- GROUPING SETS
  - 关键字：
  - 简单示例：
  - presto中grouping sets函数
- cube
- rollup

GROUPING SETS

关键字：

- GROUPING SETS： 根据不同的维度组合进行聚合，等价于将不同维度的GROUP BY结果集进行UNION ALL
- GROUPING__ID：表示结果属于哪一个分组集合，属于虚字段

简单示例：

关于grouping sets的使用，通俗的说，grouping sets是一种将多个group by 逻辑写在一个sql语句中的便利写法

```
create table temp.score_grouping as
select
  grouping__id, ---grouping__id是两个下划线
    class,
    sex,
    course,
   avg(score)
from
    tableName
group by  --group by包含所需所有维度字段
    class,
    sex,
    course -------此处无逗号
grouping sets
(
   (class,  course),
   ( class,sex),
   (sex,course),
   (course)
)
select *
from temp.score_grouping
where grouping__id in ('5','6')
```

说明：

- 1、select子句中的GROUPING__ID是两个下划线；
- 2、group by后面放的字段表示要分组聚合的全部字段；
- 3、grouping sets前没有逗号；
- 4、grouping_id的计算： 它是根据group by后面声明的顺序字段是否存在于当前group by中的一个二进制位组合数据,若组合中出现即为1，反正则为0，group by 后字段先出现的放在最低位，依次排开： 比如 group by class,sex,course,则二进制的顺序为：course sex class ,grouping sets字段出现则为1，反之

则为0，

- 比如(class, course), 二进制为 101，十进制则为5，则grouping__id为5，同理grouping__id为6，则组合为（sex,course），二进制为110；

实例**1**

```
-- 正确语句
select province
 ,city
 ,catgory_id
 ,catgory_name
 ,goodsid
 ,goodsname
 ,sum(sales_qty) as sales_qty
 ,sum(sales_amt) as sales_amt
 ,GROUPING__ID
from
(
select t1.province
 ,t1.city
 ,t2.catgory_id
 ,t2.catgory_name
 ,t1.goodsid
 ,t1.goodsname
 ,sales_qty
 ,sales_amt
from temp.goods_sale_info t1
left join
temp.goods_info t2
on t1.goodsid=t2.goodsid
) t  --在表t的基础上使用grouping sets函数
group by
 province
 ,city
 ,catgory_id
```

```
 ,catgory_name
 ,goodsid
 ,goodsname
 grouping sets(
 (province,catgory_id,catgory_name)
 --这里仅有3个字段，但select中列有不在sets中的非分组字段city、goodsid、goodsname，hive不报错，presto会报错
 )
```

注意：

1.双表联结的结果出来之后再用grouping sets,即先有t 表，在表t的基础上使用grouping sets函数； 2.sets中的字段不应该含表名；

presto中grouping sets函数

示例：

```
-- 如果group by写上单一字段
select province
 ,city
 ,catgory_id
 ,catgory_name
 ,goodsid
 ,goodsname
 ,sum(sales_qty) as sales_qty
 ,sum(sales_amt) as sales_amt
 ,grouping(province,city,catgory_id,catgory_name,goodsid,goodsname)
from
(
select t1.province
```

```
 ,t1.city
 ,t2.catgory_id
 ,t2.catgory_name
 ,t1.goodsid
 ,t1.goodsname
 ,sales_qty
 ,sales_amt
from temp.goods_sale_info t1
left join
temp.goods_info t2
on t1.goodsid=t2.goodsid
) t  --也是要在表t的基础上使用grouping sets函数
group by
    province
 ,city
 ,catgory_id
 ,catgory_name
 ,goodsid
 ,goodsname
    ,grouping sets( --这里记得加上逗号,
 (province,catgory_id,catgory_name),
 (province,catgory_id,catgory_name,goodsid,goodsname),
 (province,city),
 (province)
 )
```

注意： 1、函数grouping要与group by、grouping sets配合使用 2、函数grouping()中列出sets中所有分组涉及的字段，运行后grouing()列生成结果为二进制转化来的十进制数字；，出现为0，不出现为1，按照顺序，早出现的放高位，依次类推；

```
SELECT origin_state, origin_zip, destination_state, sum(package_weight),
       grouping(origin_state, origin_zip, destination_state)
FROM shipping
GROUP BY GROUPING SETS (
        (origin_state)---011  3,
```

```
        (origin_state, origin_zip) --001  1,
        (destination_state));---110   6
origin_state | origin_zip | destination_state | _col3 | _col4
-------------+------------+-------------------+-------+-------
California   | NULL       | NULL              |  1397 |     3
New Jersey   | NULL       | NULL              |   225 |     3
New York     | NULL       | NULL              |     3 |     3
California   |      94131 | NULL              |    60 |     1
New Jersey   |       7081 | NULL              |   225 |     1
California   |      90210 | NULL              |  1337 |     1
New York     |      10002 | NULL              |     3 |     1
NULL         | NULL       | New Jersey        |    58 |     6
NULL         | NULL       | Connecticut       |  1562 |     6
NULL         | NULL       | Colorado          |     5 |     6
(10 rows)
```

