

数据查询模式研究及在 PowerBuilder 中的实现

牛继来,何泽恒,潘庆和

(哈尔滨商业大学 计算机与信息工程学院,黑龙江 哈尔滨 150028)

摘要:查询是数据库管理系统中不可缺少的功能,所研究的数据查询模式包括组合查询及模糊查询两种。该技术充分利用了 PowerBuilder 中数据窗口的功能,从根本上解决了传统数据查询方式的弊端,提高了开发效率,方便了用户的使用,并且在实际项目中得到了很好的应用。

关键词:数据查询;组合查询;模糊查询;数据窗口;PowerBuilder

中图分类号:TP311.1

文献标识码:A

文章编号:1673-629X(2006)07-0061-03

Study on Data - Query and Implementation in PowerBuilder

NIU Ji-lai, HE Ze-heng, PAN Qing-he

(School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China)

Abstract: Query is the indispensable function of data base management system. The data - querying mode under studied includes two kinds of technologies. They are combination query and fuzzy query which use the function of data window enough based on PowerBuilder. The technology solves defects of traditional data - querying mode thoroughly, and enhances the efficiency of development, using conveniently, and has better application in practical items.

Key words: data query; combination query; fuzzy query; data window; PowerBuilder

随着社会的不断发展和进步,人们对各种信息的需求也越来越突出。然而,目前在信息系统中经常使用的查询模式比较传统、不方便,在一定程度上影响了工作的效率,若想查找的结果准确、实用,选择一种高效、便捷的查询模式是至关重要的。文中所倡导的组合与模糊数据查询模式则解决了目前所存在的问题,对方便用户操作、提高工作效率具有很高的实用价值。该查询模式已在笔者开发的若干实际项目(如:大型制药厂质量监测系统、乳业配送管理系统)中得到了很好的应用。

1 目前存在的问题

熟悉软件的人都知道,无论开发者使用哪一种开发工具,其查询方式都不会脱离传统的查询模式,即运用多组文本框实现条件的组合查询或模糊查询。该模式虽然比较直观,但无论是从开发者还是使用者的角度来讲,其开发效率与使用效果并不理想。

2 利用数据窗口实现数据的查询

2.1 数据窗口的可行性

PowerBuilder 是专门设计企业级 C/S(客户端/服务

器)模式应用程序的开发工具,它具有开放的体系结构、友好的用户界面,同时提供了快速建立应用程序的强有力的工具数据窗口(Data Window)^[1]。它独特的 User/Database(用户/数据库)接口为程序员最大限度地节省了时间和精力。在使用 PowerBuilder 编写应用程序时,数据窗口功能强大、应用灵活,尤其是其查询模式配合检索功能,可方便地实现组合查询与模糊查询的功能。

2.2 数据窗口的查询功能

文中讨论的内容充分运用了数据窗口这一有利工具,实现了数据的组合及模糊查询功能。在传统模式下,当用户调用某一数据窗口的 Retrieve()函数查询数据时,只能检索到符合 Select 语句的数据。但用户希望真正检索到的是自己感兴趣的数据,而开发人员却往往并不清楚用户的真实需要,这时,使用 PowerBuilder 提供的查询模式,就可以根据用户的需要检索相应的数据,同时摆脱传统模式的局限性。但有一点需要注意,如果该数据窗口的数据是动态生成的,那么就不可以使用此查询模式了^[2]。

3 数据查询功能的实现

3.1 设计思想

在 PowerBuilder 中利用数据窗口实现数据的查询功能主要是运用了 Query Mode 的思想,它类似 QBE(Query By Example)查询模式,但其功能没有 QBE 那么强大^[3]。在此模式下,用户可以通过在任一系列或多列中指定 SQL 语言的 Where 语句(即查询条件),然后调用 Retrieve()函

收稿日期:2005-10-16

基金项目:黑龙江教育厅科学技术研究资助项目(10551094)

作者简介:牛继来(1980-),男,山东济南人,硕士研究生,研究方向为计算机网络;何泽恒,教授,硕士生导师,研究方向为数据库与信息

系统。

数,该函数将根据指定的 SQL 语句执行各种查询功能。

3.2 组合查询

使用数据窗口实现组合查询避免了传统查询模式带来的弊端。无论使用何种开发工具实现组合查询,其各种条件之间无非是‘与’(AND)、“或”(OR)两种最基本的关系。这在数据窗口的查寻模式下有简单的实现方式,即同一行输入的条件为‘与’关系(如图 1 所示),不同行为‘或’关系(如图 2 所示)。图 1 是在同一行中不同字段内输入查询条件,当点击【执行查询】按钮时,便会检索出品种编码包含“004”,并且品种名称包含“饮用水”字样的数据,然后显示在数据窗口中。图 2 与上述相似,只是其关系为‘或’,在此不再赘述。

