Oracle 单行转多列

作者：chenxc
原文链接：https://ld246.com/article/1538031846133
来源网站：链滴

在数据库表设计中，经常会出现一对多关系，常用的处理办法有建立中间表和在“一”的那张表中的个字段引用“多”的那张表，会以某符号分隔存储在表中。

但是在实际运用的过程中可能会用到分隔符引用的列需要转多行的需求，这时候我们一般会有两种处办法：

- 写函数方法
- 正则表达式

比较而言第二种更加方便快捷，本文详细介绍第二种方式。

regexp_substr函数

REGEXP_SUBSTR函数格式如下：

function REGEXP_SUBSTR(String, pattern, position, occurrence, modifier)

__srcstr　：需要进行正则处理的字符串

__pattern　：进行匹配的正则表达式

__position　：起始位置，从第几个字符开始正则表达式匹配（默认为1）

__occurrence：标识第几个匹配组，默认为1

__modifier　：模式（'i'不区分大小写进行检索；'c'区分大小写进行检索。默认为'c'。）

举个简单的例子，说明REGEXP_SUBSTR的用法

A：Students（学生）表

id	code	name	books
1	1	张三	1,2,3

B：Book（图书）表

code	name	price
1	Java从入门到精通	10
2	编译原理	10
2	颈椎康复指南	100

现需要统计出所有学生买图书的总价格

select user_id,user_name,

```
select stu.name, sum(k.price)
   (select name,
    REGEXP_SUBSTR(books, '[^,]+', 1, LEVEL) book
    --regexp_substr(str,reg,起始位置 第几次)
    from students
    CONNECT BY LEVEL <= LENGTH(books) - LENGTH(REGEXP_REPLACE(books, ',')) + 1
```

```
      and id = prior id
      and prior dbms_random.value is not null) stu
      --筛选掉空数据
      left join book k on k.code = stu.book
```

此次sql提取前50行执行时间为1点几秒，且数据量不大，下面为sql优化

```
select stu.name, sum(k.price)
    (select name,
      regexp_substr(books, '[^,]+', 1, level) book
      --regexp_substr(str,reg,起始位置 第几次)
      from students
      connect by level <= regexp_count(books, ',') + 1
      --regexp_count(teachers, ',') 统计字符串中,的数量
      and id = prior id
      and prior dbms_random.value is not null) stu
      --筛选掉空数据
      left join book k on k.code = stu.book
```

此次执行仅为0.0几秒，速度快了数十倍

CONNECT BY

● 它相当于是一个递归的自连接，不断地把每层的连接结果叠加到结果集中。两层之间的连接条件和归出口写在CONNECT BY中。在这里我们的数据并无父子关系，只是要让同一行数据重复出现，因我们的连接的条件只用到了表的主键id=PRIOR id, 此外再用LEVEL控制层数作为递归出口。但ORACL有个检查，如果你有前后连接条件(id=PRIOR id)，但是同一行数据再次出现，它就会报一个错：

● --ERROR:

● --ORA-01436: CONNECT BY loop in user data

● --为了欺骗它，这里用了一个PRIOR DBMS_RANDOM.VALUE, 因为DBMS_RANDOM.VALUE每调用都返回不同结果，所以它认为两行数据不一样，所以不报错了。