

## FoxPro 下实现图文数据库的原理和方法

林丕源 (西南石油学院)

**摘要:**本文介绍在 FoxPro 2.5 for MS-DOS 和 FoxPro 2.5 Windows 两个系统下,各自实现图文数据库的原理和实用方法。

### 一、引言

图文数据库是伴随多媒体技术而产生的新型数据库形式,它的主要特点是在数据库系统中能够同时管理文字和图像信息。与传统的数据库一样,图文数据库能够利用数据库信息产生的像直方图、圆饼图一类的图形。但图文数据库对图像信息管理是指对图片或图像文件的管理。这是它对传统数据库管理系统的功能的扩充。图文数据库特别适合作为人事档案管理、罪犯管理、车辆管理、文物资料管理等需要对图片进行管理的系统。图文数据库在对图像实施管理的关键是把记录中的图像文件显示在计算机的屏幕上。为此,本文简明扼要地介绍使用极为流行的数据库系统 FoxPro 2.5 for MS-DOS 和 FoxPro 2.5 for Windows 来实现对图文数据库的管理。如果我们掌握了实现原理和方法,可以在这样的系统下,设计出更加优秀的管理软件,向使用者展示图文并茂的信息。

### 二、在 FoxPro 中实现图像显示的原理和方法

实现图文数据库管理的关键是把数据库的记录中所指的图像显示在计算机的屏幕上。这里将分别介绍在 FoxPro 2.5 for MS-DOS 和 FoxPro 2.5 for Windows 两个系统显示图像文件的方法。

1. 在 FoxPro 2.5 for MS-DOS 系统中显示图像的原理和方法

众所周知, FoxPro 2.5 for MS-DOS 中并无直接显示图像的功能。为实现图像的显示,就必须在 FoxPro 2.5 for MS-DOS 中运行其他软件或命令来完成这一功能。例如,使用 Blackhask Data Corp 的数据图形工具箱 Database Graph Toolkit(DGT)来显示图像文件。不过,根据笔者程序设计的经验,直接使用 UC DOS 所提供的特

殊显示功能是更为简单实用的方法。因此,对于前者只作简单说明,后者给予重点介绍。

(1)使用工具箱 DGT 和 EXHIBIT 函数实现图像显示。其原理是把 DGT 用作 API 库,安装时给出命令:

SET LIBRARY TO DGT 然后调用 DGT 的 EXHIBIT 函数显示图像,例如:

```
= EXHIBIT(" / t:3"+photo.gif")
```

这个例子是使用 DGT 的 EXHIBIT 函数将图像 photo.gif 显示 3 秒钟时间。

DGT 提供了其他的选项来控制图像的尺寸、色彩、扫描特征及其它特性,此处不进一步说明。

(2)使用汉字系统 UC DOS 3.1 的特殊显示功能实现图像显示。UC DOS 3.1 特殊显示功能是其特色之一。只要运行在 UC DOS 3.1 下,就可在 FoxPro 2.5 for MS-DOS 中方便地实现对该功能的调用。这样就很容易地在数据库系统中实现显示各种不同大小的矢量汉字、显示图像等众多的功能。

在实际应用中注意,使用特殊显示功能前必须运行 UC DOS 3.1 的打印字库读取程序(RDSL.COM)和特殊显示模块(TX.COM)。

调用特殊显示功能的原理是,调用显示中断(INT 10H),当中断(INT 10H)显示一特殊的命令字符串时,被显示的字符串不会直接在屏幕上显示出来,而被作为命令产生了其他一些现象,如显示了一个特大的汉字、在屏幕上画了一个圆等,这样便实现了特殊显示功能。由于 FoxPro 2.5 for MS-DOS 的显示命令可实现调用 INT 10H,所以实现特殊显示的工作便是将要使用的功能翻译为字符串,然后两头加上引导字符即可。

UC DOS 3.1 提供的特殊显示控制命令的格式如下:

