

Table of Contents

| | |
|----------------------------|---|
| 1. RollUp | 1 |
| 1.1. Two level group | 2 |
| 2. Cube..... | 4 |
| 3. Grouping..... | 5 |
| 4. Grouping sets..... | 8 |

1. RollUp

The RollUp function lets you calculate multiple levels of subtotals and a grand total. The Rollup function is added to the Group By clause. You give the Rollup function one or more grouping columns for the subtotals.

Demo 01: Simple group by with a single attribute; this is not using rollup

```
select dept_id, sum(salary)
from adv_emp
group by dept_id
order by dept_id;
```

| DEPT_ID | SUM(SALARY) |
|---------|-------------|
| 10 | 235000 |
| 15 | 25000 |
| 20 | 81000 |
| 30 | 188000 |

Demo 02: The rollup function gives you department totals and a grand total (529000) with a Null for the dept id since that row does not belong to a department.

```
select dept_id, sum(salary)
from adv_emp
group by rollup( dept_id)
order by dept_id;
```

| DEPT_ID | SUM(SALARY) |
|---------|-------------|
| 10 | 235000 |
| 15 | 25000 |
| 20 | 81000 |
| 30 | 188000 |
| | 529000 |

Demo 03: Replacing the null with a label

```
select
  coalesce(to_char(dept_id, '99999'), 'Total') as Dept
, sum(salary) as SalaryTotal
from adv_emp
group by rollup( dept_id)
order by dept_id;
```

| DEPT | SALARYTOTAL |
|-------|-------------|
| 10 | 235000 |
| 15 | 25000 |
| 20 | 81000 |
| 30 | 188000 |
| Total | 529000 |

1.1. Two level group

With a two level group we get subtotals by year_hired within each department and the department total and a grand total.

Demo 04: Simple two level group

```
select dept_id, year_hired, sum(salary)
from adv_emp
group by rollup( dept_id, year_hired)
;
```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) |
|---------|------------|-------------|
| 10 | 2010 | 15000 |
| 10 | 2012 | 130000 |
| 10 | 2014 | 90000 |
| 10 | | 235000 |
| 15 | 2012 | 25000 |
| 15 | | 25000 |
| 20 | 2010 | 42000 |
| 20 | 2013 | 12000 |
| 20 | 2014 | 27000 |
| 20 | | 81000 |
| 30 | 2010 | 56000 |
| 30 | 2012 | 24500 |
| 30 | 2013 | 71000 |
| 30 | 2014 | 36500 |
| 30 | | 188000 |
| | | 529000 |

Demo 05: We can group by year_hired and department within the year_hired

```
select year_hired, dept_id, sum(salary)
from adv_emp
group by rollup( year_hired, dept_id )
;
```

| YEAR_HIRED | DEPT_ID | SUM(SALARY) |
|------------|---------|-------------|
| 2010 | 10 | 15000 |
| 2010 | 20 | 42000 |
| 2010 | 30 | 56000 |
| 2010 | | 113000 |
| 2012 | 10 | 130000 |
| 2012 | 15 | 25000 |
| 2012 | 30 | 24500 |
| 2012 | | 179500 |
| 2013 | 20 | 12000 |
| 2013 | 30 | 71000 |
| 2013 | | 83000 |
| 2014 | 10 | 90000 |
| 2014 | 20 | 27000 |
| 2014 | 30 | 36500 |
| 2014 | | 153500 |
| | | 529000 |

Demo 06: roll up on one grouping only; no grand total since we are not rolling up by department

```
select dept_id, year_hired, sum(salary), Count(*)
from adv_emp
group by ( dept_id ), rollup( year_hired );
```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT(*) |
|---------|------------|-------------|----------|
| 10 | 2010 | 15000 | 1 |
| 10 | 2012 | 130000 | 5 |
| 10 | 2014 | 90000 | 3 |
| 10 | | 235000 | 9 |
| 15 | 2012 | 25000 | 1 |
| 15 | | 25000 | 1 |
| 20 | 2010 | 42000 | 2 |
| 20 | 2013 | 12000 | 1 |
| 20 | 2014 | 27000 | 1 |
| 20 | | 81000 | 4 |
| 30 | 2010 | 56000 | 2 |
| 30 | 2012 | 24500 | 2 |
| 30 | 2013 | 71000 | 3 |
| 30 | 2014 | 36500 | 2 |
| 30 | | 188000 | 9 |