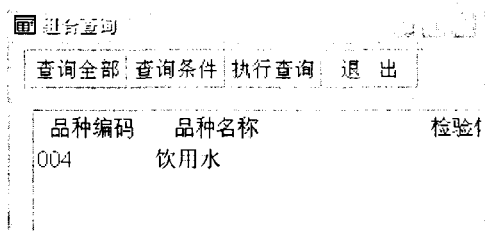


图 1 组合查询中的‘与’关系

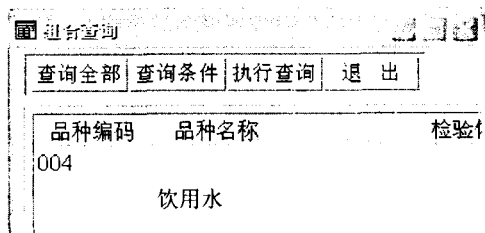


图 2 组合查询中的‘或’关系

3.3 模糊查询

模糊概念大量存在于人们的观念之中,在对事物进行分类时,总希望对集合中的成员划分等级^[4],如模糊查询中的前置模糊(即如“X%”的形式,其中,X为用户输入的模糊查询条件);后置模糊(即如“%X”);完全模糊(即如“%X%”)等。采用明确边界对系统建模的方法在实际应用中的效果有时并不好,除非能严格的控制系统,否则会带来不必要的后果,模糊查询正是处理模糊概念的有利工具。

下面针对 3 种模糊分类进行了实现,其中仅以某一字段(品种编码)为例。如果用户需要检索品种编码以“14”开头的数据集(Data Set),即以“14”进行前置模糊查询,则只需在点击【查询条件】按钮的前提下在品种编码所在列的任意一行中输入“+14”,然后点击【执行查询】即可,相应的数据就会显示在数据窗口中,如图 3 所示。

同理,如果需要检索品种编码以“14”结尾的数据集,即以“14”进行后置模糊查询,则只需在点击【查询条件】按钮的前提下在品种编码所在列的任意一行中输入“-14”,然后点击【执行查询】即可,数据就会显示在数据窗口中,如图 4 所示。

此外,如果用户并不限制输入条件出现的位置,可以

采用完全模糊的查询方式,即直接在品种编码所在列的任意一行中输入“14”即可,其结果为上述两种查询结果的集合。

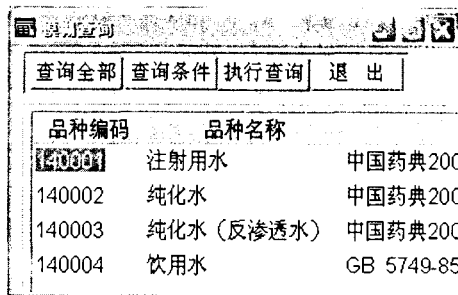


图 3 前置模糊查询

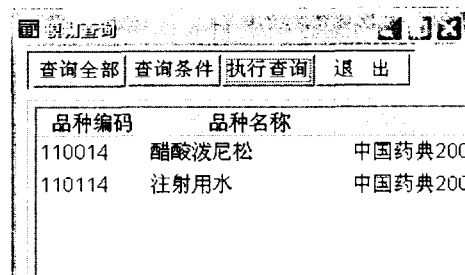


图 4 后置模糊查询

在上例中需要说明的是“+”,“-”的作用。这里采用的加减号是一种工具符号,并不是开发工具默认的,而是开发人员为了方便用户的使用,便于程序的开发而采用的。当在查询条件前加入“+”时表示是前置模糊查询,而加入“-”表示是后置模糊查询,具体的实现原理与过程将在下文做详细的说明。

4 查询功能的实现原理与应用

数据查询功能的实现原理现与应用以上述中的图 4 为例,进行具体说明。

4.1 按钮的基本功能

如图 4 所示,例中有 3 个主要按钮(【查询全部】、【查询条件】、【执行查寻】)需要说明。其中,

【查询全部】按钮是检索所有数据并在数据窗口中显示出来,并使其查询模式设为假,其实现语句为:

```
dw_query.reset()
//查询模式设为假,此时用户无法继续输入查询条件,不允许对其进行查询
dw_query.modify("datawindow.querymode=FALSE")
dw_query.settransobject(sqlca)
//检索数据
dw_query.retrieve()
```

【查询条件】按钮是用于输入用户的查询条件,因为在其它模式下,数据窗口是不可编辑状态,无法输入查询条件,这就需要修改其编辑状态,其实现语句为:

//查询模式设为真,当程序执行到该语句时,数据窗口将被清空(此时,数据缓冲区中的数据并没有被清除)^[2],数据窗口进入查询模式,用户可以在相应字段上输入相应的检索条件

```
dw_query.modify("datawindow.querymode=TRUE")
//数据窗口清空原有查询条件
```

```
dw_query.modify("datawindow.queryclear=TRUE")
```

