SQL Introduction
SQL SELECT (I)
SQL SELECT (II)
- SQL ORDER BY
- SQL GROUP BY
- SQL LIKE
- SQL Wildcards
- SQL UNION
- SQL Subquery
- SQL ANY and ALL
- SQL CASE
- SQL HAVING
- SQL EXISTS

SQL JOIN
SQL DATABASE & TABLE
SQL Insert, Update and Delete
SQL Constraints
SQL Additional Topics

SQL HAVING Clause

In this tutorial, we'll learn about the HAVING clause in SQL and how to use them with examples.

The `HAVING` clause in SQL is used if we need to filter the result set based on aggregate functions such as MIN() and MAX(), SUM() and AVG() and COUNT().

```
SELECT COUNT(customer_id), country
FROM Customers
GROUP BY country
HAVING COUNT(customer_id) > 1;
```

Run Code »

Here, the SQL command:

1. counts the number of rows by grouping them by `country`
2. returns the result set if their **count** is greater than **1**.

Table: Customers

customer_id	first_name	last_name	age	country
1	John	Doe	31	USA
2	Robert	Luna	22	USA
3	David	Robinson	22	UK
4	John	Reinhardt	25	UK
5	Betty	Doe	28	UAE

```
SELECT COUNT(customer_id), country
FROM Customers
GROUP BY country
HAVING COUNT(customer_id) > 1;
```

COUNT(customer_id)	country
2	UK
2	USA

Example: HAVING in SQL

Note: The `HAVING` clause was introduced because the `WHERE` clause does not support aggregate functions. Also, `GROUP BY` must be used before the `HAVING` clause. To learn more, visit SQL GROUP BY.

SQL HAVING Vs WHERE

HAVING Clause	WHERE Clause
The `HAVING` clause checks the condition on **a group of rows**.	The `WHERE` clause checks the condition **on each individual row**.
The `HAVING` is used with aggregate functions.	The `WHERE` clause cannot be used with aggregate functions.
The `HAVING` clause is executed after the `GROUP BY` clause.	The `WHERE` clause is executed before the `GROUP BY` clause.

Let's take a look at an example,

If we want to select rows where the value of the `amount` column in the `Orders` table is less than **500**, we can write,

```
SELECT customer_id, amount
FROM Orders
WHERE amount < 500;
```

Run Code »

Now, if we want to select rows and calculate sum off each amount, we can write,

```
SELECT customer_id, SUM(amount) AS total
FROM Orders
GROUP BY customer_id;
```

Run Code »

Thats it.

But if we need to select rows if the sum of amounts is less than 500 for any customer, we need to write,

```
SELECT customer_id, SUM(amount) AS total
FROM Orders
GROUP BY customer_id
HAVING SUM(amount) < 500;
```

Run Code »

Related Topics
SQL GROUP BY
SQL SELECT INTO Statement
SQL SUM() AND AVG()
SQL COUNT()
SQL INSERT INTO SELECT Statement
SQL CASE

Previous Tutorial: SQL CASE
Next Tutorial: SQL EXISTS

Related Tutorials

Programming — SQL GROUP BY
Programming — SQL SUM() AND AVG()
Programming — SQL SELECT INTO Statement
Programming — SQL COUNT()