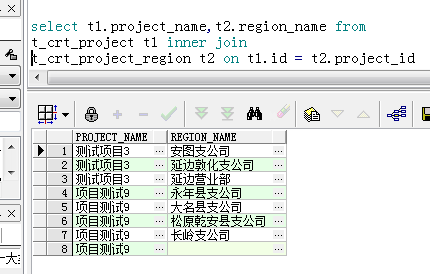
1. oracle查询多行数据合并成一行数据：

没合并之前的sql:

select t1.project\_name,t2.region\_name from t\_crt\_project t1 inner join t\_crt\_project\_region t2 on t1.id = t2.project\_id

效果：



用于合并的sql:

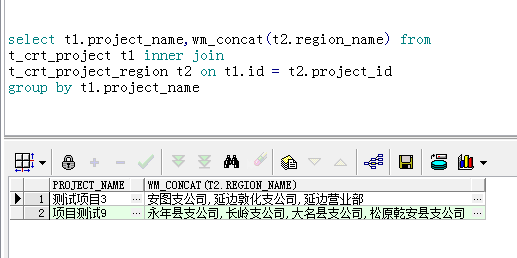
select t1.project\_name,wm\_concat(t2.region\_name) from t\_crt\_project t1 inner join t\_crt\_project\_region t2 on t1.id = t2.project\_id group by t1.project\_name

SELECT ID, listagg(NAME, ',') within group(ORDER BY ID) AS NAMES

FROM DEMO T

GROUP BY ID

效果：



1. oracle 数据库 使用（start with connect by prior）实现递归查询

原始数据：

select

t.id 分公司机构编码,

t.org\_name 分公司机构名,

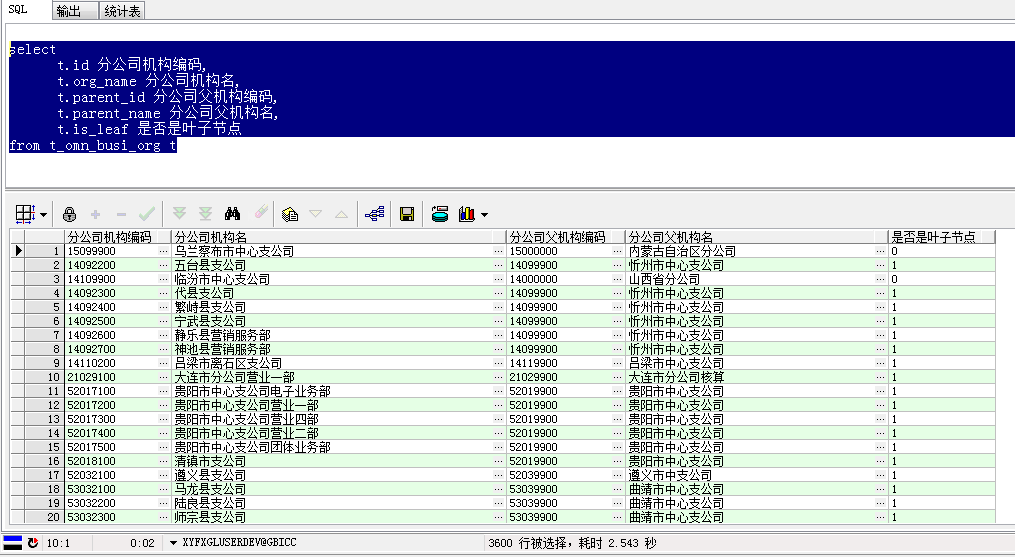
t.parent\_id 分公司父机构编码,

t.parent\_name 分公司父机构名,

t.is\_leaf 是否是叶子节点

from t\_omn\_busi\_org t

效果：



prior放的左右位置决定了检索是**自底向上还是自顶向下**

以下的sql选择了自底向上：

select

t.id 分公司机构编码,

t.org\_name 分公司机构名,

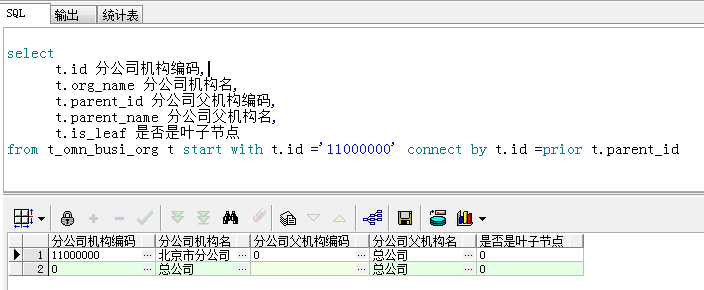
t.parent\_id 分公司父机构编码,

t.parent\_name 分公司父机构名,

t.is\_leaf 是否是叶子节点

from t\_omn\_busi\_org t start with t.id ='11000000' connect by t.id =prior t.parent\_id

效果：



以下的sql选择了**自顶向下**：

select

t.id 分公司机构编码,

t.org\_name 分公司机构名,

t.parent\_id 分公司父机构编码,

t.parent\_name 分公司父机构名,

t.is\_leaf 是否是叶子节点

from t\_omn\_busi\_org t start with t.id ='11000000' connect by prior t.id = t.parent\_id

效果：（还剩28条）

