

文章编号: 1000-5013(2009)03 0354- 03

一种单层 BOM 到多层 BOM 转化算法的实现

陈志烽, 陈永红

(华侨大学 计算机科学与技术学院, 福建 泉州 362021)

摘要: 研究当前主要的两种 BOM 表结构——单层 BOM 结构和多层 BOM 结构, 分析它们各自的优缺点. 提出一种基于存储过程的单层 BOM 结构到多层 BOM 结构转化的方法, 并给出部分实现代码. 该算法是基于存储过程的, 因此, 相对于一般的算法具有更好的计算性能.
关键词: 物料清单; 企业资源计划; 数据库; 存储过程
中图分类号: TP 311. 12 文献标识码: A

1 BOM 的数据结构

物料清单(BOM)是指产品所需零部件明细表及其结构^[1], 贯穿于产品的整个生命周期中, 是系统各个功能模块之间进行数据交换的纽带; 而 BOM 的数据结构及其算法构成了企业资源计划(ERP)系统数据模型的核心. 由于实际产品的多样性和产品结构的复杂性, 产品父部件与子部件之间并非一对多或多对一的关系, 而是多对多的网状结构关系^[2]. 对于目前应用广泛的关系型数据库来说, 网状结构的数据结构在数据库里通常难以实现. 这需要将重复使用的子部件拆开, 按不同的部件来处理, 即将数据结构规范成树状结构^[3-6].

图 1 为产品的 BOM 树状图. 常见的 BOM 数据库存储形式主要是多层 BOM (MBOM) 和单层 BOM (SBOM) 两种. 图 1 中产品 A 的多层 BOM 和单层 BOM 数据结构, 如表 1 所示. 表 1 中, ID 是记录号; Parent ID, 表 1 产品 A 的 BOM 数据表

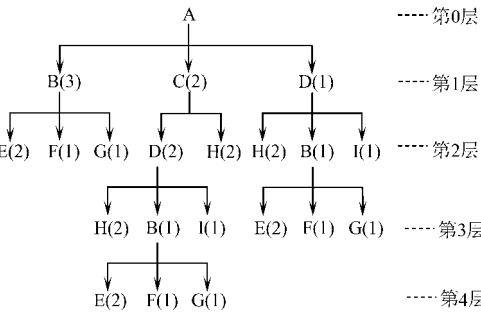


图 1 产品 BOM 结构图

Fig. 1 BOM structure diagram of the product

Tab. 1 BOM data table of the product A

| | | | | | | | | | | | | |
|-------|-----------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| M BOM | ID | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| | Parent ID | A | A | A | A | A | A | A | A | A | A | A |
| | Child ID | B | E | F | G | C | D | B | E | F | G | H |
| | Usage | 3 | 6 | 3 | 3 | 2 | 4 | 4 | 8 | 4 | 4 | 8 |
| | lvl | 1 | - 1 | - 1 | - 1 | 1 | 2 | 3 | - 1 | - 1 | - 1 | - 1 |
| | ID | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | | |
| SBOM | Parent ID | A | A | A | A | A | A | A | A | A | | |
| | Child ID | I | H | D | B | E | F | G | H | I | | |
| | Usage | 4 | 4 | 1 | 1 | 2 | 1 | 1 | 2 | 1 | | |
| | lvl | - 1 | - 1 | 1 | 2 | - 1 | - 1 | - 1 | - 1 | - 1 | | |

Child ID 分别表示父件 ID 和子件 ID, 与产品表和零部件表相关联; Usage 表示父件跟子件的用量关系; lvl 表示子件在父件的结构树中所处的层次, 如不可再分解, 即处在底层, 可以用一个特殊的符号, 如“- 1”标记出来.

多层 BOM 结构(MBOM) 采用“单父-多子”的数据结构. 它详细地记录了产品的结构信息, 即便是同样的零部件结构, 只要存在于不同的产品中, 也要再记录一次. 单层 BOM 结构(SBOM) 采用“单父-单子”的数据结构, 只是记录了各父件和子件之间的对应关系. 如表 1 所示, SBOM 只要 11 条记录就可以清楚地定义 MBOM 所描述的产品 A 结构. 如果其他产品中也含有 B, C 或 D 部件, 则不需要再重新定义. 所以, SBOM 的数据冗余度是最小的. 实际应用中, 产品所涉及的种类繁多、数量巨大, 为了适应不同条件下的需要, 常采用多种 BOM 并行的方法.

在 BOM 设计过程中需着重考虑两个要素. 其一是合理的 BOM 结构模型. 即在数据库中设计合理的方式存储 BOM 数据, 保证 BOM 数据的完整性、一致性、可靠性和无冗余; 另一个是高效便捷的算法. 它完成对产品结构的遍历和分解, 满足 MRP, ERP 运算及各种对产品结构的查询、汇总等.

2 转化步骤和算法

基于存储过程的 SBOM 到 MBOM 的转化算法, 采用临时表(temp) 来存放 BOM 中零部件的信息, 计算零部件的层次码和用量关系, 如表 2 所示. 创建的临时表结构, 其算法有如下 6 个步骤.

