

[Log in](#)[Create Account](#)

Avinash Navlani
January 30th, 2019

SQL +1

GROUP BY and HAVING Clause in SQL

In this tutorial, you will learn about the GROUP BY and HAVING Clause along with going over examples on how to put them to use.

An important component for Analyst to summarize the data such as sales, profit, cost, and salary. Data Summarization is very helpful for Analyst to create a visualization, conclude findings, and report writing. In SQL, GROUP BY Clause is one of the tools to summarize or aggregate the data series. For example, sum up the daily sales and combine in a single quarter and show it to the senior management. Similarly, if you want to count how many employees in each department of the company. It groups the databases on the basis of one or more column and aggregates the results.

After Grouping the data, you can filter the grouped record using HAVING Clause. HAVING Clause returns the grouped records which match the given condition. You can also sort the grouped records using ORDER BY. ORDER BY used after GROUP BY on aggregated column.

In this tutorial, you are going to learn GROUP BY Clause in detail with relevant examples. Here is the list of topics that you will learn in this tutorial:

- Group By Clause
- Having Clause

[Want to leave a comment?](#)

- GROUP BY With JOIN Example
- GROUP BY Comparison with other Clause
- Hands-on Practice Assignment
- Conclusion

Group By Clause

The GROUP BY Clause is utilized in SQL with the SELECT statement to organize similar data into groups. It combines the multiple records in single or more columns using some functions. Generally, these functions are aggregate functions such as min(),max(),avg(), count(), and sum() to combine into single or multiple columns. It uses the **split-apply-combine** strategy for data analysis.

- In the `split` phase , It divides the groups with its values.
- In the `apply` phase , It applies the aggregate function and generates a single value.
- In the `combiner` phase , It combines the groups with single values into a single value.

[Want to leave a comment?](#)

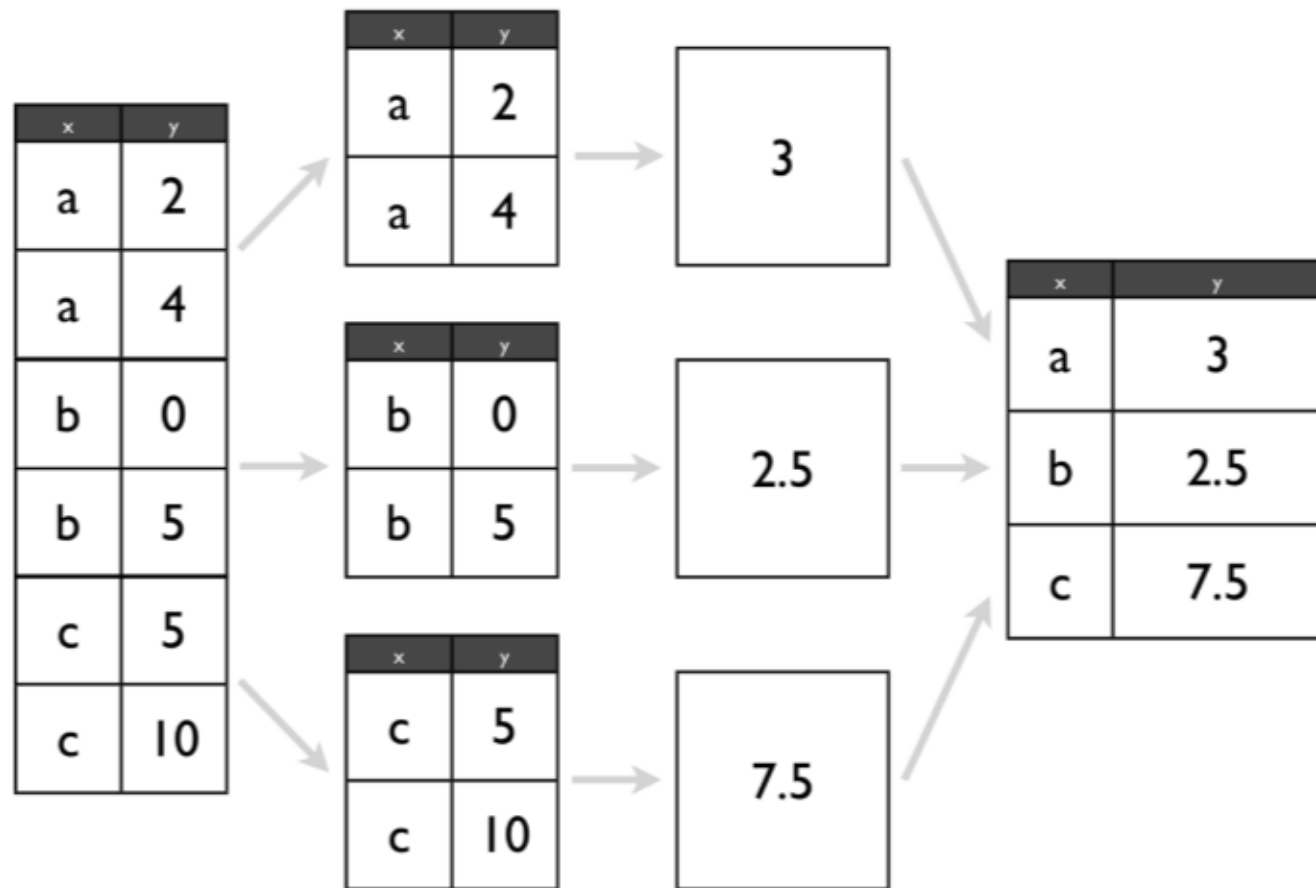


Image source

Points to Remember:

- GROUP BY Clause is utilized with the SELECT statement.
- GROUP BY aggregates the results on the basis of selected column: COUNT, MAX, MIN, SUM, AVG, etc.
- GROUP BY returns only one result per group of data.
- GROUP BY Clause always follows the WHERE Clause.
- GROUP BY Clause always precedes the ORDER BY Clause(<http://slideplayer.com/slide/15440670/>).

Want to leave a comment?

EmployeeID	Ename	DeptID	Salary
1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

```
SELECT DeptID, AVG(Salary)
FROM Employee
GROUP BY DeptID;
```

GROUP BY
Employee Table
using DeptID

DeptID	AVG(Salary)
1	3000.00
2	4000.00
3	4250.00

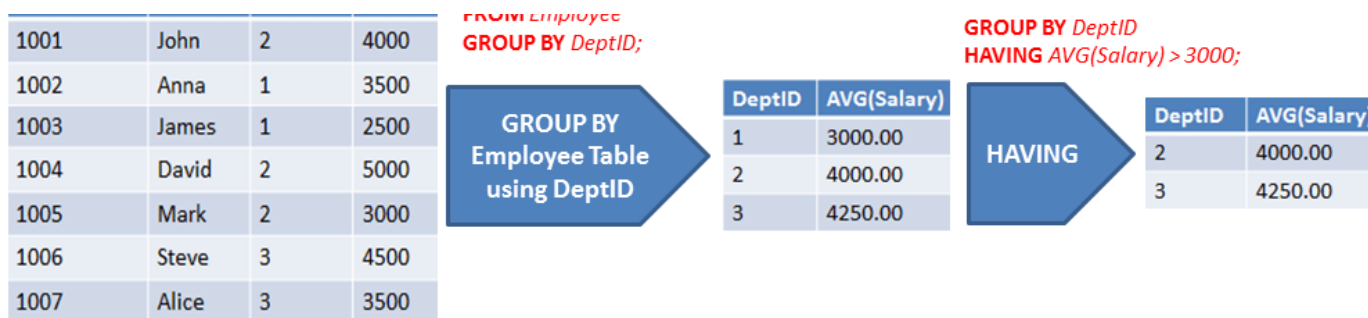
In above example, Table is grouped based on the DeptID column and Salary is aggregated department-wise.

Having Clause

HAVING Clause utilized in SQL as a conditional Clause with GROUP BY Clause. This conditional clause returns rows where aggregate function results matched with given conditions only. It added in the SQL because WHERE Clause cannot be combined with aggregate results, so it has a different purpose. The primary purpose of the WHERE Clause is to deal with non-aggregated or individual records.

- HAVING Clause always utilized in combination with GROUP BY Clause.
- HAVING Clause restricts the data on the group records rather than individual records.
- WHERE and HAVING can be used in a single query.

[Want to leave a comment?](#)



In above example, Table is grouped based on DeptID column and these grouped rows filtered using HAVING Clause with condition `AVG(Salary) > 3000`.

Aggregate Functions

Aggregate functions used to combine the result of a group into a single such as COUNT, MAX, MIN, AVG, SUM, STDDEV, and VARIANCE. These functions also known as multiple-row functions.

- `SUM()` : Returns the sum or total of each group.
- `COUNT()` : Returns the number of rows of each group.
- `AVG()` : Returns the average and mean of each group.
- `MIN()` : Returns the minimum value of each group.
- `MAX()` : Returns the maximum value of each group.

Compare Having and Where Clause in SQL

- In some cases, you need to filter out the individual records. In such cases, you can use WHERE Clause, Whereas in other cases you need to filter the groups with the specific condition. In such cases, you can use HAVING Clause.
- WHERE Clause filters the records tuple by tuple while HAVING Clause filters the whole

[Want to leave a comment?](#)

- WHERE Clause restricts records before GROUP BY Clause, whereas HAVING Clause restricts groups after GROUP BY Clause are performed.
- WHERE Clause can be utilized with SELECT, UPDATE, DELETE, and INSERT, whereas HAVING can be utilized only with SELECT statement.

