

# 事务处理统计通用程序设计方法

## ——约束条件状态矩阵方法

拱北海关技术处电脑科 郑修贵 张景良

**摘要:**在事务处理系统中,经常会遇到任意改变约束条件的统计和查询,本文提出了一种通用的程序设计方法,允许用户在菜单上定义约束条件并提出统计项目,计算机根据菜单上的逻辑“与”、逻辑“或”和它们的组合,应用约束条件状态矩阵方法自动进行处理,此方法也适用于数据之间互相制约的逻辑控制。

### 一、引言

在事务处理中,尽管各种业务完全不相同,但从软件的角度来看,其计算机的应用一般都具有数据输入、修改、删除、查询、统计、打印等功能。这些任务的实现初看起来很简单,但都相当复杂,要做得十全十美并不容易。这是因为:

1. 由於输入的数据量大,操作员多,必然会产生各种各样的误操作。因此,数据关系的逻辑检查控制就成为一个输入中的重要问题,而逻辑条件一般都很难描述完整;

2. 由於社会科学本身情况复杂,模糊概念、随机因素太多,行业的不同,客观条件的变化,政策的调整,必然要求程序不断进行修改和扩充;

3. 由於业务管理人员广阔的思路,灵活的手工处理方法,软件人员业务知识相对缺乏,使得软件质量往往不尽人意。

以上原因,使得管理软件局限性很大,生命周期较短,要解决这些问题,只能在通用性、适应性上下功夫,才能得到事半功倍的效果。

在数据处理过程中,根据我们的实践,程序人员往往忙於应付上级随时提出的各种业务统计,维护量相当大,几乎陷于无法脱身的地步。本文就此问题提出一个通用软件的设计方法——约束条件状态矩阵方法,使各式各样的统计都是面向用户,界面清晰,业务人员可以在菜单上任意定义统计条件和统计项目,计算机自动计算结果,并进行显示或打印。

### 二、事务处理统计的随机性

在管理软件中,经常使用数据文件(或数据库)来存储记录,每一文件往往要存入几千个到几万个甚至几十万个记录,每一记录存入要记载对象的具体数据,对于这些数据,业务部门经常根据实际需要,提出分门别类,名目繁多的分析统计,具有很大的随机性,以一个记录有30个字段,每个字段的可选个数10个为例,则进行业务统计时约束条件的可能个数为:

$$C = C_n^1 + C_n^2 + C_n^3 + \dots + C_n^{n-1} + C_n^n$$

其中  $n = 30 \times 10$

可能性条件看起来数目相当庞大,但这是一个粗略的估算,也许实际数据远不止如此,因为有些字段,象商品编号,目前至少约一万多个(这还是比较粗的分类),何况其它字段一般都不止10种;也许实际数据比以上结果要少些,因为有些字段不能组成合法的逻辑关系。

如何实现这些逻辑条件的任意组合,以达到随机统计的目的,本文采用了约束条件状态矩阵搜索法进行设计,展示在用户面前的是一张清单,操作员只要在屏幕上把约束条件和统计项目挑选好,计算机就自动进行统计。软件实现是根据用户提出的要求,建立约束条件表和指针表,在动态搜索中建立重复项的状态矩阵,最后按统计指针和状态矩阵之值进行统计。

### 三、约束条件和状态矩阵设计

對於用户提出的要求,首先是把它存到预先设计好的数据表中,在录入过程中同时建立指针表,以提供程序调用。

1.约束条件表设计

一般来说,对某一记录的约束可以归为三种类型:

(1)单项逻辑“与”(或称逻辑“乘”)。即对某一字段的闭区间进行约束,例如:

[1989年10月1日 1989年10月31日],  
[022001 022999]等,它表示了某段时间,某类物品等等。

(2)逻辑“或”(或称逻辑“和”)。即对某一字段进行跳跃式挑选,凡是符合其中某一条件的都应进行统计,例如:美国、日本、英国(统计某些国家);美元、日元、港币(统计某些货币);各种型号(统计某些商品)等等。

