**group by分组函数之rollup与cube用法**

原创 [Oracle](http://blog.itpub.net/oracle/) 作者：[pwz1688](http://blog.itpub.net/21251711/) 时间：2014-01-20 17:01:04  11432  0

ROLLUP，是GROUP BY子句的一种扩展，可以为每个分组返回小计记录以及为所有分组返回总计记录。

CUBE，也是GROUP BY子句的一种扩展，可以返回每一个列组合的小计记录，同时在末尾加上总计记录。  
示例：  
一、初始化表及数据代码如下：

**点击(此处)折叠或打开**

* SQL> create table group\_test (group\_id int, job varchar2(10), name varchar2(10), salary int);
* 表已创建。
* SQL>
* SQL> insert into group\_test values (10,'Coding', 'Bruce',1000);
* 已创建 1 行。
* SQL> insert into group\_test values (10,'Programmer','Clair',1000);
* 已创建 1 行。
* SQL> insert into group\_test values (10,'Architect', 'Gideon',1000);
* 已创建 1 行。
* SQL> insert into group\_test values (10,'Director', 'Hill',1000);
* 已创建 1 行。
* SQL>
* SQL> insert into group\_test values (20,'Coding', 'Jason',2000);
* 已创建 1 行。
* SQL> insert into group\_test values (20,'Programmer','Joey',2000);
* 已创建 1 行。
* SQL> insert into group\_test values (20,'Architect', 'Martin',2000);
* 已创建 1 行。
* SQL> insert into group\_test values (20,'Director', 'Michael',2000);
* 已创建 1 行。
* SQL>
* SQL> insert into group\_test values (30,'Coding', 'Rebecca',3000);
* 已创建 1 行。
* SQL> insert into group\_test values (30,'Programmer','Rex',3000);
* 已创建 1 行。
* SQL> insert into group\_test values (30,'Architect', 'Richard',3000);
* 已创建 1 行。
* SQL> insert into group\_test values (30,'Director', 'Sabrina',3000);
* 已创建 1 行。
* SQL>
* SQL> insert into group\_test values (40,'Coding', 'Samuel',4000);
* 已创建 1 行。
* SQL> insert into group\_test values (40,'Programmer','Susy',4000);
* 已创建 1 行。
* SQL> insert into group\_test values (40,'Architect', 'Tina',4000);
* 已创建 1 行。
* SQL> insert into group\_test values (40,'Director', 'Wendy',4000);
* 已创建 1 行。
* SQL>
* SQL> commit;
* 提交完成。
* SQL> select \* from group\_test;
* GROUP\_ID JOB        NAME           SALARY  
  ---------- ---------- ---------- ----------  
          10 Coding     Bruce            1000  
          10 Programmer Clair            1000  
          10 Architect  Gideon           1000  
          10 Director   Hill             1000  
          20 Coding     Jason            2000  
          20 Programmer Joey             2000  
          20 Architect  Martin           2000  
          20 Director   Michael          2000  
          30 Coding     Rebecca          3000  
          30 Programmer Rex              3000  
          30 Architect  Richard          3000  
    
    
    GROUP\_ID JOB        NAME           SALARY  
  ---------- ---------- ---------- ----------  
          30 Director   Sabrina          3000  
          40 Coding     Samuel           4000  
          40 Programmer Susy             4000  
          40 Architect  Tina             4000  
          40 Director   Wendy            4000
* 已选择16行。

二、group by一般分组与rollup语句区别  
先看一般的group by语句

**点击(此处)折叠或打开**

* SQL> select group\_id,sum(salary) from group\_test
* 2 group by group\_id;
* GROUP\_ID SUM(SALARY)
* ---------- -----------
* 30 12000
* 20 8000
* 40 16000
* 10 4000

使用rollup分组后如下

**点击(此处)折叠或打开**

* SQL> edit
* 已写入 file afiedt.buf
* 1 select group\_id,sum(salary) from group\_test
* 2\* group by rollup(group\_id)
* SQL> /
* GROUP\_ID SUM(SALARY)
* ---------- -----------
* 10 4000
* 20 8000
* 30 12000
* 40 16000
* 40000

以上sql与上一sql执行结果相比，通过group\_id分组查询，使用rollup查询结果会增加一个汇总合计列，可通过union all改写如下：

SQL> edit  
已写入 file afiedt.buf

  1  select group\_id,sum(salary) from group\_test  
  2  group by group\_id  
  3  union all  
  4  select null,sum(salary) from group\_test  
  5\* order by 1  
SQL> /

  GROUP\_ID SUM(SALARY)  
---------- -----------  
        10        4000  
        20        8000  
        30       12000  
        40       16000  
                   40000  
  
下面查看通过group\_id,job二个字段一般分组与rollup语句的区别

SQL> select group\_id,job,sum(salary) from group\_test  
  2  group by rollup(group\_id,job);

