```
19 y1 = VAL(SUBS(process,1,2))
20 x1 = VAL(SUBS(process,3,2))
21 y2 = VAL(SUBS(PROCESS,5,2))
22 x2 = VAL(SUBS(PROCESS,7,2))
23 SET COLO TO N/W
24 @y1 + 1,x1 + 1 CLEA TO y2 + 1,x2 + 1
25 SET COLO TO W/N
26 @y1,x1 CLEA TO y2,x2
27 @y1,x1 TO y2,x2 && 画框语句
28 CONT
29 DO WHILE ! EOF()
30 @line,column PROM TRIM(menuchina)
31 CONT
32 ENDDO
33 MENU TO choice
34 LOCA FOR menuname = = cdm
35 SKIP choice
36 IF SUBS(menuchina,1,1) = '0'
37 IF menuname = = '#main'
38 EXIT
39 ELSE
40 LOCA FOR process = = cdm
41 cdm = menuname
42 choice = 1
43 LOOP
44 ENDI
45 ENDI
46 IF SUBS(process,1,1) = '#'
47 cdm = process
48 choice = 1
49 ELSE
50 cx = process
51 DO &cx
52 ENDI
53 ENDD
54 USE
55 RETU
```

五、应用举例

这里通过一个“学员通迅录”管理系统的部分，来简略说明：

1.菜单库文件的数据记录

```
. use menu
. list
```

Record #	MENUNAME	MENUCHINA	LINE	COLUMN	PROCESS
1	#main	功能选择	6	32	08281651
2	#main	1.输入通迅录	10	34	d1
3	#main	2.查询通迅录	11	34	d2
4	#main	3.打印功能	12	34	#dy
5	#main	4.生成通讯录	13	34	
6	#main	0.退出系统	14	34	
7	#dy	打印功能	7	32	09271552
8	#dy	1.打印通迅录表格	11	32	d31
9	#dy	2.打印通信信封	12	32	#dyxf
10	#dy	0.退出	13	32	
11	#dyxf	打印通信信封	7	28	09171552
12	#dyxf	1.按班级打印	11	33	d321
13	#dyxf	2.按学号打印	12	33	d322
14	#dyxf	0.退出	13	33	

2.该软件系统的主程序(简)

```
* * * * * * * * * * * * * * * * * * * * *
* 文件名:main.prg                        *
* 运行方法:操作系统提示符下键入 foxplus main  *
* * * * * * * * * * * * * * * * * * * * *
SET TALK OFF
    ¦ && 软件封面处理
    DO tycd && 调通用菜单程序,完成系统的各级菜单显示及各功能程序调用
    ¦ && 软件封底处理
```

3.生成通讯录程序(主要介绍用户对本功能的自撤卸)

```
* * * * * * * * * * * * * * * * * * * * * *
* 文件名:d4.prg                           *
* 功能:自动从学籍管理库文件中舍取生成通讯录信息,*
* 为了防止用户重复操作,当生成后对本模块进行自撤卸*
* * * * * * * * * * * * * * * * * * * * * *
    SET TALK OFF
    && 生成通讯录处理
    SELE 10
    USE menu && 打开菜单库文件
    LOCA FOR process = = 'd4'
    DELE && 删除此记录,这样菜单中再也无法有“生成通
    PACK && 讯录”这选项,因此用户也不会有重复操作
    USE
```

简洁的自定义查询统计模块的设计

江　文　(中国人民建设银行)

在FoxBASE应用程序中查询统计模块的使用频率相当高,如果应用程序中此模块的程序编制仅能起到单一的或固定的查询统计作用,那此应用程序的使用面和灵活性等便可想而知。如果我们在程序设计时将查询统计模块设计成一个开放式的,由用户根据各自所需自己定义查询统计条件,这样便能大大地提高查询统计的功能和查询统计的质量,减少程序的日后繁琐的维护工作,充分体现程序应用的灵活性。目前自定义查询统计的实际方法多种多样,本文仅介绍一种很直观的并且易编写的自定义查询模块编制方法

一、编制思路

在查询统计中最直观的方法是依据查询条件将所需的数据从查询对象数据库中提出符合条件的数据新生成一个结果数据文件,然后对新生成的数据文件中的数据进行输出和统计等便能得出用户所需的结果。

自定义查询统计条件的编制思路是将用户自定义的条件

和字符串“COPY TO〈文件名〉FOR”“连接”成一字符串，然后将此字符串加入到一个只有一个类型为字符型长度为 254 字段的数据文件中，自定义条件完成后再将此字符串从数据库中利用“COPY TO〈文件名.PRG〉SDF”命令提出，这样用户自定义的查询统计条件便由此变为一个.PRG 文件，此.PRG 文件中只有一行“COPY TO〈文件名〉FOR〈自定义条件〉”，于是运行文件便可将符合自定义条件的数据从查询对象库中拷贝生成一个新的数据文件。