<命令起始标志>+<命令串及参数>+<命令结束标志>

这里<命令起始标志>由两个字符组成,它们是 CHR(14)和 "[",<命令结束标志>是 "]". CHR(14)是 ASCII 码为 14 的字符。

通过打印方式使用特殊显示功能的命令格式也完全相同,只是必须注意控制序列应送往 3 号打印口打印,也即 DOS 定义的设备文件 ipt3。

下面是在 FoxPro 2.5 for MS-DOS 中调用 UC DOS 3.1 特殊显示功能的语句表现形式:

①以缺省方式调用 UC DOS 3.1 特殊显示功能:

@0,0 SAY chr(14)+[命令串及参数]

②通过打印口 3 调用 UC DOS3.1 特殊显示功能的方法:

SET PRINT TO LPT3

SET DEVI TO PRINT

@0,0 SAY chr(14)+[命令串及参数]

SET DEVI TO SCRE

SET PRINT TO PRN

在 UC DOS 3.1 特殊显示实现显示 PCX 图像的命令及参数如下:

REx,y,f

这里,(x,y)=左上角坐标,f=文件名(以 \$ 结尾)

在 UC DOS 3.1 特殊显示实现显示部件 PCX 图像的命令及参数如下:

RPx,y,w,h,f

其中,w=宽度,h=高度,其余参数同上。

在实际使用时,还常常会用到一个与显示图像有关的“16 种基本颜色保护”特显命令:

PPn

在该命令中,n=1 表示保护 16 种基本颜色的调用板,n=0 表示不作保护。

有关使用汉字系统 UC DOS 3.1 的特殊显示功能实现图像显示,我们将在程序示例中通过实例得以体会。

2.在 FoxPro 2.5 for Windows 系统中显示图像的方法

FoxPro 2.5 for Windows,作为“客户”程序,支持 Microsoft Windows 的目标连接及嵌入(OLE)功能。这就是说,可以在 FoxPro 2.5 for Windows 应用程序中嵌入或连接由其他 Windows 程序所产生的目标,这些目标自然包括由 Windows 下的图像处理软件(如 Aldus Photo Styler)所生成的 BMP 图像文件。这就是在 FoxPro 2.5 for Windows 系统中显示图像的原理。为实现 OLE 功能,FoxPro 2.5 for Windows 为存储其它 Windows 所产生的目标提供了一类新域类型 General,并提供了几个新的命令选项来操作它们。不过在本文中,我们只介绍对 OLE 目标和 Windows 位映象 BMP 图像文件的显示操作,来说明如何实现图文数据库。

在对 OLE 目标作显示之前,首先应将它们插入数据库的 General 域中。例如,我们建立了一个结构如下的数据库:

Oletable.dbf Structure

Field	Field Name	Type	Width	Dec	Index
1	NAME	Character	12		
2	OLETYPE	Character	10		
3	OBJECT	General	10		
* * TOTAL * *			33		

如果其中某一记录的 Oletable.oletype="Picture"和 Oletable. Object="Gen",这样我们就可在 Database/Browse 时 Open 数据库 Oletable.dbf,然后在浏览表中的 Object 列对应该记录的“Gen”处,双击该标记来打开 General 域窗口;之后根据具体情况作 Edit 操作,或把 Windows Clipboard(裁剪板)的图像 Paste(粘贴)或 Special Paste 到该域中,或用 Insert Object 选择一个图像文件以嵌入该 OLE 目标,例如从 Aldus Photo Styler 中调入一个 BMP 图像文件。

一旦将 OLE 目标嵌入到 General 域的数据库 Oletable.dbf 作如下的显示图像的操作:

```
select oletable
locate for oletype="Picture"
define window mywin from 1,1 to 8,12 close float
activate window mywin
@ 0,0 say object stretch
```

