Grouping sets and cube by in Postgres

Tópicos Avançados de Bases de Dados

Grouping sets tutorial

```
CREATE TABLE my_test (
    id int,
    salary int,
    revised_salary int,
    old_salary  int,
    leave  int,
    joining_date date
);
```

Grouping sets tutorial

```
insert into my_test values(10,5000,4000,1000,5,'2019-11-15');

insert into my_test values(10,6000,9000,2000,1,'2019-11-16');

insert into my_test values(10,7000,3000,4400,2,'2019-11-17');

insert into my_test values(20,8000,4000,6600,6,'2019-11-15');

insert into my_test values(20,9000,9400,8800,10,'2019-11-16');

insert into my_test values(20,2000,7800,9400,23,'2019-11-17');

insert into my_test values(30,4400,6600,4200,44,'2019-11-15');

insert into my_test values(30,1500,3600,4300,66,'2019-11-16');

insert into my_test values(30,2000,2600,4500,77,'2019-11-17');
```

Grouping sets tutorial

```
select id, sum(salary) as total_salary,
sum(revised_salary) as total_revised_salary ,
sum(old_salary) as total_old_salary,
sum(leave) as total_leave
from my_test
group by id
order by id;
```

Grouping sets tutorial

```
select joining_date, sum(salary) as total_salary,

            sum(revised_salary) as total_revised_salary,

            sum(old_salary) as total_old_salary ,

            sum(leave) as total_leave

            from my_test

            group by joining_date

            order by joining_date;
```

Grouping sets tutorial

```
select id,joining_date,sum(salary) as total_salary,

sum(revised_salary) as total_revised_salary ,

sum(old_salary) as total_old_salary ,

sum(leave) as total_leave

from my_test

group by grouping sets ( id, joining_date)

order by id;
```

Grouping sets tutorial

We can also use empty parentheses (), which gives us the total value:

```
select id,joining_date,sum(salary) as total_salary

from my_test

group by grouping sets ( id, joining_date,())

order by id;
```

Grouping sets tutorial

CUBE(X,Y,Z) will create grouping sets like this:

(X,Y,Z)
(,Y,Z)
(X,Y,)
(X, ,Y)
(X)
(Y)
(Z)
()

```
select id,joining_date,sum(salary) as total_salary
    ,sum(revised_salary) as total_revised_salary ,
    sum(old_salary) as total_old_salary ,
    sum(leave) as total_leave
     from my_test
    group by cube ( id, joining_date)
    order by id;
```

Grouping sets tutorial

ROLLUP(id,joining_date) will generate GROUPING SETS like this:

```
(id,joining_date)
```

```
(id)
```

```
()
```

```
select id,joining_date,sum(salary) as total_salary,

sum(revised_salary) as total_revised_salary ,

sum(old_salary) as total_old_salary ,

sum(leave) as total_leave

 from my_test

group by rollup ( id, joining_date)  order by 1, 2;
```