二、应用举例

①、建立一条件库 TJ.DBF，库结构为：

Field	Field Name	Type	Width	Dec
1	Tj	Character	254	

②、自定义查询举例：

在此仅举例如何得到自定义拷贝提取符合条件数据的.PRG 文件的方法，如某应用程序中其主要数据文件的数据库结构是：

Field	Field Name	Type	Width	Dec
1	MC	Character	20	
2	CD	Character	10	
3	NB	Character	10	
4	JG	Numeric	10	2

MC　:产品名称
CD　:产地
NB　:产品类别
JG　:价格

③、介绍如何得到自定义拷贝提取符合条件数据的 PRG 程序(程序如下)。

```
SET TALK OFF
SET DELIMITED OFF
SET STATUS OFF
SET SAFETY OFF
CLEAR ALL
ch1 = 0
ch2 = 0
SELECT 1
USE tj
DELETE ALL
PACK
INSERT BLANK
STORE'copy to tjjg.dbf for'TO tj1
x = 4
i = 14
@ 02 ,2 SAY'┌──────────────────────┐'
@ 03 ,2 SAY'│                      │'
@ 04 ,2 SAY'└──────────────────────┘'
@ 08 ,11 SAY'┌──────────────────┐'
@ 09 ,11 SAY'│ =〉〈≠≥≤并且 或者 退出 │'
@ 10 ,11 SAY'└──────────────────┘'
@ 13 ,2 SAY'┌──────────────────────┐'
@ 14 ,2 SAY'│                      │'
@ 15 ,2 SAY'│                      │'
@ 16 ,2 SAY'│                      │'
@ 17 ,2 SAY'│                      │'
@ 18 ,2 SAY'│                      │'
@ 19 ,2 SAY'│                      │'
@ 20 ,2 SAY'│                      │'
@ 21 ,2 SAY'│                      │'
@ 22 ,2 SAY'└──────────────────────┘'
DO WHILE.T.
STORE SPACE(20) TO mc
STORE SPACE(10) TO cd
STORE SPACE(10) TO ib
jg = 0
@ 03, 4 PROMPT'产品名称'
@ 03,COL() + 2 PROMPT'产地'
@ 03,COL() + 2 PROMPT'产品类别'
@ 03,COL() + 2 PROMPT'价格'
@ 03,COL() + 2 PROMPT'('
@ 03,COL() + 2 PROMPT')'
@ 03,COL() + 2 PROMPT'退出'
MENU TO ch1
@ 09,13 PROMPT' = '
@ 09,COL() + 3 PROMPT'>'
@ 09,COL() + 3 PROMPT'<'
@ 09,COL() + 3 PROMPT'≠'
@ 09,COL() + 3 PROMPT'≥'
@ 09,COL() + 3 PROMPT'≤'
@ 09,COL() + 3 PROMPT'并且'
@ 09,COL() + 3 PROMPT'或者'
@ 09,COL() + 3 PROMPT'退出'
MENU TO ch2
IF ch2 = 9.AND.ch1 = 7
  REPLACE tj WITH tj1
  COPY TO'tj.prg' SDF
  RETURN
ENDIF
IF LEN(tj1)>=220
  @11 ,30 SAY'条件已满'
  tiaoj = 1
LO  OP
ENDIF
IF ch1 = 1
  IF 74 - x<20
    I = I + 1
    x = 4
ENDIF
```

```
ENDIF
IF ch1 > = 2.OR.ch1 < = 6
  IF 74 - x < 20
    I = I + 1
    x = 6
  ENDIF
ENDIF
IF I = 22
  @ 14,4 CLEAR TO 21 ,73
  I = 14
  x = 4
ENDIF
IF ch1 = 5
  @ 1, x SAY'('
  tj1 = tj1 + '('
  x = x + 2
ENDIF
IF ch2 = 7.AND.ch1 = 7
  @ 1, x SAY '并且
'   tj1 = tj1 + '.and.'
  x = x + 4
ENDIF
IF ch2 = 8.AND.ch1 = 7
  @ 1, x SAY'或者
'   tj1 = tj1 + '.or.'
  x = x + 4
ENDIF
IF ch1 = 1
  @ 1, x SAY '产品名称'
  IF ch2 = 1
   @ 1,COL () SAY' = '
ENDIF
IF ch2 = 4
  @ 1,COL () SAY''
ENDIF
IF ch2 = 1.OR.ch2 = 4
  @ 1,COL() GET mc PECTURE'xxxxxxxxxxxxx'
  READ
ENDIF
IF ch2 = 1
  tj1  = tj1 + 'mc = "'mc + '""'
ENDIF
IF ch2 = 4
  tj1 = tj1 + 'mc < >"' + mc + '""'
ENDIF
x = x + 30
ENDIF
IF ch1 = 2
@ 1, x SAY '产地'
IF ch2 = 1
  @ 1,COL() SAY' = '
ENDIF
IF ch2 = 4
  @ 1,COL() SAY''
ENDIF
IF ch2 = 1.OR.ch2 = 4
  @ 1,COL() GET cd PICTURE'xxxxxxxxxx'
  READ
ENDIF
IF ch2 = 1
  tj1 = tj1 + 'cd = "' + cd + '""'
ENDIF
IF ch2 = 4
  tj1 = tj1 + 'cd < >"' + cd + '""'
ENDIF
  x = x + 16
ENDIF
IF ch1 = 3
  @ 1, x SAY'产品类别'
  IF ch2 = 1
    @ 1,COL() SAY' = '
  ENDIF
  IF ch2 = 4
    @ 1,COL() SAY ''
  ENDIF
  IF ch2 = 1.OR.ch2 = 4
    @ 1,COL() GET lb PICTURE'xxxxxxxxxxx'
    READ
  ENDIF
  IF ch2 = 1
    tj1 = tj1 + 'lb = "' + lb + '""'
  ENDIF
  IF ch2 = 4
    tj1 = tj1 + 'lb < >' + '""'
  ENDIF
  x = x + 20
ENDIF
IF ch1 = 4
  @ 1, x SAY '价格'
  IF ch2 = 1
    @ 1,COL() SAY' = '
  ENDIF
  IF ch2 = 2
    @ 1,COL() SAY' > '
  ENDIF
  IF ch2 = 3
    @ 1,COL() SAY' < '
  ENDIF
  IF ch2 = 4
    @ 1,COL() SAY''
  ENDIF
  IF ch2 = 5
```

```
    @ 1,COL() SAY''
  ENDIF
  IF ch2=6
    @ 1,COL() SAY''
  ENDIF
  @ 1,COL() GET jg PICTURE'9999999.99'
  READ
  IF ch2=1
    tj1=tj1+'jg='+STR(jg,10,2)
  ENDIF
  IF ch2=2
    tj1=tj1+'jg>'+STR(jg,10,2)
  ENDIF
  IF ch2=3
    tj1=tj1+'jg<'+STR(jg,10,2)
  ENDIF
  IF ch2=4
    tj1=tj1+'jg<>'+STR(jg,10,2)
  ENDIF
  IF ch2=5
    tj1=tj1'jg>='+STR(jg,10,2)
  ENDIF
  IF ch2=6
    tj1=tj1'jg<='+STR(jg,10,2)
  ENDIF
  x=x+20
 ENDIF
ENDDO
```