在 FoxPro 2.5 for Windows 中,也可以直接用@...SAY 命令的变化形式来显示由 Windows 下的应用程序所产生的位映象 BMP 格式图像文件。例如,在坐标(5,5)处显示名为 d:/windows/256color.bmp 的位映象图像的命令如下:

```
@5,5 SAY "d:/windows/256color.bmp"BITMAP
```

下面是使用@...SAY 命令来显示 OLE 目标及 Windows 位映象图像(图像文件扩展名为.BMP)的变化格式之一:

```
@ < row >,< column >SAY < file > BITMAP, < general field >
[CENTER][ISOMETRIC / STRETCH]
[SIZE < ExpN1 >,< ExpN2 >][STYLE < ExpC >]
```

该命令中的参数使用说明如下:

< row >,< column >——指定目标的左上角在当前 FoxPro 2.5 for Windows 输出窗口中的坐标值;

< file > BITMAP——指定显示位映象图像,文件名 < file > 可带全路径;

< general file > ——指定显示位于打开的数据库文件中的 General 域;

SIZE < ExpN1 >,< ExpN2 > ——指定目标的显示高度(以行为单位)和宽度(以列为单位),即显示目标的区域

大小;若无 ISOMETRIC / STRETCH 项,则 SIZE 描述区域小于目标时,目标的超出部分将被裁掉;

ISOMETRIC / STRETCH——STRETCH 指定按 SIZE 区域的横向和纵向的比例变形显示整个目标,而 ISOMETRIC 则指定保持原有比例在 SIZE 区域内显示整个目标;

CENTER——指定把目标显示在 SIZE 区域的正中央;

STYLE < ExpC > ——显示目标的方式, < ExpC > 取代码"Q"为不透明方式, < ExpC > 取代码"T"为透明方式。

### 三、实现图文数据库的程序示例

懂得在 FoxPro 中实现图像显示的原理和方式之后,以下简单地举例来示范如何在 FoxPro 2.5 for MS-DOS 和 FoxPro 2.5 for Windows 两个系统下,实现图文数据库。这里的例子仅为示范之用,力求程序简单短小,不追求程序的复杂性和与说明实现图文数据库关系不大的技巧性。为便于比较两个系统在实现图文数据库方面的不同之处,两个程序使用相同的数据库 TUWEN.DBF,并完成相同功能。

表 1 数据库 TUWEN.DBF 的字段

字段名称	字段含义	类型	长度
XM	姓名	字符	6
XB	性别	字符	2
SFZH	身份证号码	字符	15
MZ	民族	字符	6
ZZMM	政治面貌	字符	8
ZC	职称	字符	10
XL	学历	字符	8
BYYX	毕业院校	字符	30
CSNY	出生年月	数字	7,2
GZDW	工作单位	字符	30
JTZZ	家庭住址	字符	30
ZP	本人照片	字符	20

程序一, FoxPro 2.5 for MS-DOS 需运行于 UC DOS 3.1 之下,数据库的 ZP 字段用于存放格式为 PCX 的图像文件名,图像的显示调用 UC DOS 的特殊显示实现;程序二, FoxPro 2.5 for Windows 下的例子,运行在中文 Windows 环境之下,数据库的 ZP 字段用于存放格式为 BMP 的图像文件名,图像的显示用 @...SAY < file > BITMAP 命令形式。两个程序在 AST P III 4.50 以及兼容机 386 DX40 VGA 上运行通过。

