

第一章 汉字 dBASE III 的基础知识

dBASE III 是美国 ASHTON-TATE 公司 1984 年 7 月推出的关系数据库管理系统,是计算机管理系统的强有力工具。dBASE III 经过汉化,得到了汉字 dBASE III,适用于 IBM-PC、IBM-PC/XT、0520 及其兼容机。汉字 dBASE III 除保留了 dBASE III 的全部功能外,又增加了对汉字的处理能力。其功能较强,现已被广泛采用。

汉字 dBASE III 简单易学,具有丰富的提示信息和各种具体的出错指示,因为各种提示信息都使用中文,这就极大地提高了其普及推广的价值。利用汉字 dBASE III 能够建立完整的数据库管理系统,对数据库中的数据能很方便地实现增加、删除、查询、修改、显示和打印功能;能从一个或多个数据库中产生多种报表;能自动地进行计数、汇总、求和、求平均等操作。汉字 dBASE III 提供的编程命令可以完成各种类型的任务,所以,汉字 dBASE III 非常适合各类管理人员使用,尤其适合编制各类家庭管理软件。

1.1 系统的配置要求

系统的配置包括硬件配置和软件配置。

1.1.1 硬件配置

对中小型企事业单位、家庭的管理较好的微机型机仍然是广泛使用的 IBM-PC 及 IBM-PC/XT 机,因此硬件需要的配置有:

- 1、内存容量要在 640K 字节以上的主机;
- 2、有两个容量不小于 360K 字节的软盘驱动器,或一个容量不小于 360K 字节的软盘驱动器加一个硬盘存储器;
- 3、彩色或单色显示器一台,彩色显示器有利于汉字显示;
- 4、并行打印机一台,最好是 24 针打印机;
- 5、键盘一个,它是人机对话的输入设备。

1.1.2 软件配置

IBM-PC 及其兼容机均可配置汉字操作系统 CCDOS 和汉字 dBASE III 系统。

- 1、汉字操作系统:

汉字操作系统 CCDOS,其中包括 CCDOS 引导程序、批命令文件、汉字输入/输出等模块,通常配有汉字库模块。

- 2、汉字 dBASE III 系统

汉字 dBASE III 是在 CCDOS 控制下进行工作的,它由一系列文件组成,全部系统文件放在软盘或硬盘上,其主要文件有:

- 1)DBASE.EXE

总控程序及常驻内存模块,是一种系统扩展的可执行命令。

2)DBASE. OVL

dBASE Ⅲ的可覆盖模块,是解释执行各种命令的软件。

3)HELP. DBS

HELP 命令使用文件,帮助用户自学 dBASE Ⅲ各种命令,函数的语法、功能、使用方法说明。

4)ASSIST. HLP

帮助自学各种命令的软件。

5)CONFIG. SYS

操作系统参数设置文件,其内容一般可设为:

FILES=20

BUFFERS=24

上面的参数设置表示,在 dBASE Ⅲ运行时,要求 CCDOS 能够同时打开 20 个文件,24 个缓冲区。如果操作系统缺省 CONFIG. SYS 文件,系统默认为可同时打开 8 个文件,除了系统本身的输入输出操作占用的 5 个文件以外,只有 3 个文件可供用户使用,显然系统的默认状态不能满足需要。

在 CCDOS 状态下,可使用编辑 EDLIN 程序,建立或修改 CONFIG. SYS 文件。

1.2 汉字 dBASE Ⅲ 的启动、工作方式和退出

为了启动和运行汉字 dBASE Ⅲ,需要准备三张软盘,即 CCDOS 盘、汉字 dBASE Ⅲ 系统盘和用户工作盘,然后按以下步骤操作。

1.2.1 启动

1、启动 CCDOS 操作系统

将 CCDOS 盘插入 A 驱动器,打开主机电源,当在屏幕左上方出现提示符

A)

表示 CCDOS 启动成功,系统处于中西文输入状态。

若机器已处于通电状态,重新启动可把 CCDOS 盘插入 A 驱动器,同时按下<Alt>+<Ctrl>+三个键,可重新启动,从而进入 CCDOS。

如果 CCDOS 系统在硬盘上,则不用插入软盘面直接启动,进入 CCDOS 时,屏幕左上角显示

C)

