

多级编码树编码的整理方法

A Way to Reform the Coding of Multilevel Coding

夏明伟 (中南大学现代教育技术中心 湖南长沙 410075)

摘要:在实际应用中经常会有些类似财务科目式的把结点的上下层关系置入编码中的数据,它们以多级编码的形式存储在库表中,但有时由于各种原因需要对原编码进行按显示顺序有规律地重新编码,但又需要保持原先的上下层关系。本文通过对 Oracle 的 SQL 语句的灵活运用,提出用一句 Oracle 的 SQL 语句完成多级编码的重新编码,此法在多个系统的初始化过程中得到实际应用,简便实用大大提高了初始化的速度与可靠性,也为其他多级编码式的树型关系数据进行编码重组提供一种新颖的方法。

关键词:树 查询语句 多级编码 重组

1 问题的提出

在实际应用中经常会碰到以多级编码形式组织的数据,但经过一段时间的增删改或剪裁后编码会顺序变得毫无连续性,在系统移植或初始化时可能需要对多级编码进行重新按新的规律重新排序并编码。此时我们一般会想到用程序实现多级编码的重编,那么有没有更好的办法实现多级编码的重编问题呢?并且最好采用 Oracle 的 SQL 语句来完成多级编码的重编。

带着这个问题笔者通过对多级编码的重编原理进行分析,最终找到一个实用并切实可行的用一句 Oracle 的 SQL 语句实现多级编码的实现方法,在此把它介绍给大家。

2 问题的分析及解决方法

首先假定我们有表 1 所示的多级编码数据表,其中 id 为多级编码代码,其中开头的 1-2 位代表一级结点的编码,3-4 位代表二级结点的编码,5-6 位代表三级结点的编码,7-8 位代表四级结点的编码,依次类推,每一级占两个字节, name 为结点名称, seq 兄弟排行(后面详解)。

假定我们要把表 1 所示的多级编码转换为保持原来的上下级关系不变但编码重组后的表 2 的多级编码,我们应该如何考虑呢?

首先我们发现表 1 的内容其实就是一棵树,每一行是树中的一个结点。如果要重组编码,原则是每一

层上的亲兄弟要分别重新编码,新编码的顺序可以按老编码的顺序也可以按结点名称排序,甚至可以按编码表中的其他字段(如果还有其他可以排序的字段)进行重新排序。某一级的编码的递增步长可以考虑按结点的不同级别设置不同的值。

表 1 多级编码表结构与内容

id	name	seq
01	董事长	
0119	总经理	
011901	李副总经理	
01190105	软件部经理	
01190107	质保部经理	
0119010701	配置管理员	
011906	柳副总经理	
011907	吴副总经理	
011911	刘副总经理	
01191102	采购部经理	
01191104	行政部经理	
0119110401	门卫	
02	法律顾问	
03	物业公司	
0301	车库管理部	
030101	东门门卫	

表 2 多级编码表结构与内容

id	name	seq
01	董事长	
0101	总经理	
010101	李副总经理	
01010101	软件部经理	
01010102	质保部经理	
0101010201	配置管理员	
010102	柳副总经理	
010103	吴副总经理	
010104	刘副总经理	
01010401	采购部经理	
01010402	行政部经理	
0101040201	门卫	
02	法律顾问	
03	物业公司	
0301	车库管理部	
030101	东门门卫	

根据以上原则如果要用一句 SQL 语句来完成编码重组必须同时解决以下几个问题: 正确判断结点属于哪一级; 每一级的编码递增步长; 本结点的新编号; 上层结点的新编号。