两个程序使用的数据库 TUWEN.DBF 的字段表如表 1 所示。

程序一 FoxPro 2.5 for MS-DOS 在 UC DOS 3.1 下实现图文数据库的例子

说明:本程序必须在可以使用 640×480×256 色的微机上使用,如果使用直接写屏型的数据库管理系统时,必须在 CONFIG.SYS 中加入 QEMM.SYS 驱动程序。

```

sele 1
use tuwen
set talk off
set prin to 1pt3
set colo to +7 / 1
clear
@ 1,2 say' /
@ 2,2 say' /
@ 3,2 say' /
@ 4,2 say' /
@ 5,2 say' /
@ 6,2 say' /
@ 7,2 say' /
@ 8,2 say' /
@ 9,2 say' /
@ 10,2 say' /
@ 11,2 say' /
@ 12,2 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' /
@ 23,15 say'光标上称键:上一记录光标下称键:下一记录 ESC:退出"
modify = .t.
do while .not.eof()
if modify
@ 6,17 say XM
@ 6,43 say XB
@ 8,17 say left(str(CSNY,7,3),4)+'年'+right(str(CSNY,7,3),2)+'月'
@ 8,43 say SFZH
@ 10,17 say MZ
@ 10,43 say ZZMM
@ 12,17 say ZC
@ 12,43 say XL
@ 14,17 say BYYX
@ 16,17 say GZDW
@ 18,17 say JTZZ
* 调用 UC DOS 的特殊显示实现照片显示
set devi to prin
@ 0,0 say chr (14)+'[pplrp479,105,108,128'+ZP+']
set devi to scre
    
```

西南石油学院人事档案卡

姓 名	性 别		
出生年月	身份证号		
民 族	政治面貌		
职 称	学 历		
毕业院校			
工作单位			
家庭住址			

```

        modify = .f.
    endif
    key = inkey(0)
    do case
    case key = 1          &&Home
        if recno() = 1
            loop
        endif
        go top
    case key = 5          &&Up
        if recno() > 1
            skip -1
        else
            loop
        endif
    case key = 24         &&Down
        if recno() < reccl()
            skip +1
        else
            loop
        endif
    case key = 6          &&End
        if recno() = reccl()
            loop
        endif
        go bott
    case key = 27         &&Esc
        exit
    othe
        loop
    endcase
    modify = .t.
enddo

```

set print to prn  
程序二在 FoxPro 2.5 for Windows 下实现图文数据库的例子

```

sele 1
use tuwen
set talk off
set colo to +7 / 1
clear
@ 4,2 say'
@ 5,2 say'
@ 6,2 say'
@ 7,2 say'
@ 8,2 say'
@ 9,2 say'
@ 10,2 say'
@ 11,2 say'
@ 12,2 say'
@ 13,2 say'
@ 14,2 say'
@ 15,2 say'
@ 16,2 say'
@ 17,2 say'
@ 18,2 say'
@ 19,2 say'

```

西南石油学院人事档案卡

姓 名	性 别		
出生年月	身份证号		
民 族	政治面貌		
职 称	学 历		
毕业院校			
工作单位			
家庭住址			

```

@20,2 say' /
@21,2 say' /
@22,2 say' /
@23,2 say' /
@24,2 say' /
@28,15 say'光标上移键:上一记录光标下移键:下一记录 ESC:退出'
modify = .t.
set colo to +7 / 1
do while .not.eof()
    if modify
        @9,17 say XM
        @9,43 say XB
        @ 11,17 say left(str(CSNY,7,3),4)+'年 '+right (str (CSNY,
7,3),2)+'月'
        @11,43 say SFZH
        @13,17 say MZ
        @13,43 say ZZMM
        @15,17 say ZC
        @15,43 say XL
        @17,17 say BYYX
        @19,17 say GZDW
        @21,17 say JTZZ
        @9,61 say ZP BITMAP STRETCH SIZE 7,12
        modify = .f.
    endif
    key = inkey(0)
    do case
    case key = 1          &&Home
        if recno() = 1
            loop
        endif
        go top
    case key = 5          &&Up
        if recno() > 1
            skip -1
        else
            loop
        endif
    case key = 24         &&Down
        if recno() < reccl()
            skip +1
        else
            loop
        endif
    case key = 27         &&End
        if recno() = reccl()
            loop
        endif
        go bott
    case key = 27         &&Esc
        exit
    othe
        loop
    endcase
    modify = .t.
enddo

```