2、启动汉字 dBASE Ⅲ

取出 CCDOS 盘,将 dBASE Ⅲ 系统盘插入 A 驱动器中,这时在 CCDOS 的提示符 A)下键入 dBASE Ⅲ 的启动命令。

A)DBASE

按回车键后;A 盘指示灯亮,不久屏幕上显示版本说明等文字信息,并在屏幕最左边出现圆点提示符

表示机器已在 dBASE Ⅲ 状态下,可以执行各种 dBASE Ⅲ 的合法命令。

如果 dBASE Ⅲ 已装在硬盘,则在 C)下直接键入

C)DBASE

也可进入 dBASE III 状态。

1.2.2 工作方式

进入 dBASE III 状态后,可以采用两种工作方式。

1. 单命令工作方式

也称键入命令方式。根据需要,在圆点提示符下,用户可以一次键入一条命令,得到相应的回答后,系统又回到圆点提示符,等待输入新的命令;这样逐条进行下去,直到完成任务为止。系统在接收命令时,若用户输入命令有错误,系统会提示用户输入错误,无法执行,并询问是否需要帮助。这种方式对于初学者学习,理解掌握 dBASE III 命令是很有好处的。这个特点是 dBASE III 的主要特点之一,是其它高级语言所缺乏的。

2. 批命令工作方式

也称程序方式。这种工作方式是有顺序地执行一批命令,这些命令预先已经存放在一个命令文件中,文件中的命令序列就是一个应用程序,其文件的扩展名为 .PRG。运行时,可用 DO 命令执行 .DO(文件名)

1.2.3 退出

当需要从 dBASE III 工作状态返回操作系统时,只要在圆点提示符下,键入 QUIT 命令即可实现。

.QUIT

C→dBASE III 运行结束。

屏幕上又出现提示符:

A)或 C)

表示系统已回到 CCDOS 状态。

1.3 汉字 dBASE III 的数据类型

在数据库中,存在着各种信息,信息的表现形式是数据。数据类型是指简单数据的基本属性,因为只有类型相同的数据之间才能进行操作,所以正确地掌握和运用数据类型是非常重要的。

1.3.1 数据类型

汉字 dBASE III 定义的数据类型共有五种,它们是:

1. 字符(C)型数据

字符型数据是指一切可印刷的中西文字符,包括字母、汉字、阿拉伯数字、各种符号、空格等。

2. 数字(N)型数据

数字型数据是指可以进行数字运算的数据,由阿拉伯数字、小数点、正负号组成。又分为整型、小数型两种。

3. 日期(D)型数据

日期型数据用于存放表示日期的数据,一般形式为“月/日/年”。

4. 逻辑(L)型数据

逻辑型数据用来表示逻辑判断结果的值,只能取真和假两个值。

5、记忆(M)型数据

记忆型数据是数据库文件特有的数据类型。是为实际应用中有时需存储更多字符而设置的。

1.3.2 常数

常数是在执行命令过程中不变的数据。在 dBASE Ⅲ 中常数有四种类型:字符型、数字型、日期型、逻辑型。

1、字符型常数

字符型常数是用定界符括起来的字符数字串,定界符有三种,单引号' '、双引导" "和方括号[]。当某一定界符本身是字符型常数的一部分时,就应当选择另一种定界符。

如"中国"、'北京'、[15ab]等都是字符型常数,"abc'adc'"也是字符型常数;但'abc'abc'"是不合法的。

2、数字型常数

数字型常数可以是整数、实数。

例如: 36.78、1000、-17.3 等。

3、逻辑型常数

逻辑型常数只有两个值,真值和假值。

真值可用.T.、.t.、.Y.、.y.表示。

假值可用.F.、.f.、.N.、.n.表示。

4、日期型常数

日期型常数用来表示一个确定的日期,其形式为月/日/年。

如: 12/25/91 表示 1991 年 12 月 25 日。

1.3.3 变量

变量是一种其内容可能要发生变化的数据。包括内存变量和字段变量。这里主要介绍内存变量。

内存变量是一种临时工作单元,需要时可以临时定义,用于存放常数、中间结果和最终结果,也可作为控制变量用于控制程序运行。内存变量存放什么类型的数据,该变量就有相应的属性。在汉字 dBASE Ⅲ 中,设置的内存变量的个数不超过 256 个,一个内存变量最多只能存放 254 个字符。

可以利用 STORE 命令建立一个内存变量并向它赋值,命令格式为:

STORE <表达式> TO <内存变量名>

或<内存变量名>=<表达式>

这里<内存变量名>代表一个内存变量,是由以字母或汉字开头的字符、数字、下划线组成的字符数字串,其长度不超过 10 个字符。例如:

.STORE "林一木" TO 姓名

.STORE 1992 TO VA

.STORE .F. TO L

.STORE 0 TO X, Y, Z

在上面这些命令中,内存变量的建立和赋值是同时进行的,在建立内存变量的同时,也赋给它一个确定的值,内存变量的数据类型与所赋数据类型是一致的。也可以说,赋值是将常数存储到内

存变量名标识的工作单元中;内存变量是由赋值语句定义的,它的数据类型取决于赋值数据类型。

上例中,姓名即是字符型变量,VA、X、Y、Z是数字型变量,L是逻辑型变量。

在单命令或批命令方式下,均可直接用内存变量名对其进行访问。但如果某一内存变量名与数据库中某字段名完全相同,为了区别这个同名变量是属于内存变量还是字段变量,在访问内存变量时,可以用下面格式输入内存变量:

M->内存变量名

如果不特别指定,则认为是字段变量。

1.4 汉字 dBASE III 的函数

函数是与常数、变量不同的另一种数据。在汉字 dBASE III 中共提供 37 个标准函数。使用函数时应注意,因为要处理各种不同类型的数据,所以函数本身要求的参数要有一定的数据类型,若类型不对,就会出现语法错误;对于合法的参数,函数必定有一个返回值,返回值也有其确定的类型。

要使用函数,最重要的就是掌握函数数据类型,要搞清要求的参数及返回值是什么类型,使得数据间操作不发生数据类型不对的错误。

这些函数按其功能可划分为五类:字符函数、数字函数、日期与时间函数、各种转换函数和专门测试函数。

1.4.1 字符函数

字符函数共有 8 个。

1.& 宏代换函数

格式: &(<字符型内存变量>)

功能:取内存变量的值。

例 1: .单位="中央财政金融学院"

.?"&.单位.信息系"

中央财政金融学院信息系

宏代换函数的作用范围是从符号 & 起,直到遇见一个句号“.”或空格为止。这里“.”作为内存变量结束符号。如果 & 后面不跟有效内存变量名,则 & 作为一普通字符。

2.LOWER 函数

格式: LOWER(<字符串>)

功能:将大写字母转为小写字母。

例 2: .? LOWER("This Is A Example")

this is a example

3.UPPER 函数

格式: UPPER(<字符串>)

功能:将小写字母转换为大写字母。

例 3: .? UPPER("This Is An Example")

THIS IS A EXAMPLE

4.SPACE 建立空格函数

格式: SPACE (<数字表达式>)

功能:产生一个由空格组成的字符串,空格个数由<数字表达式>给定。

例 4: . ? "北京" + SPACE(10) + "天安门"
北京 天安门

5、SUBSTR 子串选择函数

格式: SUBSTR (<字符串>[,<起始位置>[,<子串长度>]])

功能:从字符串中取出一个子字符串,并规定了子串的起始位置和长度。若起始位置大于字符串长度,则输出一个空串;若缺省子串长度,或子串长度大于起始位置到末尾长度时,则输出从起始位置到字符串结束。

例 5: . ? SUBSTR("中国民航",5,4)
民航

6、AT 子串寻找函数

格式: AT (<字符串 1>,<字符串 2>)

功能:当<字符串 1>是<字符串 2>的子字符串时,返回值为<字符串 1>在<字符串 2>的开始位置。若<字符串 1>不是<字符串 2>的子字符串,则返回值为 0。

例 6: . ? AT ("民航","中国民航")
5

7、TRIM 删除字符串尾部空格函数

格式: TRIM (<字符串>)

功能:删除字符串尾部的空格

例 7: . STORE "河南省" TO S
. STORE "郑州市" TO D
. ? S+D
河南省 郑州市
. ? TRIM (S) + D
河南省郑州市

8、LEN 字符串长度函数

格式: LEN (<字符串>)

功能:给出字符串的长度,返回值为<数字>。

例 8: . ? LEN ("中国民航")
8

1.4.2 数字函数

数字函数共有 5 个。

1、EXP 自然指数函数

格式: EXP (<数字表达式>)

功能:给出 e^x 的值。

例 9: . ? EXP (1.000)
2.718

2、LOG 自然对数函数

格式: LOG ((数字表达式))

功能: 给出自然对数 $\ln X$ 的值。

例 10: . ? LOG (2.718)

1.000

3、SQRT 平方根函数

格式: SQRT ((数字表达式))

功能: 给出平方根的正值。

例 11: . ? SQRT (2)

1.414

4、INT 取整函数

格式: INT ((数字表达式))

功能: 给出数字表达式的整数部分。

例 12: . ? INT (15.69)

15

5、ROUND 四舍五入运算函数

格式: ROUND ((数字表达式),i)

功能: 给出数字表达式相应的四舍五入值, i 为结果中保留小数的位数, 当 i 为负数时, 则四舍五入到对应的整数位, 其绝对值为整数部分的位数。

例 13: . ? ROUND (3.14159,4)

3.1416

. ? ROUND (456.12, -1)

460.00

1.4.3 日期与时间函数

日期与时间函数共有 8 个。

1、DATE 系统日期函数

格式: DATE ()

功能: 给出系统日期。格式为 mm/dd/yy。

例 14: . ? DATE()

12/15/92

2、TIME 系统时间函数

格式: TIME()

功能: 给出系统时间, 以 hh: mm: ss 表示。

例 15: . ? TIME()

11:01:34

3、MONTH 月份函数

格式: MONTH ((日期型表达式))

功能: 给出代表月份的数字。

例 16: . ? MONTH (DATE())

