

# 多功能通用综合查询模块的开发

郑志彬 吴昊 (哈尔滨工业大学 150001)

翁振武 葛林 (广丰县人民银行 334600)

**摘要:** 本文介绍了如何在 MIS 开发中建立了一个实用的通用查询模块。叙述了该查询模块的程序设计思想、程序设计和程序调用方法,该模块充分考虑了数据库软件使用的实际情况,具有良好的实用性及通用性。

**关键字:** 查询 模块 数据库 代码

## 一、引言

通常一个管理信息系统包括若干个子系统,各子系统都必须建立自己的查询模块,而从查询模块本身来看,其功能对各子系统来说都是一样的,即都是查询出符合某种条件的所有内容。因此对于开发人员来说,建立一个适用于各子系统使用的通用查询模块是非常有意义的。本文介绍了网络环境下用 Foxpro2.5 for Windows 创建一个通用综合查询模块。

## 二、代码库设计

在传统的查询功能模块设计中,查询条件往往是这样设计的,即左边为某一个字段名,右边要求用户输入相应字段的查询内容,然后进行条件组合形成某一综合查询条件。对于用户来说每写一个查询条件都必须手工输入一次相应内容,特别是有汉字输入时会带来诸多的不便。因此对于某些内容比较固定、种类不多的字段,我们考虑能否在查询时自动显示字段内容,只需用鼠标进行选择,而不用手工输入,可以大大减少用户的操作,从这一点出发我们考虑建立一个代码数据库来辅助查询模块的应用。

记录\字段	性别	技术职称	学历
1	男	高级工程师	研究生
2	女	工程师	本科
3		助理工程师	大专
4		高级经济师	中专
5		经济师	
6		助理经济师	

代码数据库是一个不平衡数据库,即数据库中各字段的实际记录数是不相等的。其字段名必是主数据库的某一字段名(注意:代码库中字段名若要生效必须与主数据库某一字段名相同,否则该字段在查询模块中无任何意义),字段内容则是这一字段的所有可能内容,下例为一简单代码数据库的设计,其字段名分别是性别、职称和学历,相应的字段内容都比较固定规范。

## 三、查询模块设计

### 1. 程序设计思想及特点

(1)通用性。该模块不仅是为某一特定用户服务的,而且也是为管理信息系统中所有用户服务的。因此通用模块程序必须是一个可调用的公用子程序,程序中包括三个起始字符型参数,分别表示查询主数据库名、代码数据库名和打印的报表名。各子系统在调用该模块时只要设置相应的参数,该模块即可满足子系统的查询浏览以及查询打印的需要,达到通用的要求。

(2)条件完备性。作为一个综合查询模块条件,它必须能够查询任何复杂条件的合理查询要求(即必须是对字段内容进行查询)。因此在模块中包含了"与(AND)"和"或(OR)"逻辑符以及"="、"=="、"<>"、">"、">="、"<"、"<="七种比较符,用户只要进行合理的条件组合,一定可以查询出符合条件的记录。

(3)直观方便性。这首先要求用户界面友好,即要求该查询窗口的使用简单明了,应用 Foxpro for Windows 中的屏幕生成器很容易做到这一点。其次在查询模块的输入设计中,应用辅助的代码数据库可以很直观的显示字段内容。在条件输入栏同时定义了两种格式:字段域(Field)和弹出菜单(Popup)。

(4)适用性。调用该模块时需传送查询的主数据库、代码数据库和打印的报表名三个参数。如果在实际应用

时只有主数据库而不需要代码数据库,那么只要在对应于代码库的参数输入字符型参数 nulldbf,以该字符串代替代码库参数,查询即可正常进行,在整个查询过程中尽管没有代码库,但对查询不产生任何影响,与传统的查询模式一样。

## 2. 程序设计方法

查询功能模块的使用是以窗口形式显示的。如下画出一个查询显示窗口,它能符合整个综合查询的需要。在这个窗口设计时主要包括以下五类应用要素。

(1) 查询条件输入。查询条件输入包括查询字段、比较符和字段内容三部分(即实窗口中第一行中的三个框)。查询字段采用弹出菜单式,它包括需查询数据库的所有字段,菜单内容取于主数据库字段。比较符亦采用弹出菜单式,它包括“=”、“<>”、“>”、“>=”、“<”、“<=”和“==”七种比较符,程序设计时可设计为七维数组,菜单内容来源于该数组。字段内容的设计有两种格式,一种是字段域式的,字段的查询内容由用户手工输入;另一种是弹出菜单式的,它直接显示查询字段的所有字段内容,该查询菜单的内容取自于代码数据库。当选“查询字段”项中某一字段时,它将自动搜索代码数据库,如果在代码库中有对应的字段,代码库中该字段的记录内容显示在“字段内容”的弹出菜单上,否则若无对应字段,“字段内容”显示为字段域,由用户输入字段选择条件。