下面我们一个问题一个问题加以分析, 首先结点属于哪一级可以通过原编码的长度进行换算, 用 Oracle 的表达式 $\text{length}(\text{id})/2$ 即可。每一级的编码递增步长有编码者自行决定, 如果一级至五级的编码递增步长分别为 5、4、3、2、1 那么, 我们可以通过 Oracle 的表达式 $\text{decode}(\text{length}(\text{id})/2, 1, 5, 2, 4, 3, 3, 4, 2, 1)$ 来动态绑定每一级的结点。对本结点的新编号我们假定在维护数据时同时记录了一个按要求排序字排序的兄弟排行 seq, 那么我们就可以通过 Oracle 的表达式 $\text{seq} * \text{decode}(\text{length}(\text{id})/2, 1, 5, 2, 4, 3, 3, 4, 2, 1)$ 来动态形成本结点的新编号; 对于上层结点的新编号如果我们能够得到它那么它的新编号的形成也与本结点的新编号形成一样。

那么如何能够得到上层结点的兄弟排行呢? 由于在实际应用中树的层数总是有限的, 下面我们考虑最大层数为 5 层的树的上层结点的获取方法, 最大层数为 n 层的树的上层结点的获取方法可照此方法类推。

对 5 层树我们考虑形成带条件的五张表的笛卡儿积, 如果表 1 的表名为 xmw_tree_bm, 那么 Oracle 语法:

```

from xmw_tree_bm a, xmw_tree_bm b, xmw_
tree_bm c, xmw_tree_bm d, xmw_tree_bm e
where substr(a. ID, 1, length(a. id) - 2) = b. ID
(+) and substr(b. ID, 1, length(b. id) - 2) = c. ID
(+) and
substr(c. ID, 1, length(c. id) - 2) = d. ID
(+) and substr(d. ID, 1, length(d. id) - 2) = e. ID
(+)
    
```

将形成树结构的五层父子级联关系, 其中 e 表结点处在最上层, 依次为 d 表结点、c 表结点、b 表结点和 a 表结点, 注意级联要使用外链接, 否则只能得到五级编码的记录。这样的话我们就能轻易获取 a 表中各结点的上层结点的兄弟排行了, 因为我们的例子考虑最大为 5 层的树, 所以表 2 中某一行的 id 值最大可有五段每段的形成如下:

第一段, 若 e 表结点存在则 e 表结点的兄弟排行为 e. seq, 否则为空。

第二段, 若 d 表结点存在则 d 表结点的兄弟排行为 d. seq, 否则为空。

第三段, 若 c 表结点存在则 c 表结点的兄弟排行为 c. seq, 否则为空。

第四段, 若 b 表结点存在则 b 表结点的兄弟排行为 b. seq, 否则为空。

第五段, 无条件取 a 表结点的排行 a. seq。

用 Oracle 的表达式表示分别为:

```

第一段, decode(e. id, null, '', trim(to_char((e.
seq - 1) * decode(length(e. id)/2, 1, 5, 2, 4, 3, 3, 4,
2, 1) + 1, '09'))))
    
```

```

第二段, decode(d. id, null, '', trim(to_char((d.
seq - 1) * decode(length(d. id)/2, 1, 5, 2, 4, 3, 3, 4,
2, 1) + 1, '09'))))
    
```

```

第三段, decode(c. id, null, '', trim(to_char((c.
seq - 1) * decode(length(c. id)/2, 1, 5, 2, 4, 3, 3, 4,
2, 1) + 1, '09'))))
    
```

```

第四段, decode(b. id, null, '', trim(to_char((b.
seq - 1) * decode(length(b. id)/2, 1, 5, 2, 4, 3, 3, 4,
2, 1) + 1, '09'))))
    
```

```

第五段, trim(to_char((a. seq - 1) * decode
(length(a. id)/2, 1, 5, 2, 4, 3, 3, 4, 2, 1) + 1, '09'))
    
```