【执行查询】按钮是在用户已经输入查询条件的前提下,根据输入的查询条件转换为指定的 SQL 语句进行查询,如果查询条件为空,则按【查询全部】按钮功能进行操作,其实现语句为:

```
long li_row
//将查询条件所在行赋给变量 li_row
li_row = dw_query.insertrow(0)
//使焦点回滚到 li_row 行
dw_query.scrolltorow(li_row)
dw_query.modify("datawindow.querymode=FALSE")
dw_query.retrieve()
```

另外,如果用户希望强制某一字段的输入情况,使其不允许为空,可加入如下限制语句:

```
dw_query.modify("column.criteria.required=TRUE")
```

4.2 组合查询的实现

文中,组合查询是基于 QBE 的思想设计而成的,这也是 PowerBuilder 的优势所在。另外,在组合查询中,用户可以在查询条件中使用各种操作符(>, >=, <, <=, <>, is, like 等)的有效组合,若只输入了值而未指定操作符,系统默认为 '='^[5]。

4.3 模糊查询的实现

在模糊查询中,系统(DataWindow)一方面接收来自用户的查询条件,另一方面还需将其转化为 SQL 语句,返回系统并执行操作。这一过程应在哪一事件下完成?系统如何判断用户是否已经完成输入操作? PowerBuilder 为人们提供了一种事件,可以同时解决这两个疑问,这就是 ItemChanged 事件。该事件是在数据窗口被修改过或失去焦点(例如:用户按了 Enter 键、Tab 键,或是在其它位置单击鼠标等)时激发,同时,系统的 Accepttext() 函数将数据窗口中漂浮在当前行/列上的编辑框中的内容放入数据窗口的主缓冲区当前项中,然后便可生成相应的 SQL 语句,提交返回系统,执行查询功能。在这里要实现上述模糊查询的 3 种方式,必须编程控制,其代码实现如下所示:

```
//变量声明(ls_text 文本内容 ls_cxtj 查询条件 ls_cxbz 查询标志)
```

```
string ls_text, ls_cxtj, ls_cxbz
```

```
//从条件内容中取出最左边第一位,作为查询判断
```

```
ls_cxbz = left(trim(dw_1.gettext()), 1)
```

```
Choose case ls_cxbz
```

```
case "+" //前置模糊查询
```

```
if dw_1.object.datawindow.querymode = 'yes'
```

```
then //从数据窗口获得查询条件
```

```
ls_text = dw_1.gettext()
```

```
ls_cxtj = right(ls_text, len(ls_text) - 1)
```

```
ls_cxtj = 'like' + ' ' + ls_cxtj + '%'
```

```
//将悬浮在数据窗口控件当前行/列上编辑框的内容替换为指定内容
```

```
dw_1.settext(ls_cxtj)
```

```
dw_1.accepttext()
```

```
end if
```

```
case "-" //后置模糊查询
```

```
if dw_1.object.datawindow.querymode = 'yes'
```

```
then
```

```
ls_text = dw_1.gettext()
```

```
ls_cxtj = right(ls_text, len(ls_text) - 1)
```

```
ls_cxtj = 'like' + ' ' + ' ' + ls_cxtj
```

```
dw_1.settext(ls_cxtj)
```

```
dw_1.accepttext()
```

```
end if
```

```
case else //完全模糊查询
```

```
if dw_1.object.datawindow.querymode = 'yes'
```

```
then
```

```
ls_text = dw_1.gettext()
```

```
ls_text = 'like' + ' ' + ' ' + ls_text + ' %'
```

```
dw_1.settext(ls_text)
```

```
dw_1.accepttext()
```

```
end if
```

```
End choose
```

至此,3 种模糊查询方式便已实现,用户可根据具体情况选择不同的查询方式来满足实际工作中的不同需要。

5 结 论

数据查询是管理信息系统的主要功能,其通用性程度与使用是否方便,直接关系到数据库应用系统开发的成败^[6]。文中阐述的各种查询模式从根本上解决了传统的通用及万能查询模式存在的缺陷,有助于程序员提高开发速度、缩短开发周期、减少开发成本,同时,也有利于用户快速掌握应用软件、提高工作效率,相信对实际工作具有很大的现实意义和应用价值。

参考文献:

- [1] 严怀成,王 敏.基于 PowerBuilder 的通用查询的设计与实现[J].微机发展,2005,15(6):124-126.
- [2] 刘 靓,王继东,王东阳,等. PowerBuilder8.0 实例(上册)[M].长沙:国防科学技术大学出版社,2002.
- [3] 张长富. PowerBuilder9.0 实例教程[M].成都:电子科技大学出版社,2004.
- [4] 张红兵,赵杰煜,罗雪山.综合模糊逻辑的贝叶斯网络在态势评估中的应用[J].计算机应用研究,2004(7):167-170.
- [5] 杨 昭. PowerBuilder9.0 对象与控件技术详解[M].北京:中国水利水电出版社,2003.
- [6] 李元松. PowerBuilder 应用中一种通用的查询系统设计[J].电脑开发与应用,2003,16(2):30-32.