(1) 从 SBOM 表中取得当前产品信息(即根物料信息), 置其层次码 lvl= 0(代表第 0 层), 用量 usage= 1, 将这些信息插入到 temp 表中. (2) 判断 lvl ≥ 0(用于判断 temp 表中是否还存在记录, lvl 小于零表明不存在), 是则转到步骤(3); 否则, 退出. (3) 根据 lvl 值在 temp 表中查询记录, 存在则转到步骤(4); 否则, 返回当前层次的上一层次(即置 lvl= lvl- 1)并转到步骤(2). (4) 取得 temp 表中符合条件的第一条记录, 插入到 MBOM 表, 同时删除该记录. (5) 在 SBOM 表中查询上述被删除的记录是否有子物料, 是则转到步骤(6); 否则, 转到步骤(2). (6) 从 SBOM 表中取得子物料信息, 并计算层次码(即置 lvl= lvl+ 1)以及用量关系, 插入到 temp 表, 转到步骤(2).

表 2 临时表
Tab. 2 Temporary table

| 字段 | 类型 | 字段说明 |
|--------|--------------|------|
| lvl | int | 层次码 |
| itemid | varchar(50) | 物料代码 |
| usage | float | 物料用量 |

基于存储过程的 SBOM 到 MBOM 的转化算法的部分代码如下:

```
PROCEDURE [ dbo].[ SbomToM bom]
@itemid varchar( 50) -- 产品编码
AS
create table [ temp] ( ...)-- 创建临时表用于确定物料的层次码以及用量关系
create table [ mBom] ( ...)-- 创建产品的多层 BOM 数据表(MBOM)
declare @ lvl int, @ curr varchar( 50), @ usage float, @ parentUsage float, @ lowest int -- @
curr: 当前参与计算的物料; @ parentU sage: 物料用量计算因子;
set @ lowest= - 1 -- 用- 1 标识不可再分物料
select top 1 @ lvl= 0, @ curr= @ itemid , @ usage= 1 from sBom
insert into [ temp] ( lvl, itemid, usage) values ( @ lvl, @ curr, @ usage)
while( @ lvl> = 0)
begin
    if exists ( select * from [ temp] where lvl= @ lvl)
    begin
        select top 1 @ curr= itemid, @ usage= usage from [ temp] where lvl= @ lvl
        if exists ( select * from [ sBom] where Parentid= @ curr and Parentid <> Childid) --
            判断物料是否可再分
        begin
            insert into [ mBom]
```

```

        :
    end
else
    :
    insert into [ temp]
    select @lvl+ 1, Childid, usage* @ parentU sage
    from sBom where Parentid= @ curr and Parentid< > Childid
    if( @@ row count> 0)
        set @lvl= @ lvl+ 1
    end
else
    set @ lvl= @ lvl- 1
end
end

```

3 结 束 语

BOM 作为 ERP 系统的数据模型的核心, 其构造方法直接影响到系统启动的快慢、数据的准确性和运行的效率. 本文提出的算法为 BOM 结构在 ERP 系统中的应用提供了一个有效的途径. 由于该算法是基于存储过程的, 相对于一般的算法具有更好的计算性能.

参考文献:

[1] 陈 庄, 毛华扬. ERP 原理与应用教程[M]. 2 版. 北京: 电子工业出版社, 2006: 33-34.

[2] GOU Ling-yi, WEI Sheng-min. Database designing and optimizing for dynamic product BOM based on relational database [J]. Modular Machine Tool & Automatic Manufacturing Technique, 1999(5): 6-9.

[3] 严蔚敏, 吴伟民. 数据结构[M]. 2 版. 北京: 清华大学出版社, 1997: 118-120.

[4] JI Guo-li, GONG Da-xin, TSUI F. Analysis and implementation of the BOM of a tree type structure in MRP II [J]. Journal of Materials Processing Technology, 2003, 139(1/3): 535-538.

[5] WU Mu-huacheng, HSU Yang-kang. Design of BOM configuration for reducing spare parts logistic costs[J]. Expert Systems with Applications, 2008, 34(4): 2417-2423.

[6] 陈维斌, 喻小光. 一种 XML 数据到结构化数据的转换方法[J]. 华侨大学学报: 自然科学版, 2003, 24(2): 201-207

An Algorithm Implementation of Transformation from
Single BOM to Multiple BOM

CHEN Zhifeng, CHEN Yonghong

(College of Computer Science and Technology, Huaqiao University, Quanzhou 362021, China)

Abstract: Two main bill of material (BOM) structures which are single BOM and multiple BOM are studied. The advantages and disadvantages of these two structures are analyzed. An algorithm which is based on store procedure and transforms single BOM to multiple BOM is presented. At the same time, part codes of the algorithm is given. The algorithm based on store procedure has higher computational performace that of the general algorithm.

Keywords: bill of material; enterprise resource planning; database; store procedure

(责任编辑: 鲁 斌 英文审校: 吴逢铁)