通过以上逐步分析,我们最后形成如下语句

(一),至此问题也就基本解决了。

语句(一)完成多级编码重编的 SQL 语句

```
select decode(e.id,null,'',trim(to_char((e.seq-1)
*decode(length(e.id)/2,1,5,2,4,3,3,4,2,1)+1,
'09')))||
decode(d.id,null,'',trim(to_char((d.seq-1)*de-
code(length(d.id)/2,1,5,2,4,3,3,4,2,1)+1,'09
')))||
decode(c.id,null,'',trim(to_char((c.seq-1)*de-
code(length(c.id)/2,1,5,2,4,3,3,4,2,1)+1,'09
')))||
decode(b.id,null,'',trim(to_char((b.seq-1)*de-
code(length(b.id)/2,1,5,2,4,3,3,4,2,1)+1,'09
')))||
trim(to_char((a.seq-1)*decode(length(a.id)/
2,1,5,2,4,3,3,4,2,1)+1,'09')) id,a.name
from xmw_tree_bm a,xmw_tree_bm b,xmw_tree_
bm c,xmw_tree_bm d,xmw_tree_bm e
where substr(a.ID,1,length(a.id)-2)=b.ID(+ )
and substr(b.ID,1,length(b.id)-2)=c.ID(+ )
and
substr(c.ID,1,length(c.id)-2)=d.ID(+ ) and
substr(d.ID,1,length(d.id)-2)=e.ID(+ );
```

最后需要补充的是如果我们预先在维护表 1 的数据时没有同时考虑记录一个按要求排序字排序的兄弟排行 seq 值,那么现在只要在表 1 中增设一个兄弟排行字段 seq,然后根据排序字对同父结点的兄弟进行排序,把本结点的序号登记到 seq 字段中即可。获取 seq 值的 SQL 语句为如下语句(二)或语句(三)。

语句(二)获取结点按原 id 排序字排序的兄弟排行 seq 值的 SQL 语句

```
select a.id,count(*)
from xmw_tree_bm a,xmw_tree_bm b where
decode(length(a.id),2,'$',substr(a.ID,
1,length(a.ID)-2))=
decode(length(b.id),2,'$',substr(b.ID,
1,length(b.ID)-2)) and b.id <= a.id group by a.
id;
```

语句(三)获取结点按原 name 排序字排序的兄弟排

行 seq 值的 SQL 语句

```
select a.name,count(*)
from xmw_tree_bm a,xmw_tree_bm b where
decode(length(a.id),2,'$',substr(a.ID,
1,length(a.ID)-2))=
decode(length(b.id),2,'$',substr(b.ID,
1,length(b.ID)-2)) and b.name <= a.name
group by a.name;
```

那么一次初始化表 1 中的 seq 的值可分别用如下 up-
date 语句(四)或语句(五)实现。

语句(四)一次初始化兄弟排行字段 seq 的语句(按
原 id 排序字排序)

```
update xmw_tree_bm a set seq =
(select count(*) from xmw_tree_bm b where
decode(length(a.id),2,'$',substr(a.ID,1,
length(a.ID)-2))=
decode(length(b.id),2,'$',substr(b.ID,1,
length(b.ID)-2)) and b.id <= a.id);
```

语句(五)一次初始化兄弟排行字段 seq 的语句(按
原 name 排序字排序),代码与语句(四)相似,略。

3 结论

本论文的创新点就是充分利用 ORACLE 的 SQL 语句的灵活性、内部函数的丰富性及强大的运行效率用一句 SQL 语句实现多级编码的重新按序编码,完全是本人从实际需要出发的一种奇想,相信对广大 SQL 爱好者及奋战在软件开发一线的人员对 Oracle 的 SQL 语句的作用有新的认识,希望本文的内容对大家有所帮助,谢谢。

参考文献

- 1 夏明伟、林云、胡铁,用 ORACLE 的 SQL 语句实现多
栏分页输出[J],计算机系统应用,2004,0(9)。
- 2 [美]Christopher Allen 著. Oracle PL/SQL 程序设
计基础教程 - 北京:机械工业出版社,2001. 5。
- 3 杨安祺、单振芳、杨竞澜,基于 C/C++ 与 ORACLE9i
的嵌入式 SQL 编程技术[J],微计算机信息,2005,
2。
- 4 萨师焯、王珊,数据库系统概论(第三版),高等教
育出版社,2000,2。