

将一个表的查询结果作为另一查询的字段（动态查询列）

原创

展开



zhangchu_63

最后发布于2010-07-12 10:23:00

阅读数 17803

收藏

接着上面IP地址字段查询问题，那就是统计结果的展示格式。朋友要的格式是：

	城市1	城市2	城市3	城市4
2010-06	0	1	0	0
2010-08	0	1	0	2
2010-07	0	0	1	0
2010-05	1	0	0	0

时间和记录都是从SINO_USER查询出来的，而列名是从SINO_IP表中查询出来。本来用枚举法，直接把所有城市都用CASE WHEN进行判断一遍就OK，但是，SINO_IP表中城市个数是不定的，就是说今天查询出来时4个城市，明天也许就是5个城市。这就需要我们的统计能支持列的动态添加。我第一想到的是视图（VIEW），但是这个视图怎么写，才能保证动态生成列呢，这就要使用动态SQL，而动态SQL在哪里生成能，嘿嘿，说了半天总算到正题了，没错，就是存储过程。可以使用存储过程动态生成VIEW，然后根据VIEW来查询。

1、数据基础

表结构我就不写了，和上篇博文一样的。

2、问题分析

```
select s.dd
,sum(city1) 城市1
,sum(city2) 城市2
,sum(city3) 城市3
,sum(city4) 城市4
,sum(city5) 城市5
from (select substr (to_char(t.update_time,'yyyy-mm-dd'),1,7) dd
,case when t1.city = '城市1' then 1 else 0 end city1
,case when t1.city = '城市2' then 1 else 0 end city2
,case when t1.city = '城市3' then 1 else 0 end city3
,case when t1.city = '城市4' then 1 else 0 end city4
,case when t1.city = '城市5' then 1 else 0 end city5
from sino_user t , sino_ip t1
where f_ip2number(t.ip) between f_ip2number(t1.ip_begin) and f_ip2number(t1.ip_end)) s group by s.dd
```

我们可以发现，SUM部分和CASE WHEN 部分都可以通过循环来添加，因此，我们可以把从城市对应IP段表中查询出来的数据存放在游标中，通过循环来生成统计SQL。

LET'S DO IT。

3、编写过程

```

create or replace procedure p_count is
v_sql dbms_sql.varchar2s;
l_cursor integer default dbms_sql.open_cursor;
l_rows number default 0;
i number default 2;
begin
v_sql(1) := 'create or replace view v_count as select s.dd';

for c_col in (select id,city from sino_ip order by id) loop
v_sql(i) := 'sum(city' ||c_col.id||') '||c_col.city;
i := i+1;
end loop;

v_sql(i) := 'from (select substr (to_char(t.update_time,"yyyy-mm-dd"),1,7) dd';

for c_col2 in (select id,city from sino_ip order by id) loop
v_sql(i+1) := ',case when t1.city = "' ||c_col2.city||'" then 1 else 0 end city' || c_col2.id;
i:= i + 1;
end loop;

v_sql(i+1) := 'from sino_user t , sino_ip t1 ';
v_sql(i+2) := 'where f_ip2number(t.ip) between f_ip2number(t1.ip_begin) and f_ip2number(t1.ip_end)) s group by s.dd';

dbms_sql.parse(c => l_cursor,statement => v_sql,lb => v_sql.first,ub => v_sql.last,lfflg => TRUE,language_flag => dbms_sql.native
);
l_rows := dbms_sql.execute(l_cursor);
dbms_sql.close_cursor( l_cursor );
dbms_output.put_line(l_rows);
exception
    when others then
        dbms_output.put_line(sqlerrm);
end p_count;

```

注：过程中用到的f_ip2number函数可以到另一个博文“如何使用SQL查询IP地址所属IP段”里面去看，我就不再赘述。

大家可能注意到，我的在存储过程中用来存放动态SQL不是用VARCHAR2，用的是DBMS_SQL.VARCHAR2S，这是因为VARCHAR2只能存放32K的字符，而我们的SQL是很长，将来会更长，因此使用DBMS_SQL.VARCHAR2分行来存储SQL。

同时，SQL的执行也不能使用EXECUTE IMMEDIATE，因为它也只能执行32K的SQL，我们使用DBMS_SQL.PARSE，通过游标来执行。

OK，执行。

报错了.....

没有足够的权限。

呵呵，我们在PL/SQL中创建视图需要创建视图的权限。

OK，我们给账号个CREATE ANY VIEW的权限。

执行

没报错。

我们来查询VIEW

```
select * from v_count t
```

结果出来了，和我上面写的要求一样。

那我们来验证它能不能动态生成列。

我们在SINO_IP插入城市5，一个新的IP段，然后在SINO_USER表中插入一条IP数据城市5的记录。再来查询：

	城市1	城市2	城市3	城市4	城市5
2010-06	0	1	0	0	0
2010-08	0	1	0	2	0
2010-07	0	0	1	0	0
2010-05	1	0	0	0	1

OK，成功了。哈哈

亏了今天胡老大到我们那视察，不用上班，我才有空把最近做的东西记下来。