Note: This information was not created by me, but is very useful in understanding.

YouTube Video Version:
https://www.youtube.com/watch?v=bbkybT9qnTg

The Difference Between Rollup and Cube

By Ben Richardson, 2017/10/16

The GROUP BY clause is used to group the results of aggregate functions according to a specified column. However, the GROUP BY clause doesn't perform aggregate operations on multiple levels of a hierarchy. For example, you can calculate the total of all employee salaries for each department in a company (one level of hierarchy) but you cannot calculate the total salary of all employees regardless of the department they work in (two levels of hierarchy).

ROLLUP operators let you extend the functionality of GROUP BY clauses by calculating subtotals and grand totals for a set of columns. The CUBE operator is similar in functionality to the ROLLUP operator; however, the CUBE operator can calculate subtotals and grand totals for all permutations of the columns specified in it.

In this article, we will look at both the ROLLUP and CUBE operators with the help of a simple example. This will let us see the practical differences between the two and when we should use each of them.

Creating Dummy Data

Let's create some dummy data which we can then execute our example queries on. Create a new database called "company" and then run the code below to create an "employee" table.

```sql
USE company;

CREATE TABLE employee
(
    id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    gender VARCHAR(50) NOT NULL,
    salary INT NOT NULL,
    department VARCHAR(50) NOT NULL
)
```

Now that we have our database and table set up we need to populate it with some dummy data to work with.

```sql
INSERT INTO employee
VALUES
(1, 'David', 'Male', 5000, 'Sales'),
(2, 'Jim', 'Female', 6000, 'HR'),
(3, 'Kate', 'Female', 7500, 'IT'),
(4, 'Will', 'Male', 6500, 'Marketing'),
(5, 'Shane', 'Female', 5500, 'Finance'),
(6, 'Shed', 'Male', 8000, 'Sales'),
(7, 'Vik', 'Male', 7200, 'HR'),
(8, 'Vince', 'Female', 6600, 'IT'),
(9, 'Jane', 'Female', 5400, 'Marketing'),
(10, 'Laura', 'Female', 6300, 'Finance'),
(11, 'Mac', 'Male', 5700, 'Sales'),
(12, 'Pat', 'Male', 7000, 'HR'),
(13, 'Julie', 'Female', 7100, 'IT'),
(14, 'Elice', 'Female', 6800,'Marketing'),
(15, 'Wayne', 'Male', 5000, 'Finance')
```

Simple GROUP BY Clause

Let's start with a simple GROUP BY clause to calculate the sum of the salaries of all the employees grouped by their department.

```sql
SELECT department, sum(salary) as Salary_Sum
FROM employee
GROUP BY department
```

This will return the following:

Department	Salary_Sum
Finance	16800
HR	20200
IT	21200
Marketing	18700
Sales	18700

Here you can see the sum of the salaries of all employees grouped by their department. However, we cannot see the grand total, which is the sum of the salaries of all the employees belonging to all the departments in the company.

An alternative way to look at it is to say that the GROUP BY clause did not retrieve the total sum of the salaries of all the employees in the company. This is where the ROLLUP operator comes handy.

The ROLLUP Operator

As mentioned earlier, the ROLLUP operator is used to calculate sub-totals and grand totals for a set of columns passed to the "GROUP BY ROLLUP" clause.

Let's see how the ROLLUP clause helps us calculate the total salaries of the employees grouped by their departments and the grand total of the salaries of all the employees in the company. To do this we will work through a simple example query.

```sql
SELECT coalesce (department, 'All Departments') AS Department,
sum(salary) as Salary_Sum
FROM employee
GROUP BY ROLLUP (department)
```

In this code, we used the ROLLUP operator to calculate the grand total of the salaries of the employees from all the departments. However, for the grand total ROLLUP will return a NULL for department. To avoid this, we have used the "Coalesce" clause. This will replace NULL with the text "All Departments" and display the department name of each department in the Department column. For more details on using "Coalesce" see this article.

Department	Salary_Sum
Finance	16800
HR	20200
IT	21200
Marketing	18700
Sales	18700
All Departments	95600

Finding Subtotals Using ROLLUP Operator

The ROLLUP operator can also be used to calculate sub-totals for each column, based on the groupings within that column.

Let's look at an example where we want the sum of employee salaries at a department and gender level along with a sub-total along with a grand total for all salaries of all male and female employees belonging to all departments`.

```sql
SELECT
coalesce (department, 'All Departments') AS Department,
coalesce (gender,'All Genders') AS Gender,
sum(salary) as Salary_Sum
FROM employee
GROUP BY ROLLUP (department, gender)
```

This query returns the table below. As you can see it returns the sum of the salaries of the employees of each department divided into three categories: Male, Female and All Genders. The sub-totals are the lines with "All" in them. The last line is the grand total and so has an "All" in both columns.

NB: For the purposes of this article I have bolded the "All"s in the table below to make them stand out, this is not how they appear in reality.