12

4、DOW 星期几函数

格式: DOW ((日期型表达式))

功能: 给出表示星期几的一个数字。

例 17: . ? DOW (DATE())

2

5、DAY 日期函数

格式: DAY ((日期型表达式))

功能: 给出一个代表日期号的一个数字。

例 18: . ? DAY (DATE())

15

6、YEAR 年函数

格式: YEAR ((日期型表达式))

功能: 给出一个四位表示年份的数字。

例 19: . ? YEAR (DATE())

1992

7、CMONTH 月份名函数

格式: CMONTH ((日期型表达式))

功能: 给出月份的中文名称。

例 20: . ? CMONTH (DATE())

十二月

8、CDOW 星期名函数

格式: CDOW ((日期型表达式))

功能: 给出星期几的中文名称。

例 21: . ? CDOW (DATE())

星期二

1.4.4 各种转换函数

转换函数共有 6 个。

1、ASC 求 ASCII 码函数

格式: ASC ((字符串))

功能: 将(字符串)中最左边字符转换成 ASCII 码值。

例 22: . ? ASC ("ABC")

65

2、CHR 求字符函数

格式: CHR ((n)) ($0 \leq n \leq 255$)

功能: 显示代码为 n 的 ASCII 码字符。

例 23: . ? CHR (65)

A

3、STR 数字型转换成字符型函数

格式: STR ((数字表达式) [, 长度] [, 小数位数])

功能:计算数字表达式的值,转换成字符型数据。其中长度为输出字符串字符个数,包括小数点和小数部分;小数位数决定小数部分输出的位数,若无此项,原数据中小数部分将被舍弃。

若所定义的长度不够,则输出一串" * "号。

例 24: . ? STR (123.456,7,3)

123.456

. ? STR (123.456,3)

123

4、VAL 字符串转换成整数函数

格式: VAL (<字符串>)

功能:将(<字符串>)前面的数字字符转换成整数。若其前面为非数字字符,返回值为 0。对含有小数的数字序列,将删去小数点及小数部分(可用 STR 来恢复)。

例 25: . ? VAL ("3A")

3

5、CTOD 字符型转换日期型函数

格式: CTOD(<字符串>)

功能:将格式为"xx/xx/xx"形式的字符串转换成日期型数据。

例 26: . Y=CTOD("03/21/92/")

. ? Y+10

03/31/92

6、DTOC 日期型转换为字符型函数

格式: DTOC (<日期型表达式>)

功能:把日期型数据转换成字符型数据。

例 27: . ? DTOC (DATE())

12/15/92

1.4.5 专门测试函数

测试函数共有 10 个,利用测试函数可以随时了解系统的某些时实状态。

1、TYPE 类型测试函数

格式: TYPE (<表达式>)

功能:测试(<表达式>)的数据类型,返回一个表示类型的字符,其中 C、N、D、L、M 分别表示字符型、数字型、日期型、逻辑型和记忆型。若(<表达式>)不存在或有错误,用 U 表示。

例 28: . ? TYPE (DATE())

D

2、BOF 测试文件起始函数

格式: BOF ()

功能:测试文件是否开头,返回值为一个逻辑值。若记录指针位于文件开头,则返回值为真值,否则为假值。

3、EOF 测试文件结束函数

格式: EOF ()

功能:测试文件是否结束,返回值为一个逻辑值。若记录指针位于文件末尾,返回值为真值,否

则为假值。

4、DELETED 测试删除标记函数

格式：DELETED ()

功能：测试记录是否有删除标记。如果当前记录已被加上删除标记，则返回值为真值，否则为假值。

5、FILE 测试文件函数

格式：FILE ("文件名")

功能：测试指定的文件是否存在，若存在则返回值为真值，否则为假值。

6、RECNO 测试记录号函数

格式：RECNO ()

功能：给出当前记录的记录号，若库文件中无记录，则返回值为 0，并置 EOF () 为真。

7、COL 测试屏幕列坐标函数

格式：COL ()

功能：给出屏幕上光标所处位置的列坐标，用于对光标的控制。

8、ROW 测试屏幕行坐标函数

格式：ROW ()

功能：给出屏幕上光标所处位置的行坐标，用于对光标的控制。

9、PCOL 测试打印机当前列位置函数

格式：PCOL ()

功能：给出打印机头所处位置的列坐标。

10、PROW 测试打印机当前行位置函数

格式：PROW ()

功能：给出打印机头所处位置的行坐标。

1.5 汉字 dBASE III 表达式与条件表示方法

表达式是汉字 dBASE III 语句的重要组成部分，它由运算符、常数、内存变量、函数等组成。上节已经讲述汉字 dBASE III 具有丰富的函数，通过对函数的使用，可以组成各种形式的表达式，将增强系统的计算、处理能力。