Demo 07: With additional formatting for labeling the result set

```
select
  case when dept_id is null then ' Grand total'
  else to_char(dept_id, '99999')
  end as "Department"
,
  case when year_hired is null and dept_id is null then ''
  when year_hired is null then ' Dept total'
  else to_char(year_hired, '9999')
  end as "YearHired"
, sum(salary) as "SalaryTotal"
, count (*) as "EmpCount"
from adv_emp
group by Rollup( dept_id, year_hired );
```

| Department | YearHired | SalaryTotal | EmpCount |
|-------------|------------|-------------|----------|
| 10 | 2010 | 15000 | 1 |
| 10 | 2012 | 130000 | 5 |
| 10 | 2014 | 90000 | 3 |
| 10 | Dept total | 235000 | 9 |
| 15 | 2012 | 25000 | 1 |
| 15 | Dept total | 25000 | 1 |
| 20 | 2010 | 42000 | 2 |
| 20 | 2013 | 12000 | 1 |
| 20 | 2014 | 27000 | 1 |
| 20 | Dept total | 81000 | 4 |
| 30 | 2010 | 56000 | 2 |
| 30 | 2012 | 24500 | 2 |
| 30 | 2013 | 71000 | 3 |
| 30 | 2014 | 36500 | 2 |
| 30 | Dept total | 188000 | 9 |
| Grand total | | 529000 | 23 |

2. Cube

Demo 08: This is one of the Rollups we did earlier

```
select dept_id, year_hired, sum(salary), count (*)
from adv_emp
group by Rollup( dept_id, year_hired )
;
```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT (*) |
|---------|------------|-------------|-----------|
| 10 | 2010 | 15000 | 1 |
| 10 | 2012 | 130000 | 5 |
| 10 | 2014 | 90000 | 3 |
| 10 | | 235000 | 9 |
| 15 | 2012 | 25000 | 1 |
| 15 | | 25000 | 1 |
| 20 | 2010 | 42000 | 2 |
| 20 | 2013 | 12000 | 1 |
| 20 | 2014 | 27000 | 1 |
| 20 | | 81000 | 4 |
| 30 | 2010 | 56000 | 2 |
| 30 | 2012 | 24500 | 2 |
| 30 | 2013 | 71000 | 3 |
| 30 | 2014 | 36500 | 2 |
| 30 | | 188000 | 9 |
| | | 529000 | 23 |

Demo 09: Then change the Rollup to Cube and we get more rows of data. In addition to the Grand total and the total by department, we also get a total by year_hired. The Cube function aggregates for all possible combinations of the specified columns. With two columns we have four possible combinations: dept_id, year_hired, both dept_id and year_hired, all- the grand total.

First we get the GrandTotal, then the Year totals, and department totals and Dept + year totals

```
select dept_id
, year_hired
, sum(salary)
, count (*)
from adv_emp
group by CUBE( dept_id , year_hired )
;
```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT (*) |
|---------|------------|-------------|-----------|
| | | 529000 | 23 |
| | 2010 | 113000 | 5 |
| | 2012 | 179500 | 8 |
| | 2013 | 83000 | 4 |
| | 2014 | 153500 | 6 |
| 10 | | 235000 | 9 |
| 10 | 2010 | 15000 | 1 |
| 10 | 2012 | 130000 | 5 |
| 10 | 2014 | 90000 | 3 |
| 15 | | 25000 | 1 |
| 15 | 2012 | 25000 | 1 |
| 20 | | 81000 | 4 |
| 20 | 2010 | 42000 | 2 |

| | | | |
|----|------|--------|---|
| 20 | 2013 | 12000 | 1 |
| 20 | 2014 | 27000 | 1 |
| 30 | | 188000 | 9 |
| 30 | 2010 | 56000 | 2 |
| 30 | 2012 | 24500 | 2 |
| 30 | 2013 | 71000 | 3 |
| 30 | 2014 | 36500 | 2 |

3. Grouping

One of the features of the use of RollUp and Cube is that we do not get labels that indicate which rows are totals- we simply get blanks. To remedy this, we use the GROUPING function with either Decode or Case to display labels. The Grouping function takes a single argument and returns either 0 (if this is not a totaling row) or 1 (if this is a totaling row).

The next query shows how the Grouping function works. You would not normally display the value of Grouping.

The first two return rows are grouping row for both dept and year hired- the two grouping functions return a 1.

The next row is a grouping row for dept but not for year hired. The grp_dept column has a 0 and the grp_year column has a 1

The last row is not a grouping row for either dept or year hired- both grouping functions return 1.