Department	Gender	Salary_Sum
Finance	Female	11800
Finance	Male	5000
Finance	**All Genders**	16800
HR	Female	6000
HR	Male	14200
HR	**All Genders**	20200
IT	Female	21200
IT	**All Genders**	21200
Marketing	Female	12200
Marketing	Male	6500
Marketing	**All Genders**	18700
Sales	Male	18700
Sales	**All Genders**	18700
All Departments	**All Genders**	95600

Don't worry that you don't see total salaries for male employees in the IT department. This is because there aren't any, as is the case for female employees in the sales department.

The CUBE Operator

The CUBE operator is also used in combination with the GROUP BY clause, however the CUBE operator produces results by generating all combinations of columns specified in the GROUP BY CUBE clause.

Let's use it to find salaries grouped by department and gender. If we look at these two columns carefully we can see that there are four possible combinations by which we can group salary by department and gender. They are as follows:

1. Salary grouped by both department and gender
2. Salary grouped by gender only
3. Salary grouped by department only
4. Grand total of all salaries
5.

Execute the following script to see these four combinations in the result set.

```sql
SELECT
coalesce (department, 'All Departments') AS Department,
coalesce (gender,'All Genders') AS Gender,
sum(salary) as Salary_Sum
FROM employee
GROUP BY CUBE (department, gender)
```

The output of the above script is as follows:

Row No	Department	Gender	Salary_Sum
1	Finance	Female	11800
2	HR	Female	6000
3	IT	Female	21200
4	Marketing	Female	12200
5	All Departments	Female	51200
6	Finance	Male	5000
7	HR	Male	14200
8	Marketing	Male	6500
9	Sales	Male	18700
10	All Departments	Male	44400
11	All Departments	All Genders	95600
12	Finance	All Genders	16800
13	HR	All Genders	20200
14	IT	All Genders	21200
15	Marketing	All Genders	18700
16	Sales	All Genders	18700

Let's us find the four combinations by which salary is grouped in the above output.

NB: I have added the row numbers to make referencing records in the result set clear. They do not actually exist in the result set.

1. In the first four rows and from row 6 to row 9, salaries are grouped by both department and gender.
2. In the 5th and 10th row, the salaries are grouped by gender only i.e. Female and Male employees of all departments.
3. In the 11th row, we can see the grand total which is total of salaries of employees of all genders and all departments.
4. In the last five rows i.e. rows 12 to 16, salaries are grouped by department only.

So, we can see all the four combinations that we discussed earlier in the output above.

The Difference between ROLLUP and CUBE

There is only one major difference between the functionality of the ROLLUP operator and the CUBE operator. ROLLUP operator generates aggregated results for the selected columns in a hierarchical way. On the other hand, CUBE generates a aggregated result that contains all the possible combinations for the selected columns.

To understand this, look at the result set for the ROLLUP operator where the sum of the salaries of the employees were grouped by department and gender:

Row Number	Department	Gender	Salary_Sum
1	Finance	Female	11800
2	Finance	Male	5000
3	Finance	All Genders	16800
4	HR	Female	6000
5	HR	Male	14200
6	HR	All Genders	20200
7	IT	Female	21200
8	IT	All Genders	21200
9	Marketing	Female	12200
10	Marketing	Male	6500
11	Marketing	All Genders	18700
12	Sales	Male	18700
13	Sales	All Genders	18700
14	All Departments	All Genders	95600

Here data is aggregated in hierarchical manner. In rows 1, 2, 4, 5, 7, 9, 10 and 12, salaries are grouped by department and gender. In rows 3, 6, 8, 11 and 13, salaries are grouped by Department only.

Finally, in row 14 we have the grand total of the salaries of all of the employees of all genders from all departments. Here we have three combinations that are hierarchical in nature. They are as follows:

1. Department and Gender
2. Department
3. Grand Total

We do not have salary grouped by Gender only. This is because gender is lowest in hierarchy.

On the other hand, if you look at the aggregated result of the CUBE operator where the sum of the salaries of the employees were grouped by department and gender, we had all four possible combinations:

1. 1- Department and Gender
2. 2- Department only
3. 3- Gender Only
4. 4- Grand Total.

o

Note: It is important to mention here that the result of both the ROLLUP and the CUBE operators will be similar if your data is grouped by only one column.

Which One Should I Use?

ROLLUP and CUBE are performance tools. You should use ROLLUP if you want your data hierarchically and CUBE if you want all possible combinations.

For example, if you want to retrieve the total population of a country, state and city. ROLLUP would sum the population at three levels. First it would return the sum of population at Country-State-City Level. Then it would sum the population at Country-State level and finally it would sum the population at Country level. It would also provide a grand total level.

CUBE groups data in all possible combinations of columns so the population would be summed up in following levels:

1. Country-State-City
2. State-City
3. City
4. Country-State
5. State
6. Country-City
7. Country
8. All

It all depends what you need as to which you would choose. A simple rule of thumb is that if you have hierarchical data (for example, country->state->city or Department->Manager-Salesman, etc.), you usually want hierarchical results, and you use ROLLUP to group the data.

If you have non-hierarchical data (for example, City-Gender-Nationality), then you don't want hierarchical results and so you use CUBE as it will provide all possible combinations.

Keep up to date with SQL Server - new articles every day.

Sign up