本文通过演示给出Oracle ROLLUP分组函数的用法，体验一下Oracle在统计查询领域中的函数魅力。ROLLUP分组函数可以理解为Group By分组函数封装后的精简用法，这里同时给出ROLLUP的Group By的改写思路。

1.初始化实验坏境

1）创建测试表group_test

```
SECOOLER@ora11g> create table group_test (group_id int,
job varchar2(10), name varchar2(10), salary int);

Table created.
```

2）初始化数据

```
insert into group_test values (10,'Coding',
'Bruce',1000);
insert into group_test values
(10,'Programmer','Clair',1000);
insert into group_test values (10,'Architect',
'Gideon',1000);
insert into group_test values (10,'Director',
'Hill',1000);

insert into group_test values (20,'Coding',
'Jason',2000);
insert into group_test values
(20,'Programmer','Joey',2000);
insert into group_test values (20,'Architect',
'Martin',2000);
insert into group_test values (20,'Director',
'Michael',2000);
```

```
insert into group_test values (30,'Coding',
'Rebecca',3000);
insert into group_test values
(30,'Programmer','Rex',3000);
insert into group_test values (30,'Architect',
'Richard',3000);
insert into group_test values (30,'Director',
'Sabrina',3000);

insert into group_test values (40,'Coding',
'Samuel',4000);
insert into group_test values
(40,'Programmer','Susy',4000);
insert into group_test values (40,'Architect',
'Tina',4000);
insert into group_test values (40,'Director',
'Wendy',4000);

commit;
```

3）初始化之后的数据情况如下：

```
SECOOLER@ora11g> set pages 100
SECOOLER@ora11g> 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
        30 Director   Sabrina              3000
        40 Coding     Samuel              4000
        40 Programmer Susy                4000
        40 Architect  Tina                4000
        40 Director   Wendy               4000

16 rows selected.
```

2.先看一下普通分组的效果：对group_id进行普通的group by操作---按照小组进行分组

```
SECOOLER@ora11g> select group_id,sum(salary) from
group_test group by group_id;

  GROUP_ID SUM(SALARY)
---------- -----------
        30       12000
        20        8000
        40       16000
        10        4000
```

3.对group_id进行普通的roolup操作---按照小组进行分组，同时求总计

```
SECOOLER@ora11g> select group_id,sum(salary) from
group_test group by rollup(group_id);

  GROUP_ID SUM(SALARY)
```

```
---------- -----------
        10        4000
        20        8000
        30       12000
        40       16000
                 40000
```

使用Group By语句翻译一下上面的SQL语句如下（union all一个统计所有数据的行）：

```
SECOOLER@ora11g> select group_id,sum(salary) from
group_test group by group_id
  2  union all
  3  select null, sum(salary) from group_test
  4  order by 1;

  GROUP_ID SUM(SALARY)
---------- -----------
        10        4000
        20        8000
        30       12000
        40       16000
                 40000
```

4.再看一个rollup两列的情况

```
SECOOLER@ora11g> select group_id,job,sum(salary) from
group_test 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
        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

21 rows selected.
```

上面的SQL语句该如何使用Group By进行翻译呢？

答案如下：

```
SECOOLER@ora11g> select group_id,job,sum(salary) from
group_test group by group_id, job
  2  union all
  3  select group_id,null,sum(salary) from group_test
group by group_id
  4  union all
  5  select null,null,sum(salary) from group_test
```

```
  6  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
        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

21 rows selected.
```

5.补充一步，体验一下GROUPING函数的效果

直接看效果就OK啦：

```
SECOOLER@ora11g> select
group_id,job,grouping(GROUP_ID),grouping(JOB),sum(salary)
from group_test 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
```

```
         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

21 rows selected.
```

如果显示"1"表示GROUPING函数对应的列（例如JOB字段）是由于ROLLUP函数所产生的空值对应的信息，即对此列进行汇总计算后的结果。

如果显示"0"表示此行对应的这列参未与ROLLUP函数分组汇总活动。

进一步体验CUBE的魅力

```
sec@ora10g> select
group_id,job,grouping(GROUP_ID),grouping(JOB),sum(salary)
from group_test group by cube(group_id, job) order by 1;

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

```
        30 Coding                          0
0        3000
        30 Director                        0
0        3000
        30 Programmer                      0
0        3000
        30                                 0
1       12000
        40 Architect                       0
0        4000
        40 Coding                          0
0        4000
        40 Director                        0
0        4000
        40 Programmer                      0
0        4000
        40                                 0
1       16000
           Architect                       1
0       10000
           Coding                          1
0       10000
           Director                        1
0       10000
           Programmer                      1
0       10000
                                           1
1       40000

25 rows selected.
```

仔细观察一下，CUBE与ROLLUP之间的细微差别

rollup(a,b)　　统计列包含：(a,b)、(a)、()

rollup(a,b,c)　统计列包含：(a,b,c)、(a,b)、(a)、()

……以此类推ing……

cube(a,b)　　统计列包含：(a,b)、(a)、(b)、()

cube(a,b,c)　　统计列包含：(a,b,c)、(a,b)、(a,c)、(b,c)、(a)、(b)、(c)、()

……以此类推ing……

So，上面例子中CUBE的结果比ROLLUP多了下面关于第一列GROUP_ID的统计信息：

```
            Architect                      1
0        10000
            Coding                         1
0        10000
            Director                       1
0        10000
```

CUBE在ROLLUP的基础上进一步从各种维度上给出细化的统计汇总结果。