表达式中使用的运算符包括算术运算符、关系运算符、字符运算符和逻辑运算符。参与运算的运算符、常数、变量、函数等必须有相符合的类型。

1.5.1 数字表达式

数字表达式是由算术运算符、数字型常数、变量、函数组成，其运算结果仍为数字型数据。算术运算符有：

()	分组括号	*	乘法运算
+	加法运算	/	除法运算
-	减法运算	** 或 ^	乘方运算

优先顺序为：分组括号、乘方、乘除和加减，同级运算从左至右依次进行。

例: $10+25*6/(5+15*2)$

LOG (15)

均是算术表达式。

1.5.2 字符表达式

字符表达式是由字符运算符和字符型常数、变量、函数组成,其运算结果仍为字符型。

1、完全连接运算

格式: 〈字符串1〉+〈字符串2〉

功能:将两个字符串首尾连接成一个新字符串。

例: . ? "中国"+"北京"

中国北京

2、不完全连接运算

格式: 〈字符串1〉-〈字符串2〉

功能:删除(字符串1)尾部的空格后,再与(字符串2)连接在一起而成为一个新字符串。

例: . ? "中国" - "北京"

中国北京

3、包含运算

格式: 〈子字符串〉\$〈字符串〉

功能:判断(子字符串)是否包含在(字符串)中,如果被包含,结果值为真,否则值为假。

例: . ? "a b" \$ "a b c"

. T.

1.5.3 关系表达式

关系表达式是由关系运算符与字符表达式或数字表达式组成,其运算结果是一个逻辑值,因此关系表达式属于逻辑表达式。

关系运算符有:

<	小于	<=	小于等于
>	大于	>=	大于等于
=	等于	<>	不等于

关系运算符两边可以是数字表达式和字符表达式,两边的数据类型要一致。

对于数字表达式,应先计算表达式的值,再进行比较,关系成立时取真值,否则为假。

英文字母比较大小,是按其 ASCII 码值的大小进行比较的,"B">"A"为真值,因为"B"的 ASCII 码值为 66,"A"的 ASCII 码值为 65。在英文字符串中,字母挨个比较,因此表达式"AZ">"AA"值为真。

汉字字符串比较大小,在 CCDOS 下采用相应的汉语拼音字母替代汉字,然后再按英文字母比较大小的规则进行比较。

例: . ? (36+75*9)/(4+2*3) >= 50

. T.

. ? "姓名" < "名称"

.F.

由于“姓名”的汉语拼音为“XINGMING”,“名称”的汉语拼音为“MINGCHENG”,“X”的 ASCII 码大于“M”的 ASCII 码。

1.5.4 逻辑表达式

逻辑表达式是由逻辑运算符和逻辑型常数、变量、函数组成,它实际上是一个判断条件,结果为真或假值。

逻辑运算符有:

()	分组	.AND.	逻辑与
.NOT.	逻辑非	.OR.	逻辑或

优先级顺序为分组括号、逻辑非、逻辑与、逻辑或。

逻辑表达式的一般形式为:

〈关系表达式〉〈逻辑运算符〉〈关系表达式〉

运算时,先处理〈关系表达式〉,后进行逻辑运算。

1.5.5 条件表示方法

逻辑表达式常用于对一些条件的判断,而条件本身也是由关系表达式和逻辑表达式组成的。以后我们使用汉字 dBASE III 命令编写程序,计算处理数据时,条件及其应用是非常重要的,同时也是必不可少的,所以要求能够熟练地描述一些判断条件。

例: A、B、C、D 都不为零。

$A < > 0 .AND. B < > 0 .AND. C < > 0 .AND. D < > 0$

例:工资高于 40 元,且低于 100 元者。

$工资 > 40 .AND. 工资 < 100$

例:文件没有结束。

$.NOT. EOF ()$

例:25 岁以上的未婚女性。

$年龄 > 25 .AND. 婚否 = .F. .AND. 性别 = "女"$

例:工资在 100 元以下和 400 元以上者。

$工资 < 100 .OR. 工资 > 400$

例:具有讲师以外职称的男教师。

$.NOT. 职称 = "讲师" .AND. 性别 = "男" .AND. 职业 = "教师"$

1.6 汉字 dBASE III 的命令结构

命令也称作语句。在汉字 dBASE III 中一共提供了 117 条命令,其中大多数命令可以从键盘直接输入使系统完成规定的操作,另外一小部分专为程序模式提供的。对用户来说,首先要掌握各种命令的意义、书写格式及使用方法,并在此基础上学会程序设计的基本方法和技能。

1.6.1 命令结构

前面我们已经介绍了 STORE 命令,从这个命令中可以看出,命令是由两部分组成的。第一部分是命令动词,也称命令关键字,用来指示计算机完成什么样的任务;第二部分是跟随在关键字后面的动词短语,是用于限制和说明命令关键字的。

命令结构一:

〈命令关键字〉 〈动词短语〉

例: . STORE 0 TO X,Y,Z
. CREATE ADDRESS
. USE ADDRESS

在汉字 dBASE III 命令中,相当一部分是对库文件操作的命令,其命令结构的一般形式为:

命令结构二:

〈命令关键字〉〔〈范围〉〕〔〈表达式〉〕〔FOR/WHILE 〈条件表达式〉〕

首先对格式中符号做一个约定:

〔 〕 表示任选项,如不选择时,使用系统的缺省值。

〈 〉 表示必选项,由用户根据问题的需要输入具体内容,不可缺少。

/ 表示二者可选其一。

1、〈命令关键字〉项

〈命令关键字〉是一个英文动词,直接表示要进行什么操作,用户不得随意修改。

例如,LIST 是显示命令,REPLACE 是替换命令。

2、〔〈范围〉〕项:

指定命令所执行的记录范围,一般有三种选择。

ALL 表示全体记录。

NEXT N 从指针指示的当前记录号开始后的 N 个记录。N 是一个十进制数。

RECORD N 指第 N 条记录。

3、〔〈表达式〉〕项:

也称〔〈表达式清单〉〕、〔〈字段名清单〉〕或〔〈字段名表〉〕。此项中可列出一些表达式,往往是库文件中字段名的清单或包含字段名的表达式,其中各项以“,”分隔,若省略此项,一般等价于库文件中全部字段。

4、〔FOR/WHILE 〈条件表达式〉〕项:

条件短语,表示筛选出满足表达式的记录。条件项以 FOR 开始时,将对范围内符合条件的所有记录进行操作;条件项以 WHILE 开始时,只要遇到一个不符合条件的记录,不管剩下的范围内是否还有符合条件的记录,都停止执行命令,返回圆点状态。

例: .LIST ALL 姓名,出生日期 FOR 职称="教授"

命令中,LIST 是命令关键字,是“显示”的意思,指出该命令的功能;ALL 是范围项,指出是全部记录;表达式项规定显示的内容是姓名和出生日期;最后是由逻辑表达式所规定的条件。只显示职称为“教授”的记录内容。不难看出,此命令的功能即是显示当前数据库文件中,所有职称为“教授”的姓名、出生日期内容。

以上给出的命令结构的一般形式,只是对大多数命令而言的,各个命令本身还有一些特殊规定,在以后的章节中将详细说明。

1.6.2 命令的书写规则

- 1、每条命令必须以命令动词开头,当一个命令动词的字母个数超过四个时,可以从第五个起省略;动词后面的短语先后顺序可以是任意的,但必须符合规定的书写格式。
- 2、命令中的各动词、短语之间至少要有一个空格。
- 3、一条命令的长度不超过 254 个字符。
- 4、可以使用大写、小写字母。
- 5、给变量命名时,尽可能不使用 dBASE III 系统中的命令动词和关键字,以免引起不必要的混乱。
- 6、一条命令若太长,一行内写不下时,可以使用分界符“;”,在行末尾分行。
- 7、一行内只能写一条命令。
- 8、每条命令无标点符号作命令结束标志,结束标志是换行键。

1.7 汉字 dBASE III 文件名和文件类型

利用汉字 dBASE III 建立的各类文件都是磁盘文件,可存放在磁盘上,需要时才调入内存。汉字 dBASE III 可根据用户提供的文件名,在磁盘上对文件进行标识、查找、调用和操作。

1.7.1 文件名

汉字 dBASE III 是在 CCDOS 操作系统支持下管理文件的,因此文件名的规定与 CCDOS 相同。在汉字 dBASE III 中,一个文件名由两部分组成,即文件名和扩展名,扩展名也称后缀。其一般形式为:

文件名.扩展名

如 ADDRESS.DBF,其中 ADDRESS 为文件名,.DBF 为扩展名。汉字 dBASE III 规定,文件名本身必须以汉字或英文字母开头的字符串,最多可由八个字符组成(一个汉字占两个字符),可以有汉字、英文字母、数字符号和下划线,但不能有空格等其它符号。扩展名由三个字符组成,是文件名不可分割的部分。如果两个文件的扩展名不同,不管文件名是否相同,它们是两个不同的文件。

扩展名不仅是文件名的一部分,而且可用于区分文件的类型,扩展名相同的文件被认为是同一类文件。

1.7.2 文件类型

在汉字 dBASE III 中共有九种类型的文件,也就有九种特定的扩展名。数据库管理系统对数据库的管理就是通过对这九类文件管理而实现的。

1.数据库文件(.DBF)

数据库中最基本的文件,以记录和字段组成的二维表格的形式存储数据。每个数据库文件包含一系列的记录,每个记录由若干字段组成,记录的格式是相同的。记录中的字段个数、字段名、字段类型及字段宽度组成数据库文件的结构。用户输入数据之前需要首先定义数据库的结构,然后再逐记录输入数据。数据库文件一般由 CREATE 命令建立。