(2) 条件显示。条件显示包括一个列表显示框,即“查询条件显示框”下的列表框(List)。该框将直观显示出用户输入的复合查询条件,选用列表框的好处是:如果

该框显示的查询条件已装满,它将自动激活其右边的滚动块,可以显示更多的查询条件。

(3) 条件选择。它包括选中条件、重选条件两个按钮(Button),即查询窗口左边竖列的四个按钮的上面两个按钮。当已输入查询条件后,选择“选中条件”按钮时,会把对应的查询条件显示到查询条件显示框中,此时该条件也被写入查询条件过滤器。当选择“重选条件”按钮时,则清除所有查询条件,查询条件显示框变成空白,查询条件过滤器的条件也被清除,可以重新进行查询条件的选择。

(4) 条件符。条件符包括并且、或者两个按钮(Button),即查询窗口左边竖列的四个按钮的下面两个按钮。当选择“并且”按钮时,表示将生成一个“与(and)”条件,相应的把“并且”显示到条件显示框中去,此时将等待再输入组合查询条件。当选择“或者”按钮时,表示将生成一个“或(or)”条件,相应的把“或者”显示到条件显示框中去,此时同样等待输入组合查询条件。

(5) 方式选择。方式选择包括查询浏览、查询打印和退出三个按钮,即查询窗口最下面一行的三个按钮(注:如果在调用该模块时报表名参数是 nullfrx,那么“查询打印”按钮将是灰色,即是无效按钮)。当选择“查询浏览”后将在已定义的数据浏览窗口中显示查询数据库的结果,并可以进行数据的修改和记录的删除。当选择“查询打印”按钮时,将根据查询结果打印对应的报表。如果选择“退出”按钮时,退出通用查询模块,该窗口被清除。

## 3. 程序调用方法

实际上该通用综合查询模块不仅适用于开发管理信息系统,也适用于一个 Foxpro for Windows 数据库软件的开发。如果在一个管理信息系统中,所有的子系统都可能调用该模块,那么就on应该把该通用程序放在管理信息系统的公共目录中,所有的子系统都有权去访问它,欲调用它时只需在子系统主程序实现查询功能的对应位置加入调用子程序语句“do <查询子程序名> with 参数1,参数2,参数3”,当程序执行到这条指令时即可调用查询模块。其中参数1是查询主数据库名,参数2是代码数据库名,参数3是报表名(注意数据库名后不用加扩展名“.dbf”)。当不需代码数据库时,参数2为“nulldbf”,当不需打印报表时,参数3为“nullfrx”。当查询结束后,它可回到主程序并执行该语句的下一条命令。

## 四、查询模块源程序

由于该通用查询模块的源程序比较长,以下我们将

提供部分关键性源程序,以供读者使用参考。

```

*****
* 通用查询模块 COMCX.PRG
*****
* 初始环境设置
*****
parameter dbf1, dbf2, frx
define window comcx ;
    at 2.846, 18.600 size 21.000, 78.400 ;
    font "MS Sans Serif", 8 float noclose ;
    title "通用查询窗口" minimize system
if used(dbf1)
    select &dbf1
else
    select 0
    use &dbf1
endif
if(&dbf2<>"nulldbf")
    if used(dbf2)
        select &dbf2
    else
        select 0
        use &dbf2
    endif
endif

*****
* 部分变量初始设置
*****
.....
.....
select &dbf1
z = afields(fld)
dimension bm(z)
for i = 1 to z
    bm(i) = fld(i, 1)
next
.....
.....
.....

```

```

*****
* 在代码库中查找是否有与主数据库字段相符合的字段,若有的话,激活对应字段内容的弹出菜单,否则
* 激活对应的字段域
*****
function cdpro1 && cd1 VALID
sele &dbf1
fiename = fld(cd1, 1)
select &dbf2
count to nm
go top
num2 = fcount()
for i = 1 to num2
    if bm(cd1) = field(i)
        for j = 1 to nm
            -cd1 = bm(cd1)
            if alltrim(&-cd1) = ""
                -num1 = j
                exit
            endif
            -num1 = nm
            skip
        next
        dimension zdsr(-num1 - 1, 1)
        copy to array zdsr field &-cd1
        -curobj = 2
        show get cd enable
        show get tj disable
        exit
    else
        if i = num2
            tj = ""
            sele 1
            if fld(cd1, 2) = "D"
                tj = { / / }
            endif
            -curobj = 1
            show get tj enable
            show get cd disable
        endif
        show get lj, 1 disable
    endif
endif
next

```

(来稿时间:1997年1月)