Sno	Where Clause	Having Clause
1	The WHERE clause specifies the criteria which individual records must meet to be selected by a query. It can be used without the GROUP BY clause	The HAVING clause cannot be used without the GROUP BY clause.
2	The WHERE clause selects rows before grouping.	The HAVING clause selects rows after grouping.
3	The WHERE clause cannot contain aggregate functions	The HAVING clause can contain aggregate functions.
4	WHERE clause is used to impose condition on SELECT statement as well as single row function and is used before GROUP BY clause	HAVING clause is used to impose condition on GROUP Function and is used after GROUP BY clause in the query
5	SELECT Column,AVG(Column_name)FROM Table_name WHERE Column > value GROUP BY Column_name	SELECT Column, AVG(Column_name)FROM Table_name WHERE Column > value GROUP BY Column_name Having column_name>or<value

[Image Source](#)

GROUP BY With JOIN Example

The normalized relational database breaks down the complex table into small tables, which helps you to eliminate the data redundancy, inconsistency and ensure there is no loss of information. Normalized tables require joining data from multiple tables.

[Want to leave a comment?](#)

1001	John	2	4000
1002	Anna	1	3500
1003	James	1	2500
1004	David	2	5000
1005	Mark	2	3000
1006	Steve	3	4500
1007	Alice	3	3500

Department

DeptID	Dname	Dlocation
1	HR	Mumbai
2	IT	New Delhi
3	Finance	Mumbai



JOIN using
Common
Column

**FROM Employee JOIN Department
USING(DeptID)**

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai
1007	Alice	3	3500	Finance	Mumbai

In above example, Employee and Department are joined using the common column DeptID.

SELECT *
FROM Employee JOIN Department
USING(DeptID)

EmployeeID	Ename	DeptID	Salary	Dname	Dlocation
1001	John	2	4000	IT	New Delhi
1002	Anna	1	3500	HR	Mumbai
1003	James	1	2500	HR	Mumbai
1004	David	2	5000	IT	New Delhi
1005	Mark	2	3000	IT	New Delhi
1006	Steve	3	4500	Finance	Mumbai
1007	Alice	3	3500	Finance	Mumbai

SELECT Dname, AVG(Salary)
FROM Employee JOIN Department
USING(DeptID)
GROUP BY Dname



GROUP BY
using Dname

Dname	AVG(Salary)
HR	3000.00
IT	4000.00
Finance	4250.00

In the above example, JOIN and GROUP BY both clauses used together in a single query. After joining both tables(Employee and Department), joined table grouped by Department name.

GROUP BY Comparison with Other Clause

Want to leave a comment?

see the DISTINCT values in the dept table.

```
mysql> SELECT DISTINCT dept_id
-> FROM dept;
+-----+
| dept_id |
+-----+
|      1  |
|      2  |
|      3  |
+-----+
```

Compare GROUP BY and ORDER BY

ORDER BY returns sorted items in ascending and descending order while GROUP BY returns unique items with the aggregate resultant column. In the following example, you can see the ORDER BY or sorted salary table.

```
mysql> SELECT *
-> FROM emp
-> ORDER BY salary;
+-----+-----+-----+-----+-----+
| eid | ename | dept_id | salary | bonus |
+-----+-----+-----+-----+-----+
| 103 | michell | 1 | 2500 | 400 |
| 105 | mark | 2 | 3000 | 1000 |
| 102 | anna | 1 | 3500 | 600 |
| 101 | john | 2 | 4000 | 0 |
| 107 | harshal | 3 | 4000 | 100 |
| 106 | steve | 3 | 4500 | 0 |
| 104 | david | 2 | 5000 | 500 |
+-----+-----+-----+-----+-----+
```

Hands-on Practice Assignment

Table Name: Books

[Want to leave a comment?](#)

1. Determine how many books are in each category.
2. Determine how many books are in the Management category.
3. Determine the average book price of each category.
4. List the price of the least expensive book in each category.

Source: *This Assignment is inspired from the book "Oracle 11g SQL" by John Casteel.*

Conclusion

Congratulations, you have made it to the end of this tutorial!

In this tutorial, you have covered a lot of details about the GROUP BY and HAVING Clause. You have learned what the GROUP BY and HAVING Clause are with examples, Comparison between HAVING and WHERE Clause in SQL, GROUP BY with JOIN, and GROUP BY Comparison with DISTINCT and ORDER BY. In the last section, you have a Hands-on practice assignment to assess your knowledge.

Hopefully, you can now utilize GROUP BY and HAVING Clause concept to analyze your own datasets. Thanks for reading this tutorial!

If you are interested in learning more about SQL, take DataCamp's [Intermediate SQL](#) course.

▲
29

💬
2



COMMENTS

[Want to leave a comment?](#)

Vivek Kumar

04/07/2019 03:31 PM

I was so confident in this game performance, yes it is surly make my day in exciting. [play freecell](#) on your gadget in online and create highest score in this game.

▲ 1 ↩ REPLY

 [Subscribe to RSS](#)



[About](#) [Terms](#) [Privacy](#)

[Want to leave a comment?](#)