Demo 10: The GROUPING function returns a 0 or a 1

```
select
  grouping(dept_ID) as grp_dept
, grouping (year_hired) as grp_year
, dept_id, year_hired
, sum(salary)
, count (*)
from adv_emp
group by CUBE( dept_id , year_hired )
order by dept_id, year_hired;
```

| GRP_DEPT | GRP_YEAR | DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT (*) |
|----------|----------|---------|------------|-------------|-----------|
| 0 | 0 | 10 | 2010 | 15000 | 1 |
| 0 | 0 | 10 | 2012 | 130000 | 5 |
| 0 | 0 | 10 | 2014 | 90000 | 3 |
| 0 | 1 | 10 | | 235000 | 9 |
| 0 | 0 | 15 | 2012 | 25000 | 1 |
| 0 | 1 | 15 | | 25000 | 1 |
| 0 | 0 | 20 | 2010 | 42000 | 2 |
| 0 | 0 | 20 | 2013 | 12000 | 1 |
| 0 | 0 | 20 | 2014 | 27000 | 1 |
| 0 | 1 | 20 | | 81000 | 4 |
| 0 | 0 | 30 | 2010 | 56000 | 2 |
| 0 | 0 | 30 | 2012 | 24500 | 2 |
| 0 | 0 | 30 | 2013 | 71000 | 3 |
| 0 | 0 | 30 | 2014 | 36500 | 2 |
| 0 | 1 | 30 | | 188000 | 9 |
| 1 | 0 | | 2010 | 113000 | 5 |
| 1 | 0 | | 2012 | 179500 | 8 |
| 1 | 0 | | 2013 | 83000 | 4 |
| 1 | 0 | | 2014 | 153500 | 6 |
| 1 | 1 | | | 529000 | 23 |

20 rows selected.

This is the order we want for the rows and the labels to be displayed; we want the labels in the first column, replacing the two Grouping columns. This is sample data only.

| GRP_DEPT | GRP_YEAR | DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT(*) | |
|----------|----------|---------|------------|-------------|----------|-------------|
| 0 | 0 | 100 | 2000 | 99999 | 9 | |
| 0 | 0 | 100 | 2007 | 9999 | 9 | |
| 0 | 1 | 100 | | 91999 | 99 | Dept total |
| 0 | 0 | 200 | 2008 | 99999 | 9 | |
| 0 | 1 | 200 | | 99999 | 99 | Dept total |
| 0 | 0 | 300 | 2007 | 999999 | 99 | |
| 0 | 1 | 300 | | 999999 | 99 | Dept total |
| 1 | 0 | | 2000 | 99999 | 99 | Year total |
| 1 | 0 | | 2007 | 999999 | 99 | Year total |
| 1 | 0 | | 2008 | 99999 | 99 | Year total |
| 1 | 1 | | | 999999 | 999 | Grand total |

11 rows selected

The pattern is that if both grouping are 1 then this is the **grand total line**; if the dept grouping is 0 and the year grouping is 1 then this is a **Dept total line**; if the dept grouping is 1 and the year grouping is 0 then this is a **Year total line**; if they are both 0, then we do not have a label. We can use a case statement for this. I added an Rpad to control the width of the first column.

Demo 11:

```
Column Descr format A20
select
  Rpad( CASE
    when GROUPING(dept_ID) = 1 and Grouping (year_hired) = 1
      then 'Grand Total'
    when GROUPING(dept_ID) = 1 and Grouping (year_hired) = 0
      then '  Year Total'
    when GROUPING(dept_ID) = 0 and Grouping (year_hired) = 1
      then '    Dept Total'
    when GROUPING(dept_ID) = 0 and Grouping (year_hired) = 0
      then '      '
    end, 20) AS Descr
, dept_id, year_hired
, sum(salary), count ('emp')
from adv_emp
group by Cube( dept_id , year_hired )
order by dept_id , year_hired
;
```

| DESCR | DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT('EMP') |
|-------|---------|------------|-------------|--------------|
| | 10 | 2010 | 15000 | 1 |
| | 10 | 2012 | 130000 | 5 |
| | 10 | 2014 | 90000 | 3 |

| | | | | |
|-------------|----|------|--------|----|
| Dept Total | 10 | | 235000 | 9 |
| | 15 | 2012 | 25000 | 1 |
| Dept Total | 15 | | 25000 | 1 |
| | 20 | 2010 | 42000 | 2 |
| | 20 | 2013 | 12000 | 1 |
| | 20 | 2014 | 27000 | 1 |
| Dept Total | 20 | | 81000 | 4 |
| | 30 | 2010 | 56000 | 2 |
| | 30 | 2012 | 24500 | 2 |
| | 30 | 2013 | 71000 | 3 |
| | 30 | 2014 | 36500 | 2 |
| Dept Total | 30 | | 188000 | 9 |
| Year Total | | 2010 | 113000 | 5 |
| Year Total | | 2012 | 179500 | 8 |
| Year Total | | 2013 | 83000 | 4 |
| Year Total | | 2014 | 153500 | 6 |
| Grand Total | | | 529000 | 23 |

