# MySQL - Having Clause

## MySQL Having Clause

The MySQL HAVING Clause is used to filter grouped rows in a table based on conditions.

This clause is used with the GROUP BY clause to group the rows based on one or more columns and then filter them based on the conditions specified in the HAVING clause. So, the HAVING clause must always be followed by the GROUP BY clause.

The HAVING clause was added to MySQL because the WHERE keyword cannot be used with aggregate functions such as COUNT(), SUM(), AVG(), etc.

This clause is similar to the MySQL WHERE clause. The difference between both of them is that the WHERE clause filters individual rows in a table, whereas the HAVING clause filters grouped rows based on conditions.

### Syntax

**Following is the basic syntax of the HAVING clause in MySQL –**

```
SELECT column1, column2, aggregate_function(column)
FROM table_name
GROUP BY column1, column2, ...
HAVING condition
ORDER BY column1, column2, ...;
```

### Example

**Let us begin with creating a table named CUSTOMERS using the following query –**

```
CREATE TABLE CUSTOMERS (
   ID INT NOT NULL,
   NAME VARCHAR (20) NOT NULL,
   AGE INT NOT NULL,
   ADDRESS CHAR (25),
   SALARY DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

```
INSERT INTO CUSTOMERS (ID,NAME,AGE,ADDRESS,SALARY) VALUES
(1, 'Ramesh', 32, 'Ahmedabad', 2000.00 ),
(2, 'Khilan', 25, 'Delhi', 1500.00 ),
(3, 'Kaushik', 23, 'Kota', 2000.00 ),
(4, 'Chaitali', 25, 'Mumbai', 6500.00 ),
(5, 'Hardik', 27, 'Bhopal', 8500.00 ),
(6, 'Komal', 22, 'Hyderabad', 4500.00 ),
(7, 'Muffy', 24, 'Indore', 10000.00 );
```

```
SELECT * FROM CUSTOMERS;
```

**HAVING clause with ORDER BY clause**

In MySQL, the HAVING clause filters the groups, and the ORDER BY clause sorts the results. When we used both of them together, HAVING is executed first, then the result set is sorted according to the ORDER BY criteria.

**Example**

In the following query, we are retrieving all the records from the CUSTOMERS table where the sum of their SALARY is less than 4540, ordered by their name in ascending order –

```
SELECT NAME, SUM(SALARY) as total_salary
FROM CUSTOMERS
GROUP BY NAME
HAVING SUM(SALARY) < 4540
ORDER BY NAME;
```

**HAVING clause with COUNT() function**

We can use the MySQL HAVING clause in conjunction with the COUNT() function to filter the groups based on the number of rows they contain.

**Example**

In this query, we are fetching a record where the count of similar age is greater than or equal to 2.

```
SELECT AGE
FROM CUSTOMERS
GROUP BY age
HAVING COUNT(age) >= 2;
```

**HAVING clause with AVG() function**

The MySQL HAVING clause can also be used with the **AVG()** function to filter groups based on the average value of a specified column.

**Example**

In the following query, we are trying to return the names of the customers whose salary is greater than 3000 –

```
SELECT NAME, AVG(salary) as avg_salary
FROM customers
GROUP BY NAME
HAVING AVG(salary) > 3000;
```

**HAVING clause with MAX() function**

In MySQL, we can also use the HAVING clause with **MAX()** function to filter groups based on the maximum value of a specified column.

**Example**

In this query, we are retrieving the customer names whose maximum SALARY is less than 4000 –

```
SELECT NAME, MAX(salary) as max_salary
FROM customers
GROUP BY NAME
HAVING MAX(salary) < 4000;
```