(3)逻辑“与”中的逻辑“或”。对一个记录的约束,一般来说,“与”和“或”都会同时存在,对多字段的约束以逻辑“与”出现,而每一个字段的约束则有(1)和(2)两种。即 AND 中有 AND,AND 中有 OR 或只有 OR。

当然,还有其它形式的条件,例如排它条件,即除了某某以外的都统计,但都可以归结为以上三种。

假设一个记录的数据项个数为 n(包括重复出现的项目),每一项使用逻辑“或”可供挑选的项目数最多为 m(根据业务具体情况,自行设定),则约束条件表可描述为:

| 字段名   | 开始条件      | 结束条件      | 逻辑“或”条件    |       |            |
|-------|-----------|-----------|------------|-------|------------|
| NAME1 | MANE1 BEG | MANE1 END | NAME1 REP1 | ..... | NAME1 REPm |
| NAME2 | NAME2 BEG | NAME2 END | NAME2 REP1 | ..... | NAME2 REPm |
| ..... | .....     | .....     | .....      | ..... | .....      |
| NAMEn | NAMEn BEG | NAMEn END | NAMEn REP1 | ..... | NAMEn REPm |

表中元素之数据类型应与你的记录定义中字段数据类型保持一致。元素的多少还可以根据需有所增减,好象时间,一般不大可能跳着统计,则可以把逻辑“或”条件去掉(即取消 NAMEi REP1--NAMEi REPm);象名字,不可能有统计区间,则可以把开始和结束条件去掉(即取消 NAMEi BEG,NAMEi END)。

用户提出的约束条件,全部存放在这个表中,它的内容提供给程序调用,以 COBOL 语言为例(以下同),设计如下:

```
WORKING-STORAGE SECTION.
01 NAME.
    03 NAME1.
```

```
05 NAME1 _ BEG PIC 9(6).
05 NAME1 _ END PIC 9(6).
03 NAME2.
05 NAME2 _ BEG PIC 9(4).
05 NAME2 _ END PIC 9(4).
05 NAME2 _ NUM PIC 9(2).
05 NAME2 _ REP OCCURS 30 TIMES PIC 9(4).
03 NAME3.
05 NAME3 _ NUM PIC 9(2).
05 NAME3 _ REP OCCURS 30 TIMES PIC
X(10).
```

其中, m = 30, NAMEi NUM 存放逻辑“或”的个数,其实, m 的值在 COBOL 中完全可以变动,只要按变量可能变化的个数进行定义就可以。

2.统计项目指针表设计

当用户输完约束条件(或称统计条件)以后,到底要统计什么呢,这就需要操作员向计算机提出要求,由于在终端上有清楚的一张菜单,用户可以不用指定具体项目的内容,只要输入菜单上的项目序号即可,但要注意,统计的项目必需是计算型的,否则统计没有意义,计算机也不会接受。

假设可统计项目的个数为 maxa,则统计项目指针可用一维表定义:

(STATI \_ PONITI) 其中, i = 1,2,.....maxa; H. maxa < n.

在 COBOL 中,可以定义如下:

```
WORKING-STORAGE SECTION.
01 DEF _ STATI.
    03 STATI _ NUM PIC 9(2).
    03 STATI _ POINT OCCURS 15 TIMES PIC 9(2).
```

其中, 维数 maxa = 15, STATI NUM 存放用户要统计的个数。

3.约束条件指针表设计

在任何时候,用户不可能提出包罗万象的约束条件,只能按实际情况,按上级要求的具体内容向计算机提出。因此,软件的设计只能按有多少条件就判断多少条件,这样不但可以节省工作量,还可加快搜索速度。为

此,建立约束条件指针表作为辅助工具就很必要。对多个字段的逻辑“与”,可以按此指针表逐一扫描,直至完毕,我们称之为指针定位法。

设多个字段约束的逻辑“与”判据最大可能个数为  $\max b$ (最大只能等于记录的字段个数),则约束条件指针可用一维表定义:

(LOG POINT<sub>i</sub>) 其中,  $i = 1, 2, \dots, \max b$ ; 且  $\max b < n$ .