2.报表格式文件(.FRM)

按照报表格式设计的要求,将库文件数据转换成报表格式文件,以产生报表。

3. 命令文件 (.PRG)

有汉字 dBASE III 命令编写的程序文件是对数据库进行数据处理,实现各种管理的文件。

4. 索引文件 (.NDX)

索引文件提供一种按逻辑次序使用数据库的方法。是用户通过 INDEX 命令对库文件建立的索引。该文件只包含排序过的关键字以及对应的指针,常用于各类信息数据的迅速查找。

5. 内存变量文件 (.MEM)

汉字 dBASE III 在处理数据时,常常用一些内存变量来存放常数、中间结果、最终结果,一次最多可以使用 256 个内存变量。这些内存变量可以通过 SAVE 命令保存到内存变量文件中去以备后用,以空出内存空间给其它内存变量使用,从而提高内存空间的使用效率。需要时可用 RESTORE 命令将内存变量文件中的内容恢复到内存中去。

6. 文本文件 (.TXT)

是一种 ASCII 码文件,主要作用是实现汉字 dBASE III 与其它高级语言之间的数据传递。

7. 格式文件 (.FMT)

格式文件是用户建立的用来决定数据输入和输出时的格式,以提高输入输出的质量和效率。格式文件主要包含 @...SAY, @...GET 和 NOTE 三种命令。

8. 记忆文件 (.DBT)

记忆文件是数据库文件的辅助文件,用来存放数据库文件中记忆型字段的内容。当用户建立的数据库文件中包含有记忆型字段时,系统自动生成同名的 .DBT 文件。如含有记忆型字段的数据库文件被打开或关闭的同时,与其同名的记忆文件也同时被打开或关闭。如记忆型文件被删除,则与同名的数据库文件便打不开,这时必须再建立同名的记忆文件才能打开数据库文件。

9. 标签文件 (.LBL)

按照要求从库文件中提取有关数据,转换为具有标签格式的文件。

以上是按照文件功能、性质进行分类。如果按照数据库文件的操作特点,又可分为结构式文件和非结构式文件两大类。

结构式文件:数据库文件、内存变量文件、索引文件、记忆文件。

非结构式文件:命令文件、文本文件、格式文件。

1.8 汉字 dBASE III 的数据库文件

在汉字 dBASE III 的九类文件中,数据库文件是最基本的文件。数据库文件又称为库文件,习惯上也称为数据库。表 1—1 是计算机输出的一个实际数据库文件,其文件名为“ADDRESS1.DBF”。

表 1—1 数据库文件“ADDRESS1.DBF”

Rrecord #	姓名	性别	出生日期	工作单位	工作电话	婚否	职称	年龄
1	王勇	男	12/25/55	化工部设计院	690084-214	.F.	助理工程师	27
2	王立育	女	01/12/63	企管干部中心	5171770	.T.	工程师	29
3	宁志明	男	11/30/56	北京市统计局	8011122-1234	.T.	副研究员	36
4	蔡芸	女	05/17/70	标致汽车公司	2277789-218	.F.		22
5	冯小刚	男	03/22/42	北京建筑公司	2033604-财务	.T.	高级工程师	50
6	李景	男	12/13/60	南昌市农科院	557799	.T.	工程师	32
7	李克宏	男	07/27/69	钢铁研究总院	3046588-2123	.F.		23

8	王民安	男	09/19/51	燕碧企业公司	617057	.T.	经济师	41
9	尹立	女	03/21/57	建行郑州分行	337440	.T.		35
10	孙立杰	男	10/06/61	浙江人民银行	724938	.T.		31
11	陈吕昌	男	12/31/45	国家教委	6033344 - 2042	.T.		47
12	李小阳	女	05/15/65	农垦企业集团	7705740 - 216	.F.	经济师	27

1.8.1 数据库文件的文件名

在汉字 dBASE III 中,系统提供的数据库文件名格式为:

文件名.DBF

其中“.DBF”是系统默认的数据库文件的扩展名,是库文件的标志。如果用户不采用系统默认的扩展名,则必须明确声明。

1.8.2 字段、字段变量、字段值和记录

1、字段、字段名、字段值

表 1—1 由九列组成,除了第一列“Record #”项以外,其余八列每一列都称为一个字段,每列第一行上显示了该字段的字段名,如表中的“性别”,“姓名”等,每列其它各行上的内容都是字段值,由各种字符组成。

