**润乾迁移HANA递归sql解决方案**

**By hujue**

1. 概述

递归语句是数据库中用于解决父子节点关系的组织结构数据，在oracle及informix中都是采用start with … connect by …的递归语句进行处理；

而在HANA 1.0.33中还支持的start with … connect by …，在1.0.7000版本中已经不被支持，而在润乾报表数据集以及存储过程中大量使用了该递归语句进行处理，这就需要自行进行处理。

1. 解决方案
   1. 解决方案说明

在该方案中，特别为使用TF\_RPT\_HN\_STAT\_BSHZQK进行递归查询而定制了hana函数FUNC\_TF\_RPT\_HN\_STAT\_BSHZQK，该函数只能查询TF\_RPT\_HN\_STAT\_BSHZQK数据表，输入参数为 issue，rpt\_id，和dept\_id，另外，该函数只能从上到下进行查询。

输入参数如下：

* **issue**：格式为’120130000’，’320131200’ ，’620131212’等，用于对数据表进行预先筛选；
* **rpt\_id**：格式为’HDY01’等，用于对数据表进行预先筛选；
* **dept\_id**：用于递归sql的起始dept\_id

函数调用后，返回一个table类型的数据集，在hana中，table类型数据集可以作为sql中from后的数据来源。

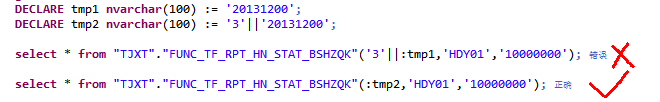
返回table类型字段如下：

* **dept\_id**：**nvarchar**(100)，单位编码，来源于BSHZQK数据表dept\_id字段
* **dept\_name**：**nvarchar**(100)，单位名称，来源于BSHZQK数据表dept\_id字段
* **px**：int，单位排序，来源于BSHZQK数据表px字段
* **CONNECT\_BY\_LEVEL**：int，单位级别，在函数中生成，用于标识单位的层级
* **CONNECT\_BY\_ISLEAF**：int，是否叶子节点，是=1，否=0，预留
* **CONNECT\_BY\_PATH**：**varchar**(5000)，dept\_id结合\_形成的path
* **CONNECT\_BY\_SORT**：**varchar**(5000)，px结合\_形成的sort，用于实现按目录树排序
  1. SQL调用示例

|  |  |
| --- | --- |
| HANA | ORACLE |
| **select** \* **from** FUNC\_TF\_RPT\_HN\_STAT\_BSHZQK  ('320131200','HDY01','10000000') | select \* from  (select \* from TF\_RPT\_HN\_STAT\_BSHZQK t  where t.issue='320131200' and t.rpt\_id='HDY01') a  start with a.dept\_id = '10000000'  connect by prior a.dept\_id=a.dept\_mng\_id  order siblings by a.px |
|  |  |

如表格中示例所示，'320131200','HDY01'这个2个参数是用于对BSHZQK数据表的预先筛选，'10000000'是用于确定start单位编码

需要注意的是，FUNC\_TF\_RPT\_HN\_STAT\_BSHZQK(issue\_,rpt\_id\_,dept\_id\_)中，issue\_和rpt\_id\_不能是表达式，只能是参数，如下