在 COBOL 中,可以定义如下:

```
WORKING-STORAGE SECTION.
01 DEF...LOG.
    03 LOG...NUM PIC 9(2).
    03 LOG...POINT OCCURS 30 TIMES PIC 9(2).
```

其中,维数  $\max b = 30$ , LOG\_NUM 存放用户逻辑“与”判据个数。

#### 4.重复项约束条件状态矩阵设计

在事务处理记录的定义中,除了单字段的定义之外,还有重复项的定义,也即整个记录除了表头的描述项之外,还有重复项的定义,也即整个记录除了表头的描述项之外,还有重复出现的表体描述。一般来说,如果统计表体项目,其约束条件除了表头某些条件之外,还有对表体内容的某些限制。也就是说,同一个记录内容,符合条件的某些重复项要统计,不符合的就不能统计,与平记录的情况不一样。

假设重复项的最大重复数为  $k$ ,重量项中的字段数为  $l$ ,则可定义一个二维状态矩阵,用它存放检测记录符合条件时的状态。

(STATU<sub>ij</sub>) 其中,  $i = 1, 2, \dots, k$ ;  $j = 1, 2, \dots, l$

STATU<sub>ij</sub> 之值只有两种,符合匹配条件的为 1,否则为 0。

在表头条件满足的基础上,设用户定义的重复项的约束条件为  $h$  个( $h \leq l$ ),当

STATU<sub>i1</sub> = STATU<sub>i2</sub> = ... = STATU<sub>ih</sub> = 1 其中,  $i = 1, 2, \dots, k$

时,则该重复项满足条件,应该统计,否则,无需统计,在 COBOL 中,可以定义如下:

```
WORKING-STORAGE SECTION.
01 DEF STATU.
    03 STA OCCURS 5 TIMES.
```

05 STATU OCCURS 7 TIMES PIC 9.

其中,  $k = 5$ ;  $l = 7$ .

### 四、通用统计方法综述

对于约束条件和统计指针的输入,程序设计是很容易的,首先,在屏幕上设计好一张菜单,操作员按序号挑选,每输入一个项目,在屏幕上进行提示,直至输完为止。三种条件类型的输入格式,可以这样设计,一个字段的逻辑“与”用“-”号标识,逻辑“或”用“,”号分开,多个字段的逻辑“与”由连续输入单字段的约束实现,例如:

891001-891031--开始日期:1989年10月1日,结束日期:1989年10月31日.NATIONAL, HITACHI--型号为乐声或日立.

以上两个条件合起来可表示对该记录字段之间的逻辑“与”要求,这里要注意的是:输入过程中,对数据一定要进行控制,预防非法数据或误操作,以免空统计而浪费时间。

问题的关键是读到一个记录以后,如何来判断它是否符合约束条件,除了前面提过的指针定位法之外,这里再提供三种切实可行的设计方法:

#### 1.多字段逻辑“与”条件分割法

对于逻辑“与”条件,一般是用 IF 语句加上连续的几个“AND”来实现,但这种编程方法不够灵活,且无法实现任意的逻辑“与”组合。因此,必需把每一个条件进行分割,变成一个个的程序段。例如,对时间区间的约束可以这样编写:

```
INP...DATE...RES.
IF (INP...DATE < INP...DATE...BEG) OR (INP...DATE > INP...DATE...END) THEN
GO TO READ...RECORD
END-IF.
```

执行这一段的结果只有两种情况,一种是不满足,则其它逻辑“与”也不用判断,整个记录不符合条件,只好去读下一个记录;一种是满足,则按指针表判断下一个条件。可以想象,指针表执行完以后,这个记录是符合统计条件的。

