

1 对多关系在会计核算文件设计中的应用

马 慧 (首都经济贸易大学信息学院 100026)

平行登帐是手工会计的重要一环,平行记帐即每一笔经济业务都要平行记录总分类帐和明细分类帐。在电算化会计中,输出的帐簿均来源于凭证文件和辅助文件,即数据来源是一致并唯一的,不需要进行平行登帐。

1 凭证文件的设计

在会计分录中,像日期、凭证号、及附件等数据项在一张凭证仅需要设计一个即1对1的关系,而借、贷科目的数量差距较大,有一借多贷,一贷多借等多种形式。假设凭证文件中设计了5个借方科目,5个贷方科目,一旦在某个分录中含有6个贷方科目,文件设计就不够用了,而对于那些只有一借二贷、一借一贷、一贷二借的业务又浪费了大量的空间。对于业务和会计科目之间“一对多”的关系,是定义一个文件组成的固定结构还是设计两个文件组成的变动结构是文件设计的焦点,结构如图1、图2所示。

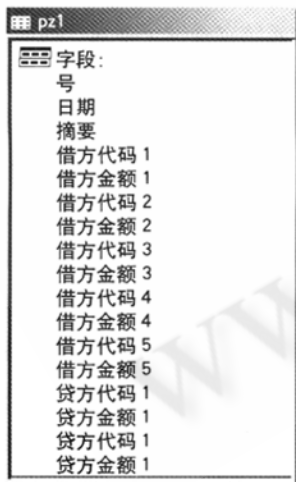


图1 固定结构

可以看出,采用变动结构的文件具有很大灵活性,由于文件中一条纪录对应唯一的一个会计科目,这对于基于会计科目而进行的各种索引、汇总与查询、以及报表数据的输出是及其有利的。此结构还易于数据加工过程并节约存储空间。

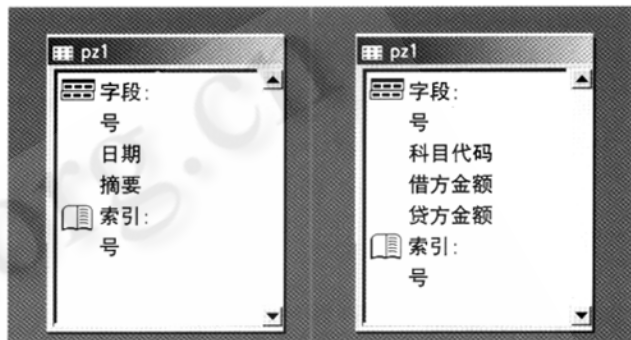


图2 变动结构

E.F. Codd在1970年以关系集合理论中重要的数学原理为基础,提出不规范的数据库文件会带来冗余太长、插入异常、删除异常的弊病,它将造成编程困难,模块互相依赖性大,产生水波效应等不良反应。“一事一地”是解决不良设计的重要出发点。第一范式:消除重复字段并满足关系的性质;第二范式:消除非主关键字对主关键字的部分依赖。

在固定结构中主关键字是凭证号+科目代码I,金额项对主关键字是完全依赖,日期、摘要、审核人等数据项对主关键字是部分依赖。为了消除部分依赖对固定结构1进行分解,形成只包含完全依赖的两个文件(图2),规范化过程如图3所示。

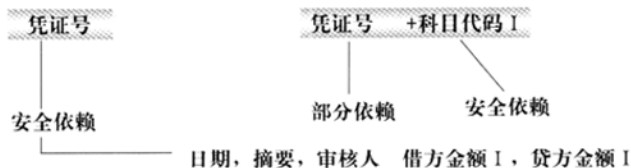


图3 完全依赖形成可变结构

2 1对多关系和审核功能的实现

Visual Foxpro提供了良好的数据库规范化的环境。对于1对多的关系提供向导技术,它支撑表单、报表的功能,可以让我们编写很少的命令便可以实现凭证文件1对多的

数据加工工作。如图 4 所示。

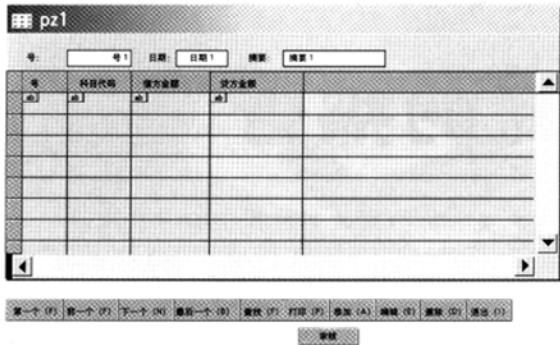


图 4 1 对多结构的表单

在实践中,对凭证的审核是必不可少的环节。在大量的实践过程中,我们利用 1 对多的向导创建表单,利用 sele2 命令和向导建立联接,对凭证文件进行操作,计算累计借方金额和累计贷方金额,检查是否相等。在处理前保留纪录指针。审核命令如下:

```

sele 2
n1=recno()
sum 借方金额, 贷方金额 to a1, b1 for 号=thisform.号
1.value
if a1<>b1
messagebox("不平衡")
endif
go n1
retu
    
```

相比而言,如果用固定结构设计凭证文件,审核功能实现起来要麻烦得多,命令如下:

```

A=thisform.text6.value+thisform.text8.value+thisform.text10.value+
Thisform.text12.value+ thisform.text14.value
B=thisform.text16.value+thisform.text18.value+thisform.text20.value+
Thisform.text22.value+ thisform.text24.value
If a<> b
Messagebox("不平衡")
Endif
Retu
    
```

3 辅助帐的设计

会计软件正从核算型向管理型发展而为管理者提供

会计信息。不仅可以利用科目代码处理完成基本的会计核算,而且部门核算、个人往来、项目核算以及部门收支分析等辅助核算也是必不可少的组成部分。下面以部门辅助帐设计为例,说明“一事一地”规范化思想的应用。

进行辅助核算有两种设计方案:①增设明细科目;②定义辅助核算科目并且设计辅助核算帐目文件。

例如 (501) 产品销售收入、(502) 产品销售成本、(503) 产品销售费用需要进行部门核算。其中部门包括 01 厂部、02 供应部、03 销售一部、04 销售二部、05 车间。

方案 1: 设计明细科目

- 501 产品销售收入
- 50101 厂部产品销售收入
- 50102 供应部产品销售收入
- 50103 销售一部产品销售收入
- 50104 销售二部产品销售收入
- 50105 车间产品销售收入

.....

如果采用方案 1 设计科目文件,按部门归集很不方便;当部门核算科目数量逐渐增大时,所增加的明细科目成几何级数增加。数据查询、处理效率较低。

方案 2: 设计部门核算并进行规范化



图 5 辅助核算文件的规范化

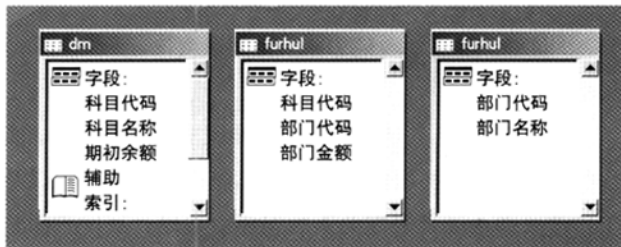


图 6 部门核算文件的设计

综上所述,规范化的方法起到了重要的指导作用;面向对象的开发环境是方案实现的重要保证,而设计良好的文件结构是最重要的基础,可以减轻编程工作量、提高输入输出的灵活性和系统维护性。■