MySQL HAVING

**Here**, you will learn how to use MySQL HAVING clause to specify a filter condition for groups of rows or aggregates.

## **Introduction to MySQL HAVING clause**

The HAVING clause is used in the SELECT statement to specify filter conditions for a group of rows or aggregates.

The HAVING clause is often used with the GROUP BY clause to filter groups based on a specified condition. If you omit the GROUP BY clause, the HAVING clause behaves like the WHERE clause.

The following illustrates the syntax of the HAVING clause:

SELECT

select\_list

FROM

table\_name

WHERE

search\_condition

GROUP BY

group\_by\_expression

HAVING

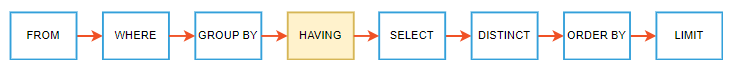
group\_condition;

In this syntax, you specify a condition in the HAVING clause.

The HAVING clause evaluates each group returned by the GROUP BY clause. If the result is true, the row is included in the result set.

Notice that the HAVING clause applies a filter condition to each group of rows, while the WHERE clause applies the filter condition to each individual row.

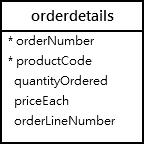
MySQL evaluates the HAVING clause after the FROM, WHERE, SELECT and GROUP BY clauses and before ORDER BY, and LIMIT clauses:



Note that the SQL standard specifies that the HAVING is evaluated before SELECT clause and after GROUP BY clause.

## **MySQL HAVING clause examples**

Let’s take some examples of using the HAVING clause to see how it works. We’ll use the orderdetails table in the sample database for the demonstration.



The following uses the GROUP BY clause to get order numbers, the number of items sold per order, and total sales for each from the orderdetails table:

SELECT

ordernumber,

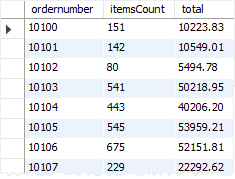
SUM(quantityOrdered) AS itemsCount,

SUM(priceeach\*quantityOrdered) AS total

FROM

orderdetails

GROUP BY ordernumber;



Now, you can find which order has total sales greater than 50000 by using the HAVING clause as follows:

SELECT

ordernumber,

SUM(quantityOrdered) AS itemsCount,

SUM(priceeach\*quantityOrdered) AS total

FROM

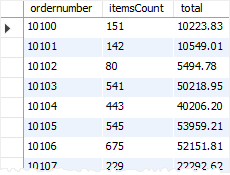
orderdetails

GROUP BY

ordernumber

HAVING

total > 50000;



It’s possible to form a complex condition in the HAVING clause using logical operators such as OR and AND.

The following example uses the HAVING clause to find orders that have total amounts greater than 1000 and contain more than 600 items:

SELECT

ordernumber,

SUM(quantityOrdered) AS itemsCount,

SUM(priceeach\*quantityOrdered) AS total

FROM

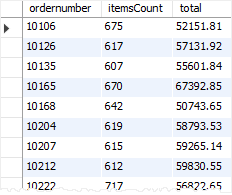
orderdetails

GROUP BY ordernumber

HAVING

total > 1000 AND

itemsCount > 600;



Suppose that you want to find all orders that already shipped and have a total amount greater than 1500, you can join the orderdetails table with the orders table using the INNER JOIN clause and apply a condition on status column and total aggregate as shown in the following query:

SELECT

a.ordernumber,

status,

SUM(priceeach\*quantityOrdered) total

FROM

orderdetails a

INNER JOIN orders b

ON b.ordernumber = a.ordernumber

GROUP BY

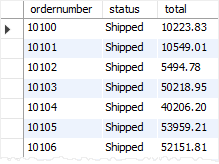
ordernumber,

status

HAVING

status = 'Shipped' AND

total > 1500;



The HAVING clause is only useful when you use it with the GROUP BY clause to generate the output of the high-level reports. For example, you can use the HAVING clause to answer the questions like finding the number of orders this month, this quarter, or this year that have a total amount greater than 10K.

## **Summary**

* Use the MySQL HAVING clause with the GROUP BY clause to specify a filter condition for groups of rows or aggregates.