#### 2.单字段逻辑“或”条件扫描法

在前面的约束条件表中,逻辑“或”条件是存放在一维数组之中,所以只要对该字段的内容与数组元素逐一

进行匹配,只要有一个满足,即认为条件成立,可以再转下一个指针。例如,对贸易国别的扫描,可以这样编写:

```
TRADE_COUNTRY__RES__BEG.
PERFORM VARYING I FROM 1 BY 1 UNTIL I>
TRADE_COUNTRY_NUM
IF TRADE_COUNTRY = TRADE_COUNTRY__REP (I)
THEN
GO TO TRADE_COUNTRY__RES__END
END-IF
END-PERFORM.
GO TO READ_RECORD.
TRADE_COUNTRY__RES__END.
EXIT.
```

其中,I是下标变量,TRADE\_COUNTRY是记录中存放国别的字段名,TRADE\_COUNTRY\_\_NUM是逻辑“或”条件的个数,TRADE\_COUNTRY\_\_REP是存放条件的数组,大家可以看到,只要有一个相等,则马上转入下一个指针,如果执行完,相等条件一个也没有,则只好去读下一个记录。

### 3.状态矩阵方法

前面已经定义过重复项的状态有,如何存储它的值并使用它,现先分析下的一种情况:如果有一种统计,它的约束条件既有表头部分,也有表体部分,而统计的内容是表体的项目。在表头满足的情况下,后一步有两种实现方法,一种是取一重复项,判断剩下条件是否满足,如果都满足则统计,有一不满足则罢,再取第二项,做同样的工作,直至完毕。

另一种是不管表头的还是表体的,一视同仁地进行判断,只是判断表体时,要把情况记入状态矩阵中,等所有条件都满足时再进行统计,这样实现起来方便。我们使用的是后者。

若一个记录的表头部分的条件已经满足,则按表体的第一个条件对所有的重复项循环一遍,结果按0或1记入状态矩阵的第一列中,其次再做第二条件,于此类推。

经过执行以上判据之后,一个记录需要统计的情况分三类(设表头项目数为 head num):

1.判据条件最大序号 $\leq$  head\_num,统计条件最大

序号 $\leq$  head\_num。这时只作表头有关统计项目的累加。

2.判据条件最大序号 $\leq$  head\_num,统计条件最大序号 $>$  head\_num。此时除了累加表头有关项目外,还要进行所有重复项的累加。

3.判据条件最大序号 $>$  head\_num,统计条件最大序号 $>$  head\_num。此时还要寻找大于 head\_num 的约束条件数 h(前面已经定义),对于每一个 i(i = 1,2,...k),当所有的

STATU<sub>ij</sub> = 1 其中 j = 1,2,...,h.

时,才作重复项有关可计算型字段的累加。至于判据条件最大序号 $>$  head\_num,而统计条件最大序号 $<$  head\_num 则为不合法:

根据以上方法,整个统计过程分为下列五步:

- 1.输入约束条件。
- 2.输入统计项目。
- 3.顺序读一记录(或按关键词顺序读),至文件末转6。
- 4.根据条件指针表,判断是否符合逻辑条件,合法时转5,否则转3。
- 5.对每一记录的有关字段进行累加,做完转3。
- 6.显示或打印结果。

## 五、结论

这个方法,可以用于任何事务处理,方法是通用的,但不等于程序对任何项目都通用,每一个项目都必需自己编一个,只是对某一个任务通用而已。这是因为数据文件存储模式,文件个数及其命名法不同,记录的大小,字段的差异,数据类型和长度不同,代码意义不一样,每一事务统计的业务要求完全不同等缘故。

以上介绍的虽然只讲通用统计程序的设计,但可以推广到通用逻辑查询,只要把统计条件去掉,加上屏幕显示功能即成。

此方法还可以推广到通用数据逻辑控制,即定义一个数据文件,其记录的定义为带指针的约束条件,然后,设计一套包括输入修改删除查询等功能的维护程序,把每一逻辑控制条件作为一个记录存入文件中,以后如有修改增加或减少,可直接运行以上的维护程序。