3、group by后面只跟grouping sets()，不加select中的单一字段，否则函数grouping sets无作用

```
-- 如果group by写上单一字段
select province
 ,city
 ,catgory_id
 ,catgory_name
 ,goodsid
 ,goodsname
 ,sum(sales_qty) as sales_qty
 ,sum(sales_amt) as sales_amt
 ,grouping(province,city,catgory_id,catgory_name,goodsid,goodsname)
from
(
select t1.province
 ,t1.city
 ,t2.catgory_id
```

```
 ,t2.catgory_name
 ,t1.goodsid
 ,t1.goodsname
 ,sales_qty
 ,sales_amt
from temp.goods_sale_info t1
left join
temp.goods_info t2
on t1.goodsid=t2.goodsid
) t  --也是要在表t的基础上使用grouping sets函数
group by
    province
 ,city
 ,catgory_id
 ,catgory_name
 ,goodsid
 ,goodsname
    ,grouping sets( --这里记得加上逗号,
 (province,catgory_id,catgory_name),
 (province,catgory_id,catgory_name,goodsid,goodsname),
 (province,city),
 (province)
 )
```

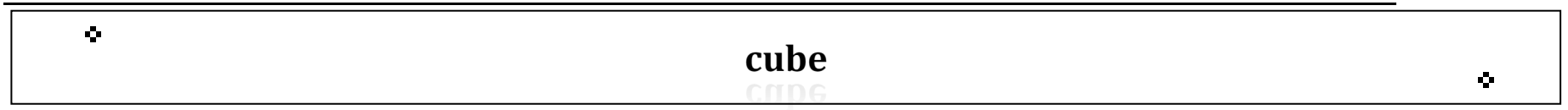
4、不用的分组字段不要在**select**子句中写出

```
-- 与hive不同，如果不出现在grouping sets中的字段，select子句写上会报错
-- 比如sets中不涉及city、goodsid、goodsname，select子句中写出来报错

select province
--  ,city
 ,catgory_id
```

```
 ,catgory_name
--  ,goodsid
--  ,goodsname
 ,sum(sales_qty) as sales_qty
 ,sum(sales_amt) as sales_amt
 ,grouping(province,catgory_id,catgory_name)
from
(
select t1.province
 ,t1.city
 ,t2.catgory_id
 ,t2.catgory_name
 ,t1.goodsid
 ,t1.goodsname
 ,sales_qty
 ,sales_amt
from temp.goods_sale_info t1
left join
temp.goods_info t2
on t1.goodsid=t2.goodsid
) t
group by
 grouping sets(
 (province,catgory_id,catgory_name)
 )
```

5、函数**grouping**中要将**grouping sets**所有分组组合用到的字段取并集列出



cube同样是HIVE提供的多维分析的内置函数，可以看作是全组合版的grouping sets

```
select
  k1,k2,k3,sum(v1)
from
  tbl
group by
  k1,k2,k3
with cube
```

举例 还是上述的例子，对于department、name、age，如果需要求所有的组合聚合结果，使用grouping sets实现，如下：

```
select
  department,name,age,count(1)
from tbl
  group by department,name,age
grouping sets ((department,name,age),(department,name),(department,age),(department),(name,age),(name),(age),())
```

而函数cube直接就帮你全组合了字段，使用cube实现，如下：

```
select
  department,name,age,count(1)
from tbl
  group by department,name,age
with cube
```

> 这里with cube就等同于grouping sets ((department,name,age),(department,name),(department,age),(department),(name,age),(name),(age),())

> 注意，cube会返回所有group by后的字段全组合的聚合结果，对于未使用到的组合，会用null值填充

rollup

> rollup的含义是卷曲的意思，顾名思义，就是会从右向左的组合字段，得到聚合结果

```
select
  k1,k2,k3,sum(v1)
from
  tbl
group by
  k1,k2,k3
with rollup
```

举例 对于cube，虽然很方便的返回全组合的聚合结果，但是在实际使用中可能不需要这样，还是上个例子，对于字段department、name、age，如果想要得到(department,name,age), (department, name),(department),()的组合，使用grouping sets实现，如

```
select
  department,name,age,count(1)
from tbl
  group by department,name,age
grouping sets ((department,name,age),(department,name),(department,age),(department),())
```

如果使用rollup实现，如下：

```
select   department,name,age,count(1) from tbl   group by department,name,age with rollup
```

这里的with rollup就等同于grouping sets ((department,name,age),(department,name),(department,age),(department),())

> 注意，rollup的组合顺序是按照group by后的字段，从左到右迭代的，对于未使用到的字段，会使用null值填充

- END -