2、记录

表 1—1 中除第一行外,每行内容称为一个记录。每个记录在各字段上的内容为该记录在该字段上取得的字段值,也称为数据值、数据。

3、字段变量、字段变量名和字段变量值

在库文件中,字段值是经常变化的,因此字段也称为字段变量。相应地,字段名也称为字段变量名,字段值也称为字段变量值。

在汉字 dBASE III 中,字段变量名都是由字母或汉字开头的字符串,最多可由十个字符组成。字符串中可以有字母、汉字、数字字符和下划线。

定义字段变量名时,不论输入的是大写字母还是小写字母,系统都把它们转为大写字母,并以大写字母储存,大写字母显示。

“姓名”、“性别”、“学校__1”等都是合法的字段名,而“4BD”、“婚 否”等不能作为字段变量名。

4、字段宽度

在同一个库文件中,各记录对应于同一个字段所取的字段值虽然可以不同,但同一字段上所有字段值的宽度都是相同的,这个宽度就称为字段宽度,由用户在定义数据结构时给定。以“姓名”字段为例,若给定的宽度为 8,那么各字段值都要占据 8 个字节,未满部分以空格填充。

5、记录号

表 1—1 中,各记录在“记录号:”这一列上都取得了一个具体值,1,2,……。这一列不是库文件的内容,也不是一个字段,这是系统增设的一个显示项目,系统在显示打印库文件时,根据各记录在库文件中的实际物理次序给出此值,这个值就称为记录号。如物理次序第一位的记录,其记录号为 1,以下类推。

1.8.3 字段的五种类型

定义一个字段,不仅要明确规定它的字段名和宽度,还必须规定它的类型。字段的类型就是其

字段值的类型。

不同类型的数据存储方法是不同的,且不同类型的数据所能进行的运算也是不同的。汉字 dBASE III 有五种字段类型,它们是:字符(C)型字段、数据(N)型字段、日期(D)型字段、逻辑(L)型字段、记忆(M)型字段。

1、字符(C)型字段

字符型字段的数据值是字符串。字符串中既可有可打印的 ASCII 码字符,也可有汉字和图形符号(如俄文字母、日文假名、制表符号等)。一个汉字的宽度为 2 个字节。一个字符型字段的最大宽度是 254 个字节。表 1—1 中的“姓名”、“性别”、“工作单位”、“职称”四个字段都是字符型字段。

2、数字(N)型字段

数据型字段的数据值是可以进行算术运算的数值,又分为整数型和小数型两种。

数字型字段的最大宽度为 19 个字节(其中包括小数点位)。当字段为整数型时,最多可以是 19 位的正数或是 18 位的负数;当字段为小数型的,允许小数部分(不包括小数点)最多可有 15 位,或者最多为所定义的宽度减 2(留下整数位的 0 和小数点位)。如果定义了某数字型字段宽度为 6,则其小数部分最多为 4 位。另外,对小数型字段来说,其整数部分的宽度一般不允许超过 16 位。

表 1—1 中,“年龄”字段即为数字型字段。

3、日期(D)型字段

这种字段用以存储日期型数据,系统固定其宽度为 8 个字节,用户无权更改。数据的输入和显示形式为 xx/xx/xx,系统默认其含义为月/日/年。

表 1—1 中,“出生日期”字段即为日期型字段。

日期型数据可以进行两种运算:一个日期型数据加上或减去一个十进制整数得到一个新的日期型数据;一个日期型数据减去另一个日期型数据得到一个十进制整数(表示两个日期之间的差),即是天数。

4、逻辑(L)型字段

逻辑型字段宽度固定为 1 个字节,用户也无权更改。这种字段只能取两种值:„T. (逻辑真值)或„F. (逻辑假值)。在输入时,可以用 T,t,Y,y 来输入逻辑真值,用 F,f,N,n 来输入逻辑假值。系统一律转变成„T. 和„F. 存储及显示。

表 1—1 中“婚否”字段就是逻辑型字段,„T. 代表已婚,„F. 代表未婚。

以上四种字段类型都有各自不同的运算规则。根据这四种字段类型的数据值,可以在库文件中查找、检索、统计等操作。所以上述四种类型字段是库操作的基本字段。

5、记忆(M)型字段

记忆型字段是一种助忆字段,常用于扩大库文件容量。在数据库文件中,所有记忆型字段的宽度都是 10,这是系统固定的。但这 10 个字节并不是用来存放该记忆型字段的真正数据,而是用来存放一个指针。这个指针指向另一个辅助磁盘文件,这个辅助文件与库文件同名,但其扩展名为„DBT. 这种文件即是记忆文件,也叫明细文件。在记忆文件中,以 512 个字节为单位划分存储空间,当一个记忆型字段要存储数据时,系统就在同名记忆文件中分配一个单位的存储空间给此字段;如一个单位不够,再分配一个单位,但一个记忆型字段的值最多只能占八个单位,即 4K 字节。而记忆型字段上 10 个字节组成的指针指出了本字段数据在记忆文件存放的单位号,在列表输出库文件时,系统一般只在记忆型字段中显示“Memo”,表明它是记忆型字段。系统提供了专门的手段输入、输出和编辑这种字段,下一章中将详细介绍。