**MySQL Group by Having Clause**

SQL GROUP BY clause

The GROUP BY is an optional clause of the [SELECT](https://www.sqltutorial.org/sql-select/) statement. The GROUP BY clause allows you to group rows based on values of one or more columns. It returns one row for each group.

The following shows the basic syntax of the GROUP BY clause:

SELECT

column1,

column2,

aggregate\_function(column3)

FROM

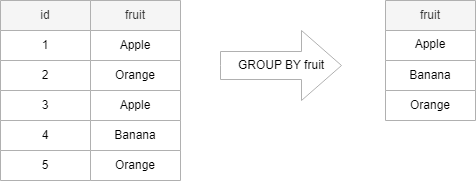
table\_name

GROUP BY

column1,

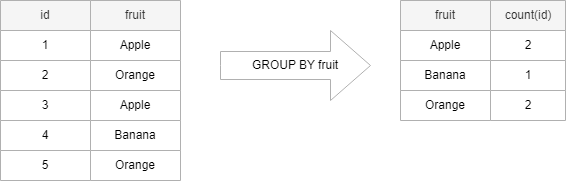
column2;

**The following picture illustrates shows how the GROUP BY clause works:**



In practice, you often use the GROUP BY clause with an [aggregate function](https://www.sqltutorial.org/sql-aggregate-functions/) such as [MIN](https://www.sqltutorial.org/sql-aggregate-functions/sql-min/), [MAX](https://www.sqltutorial.org/sql-aggregate-functions/sql-max/), [AVG](https://www.sqltutorial.org/sql-aggregate-functions/sql-avg/), [SUM](https://www.sqltutorial.org/sql-aggregate-functions/sql-sum/), or [COUNT](https://www.sqltutorial.org/sql-aggregate-functions/sql-count/) to calculate a measure that provides the information for each group.

For example, the following illustrates how the GROUP BY clause works with the COUNT aggregate function:



SELECT

fruit, COUNT(id)

FROM

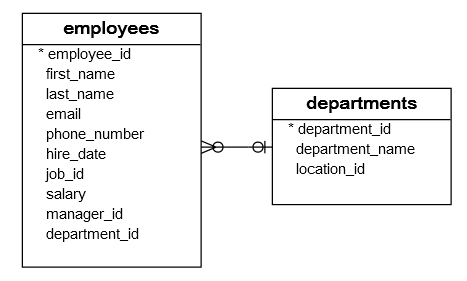
sample\_table

GROUP BY

fruit;

The columns that appear in the GROUP BY clause are called *grouping columns*. If a grouping column contains [NULL](https://www.sqltutorial.org/sql-is-null/) values, all NULL values are summarized into a single group because the GROUP BY clause considers all NULL values equal.

Use the employees and departments tables in the [sample database](https://www.sqltutorial.org/sql-sample-database/) to demonstrate how the GROUP BY clause works.



SELECT department\_id

FROM employees

GROUP BY department\_id;

**In this example:**

* First, the SELECT clause returns all values from the department\_id column of employees table.
* Second, the GROUP BY clause groups all values into groups.
* The department\_id column of the employees table has 40 rows, including duplicate department\_id values. However, the GROUP BY groups these values into groups.
* Without an aggregate function, the GROUP BY behaves like the DISTINCT keyword:

## SQL HAVING clause

1. To specify a condition for groups, you use the HAVING clause.
2. The HAVING clause is often used with the GROUP BY clause in the [SELECT statement](https://www.sqltutorial.org/sql-select/). If you use a HAVING clause without a GROUP BY clause, the HAVING clause behaves like the [WHERE clause](https://www.sqltutorial.org/sql-where/).
3. The following illustrates the syntax of the HAVING clause:

SELECT

column1,

column2,

AGGREGATE\_FUNCTION (column3)

FROM

table1

GROUP BY

column1,

column2

HAVING

group\_condition;

## HAVING vs. WHERE

The [WHERE](https://www.sqltutorial.org/sql-where/) clause applies the condition to individual rows before the rows are summarized into groups by the GROUP BY clause. However, the HAVING clause applies the condition to the groups after the rows are grouped into groups.

Therefore, it is important to note that the HAVING clause is applied after whereas the WHERE clause is applied before the GROUP BY clause.

## SQL HAVING clause examples



1. **To get the managers and their direct reports,**

**you use the GROUP BY clause to group employees by the managers and use the**[**COUNT function**](https://www.sqltutorial.org/sql-aggregate-functions/sql-count/)**to count the direct reports.**

SELECT

manager\_id,

first\_name,

last\_name,

COUNT(employee\_id) direct\_reports

FROM

employees

WHERE

manager\_id IS NOT NULL

GROUP BY manager\_id;

1. To find the managers who have at least five direct reports, you add a HAVING clause to the query above as the following:

SELECT

manager\_id,

first\_name,

last\_name,

COUNT(employee\_id) direct\_reports

FROM

employees

WHERE

manager\_id IS NOT NULL

GROUP BY manager\_id

HAVING direct\_reports >= 5;

### **SQL HAVING with SUM function example**

1. The following statement calculates the [sum](https://www.sqltutorial.org/sql-aggregate-functions/sql-sum/) of salary that the company pays for each department and selects only the departments with the sum of salary [between](https://www.sqltutorial.org/sql-between/) 20000 and 30000.

SELECT

department\_id, SUM(salary)

FROM

employees

GROUP BY department\_id

HAVING SUM(salary) BETWEEN 20000 AND 30000

ORDER BY SUM(salary);

Ques-1 To find the department that has employees with the lowest salary greater than 10000

Ques-2 To find the departments that have the average salaries of employees between 5000 and 7000