**Oracle分析函数**

**这里有两个帖子可以看看**

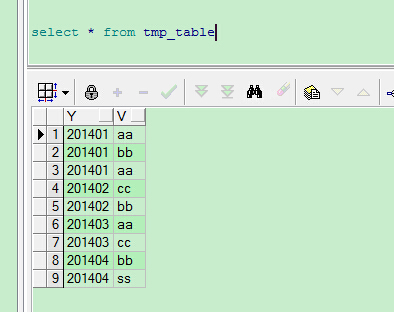
[**http://www.cnblogs.com/linjiqin/archive/2012/04/04/2431975.html**](http://www.cnblogs.com/linjiqin/archive/2012/04/04/2431975.html)

[**http://www.cnblogs.com/zfc2201/archive/2013/08/27/3285010.html**](http://www.cnblogs.com/zfc2201/archive/2013/08/27/3285010.html)

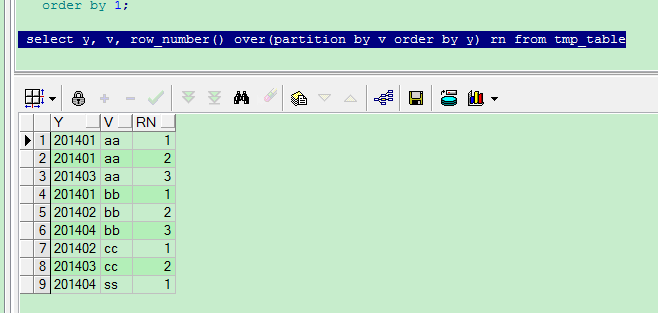
**1.row\_number() over()函数**

ROW\_NUMBER() OVER(partition by col1 order by col2) 表示根据col1分组，在分组内部根据col2排序，而此函数计算的值就表示每组内部排序后的顺序编号（组内是连续且唯一的）。

原表查询截图:



**Row\_number() over()实例：**



**2.sum() over()**

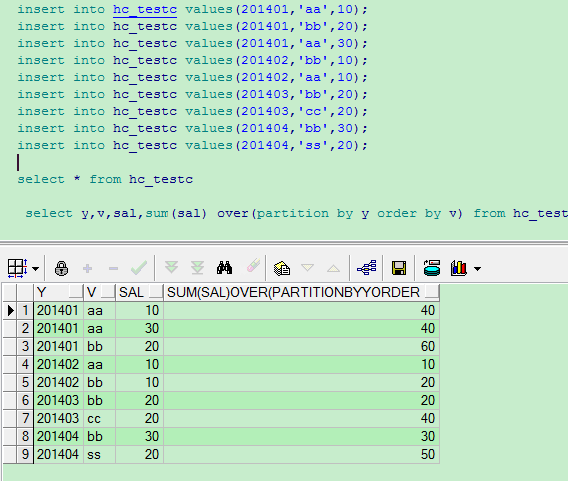
sum() OVER(partition by col1 order by col2) 表示根据col1分组，在分组内部根据col2排序，并且进行连续的追加。

**这里有个帖子有讲**

[**http://www.360doc.com/content/13/0105/10/11273733\_258308394.shtml**](http://www.360doc.com/content/13/0105/10/11273733_258308394.shtml)

**一下是我自己测试的实例**

先来看看over函数里面有分组的情况



**类似的其他函数还有**

sum(1) over() as 总记录数,

sum(1) over(partition by id) as 分组记录数,

sum(score) over() as 总计 ,

sum(score) over(partition by id) as 分组求和,

sum(score) over(order by id) as 分组连续求和,

sum(score) over(partition by id,area) as 分组ID和area求和,

sum(score) over(partition by id order by area) as 分组ID并连续按area求和,

max(score) over() as 最大值,

max(score) over(partition by id) as 分组最大值,

max(score) over(order by id) as 分组连续最大值,

max(score) over(partition by id,area) as 分组ID和area求最大值,

max(score) over(partition by id order by area) as 分组ID并连续按area求最大值,

avg(score) over() as 所有平均,

avg(score) over(partition by id) as 分组平均,

avg(score) over(order by id) as 分组连续平均,

avg(score) over(partition by id,area) as 分组ID和area平均,

avg(score) over(partition by id order by area) as 分组ID并连续按area平均,

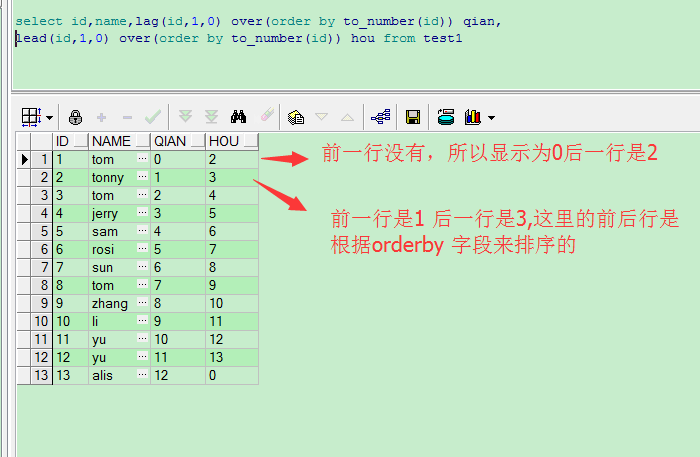
RATIO\_TO\_REPORT(score) over() as "占所有%",