用 FoxBASE+ 实现基于数据库的窗口编辑器

林　民　（内蒙师大计算机系）

摘要：本文针对 FoxBASE+ 中 MEMO 字段处理变长信息功能差的问题，用 FoxBASE+ 设计了一个基于数据库的窗口编辑器，克服了 MEMO 字段的缺点，较好的解决了变长信息的存储、编辑、显示、检索、打印等处理问题。

一、引言

在开发微机数据库应用系统中，经常会遇到处理信息量大、伸缩性也大的文字信息。例如，档案管理中个人简历、大事记，题库管理中说明性问题的答案等。若采用 FoxBASE+ 中提供的 MEMO 字段来处理这类信息，会遇到下述难以解决的问题：

1. 编辑屏幕格式单一，无法控制
2. 编辑时存在半个汉字的处理问题
3. 显示输出会破坏原有屏幕格式
4. 难于进行检索
5. 打印报表实现复杂
6. 磁盘空间利用率不高

因此，为解决这些问题，笔者开发了基于数据库的窗口编辑器，能较好的适应变长信息的存储、编辑、显示、检索、打印等方面的需要。考虑到 FoxBASE+ 与其它语言接口的复杂性以及需占用很大内存空间的限制，这里直接采用 FoxBASE+ 自身的命令和函数来实现编辑器。

二、有关数据库结构的设计

1. 辅助数据库结构

辅助数据库用来代替存放 MEMO 字段信息的辅助文件（*.DBT），其结构如下：

NO1 字段：(N)用来存储变长信息字段对应的记录号

NO2 字段：(N)用来存储变长信息字段在记录中的序号

TEXT 字段：(C)用来存储相应的变长信息

2. 编辑数据库结构

供编辑器使用，用来暂时存放变长信息，其结构如下：

TEXT 字段：(C)存储输入的变长信息

这样，编辑器直接对编辑数据库记录进行操作，再将编辑好的信息添加到辅助数据库。一个数据库里的所有变长信息字段内容都存储在一个辅助数据库中，并可以按 NO1、NO2 字段建立索引，便于检索和显示。

三、编辑器的设计方法

1. 调用方法及参数说明

DO EDITOR WITH ROW1, COL1, ROW2, COL2, MFILE

其中，EDITOR：编辑器过程文件名

ROW1，COL1：窗口左上角坐标

ROW2，COL2：窗口右下角坐标

MFILE：编辑数据库文件名

2. 主要变量说明

TEXTCOLOR：文本显示颜色

BLACKCOLOR：背景颜色

SS(400)：编辑缓冲区数组

SS-MAX：编辑数组的实用最大行号

KEY：存放按键值

XX，YY：当前行在文本中的绝对行、列号