或等于)

>=: 代表不小于(即大于或等于)

这些符号的意义,正如右边所解释的一样。所有的关系符在运算后,都会产生一个逻辑值(真或假)。假使结果是“真”,那么命令就会被完成;否则便不必做,在前面我们曾提到^LIST^命令是用来列出数据库里的所有记录(按ctl-S键,可以暂停/继续列出的动作)。在这儿,我们再介绍LIST的更多使用方法。它的完整格式是。

^LIST [OFF] [FOR<expression>]^

如果选用了[OFF],记录的编号便不会显示出来。另外选用了[FOR],dBASU II仅把能满足表达式为真的记录列出来。下面就举一些文字型数据的运算例子来看(其他型数据请看第二章):

^USE Names^

^LIST^

^LIST OFF^

^LIST FOR zip: code='9'^

^LIST OFF FOR zip: code<'8'^

^LIST FOR Name='GREEN'^

〔注意〕: 虽然你只填进了某栏的部分内容,但dBASE II仍拿该栏的所有信息来做比较。在本例中我们并不需要GREEN先生的全名,但当好几个CREEN先生出现在同一栏时,就非得写全名不可了。

• use names

• list

00001	ALAZAR, PAT	123 Crater Rd.	Everett	WA98206
00002	BROWN, JOHN	456 Minnow Pl.	Burlington	MA01730
00003	CLINKER, DUANE	789 Charles Dr.	Los Angeles	CA90036
00004	DESTRY, RALPH	234 Mahogany St.	Deerfield	FL33441
00005	EMBRY, ALBERT	345 Sage Avenue.	Palo Alto	CA94303
00006	FORMAN, ED	456 Boston St.	Dallas	TX75220
00007	GREEN, TERRY	567 Doheny Dr.	Hollywood	CA90046
00008	HOWSER, PETER	678 Dusty Rd.	Chicago	IL60631

• list off

ALZAH, PAT	123 Crater Rd.	Everett	WA98206
BROWN, JOHN	456 Minnow Pl.	Burlington	MA01730
CLINKER, DUANE	789 Charles Dr	Los Angeles	CA90036
DESTRY, RALPH	234 Mahogany St.	Deerfield	FL33441
EMBRY, ALBERT	345 Sage Avenue	Palo Alto	CA94303
FORMAN, ED	456 Boston St.	Dallas	TX75220
GREEN, TERRY	567 Doheny Dr.	Hollywood	CA90946
HOWSER, PETER	678 Dusty Rd.	Chicago	IL60631

• list for zip : code='9'

00001	ALAZAR, PAT	123 Crater Rd.	Everett	WA98206
00003	CLINKER, DUANE	789 Charles Dr.	Los Angeles	CA90036
00005	EMBRY, ALBERT	345 Sage Avenue	Palo Alto	CA94303
00007	GREEN, TERRY	567 Doheny Dr.	Hollywood	CA90046

• list for zip : code<'8'

00002	BROWN, JOHN	456 Minnow Pl.	Burlington	MA01730
00004	DESTRY, RALPH	234 Mahogany St.	Deerfield	FL33441
00006	FORMAN, ED	456 Boston St.	Dailas	TX75220
00008	HOWSER, PETER	678 Dusty Rd.	Chicago	IL60631

• list for name='GREEN'

00007	GREEN, TERRY	567 Doheny Dr.	Hollywood	CA90046
-------	--------------	----------------	-----------	---------

另外为了让你更能精细地选择数据库里的数据，这个LIST命令尚可提供其他的系统信息，例如：

LIST STRUCTURE：显示目前使用中的数据库之数据结构。