

基于 Foxpro 通用快速分类 查询程序的设计

李 晓 童 立 刘亚峰
(中科院新疆物理研究所)

摘要:本文简述了通用快速分类查询的方法和程序设计思想,并给出了应用实例和程序清单。

一、问题的提出

分类查询是按不同类型和不同层次进行查询。为满足人们这方面的需求和程序通用性,我们用 Foxpro 在微机上编写了通用快速分类查询程序。在应用系统开发中遇到分类查询时都可以使用该程序模块,缩短了应用系统的开发周期。通用快速分类查询程序适应于分类代码管理的数据库系统,如:物资管理信息系统、设备管理信息系统等。

二、实现通用快速分类查询的方法

1.快速分类查询的结构设计

快速分类查询的结构必须满足树型层次结构,叶子节点为具体内容,内节点为层次分类的描述内容。上一层节点与下一层节点是一对多的关系,每一个节点用一个代码表示,每一层的代码宽度相等,且上一层代码包含在下一层代码中。快速分类查询的结构设计就成了分类代码的设计,代码设计原则为:每一层给定一个分类代码等于该节点上一层相关节点的代码加本节点分类码,分类的层次根据具体应用对象而定。当定位到某一个节点时,与它相关的下一层节点和上层节点及叶子节点将被列出,很容易看出该节点所处的位置和相关的叶子节点。

下面以物资管理为例说明代码设计的原则:一般物资管理的代码分为大类、中类、小类、规格型号共四层,设大类、中类、小类各用两位表示,规格型号用四位表示,码的全长为十个字符位。树型结构为:树根对应所有物资,第一层为大类,第二层为中类,第三层为小类,第四层为规格型号。由于下一层对应的物资包含了上一层对应的物资,从根节点向下移动时对应的叶子节点数逐步减少,最后定位到所要查的物资。

2.快速分类查询的关系建立

为了实现快速分类查询,必须将叶子节点和内节点分开,组成分类库和明细库,然后建立叶子节点和内节点的映射关系,当内节点变动时对应的叶子节点也跟着一起变动,叶子节点和内节点的关系是靠节点代码联系的。

三、Foxpro 有关命令的简介

Microsoft Foxpro 在 1991 年 7 月 2.0 版推出后,荣获了当年度 Computer Language, Infoworld, PC Magazine, PC Computer 等杂志评选的多项优秀成果奖。它有四个平台的版本 DOS、UNIX、Windows、Macintosh。1992 年 3 月推出的 2.5 版具有更加强大的功能,其它微机上运行的数据库难以比拟的速度,完善而丰富的工具,极具友好的图形界面,独一无二的跨平台特性以及真正的可编译性等突出特点。另外它使用 Rushmore 查询优化技术,比以往的产品查询速度更快,提供的 4GL 功能使开发者无需为编程而花费多余的时间。下面介绍有关的命令。

1.@GET—列表

@GET—列表命令是 Foxpro 中使用最广泛的命令之一,它生成一个列表,列表中有系列项目,用户可以从这一系列项目中选择其一,列表显示在一个框内,用户可以用键盘和鼠标对其操作。列表中的项目来自一个数组或弹出式菜单,使用 FROM <数组名> 可以从一个数组中建立列表,使用 POPUP <弹出式菜单名> 可以从由 DEFINE POPUP 生成的弹出式菜单中建立列表。该命令用 READ 或 READ CYCLE 来激活。命令的基本格式如下:

```
@行号,列号 GET<变量>|<域名>FROM<数组名>|POPUP<弹出式菜单名>
FUNCTION"&"|PICTURE"@&"
WHEN 自定义函数
VALID 自定义函数
...
```

这里需要说明的是 WHEN 和 VALID 子句。

WHEN 子句是当光标在列表中从一个项目移入另一个项目时执行的一段用户自定义的函数,在这个函数中可以执行用户编写的程序。VALID 子句是当光标在列表中一个项目被选中时执行的一段用户自定义的函数

在这个函数中可以执行用户编写的程序。

2.SET FILTER TO

该命令是把当前打开的数据库中满足给定条件的记录过滤出来,运行 SET FILTER TO 时,只有满足条件的记录可以被显示。命令格式为:

```
SET FILTER TO <逻辑表达式>
```

3.SET RELATION TO

该命令是连接和中断两个或多个打开数据库的关系,在建立关系前,父数据库必须在当前工作区中打开,而另一个子数据库必须在另一个工作区中也打开,然后才能用 SET RELATION 来建立关系,当关系建立以后,移动父数据库的记录指针同样也将子数据库中的记录指针移到对应的记录上。如果子数据库中没有对应的记录,则子数据库中的记录指针移到子数据库的尾部。父数据库和子数据库的关系是通过两库的相同字段建立的,并子数据库必须按相同字段索引。命令格式为:

```
SET RELATION TO [<相同字段>]INTO<子数据库
1 别名或区号>
[,<相同字段>INTO<子数据库
2 别名或区号>...]
[ADDITIVE]
```

四、程序设计要点

1.分类库与明细库的建立

分类库由内节点组成,分类代码(code);分类名称(name);内部代码(intercode)。明细库由叶子节点组成。库结构如下:

域名	类型	长度	小数位
code	C	10	
mane	C	20	
intercode	C	10	

内部代码是标准的树型结构码,每一层定两位,用在分类库内。

分类代码用来建立与明细库的关系,该域名必须与明细库的代码域名相同。分类代码可根据管理对象决定。

分类名称是和分类代码对应的,表示代码的意义,用汉字表示,用户很容易一目了然。

明细库可根据用户的需要建立,必须有字段名 code,并且按该字段建立索引。

如果用户已有建好的明细库,可以重建一个分类库,分类库中的字段 CODE 可以改为明细库中代码的字段名,其它字段不要改变,程序中相应的名也要修改,或用宏替换替换该名。

2.分类库与明细库的关系建立

首先打开分类库和明细库,明细库必须打开按 code 索引的索引文件,然后进入分类库工作区,最后用 SET RELATION TO 命令建立两库的关系。命令如下:

```
SET RELATION TO ALLTRIM(CODE) INTO<
明细库名>
```

3.@ GET—列表中 WHEN 和 VALID 的 UDF 设计

该命令是程序设计的关键,是该程序的控制中心,所有的控制是靠 WHEN 和 VALID 的 UDF 完成的。WHEN 主要完成分类库与明细库的关系建立,当光标移到某一系列项时,明细库的相关内容将被列出和统计。VALID 主要完成列表内容过滤,即:当定位到某一个节点时,与它相关的下一层节点和上层节点将被列出。

WHEN 的 UDF 为:

```
SELECT 明细库区
SUM<被统计的域名>TO<变量名>FOR
CODE=ALLTRIM(CODE)
SELECT 分类库区
IF 上次代码宽度大于本次代码宽度
*取消关系后重新建立关系
SET RELATION TO
SELECT 明细库区
BROWSE LAST NOWAIT
SELECT 分类库区
SET RELATION TO ALLTRIM(CODE) INTO<明
细库名>
```

ENDIF

VALID 的 UDF 为:

```
SELECT 分类库区
m.recn1 = recn( )
code1 = alltrim(reop111.intercode)
len1 = len(code1)
cond = "len(alltrim(intercode))=0"
nn = 2
do while nn < len1
cond = cond + ".OR.intercode = substr(code1,1," + str(nn) + ". and. len( alltrim(intercode)) = " + str(nn)
```

```

nn = nn+2
enddo
set filter to intercode=code1. and. len (alltrim
(intercode))< =len1+2;
.OR. &cond
go m.recn1
choice = rep0111.name
show get choice
    
```

五、实例

下面以物资管理为例说明程序设计的过程。

1.分类库的建立

假设物资代码为八位,第一位是大类;第二位为子类;

后几位表示规格型号。分类库内容如下:

NAME	CODE	INTERCODE
物资总类		
一、金属材料	j	01
钢材	jg	0101
钢球	jq	0102
有色金属	js	0103
二、二类机电	e	02
电焊条	eh	0201
低压阀	ef	0202
小型设备	ex	0203
标准紧固件	eb	0204
电工产品	ed	0205
三、市场采购	s	03
工具	sg	0301
小五金	sx	0302
电料	sd	0303
油漆	sq	0304
四、轻化建材	q	04
化药	qv	0401
代工产品	qh	0402
橡胶产品	qx	0403
建材	qj	0404

2.建立明细库索引文件和关系

明细库域名为:code,name, model, measure, save_S

num, save_finac

关系为 ALLTRIM(cade),

3.程序生成

用屏幕生成器建立该程序或自行编写.为了在程序中不对应具体数据库,在程序执行前必须将明细库和索引文件打开。

```

select 0
use standard index standard
do wz.spr
    
```

