

今天给各位分享rollup的知识，其中也会对rollup是什么意思进行解释，如果能碰巧解决你现在面临的问题，别忘了关注本站，如果有不同的见解与看法，请积极在评论区留言，现在开始进入正题！

ROLLUP/CUBE基本语法，ROLLUP/CUBE的使用语法如下：

```
SELECT [column,] group_function(column) . .
```

```
FROM?table
```

```
[WHERE? condition]
```

```
[GROUP BY? [ROLLUP/CUBE]? group_by_expression]
```

```
[HAVING having_expression];
```

```
[ORDER BY??column];
```

使用ROLLUP/CUBE，必须具备：1.查询语句为分组函数（group function）查询。最常见的有：AVG, SUM, MAX, MIN, COUNT等都是分组函数。2. ROLLUP/CUBE的使用位置在GROUP BY后面。

### Oracle的GROUP

BY语句除了最基本的语法外，还支持ROLLUP和CUBE语句。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)。

### 扩展资料

ROLLUP：? GROUP BY A, B会对A和B出现的每种可能性进行分组。而GROUP BY ROLLUP(A,

B)会对A和B出现的每种可能性分组，A的每种可能性分组，最后计算所有的情况。

例如：

rollup(a,b) 包括：(a,b)、(a)、()

rollup(a,b,c) 包括：(a,b,c)、(a,b)、(a)、()

CUBE：? 和ROLLUP类似，GROUP BY CUBE(A, B)会对A和B出现的每种可能性分组，A的每种可能性分组，B的每种可能性分组，最后计算所有的情况。

Oracle为了解决引入ROLLUP和CUBE后，汇总和NULL值无法区分的情况，提供了GROUPING、GROUPING\_ID和GROUP\_ID三个函数。

例如：? cube(a,b) 包括：(a,b)、(a)、(b)、()? cube(a,b,c)  
包括：(a,b,c)、(a,b)、(a,c)、(b,c)、(a)、(b)、(c)、()

ROLLUP和CUBE 用法?????

????? Oracle的GROUP

BY语句除了最基本的语法外，还支持ROLLUP和CUBE语句。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。

Rollup()：分组函数可以理解为group by的精简模式，具体分组模式如下：

Rollup(a,b,c): (a,b,c),(a,b),(a),(全表)

Cube():分组函数也是以group by为基础，具体分组模式如下：

cube(a,b,c):(a,b,c),(a,b),(a,c),(b,c),(a),(b),(c),(全表)

下面准备数据比较一下两个函数的不同：

1、准备数据：

2、使用rollup函数查询

```
select group_id,job,name,sum(salary) from GROUP_TEST group by rollup(group_id,job,name) ;
```

3、使用cube函数：

```
select group_id,job,name,sum(salary) from GROUP_TEST group by cube(group_id,job,name)
```

4、对比：从最后查询出来的数据条数就差了好多，下面看一下将两个函数从转化成对应的group函数语句：

rollup函数：

```
select group_id,job,name,sum(salary) from GROUP_TEST group by rollup(group_id,job,name);
```

等价于：

```
select group_id,job,name,sum(salary) from GROUP_TEST group by group_id,job,name
```

union all

```
select group_id,job,null,sum(salary) from GROUP_TEST group by group_id,job
```

union all

```
select group_id,null,null,sum(salary) from GROUP_TEST group by group_id
```

union all

```
select null,null,null,sum(salary) from GROUP_TEST
```

cube函数：

```
select group_id,job,name,sum(salary) from GROUP_TEST group by  
cube(group_id,job,name) ;
```

等价于：

```
select group_id,job,name,sum(salary) from GROUP_TEST group by  
group_id,job,name
```

union all

```
select group_id,job,null,sum(salary) from GROUP_TEST group by  
group_id,job
```

union all

```
select group_id,null,name,sum(salary) from GROUP_TEST group by  
group_id,name
```

union all

```
select group_id,null,null,sum(salary) from GROUP_TEST group by group_id
```

union all

```
select null,job,name,sum(salary) from GROUP_TEST group by job,name
```

union all

```
select null,job,null,sum(salary) from GROUP_TEST group by job
```

union all

```
select null,null,name,sum(salary) from GROUP_TEST group by name
```

union all

```
select null,null,null,sum(salary) from GROUP_TEST
```

5、由此可见两个函数对于汇总统计来说要比普通函数好用的多，另外还有一个配套使用的函数

grouping(\*\*):当\*\*字段为null的时候值为1，当字段\*\*非null的时候值为0；

```
select grouping(group_id),job,name,sum(salary) from GROUP_TEST group
by rollup(group_id,job,name);
```

6、添加一列用来直观的显示所有的汇总字段：

```
select group_id,job,name,
```

```
?case when name is null and nvl(group_id,0)=0 and job is null???then
'全表聚合'
```

```
???when name is null and nvl(group_id,0)=0 and job is not null then
'JOB聚合'
```

```
???when name is null and???grouping(group_id)=0 and job is null then
'GROUPID聚合'
```

```
???when name is not null and nvl(group_id,0)=0 and job is null???then
'Name聚合'
```

```
???when name is not null and grouping(group_id)=0 and job is null???then
'GROPName聚合'
```

```
???when name is not null and grouping(group_id)=1 and job is not
null???then 'JOBName聚合'
```

```
???when name is???null and grouping(group_id)=0 and job is not null???then
'GROUPJOB聚合'
```

```
????else
```

```
? '三列汇总' end ,
```

```
sum(salary) from GROUP_TEST group by cube(group_id,job,name) ;
```

rollup

n.

几上归纳；卷曲；袅袅上升；

[例句]Unable to run SQL queries against cubes that have custom rollup formulas.

无法对具有自定义汇总公式的多维数据集执行sql查询

rollup 初版源码，才1000 行左右。

\_ollup 使用了 acorn 和 magic-string 两个库。为了更好的阅读 rollup 源码，必须对它们有所了解。

\_corn 是一个 JavaScript 语法解析器，它将 JavaScript 字符串解析成语法抽象树 AST。magic-string

\_agic-string 也是 rollup 作者写的一个关于字符串操作的库。

相信经过小编对rollup和rollup是什么意思的介绍，你对rollup了解更加地透彻了，感谢你对我们地支持与关注！