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        10 Coding            1000  
        10 Director          1000  
        10 Architect         1000  
        10 Programmer        1000  
        10                   4000  
        20 Coding            2000  
        20 Director          2000  
        20 Architect         2000  
        20 Programmer        2000  
        20                   8000  
        30 Coding            3000

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        30 Director          3000  
        30 Architect         3000  
        30 Programmer        3000  
        30                  12000  
        40 Coding            4000  
        40 Director          4000  
        40 Architect         4000  
        40 Programmer        4000  
        40                  16000  
                            40000

已选择21行。  
用union all方式对上面代码改写如下：

SQL> select group\_id,job,sum(salary) from group\_test  
  2  group by group\_id,job  
  3  union all  
  4  select group\_id,null,sum(salary) from group\_test  
  5  group by group\_id  
  6  union all  
  7  select null,null,sum(salary) from group\_test  
  8  order by 1,2;

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        10 Architect         1000  
        10 Coding            1000  
        10 Director          1000  
        10 Programmer        1000  
        10                   4000  
        20 Architect         2000  
        20 Coding            2000  
        20 Director          2000  
        20 Programmer        2000  
        20                   8000  
        30 Architect         3000

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        30 Coding            3000  
        30 Director          3000  
        30 Programmer        3000  
        30                  12000  
        40 Architect         4000  
        40 Coding            4000  
        40 Director          4000  
        40 Programmer        4000  
        40                  16000  
                            40000

已选择21行。

SQL> select group\_id,job,grouping(group\_id),grouping(job),sum(salary)  
  2  from group\_test  
  3  group by rollup(group\_id,job);

  GROUP\_ID JOB        GROUPING(GROUP\_ID) GROUPING(JOB) SUM(SALARY)  
---------- ---------- ------------------ ------------- -----------  
        10 Coding                      0             0        1000  
        10 Director                    0             0        1000  
        10 Architect                   0             0        1000  
        10 Programmer                  0             0        1000  
        10                             0             1        4000  
        20 Coding                      0             0        2000  
        20 Director                    0             0        2000  
        20 Architect                   0             0        2000  
        20 Programmer                  0             0        2000  
        20                             0             1        8000  
        30 Coding                      0             0        3000

  GROUP\_ID JOB        GROUPING(GROUP\_ID) GROUPING(JOB) SUM(SALARY)  
---------- ---------- ------------------ ------------- -----------  
        30 Director                    0             0        3000  
        30 Architect                   0             0        3000  
        30 Programmer                  0             0        3000  
        30                             0             1       12000  
        40 Coding                      0             0        4000  
        40 Director                    0             0        4000  
        40 Architect                   0             0        4000  
        40 Programmer                  0             0        4000  
        40                             0             1       16000  
                                       1             1       40000

已选择21行。

分析上面sql执行结果，可知grouping用法，当对某例进行了汇总合计时，grouping（此列）的值为1，否则为0。  
三、group by一般分组与cube语句区别

SQL> edit  
已写入 file afiedt.buf

  1  select group\_id,job,sum(salary) from group\_test  
  2  group by cube(group\_id,job)  
  3\* order by 1,2  
SQL> /

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        10 Architect         1000  
        10 Coding            1000  
        10 Director          1000  
        10 Programmer        1000  
        10                   4000  
        20 Architect         2000  
        20 Coding            2000  
        20 Director          2000  
        20 Programmer        2000  
        20                   8000  
        30 Architect         3000

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
        30 Coding            3000  
        30 Director          3000  
        30 Programmer        3000  
        30                  12000  
        40 Architect         4000  
        40 Coding            4000  
        40 Director          4000  
        40 Programmer        4000  
        40                  16000  
           Architect        10000  
           Coding           10000

  GROUP\_ID JOB        SUM(SALARY)  
---------- ---------- -----------  
           Director         10000  
           Programmer       10000  
                            40000

已选择25行。

你会发现，cube与rollup用法不同，cube会对所有例进行分组汇总合计。  
通过以上例子，可知：  
如果是Group by  ROLLUP(A, B, C)的话，首先会对(A、B、C)进行GROUP BY，然后对(A、B)进行GROUP BY，然后是(A)进行GROUP BY，最后对全表进行GROUP BY操作。  
如果是GROUP BY CUBE(A, B, C)，则首先会对(A、B、C)进行GROUP BY，然后依次是(A、B)，(A、C)，(A)，(B、C)，(B)，(C)，最后对全表进行GROUP BY操作。 grouping\_id()可以美化效果。除了使用GROUPING函数，还可以使用GROUPING\_ID来标识GROUP BY的结果。  
也可以 Group by Rollup(A,(B,C)) ，Group by A Rollup(B,C)，…… 这样任意按自己想要的形式结合统计数据，非常方便。