4.程序清单

```

* *****
* * 10/27/94 WZ.SPR 12:18:05
* * This program was automatically generated by GENSCRN.
* *****
* * WZ/Windows Setup Code - SECTION 1
*
#REGION 1
public code1,code3
code3="e"
code2="e"
dbfname=dbf()
#REGION 0
REGIONAL m.currarea, m.talkstat, m.compstat
IF SET("TALK") = "ON"
SET TALK OFF
m.talkstat = "ON"
ELSE
m.talkstat = "OFF"
ENDIF
m.compstat = SET("COMPATIBLE")
SET COMPATIBLE FOXPLUS
m.rborder = SET("READBORDER")
SET READBORDER ON
m.currarea = SELECT()
* *****
* * WZ/Windows Databases, Indexes, Relations
*
IF USED("rep0111")
SELECT rep0111
SET ORDER TO 0
ELSE
SELECT 0
USE (LOCFILE("\foxpro\cl\dbfs\rep0111.dbf","DBF","Where is rep0111?")
AGAIN ALIAS rep0111 ORDER 0
ENDIF
* *****
* * Windows Window definitions
*
IF NOT VEXIST("_qum0qd6pb")
DEFINE WINDOW _qum0qd6pb AT 0.000, 0.000 ;
SIZE 23,308,53,800 ;
FONT "MS Sans Serif", 8 ;
FLOAT NOCLOSE MINIMIZE SYSTEM
ENDIF
* *****
* * WZ/Windows Setup Code - SECTION 2
*
#REGION 1
jc=0.00
choice=rep0111.name
SELECT rep0111
SET RELATION OFF INTO &dbfname
SET RELATION TO alltrim(rep0111.code) INTO &dbfname ADDITIVE
select (m.currarea)
brow fields code,name,model,measure,save_num,save_finac,rep01;
normal nowait nomodi last title "库房材料"
sele rep0111
set filter to len(alltrim(intercode))<=2
#REGION 1
DEFINE POPUP _qum0qd7pe PROMPT FIELD name SCROLL
* *****
    
```

```

* * WZ/Windows Screen Layout
#REGION 1
IF WVISIBLE("_qum0qd6pb")
  ACTIVATE WINDOW _qum0qd6pb SALT
ELSE
  ACTIVATE WINDOW _qum0qd6pb NOSHOW
ENDIF
@ 11.154,34.400 SAY "结束" ;
  FONT "MS Sans Serif", 8 STYLE "T"
@ 1.923,2.600 GET choice ;
  PICTURE "@&N" ;
  POPUP _qum0qd7pe ;
  SIZE 12.692,28.600 ;
  DEFAULT " " ;
  FONT "MS Sans Serif", 8 ;
  STYLE "" ;
  WHEN _qum0qd7z5() ;
  VALID _qum0qd86x()
@ 17.538,14.400 GET qt ;
  PICTURE "@&HN 退出" ;
  SIZE 1.923,10.167,0.667 ;
  DEFAULT 1 ;
  FONT "MS Sans Serif", 8 ;
  STYLE "B" ;
  VALID _qum0qd8cu()
@ 13.000,34.000 SAY jc ;
  SIZE 1.000,18.400 ;
  FONT "MS Sans Serif", 8
IF NOT WVISIBLE("_qum0qd6pb")
  ACTIVATE WINDOW _qum0qd6pb
ENDIF
READ CYCLE
RELEASE WINDOW _qum0qd6pb
* *****
* * Windows Closing Databases
IF USED("rep011")
  SELECT rep011
  USE
ENDIF
SELECT (m.currarea)
RELEASE POPUPS _qum0qd7pe
#REGION 0
SET READBORDER &rborder
IF m.talkstat = "ON"
  SET TALK ON
ENDIF
IF m.compstat = "ON"
  SET COMPATIBLE ON
ENDIF
* *****
* * choice WHEN
* * Object Type: List
* * Snippet Number: 1
* *****
FUNCTION _qum0qd7z5 && choice WHEN
#REGION 1
SELECT (m.currarea)
sum save_finac to jc for code=alltrim(rep011.code)
go top
show gets
sele rep011
code2=code3
code3=alltrim(code)
if lenalltrim(code)<len(code2)
  set relation to
  SELECT (m.currarea)
  brow last nowait
  sele rep011
  set relation to alltrim(rep011.code) into &dbfname
endif
* *****
* * choice VALID
* * Object Type: List
* * Snippet Number: 2
* *****
FUNCTION _qum0qd86x && choice VALID
#REGION 1
code1=alltrim(rep011.intercode)
!ani=len(code1)
cond="len(alltrim(intercode))=0"

```

```

nn=2
do while nn<len1
  cond=cond+".OR.intercode=substr(code1,1,"+str(nn)+":
  ".and.len(alltrim(intercode))="+str(nn)
  nn=nn+2
enddo
set filter to intercode=code1.and.len(alltrim(intercode))<len1+1:
.OR. &cond
go m.recn1
choice=rep011.name
show get choice
retu
* *****
* * _QUMQD8CU qt VALID
* * Object Type: Push Button
* * Snippet Number: 3
* *****
FUNCTION _qum0qd8cu && qt VALID
#REGION 1
clear read

```

参考文献:

- [1]《FOXPRO 语言实用详解》亦欧主编
- [2]《FOXPRO Developer' Guide》

FoxBASE 下报表汇总系统的实现方法

周国钢 (新疆财经学院)

在会计信息系统中,报表汇总是一个重要的子系统。随着现代企业制度的建立和实施,越来越多的企业建立起下属企业和生产营销网点,总公司需要定期掌握全面的财会信息,为管理、决策提供依据。这样,在计算机会计信息系统中,对所属企业各种报表的汇总显得日益重要。本文就笔者在实践中将报表汇总系统简洁、高效、通用的实现思想和方法加以总结,供同行参考。

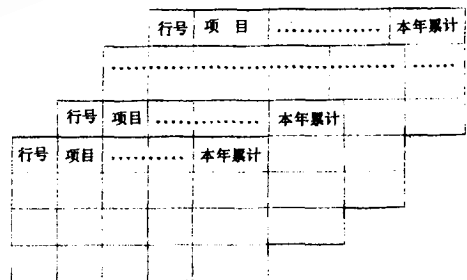


图 1

报表汇总的形式一般有两种,一种是若干张相同格式的报表在三维空间的数据叠加,如图 1 所示。每张表格是一个分公司的相关表格,相同行列的数据需要加在