Demo 12: An alternative

Column Department format A20

Column year_hired format A20

```
select
  lpad(case when Grouping(dept_id) = 1 then '--- Year Total ---'
        else To_Char(Dept_id) end, 20) as Department
, lpad(case when Grouping(year_hired) = 1 then '--- Dept Total---'
        else To_Char(year_hired) end , 20) as Year_Hired
, sum(salary)
, count ('emp')
from adv_emp
group by cube( dept_id , year_hired )
order by dept_id , year_hired
;
```

| DEPARTMENT | YEAR_HIRED | SUM(SALARY) | COUNT('EMP') |
|--------------------|-------------------|-------------|--------------|
| 10 | 2010 | 15000 | 1 |
| 10 | 2012 | 130000 | 5 |
| 10 | 2014 | 90000 | 3 |
| 10 | --- Dept Total--- | 235000 | 9 |
| 15 | 2012 | 25000 | 1 |
| 15 | --- Dept Total--- | 25000 | 1 |
| 20 | 2010 | 42000 | 2 |
| 20 | 2013 | 12000 | 1 |
| 20 | 2014 | 27000 | 1 |
| 20 | --- Dept Total--- | 81000 | 4 |
| 30 | 2010 | 56000 | 2 |
| 30 | 2012 | 24500 | 2 |
| 30 | 2013 | 71000 | 3 |
| 30 | 2014 | 36500 | 2 |
| 30 | --- Dept Total--- | 188000 | 9 |
| --- Year Total --- | 2010 | 113000 | 5 |
| --- Year Total --- | 2012 | 179500 | 8 |
| --- Year Total --- | 2013 | 83000 | 4 |
| --- Year Total --- | 2014 | 153500 | 6 |
| --- Year Total --- | --- Dept Total--- | 529000 | 23 |

4. Grouping sets

The Grouping Sets expression lets you specify which groupings you want for the aggregates.

Demo 13: Here I have a group by department, a group by year_hired, a group by both attributes and a group by all – indicated by ()

Column year_hired format 9999

```

Select dept_id, year_hired, sum(salary), count ('emp')
from adv_emp
group by grouping sets (
    dept_id
    , year_hired
    , (dept_id, year_hired)
    , ()
);

```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT('EMP') |
|---------|------------|-------------|--------------|
| ----- | | | |
| 10 | 2010 | 15000 | 1 |
| 20 | 2010 | 42000 | 2 |
| 30 | 2010 | 56000 | 2 |
| 10 | 2012 | 130000 | 5 |
| 15 | 2012 | 25000 | 1 |
| 30 | 2012 | 24500 | 2 |
| 20 | 2013 | 12000 | 1 |
| 30 | 2013 | 71000 | 3 |
| 10 | 2014 | 90000 | 3 |
| 20 | 2014 | 27000 | 1 |
| 30 | 2014 | 36500 | 2 |
| | 2010 | 113000 | 5 |
| | 2012 | 179500 | 8 |
| | 2013 | 83000 | 4 |
| | 2014 | 153500 | 6 |
| 10 | | 235000 | 9 |
| 15 | | 25000 | 1 |
| 20 | | 81000 | 4 |
| 30 | | 188000 | 9 |
| | | 529000 | 23 |

Demo 14:

```

select dept_id
, year_hired
, sum(salary)
, count ('emp')
from adv_emp
group by grouping sets (dept_id, year_hired, ())
order by coalesce(dept_id, 999999), coalesce(year_hired, 9999)
;

```

| DEPT_ID | YEAR_HIRED | SUM(SALARY) | COUNT('EMP') |
|---------|------------|-------------|--------------|
| ----- | | | |
| 10 | | 235000 | 9 |
| 15 | | 25000 | 1 |
| 20 | | 81000 | 4 |
| 30 | | 188000 | 9 |

| | | |
|------|--------|----|
| 2010 | 113000 | 5 |
| 2012 | 179500 | 8 |
| 2013 | 83000 | 4 |
| 2014 | 153500 | 6 |
| | 529000 | 23 |