set verify off   
 column file\_name format a50 word\_wrapped   
 column smallest format 999,990 heading "Smallest|Size|Poss."   
 column currsize format 999,990 heading "Current|Size"   
 column savings  format 999,990 heading "Poss.|Savings" 

break on report   
 compute sum of savings on report

SQL> break on report  
 SQL>  compute sum of BYTES/1024/1024 on report  
 SQL>  select bytes/1024/1024 from dba\_data\_files;  
  
 BYTES/1024/1024  
 ---------------  
         6.25  
          105  
          620  
          730  
      345.625  
 ---------------  
        1806.875

SQL>

=======================================

[【SQL\*Plus】使用BREAK和COMPUTE在SQL\*Plus中得到分组统计结果](http://blog.itpub.net/519536/viewspace-618344/) *2009-11-07 17:30:38*

分类： Linux

秉承KISS原则，一切简单实用的工具都是值得深入挖掘的。SQL\*Plus就是其中之一。我通过这个小文儿给大家介绍一下在SQL\*Plus中使用break和compute命令实现类似group分组函数的统计效果，这里只是抛个“砖”，期待大家“玉”的到来。  
  
1.创建样例表t，并简单初始化几条数据，以便后续的演示操作。  
sec@ora10g> create table t (x number, y varchar2(20), z number);  
sec@ora10g> insert into t values (1, 'a', 100);  
sec@ora10g> insert into t values (2, 'b', 200);  
sec@ora10g> insert into t values (2, 'c', 300);  
sec@ora10g> insert into t values (3, 'd', 400);  
sec@ora10g> insert into t values (3, 'e', 500);  
sec@ora10g> insert into t values (3, 'f', 600);  
sec@ora10g> insert into t values (4, 'g', 700);  
sec@ora10g> insert into t values (4, 'h', 800);  
sec@ora10g> insert into t values (4, 'i', 1000);  
sec@ora10g> insert into t values (4, 'j', 1100);  
  
2.最普通的查询结果，这个最简单的select语句听说地球人都知道了。  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
         2 b                           200  
         2 c                           300  
         3 d                           400  
         3 e                           500  
         3 f                           600  
         4 g                           700  
         4 h                           800  
         4 i                          1000  
         4 j                          1100  
  
10 rows selected.  
  
3.引入BREAK命令，请看select语句的输出效果，不说自明，BREAK作用是将x列重复的值进行隐藏，方便查看。似乎使用过BREAK命令的地球人不是很多。  
sec@ora10g> break on x  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
         2 b                           200  
           c                           300  
         3 d                           400  
           e                           500  
           f                           600  
         4 g                           700  
           h                           800  
           i                          1000  
           j                          1100  
  
10 rows selected.  
  
4.在上面的基础上再引入COMPUTE命令。“计算”效果明显，实现了对X列分组求和的目的。  
sec@ora10g> compute sum of z on x;  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    100  
         2 b                           200  
           c                           300  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    500  
         3 d                           400  
           e                           500  
           f                           600  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   1500  
         4 g                           700  
           h                           800  
           i                          1000  
           j                          1100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   3600  
  
10 rows selected.  
  
5.到此，分组求和的功能实现了，但是显示效果有点压抑，我们再使用“skip 1”命令在每组结果后面添加一空行（如果想加入2个空行可以使用“skip 2”，以此类推……**）。  
sec@ora10g> break on x skip 1  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    100  
  
         2 b                           200  
           c                           300  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    500  
  
         3 d                           400  
           e                           500  
           f                           600  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   1500  
  
         4 g                           700  
           h                           800  
           i                          1000  
           j                          1100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   3600  
  
  
10 rows selected.  
  
6.如果行数众多，在显示结果的尾部将无法看到每一列的列名信息，此时“skip page 1”命令就派上用场了，请看下面的效果。  
sec@ora10g> break on x skip page 1  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    100  
  
         X Y                             Z  
---------- -------------------- ----------  
         2 b                           200  
           c                           300  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                    500  
  
         X Y                             Z  
---------- -------------------- ----------  
         3 d                           400  
           e                           500  
           f                           600  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   1500  
  
         X Y                             Z  
---------- -------------------- ----------  
         4 g                           700  
           h                           800  
           i                          1000  
           j                          1100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
sum                                   3600  
  
10 rows selected.  
  
7.上面的实验结果已经满足我们仅使用SQL\*Plus命令来完成类似group by分组求和的目的。一颗追求深入的心在不停的追问：“SQL\*Plus中还有哪些类似sum的“分组函数”可用呢？”请看我继续分解。  
sec@ora10g> compute sum minimum maximum avg std variance count number of z on x;  
sec@ora10g> select \* From t;  
  
         X Y                             Z  
---------- -------------------- ----------  
         1 a                           100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
avg                                    100  
count                                    1  
minimum                                100  
maximum                                100  
number                                   1  
sum                                    100  
std                                      0  
variance                                 0  
  
         X Y                             Z  
---------- -------------------- ----------  
         2 b                           200  
           c                           300  
\*\*\*\*\*\*\*\*\*\*                      ----------  
avg                                    250  
count                                    2  
minimum                                200  
maximum                                300  
number                                   2  
sum                                    500  
std                             70.7106781  
variance                              5000  
  
         X Y                             Z  
---------- -------------------- ----------  
         3 d                           400  
           e                           500  
           f                           600  
\*\*\*\*\*\*\*\*\*\*                      ----------  
avg                                    500  
count                                    3  
minimum                                400  
maximum                                600  
number                                   3  
sum                                   1500  
std                                    100  
variance                             10000  
  
         X Y                             Z  
---------- -------------------- ----------  
         4 g                           700  
           h                           800  
           i                          1000  
           j                          1100  
\*\*\*\*\*\*\*\*\*\*                      ----------  
avg                                    900  
count                                    4  
minimum                                700  
maximum                               1100  
number                                   4  
sum                                   3600  
std                             182.574186  
variance                        33333.3333  
  
10 rows selected.  
  
上面的结果就是在SQL\*Plus中能够得到统计功能，这里再逐条分解一下：  
avg      :Average of the values in the column.（求平均值）  
 count    :Number of non-null values in the column.（统计这组中一共有多少条非空记录）  
 minimum  :Minimum value in the column.（选出这组值中最小的一个）  
 maximum  :Maximum value in the column.（选出这组值中最大的一个）  
 number   :Number of rows in the column.（统计这组中一共有多少条记录，包含空记录）  
 sum      :Sum of the values in the column.（求和）  
 std      :Standard deviation of the values in the column.（求标准差）  
 variance :Variance of the values in the column.（求方差）  
  
8.小结  
基于上面演示的BREAK和COMPUTE命令所实现的功能，SQL\*Plus的强大之处可见一斑。  
其实，单纯使用SQL\*Plus就可以生成一份非常精美的报表，有兴趣的朋友可以继续深入挖掘一下，其乐无穷也。**

“抛砖”到此结束，现在是“引玉”时间……