

# 一种完全活动报表的 设计与实现

杨扬 李亚芬 (大连理工大学电子与信息工程学院 116024)

**摘要:** 本文给出了一种利用数据表实现活动报表的设计思想及其实现方案。使用此种方法可以在不更改源程序的情况下, 灵活方便地定义报表。

**关键词:** 数据库 报表

## 1 前言

报表设计是数据库应用系统的一个重要环节, 而且实现较为繁琐。使用Crystal Reports虽然可以简化报表设计过程, 但灵活性太差。如果报表环节设计得不灵活, 当用户要求发生改变时(如改变报表内容或增删报表等), 则需要不断修改源程序。本文结合作者的体会给出一种通过数据表(Table)实现完全活动报表的设计方案, 使用这种方法可以让开发人员或数据维护人员灵活地定义报表, 让系统的使用者自由地选择报表的具体输出内容和格式。

## 2 设计思想

在数据库应用系统中, 表(Table)是一个应该充分利用的资源, 除了可以用于管理数据, 还可以用于存储和管理其他信息。如果能利用表来存储各种报表信息(生成或打印信息), 又能通过修改数据表而不更改源程序实现对报表的设计, 会大大提高应用程序的适用范围。本设计方案正是基于这一思想而提出的。

需要在数据库中添加的与报表设计有关的表有: 报表类别定义表printlevel、报表内容定义表printdes和报表列信息定义表printcol。下面简要给出这三个表的定义, 其中提到的字段均为string类型。

printlevel表用于存储报表的类别信息, 即每种报表所属类别(日报表、月报表等), 通过它来实现对众多报表的分类。它的主要字段有pl\_key和pl\_level。其中,

pl\_key是主键, 指报表的名称; pl\_level指报表的所属类别。

printdes用于定义报表内容, 它的主要字段有: pd\_key、pd\_datasource和pd\_where。其中pd\_key是主键, 它的意义同pl\_key; pd\_datasource是生成该报表需要关联的表; pd\_where指各关联表之间的关联条件。printdes表中另外还有大量的字段用于定义报表的页面设置、字体信息和表头信息等, 这里不再赘述。

printcol表用于存储各报表可以提供输出的所有列的信息。它的主要字段有: pc\_cname、pc\_fname和pc\_key。其中, pc\_cname指报表中列的显示名称; pc\_fname指列在关联表中的字段名; pc\_key指该列所属的报表, 但是这里这个字段并非主键, 因为任何报表总有不止一列。这三个表中的内容应由系统开发人员或数据维护人员根据用户的要求填入并更新。

该设计方法设计报表的灵活性主要体现在以下两个方面:

(1) 报表的增删/修改灵活。开发人员或数据维护人员随时可以通过修改后台数据库中的这三个表完成。这一环节使开发人员可以在不修改源程序的情况下完全自由地定义报表。

(2) 用户可以通过前台程序自由选择报表的具体输出列、添加过滤条件、设置表头信息等, 这一环节在一定程度上给了使用者定制报表的自由。

值得注意的是, 前台程序是通过pl\_key(或pd\_key)、

pd\_datasource、pd\_where、pc\_fname 和用户指定的其他过滤条件生成提取报表内容的 SQL 语句，因此要求这几个字段的内容要严格符合 SQL 语法规定。

### 3 实现

根据上述设计思想我们在一商业企业的物流管理系统中设计了如下图所示的报表中心。

如果说前面介绍的三个表的定义是活动报表实现的基础，那么前台的界面才使后台表的定义真正有意义。图 1 所示的窗口有机地反映了前面定义的表，并利用了它们的信息获取了报表内容。

窗口中各 ListBox 的内容在窗口初始化时填入。其中“报表类别”列表框 (List1) 中的内容通过下面的 SQL 语句得到：

```
“select distinct pl_level from printlevel”
```