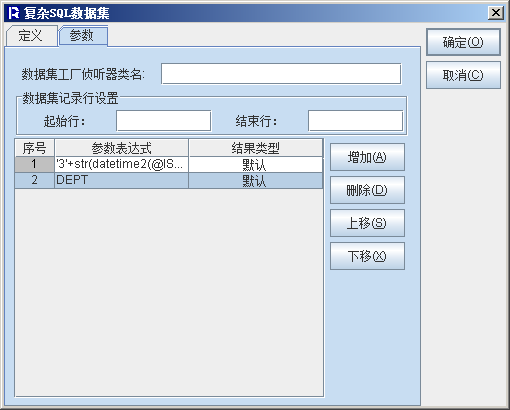
* 1. 润乾调用示例
* 润乾示例

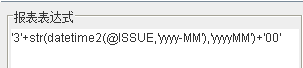
 <http://pan.baidu.com/s/1hqr8NvY>

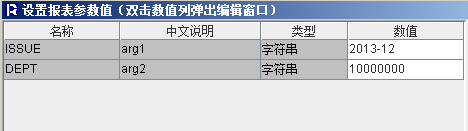
* 数据集SQL设置



* 参数设置







以上润乾调用方法有点特别，关键还是在hana该函数参数不能是表达式，只能对参数在润乾中预先处理。

1. 函数代码

|  |
| --- |
| **create** **function** FUNC\_TF\_RPT\_HN\_STAT\_BSHZQK(**IN** issue\_ **NVARCHAR**(100),**IN** rpt\_id\_ **NVARCHAR**(100),**IN** root\_id\_ **NVARCHAR**(100))  RETURNS **table**(dept\_id **nvarchar**(100),dept\_name **nvarchar**(100),px int,CONNECT\_BY\_LEVEL int,CONNECT\_BY\_ISLEAF int,CONNECT\_BY\_PATH **varchar**(5000),CONNECT\_BY\_SORT **varchar**(5000))  **LANGUAGE** SQLSCRIPT  **AS**  **begin**  **DECLARE** node\_level INT := 1;  **DECLARE** isleaf INT := 0;  **DECLARE** node\_count int := 1;  **declare** px\_length int ;  **declare** ss **nvarchar**(100):= ' issue=''' || :issue\_ || ''' and rpt\_id='''|| :rpt\_id\_ ||'''';  tmp1 = APPLY\_FILTER(TF\_RPT\_HN\_STAT\_BSHZQK,:ss) ;    **select** **length**(**max**(px)) **into** px\_length **from** :tmp1;    tmp2 = **select** dept\_id,dept\_name,  :node\_level **as** dept\_level,  px,  to\_varchar(dept\_id) || ',' **as** CONNECT\_BY\_PATH,  lpad('1',px\_length,'0') || '\_' **as** CONNECT\_BY\_SORT  **from** :tmp1  **where** dept\_id = :root\_id\_;    tmp3 = **select** dept\_id,dept\_name,  :node\_level **as** dept\_level,  px,  to\_varchar(dept\_id) || ',' **as** CONNECT\_BY\_PATH,  lpad('1',px\_length,'0') || '\_' **as** CONNECT\_BY\_SORT  **from** :tmp1  **where** dept\_id = :root\_id\_;    **while** node\_count > 0 **do**  node\_level := node\_level + 1;    **select** **count**(1) **into** node\_count **from** :tmp2;    tmp2 = **select** a.dept\_id,a.dept\_name,  :node\_level **as** dept\_level,  a.px,  b.CONNECT\_BY\_PATH || to\_varchar(a.dept\_id) || ',' **as** CONNECT\_BY\_PATH,  b.CONNECT\_BY\_SORT || lpad(a.px,px\_length,'0') || '\_' **as** CONNECT\_BY\_SORT  **from** :tmp1 a , :tmp2 b  **where** a.dept\_mng\_id = b.dept\_id ;    tmp3 = **select** dept\_id,dept\_name,dept\_level,px,CONNECT\_BY\_PATH,CONNECT\_BY\_SORT **from** :tmp3  **union** **all**  **select** dept\_id,dept\_name,dept\_level,px,CONNECT\_BY\_PATH,CONNECT\_BY\_SORT **from** :tmp2 ;      **end** **while**;  RETURN **select** dept\_id,dept\_name,px,dept\_level **as** CONNECT\_BY\_LEVEL,**null** **as** CONNECT\_BY\_ISLEAF,CONNECT\_BY\_PATH,CONNECT\_BY\_SORT **from** :tmp3 **order** **by** CONNECT\_BY\_SORT;  **end**; |