RATIO\_TO\_REPORT(score) over(partition by id) as "占分组%",

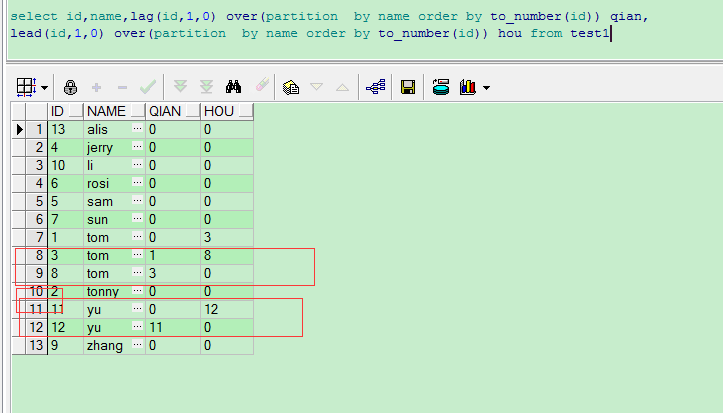
**Lag和lead函数**

Lag和Lead分析函数可以在同一次查询中取出同一字段的前N行的数据(Lag)和后N行的数据(Lead)作为独立的列。

实例截图

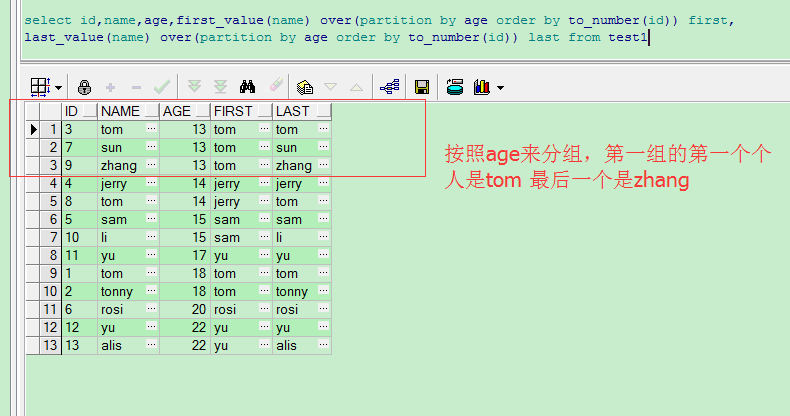


**如果再over函数内用到了分组的话(**partition by name**)**

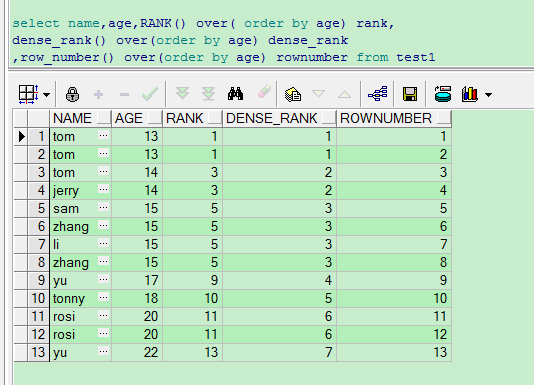
**那么这里的前后行都是在分组内部取数据的，**

**First\_value和last\_value函数**

**返回指定列首行值和末行值**

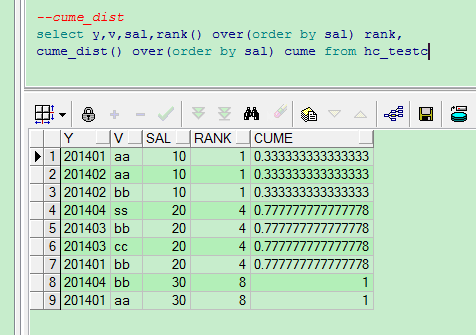


**Rank() 和dense\_rank 以及row\_number的区别**



**Cume\_dist() over()**

**这个函数属于数据分析函数，官方解释是:计算一行在组中的相对位置，cume\_dist总是返回大于0、小于1的数，该数表示改行在n行中的位置.看例子:**



**Cume\_dist。默认是用的rank()排序的规则，这一点非常重要，不然是看不懂cume这一行的。虽然遵循rank()排序规则，但是还有有差别的。先看rank一列的前3行的值是1.但是cume这一栏的是确实1/3.其实是因为rank这一栏如果有相同的值，那么就按最后一个值的相对位置来计算。也就是说有3个1，那么这3个1都是按照最后一个1，也就是按照3来计算的。起结果是3/9=1/3=0.3333333.**

**第二组数字是4,但是它有4个。那么最后一个的位子应该是7，所以计算结果是7/9.**

**第三组的数字是8，有两个。那么计算结果是9/9=1**