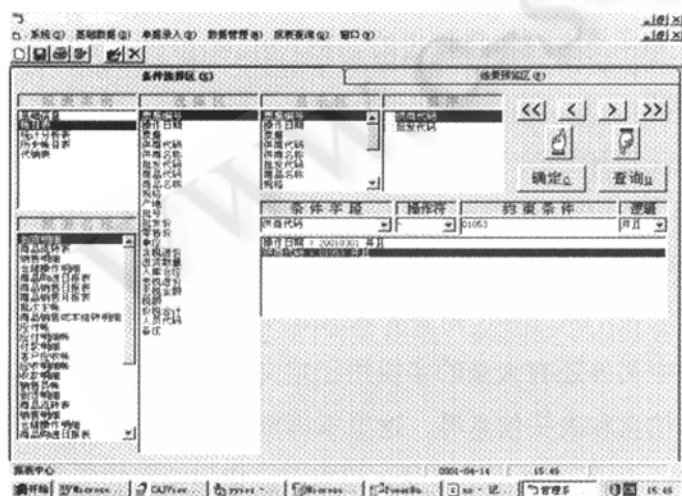


图 1 活动报表实现窗口

左下角的“报表名称”列表框 (List2) 中的内容根据选择的报表类型从 printlevel 中提取，相应的查询 SQL 语句为：

```
“select pl_key from printlevel where pl_level='” & List1.Text & “”
```

“选择区”列表框显示的是从 printcol 中提取的相应报表所有可以显示的列，它的内容通过下面的 SQL 语句获得：

```
“select pc_cname,pc_fname from printcol where pc_key='” & List2.Text & “”
```

“显示区”存放用户从“选择区”选择的所需要显示和打印的列；“排序”用于指定显示顺序；窗口右下部为条件选择区，用于用户指定过滤条件。

可以说，通过“报表名称”、“选择区”选择的最终结

果是生成了一个能查询出“显示区”所选列的 SQL 语句；通过“排序”、“条件字段”、“操作符”、“约束条件”等条件选择区的选择，给这个 SQL 语句又加上了一些过滤条件。如果把显示区中用户选择的列对应的 pc\_col 字段存入动态数组 ColArray 中，过滤条件存入字符串 strFilter 中，则生成的提取报表内容的 SQL 语句 strSQL 为：

```
For i=1 to UBound(ColArray()-1
```

```
strDisp=strDisp & ColArray(i) & “,”
```

```
Next
```

```
StrDisp=strDisp & ColArray(i)
```

```
strSQL=” select ” & strDisp & “ from ” & pd_datasource & “ where ” & pd_where & “ and ” & strFilter
```

则当点击“查询”键时，应用程序通过执行这个 SQL 语句提取查询结果显示在“报表预览区”中。当用户执行打印命令时，应用程序使用“报表预览区”中的数据和 printdes 表中对应报表的页面设置、字体设置信息进行报表的打印机输出。

在我们的实现中，同时考虑了权限问题，对不同角色用户的报表使用权限做了规定。这是通过 printdes 表中上面没有提到的一个字段 pd\_role 实现的。系统在“报表中心”窗口初始化时，对填入“报表名称”列表框中的内容使用了角色权限进行过滤，以保证不在登陆用户权限范围内的报表无法查询和显示。它们的实现相对简单，在此不再赘述。

### 4 结束语

本文给出了一种活动报表的实现方案，使用此种方法可以在不更改源程序的情况下方便地定义报表。对于不同的系统，前台程序是普遍适用的，不同系统的区别仅在于前面提到的三个表的内容不同，因此该设计思想具有一定的通用性和实用价值。本文中示例系统是使用 Visual Basic 和 SQL Server 开发的并已得到成功应用。■

#### 参考文献

- 1 Microsoft Corporation. Visual Basic 6.0 Books On Line [Z].
- 2 Eric Brierley, Anthony Prince, David Rinaldi 著, Visual Basic 6.0 开发人员指南 [M], 机械工业出版社.
- 3 Microsoft Corporation. SQL Server Books On Line [Z].