

# HAVING CLAUSE

The HAVING clause was introduced in SQL to allow the filtering of query results based on aggregate functions and groupings, which cannot be achieved using the WHERE clause that is used to filter individual rows.

In simpler terms MSSQL, the HAVING clause is used to apply a filter on the result of GROUP BY based on the specified condition. The conditions are Boolean type i.e. use of logical operators (AND, OR). This clause was included in SQL as the WHERE keyword failed when we use it with aggregate expressions. Having is a very generally used clause in SQL. Similar to WHERE it helps to apply conditions, but HAVING works with groups. If you wish to filter a group, the HAVING clause comes into action.

## Some important points

- + Having clause is used to filter data according to the conditions provided.
- + Having a clause is generally used in reports of large data.
- + Having clause is only used with the SELECT clause.
- + The expression in the syntax can only have constants.
- + In the query, ORDER BY is to be placed after the HAVING clause, if any.
- + HAVING Clause is implemented in column operation.
- + Having clause is generally used after GROUP BY.

## Syntax

```
SELECT col_1, function_name(col_2)
FROM tablename
WHERE condition
GROUP BY column1, column2
HAVING Condition
ORDER BY column1, column2;
```

Here, the function\_name is the name of the function used, for example, SUM(), and AVG().

## Example 1

Here first we create a database name as “Company”, then we will create a table named “Employee” in the database. After creating a table we will execute the query.

### Step 1: Creating a database

```
CREATE DATABASE Company;
```

### Step 2: To use this database

```
USE Company;
```

### Step 3: Creating a table

```
CREATE TABLE Employee (  
    EmployeeId int,  
    Name varchar(20),  
    Gender varchar(20),  
    Salary int,  
    Department varchar(20),  
    Experience varchar(20)  
);
```

### Step 4: Add value to the table

```
INSERT INTO Employee (EmployeeId, Name, Gender, Salary,  
Department, Experience)  
VALUES (5, 'Priya Sharma', 'Female', 45000, 'IT', '2  
years');  
INSERT INTO Employee (EmployeeId, Name, Gender, Salary,  
Department, Experience)  
VALUES (6, 'Rahul Patel', 'Male', 65000, 'Sales', '5  
years');  
INSERT INTO Employee (EmployeeId, Name, Gender, Salary,  
Department, Experience)  
VALUES (7, 'Nisha Gupta', 'Female', 55000, 'Marketing', '4  
years');  
INSERT INTO Employee (EmployeeId, Name, Gender, Salary,  
Department, Experience)  
VALUES (8, 'Vikram Singh', 'Male', 75000, 'Finance', '7  
years');  
INSERT INTO Employee (EmployeeId, Name, Gender, Salary,  
Department, Experience)  
VALUES (9, 'Aarti Desai', 'Female', 50000, 'IT', '3 years');
```

The final table is:

EmployeeId	Name	Gender	Salary	Department	Experience
5	Priya Sharma	Female	45000	IT	2 years
6	Rahul Patel	Male	65000	Sales	5 years
7	Nisha Gupta	Female	55000	Marketing	4 years
8	Vikram Singh	Male	75000	Finance	7 years
9	Aarti Desai	Female	50000	IT	3 years

### Step 5: Execute the query

This employee table will help us understand the HAVING Clause. It contains employee IDs, Name, Gender, department, and salary. To Know the sum of salaries, we will write the query.

```
SELECT Department, sum(Salary) AS Salary
FROM employee
GROUP BY department;
```

### Output

Department	Salary
Finance	75000
IT	95000
Marketing	55000
Sales	65000

Now if we need to display the departments where the sum of salaries is 50,000 or more. In this condition, we will use the HAVING Clause.

```
SELECT Department, sum(Salary) AS Salary
FROM employee
GROUP BY department
HAVING SUM= 50000;
```

### Output

Department	Salary
Finance	75000
IT	95000
Marketing	55000
Sales	65000

## Example 2

Suppose, a teacher wants to announce the toppers in class. For this, she decides to reward every student who scored more than 95%. We need to group the database by name and their percentage and find out who scored more than 95% in that year. So for this first, we create a database name “School”, and then we will create a table named “Student” in the database. After creating a table we will execute the query.

### Step 1: Creating a database

```
CREATE DATABASE School;
```

### Step 2: To use this database

```
USE School;
```

### Step 3: Creating a table

```
CREATE TABLE Student(  
    student Varchar(20) ,  
    percentage int  
);
```

Step 4: Add value to the table

```
INSERT INTO Student VALUES ('Isha Patel', 98)  
INSERT INTO Student VALUES ('Harsh Das', 94)  
INSERT INTO Student VALUES ('Rachit Sha', 93)  
INSERT INTO Student VALUES ('Sumedha', 98)  
INSERT INTO Student VALUES ('Rahat Ali', 98)
```

The final table is:

Results Messages		
	student ▾	percentage ▾
1	Isha Patel	98
2	Harsh Das	94
3	Rachit Sha	93
4	Sumedha	98
5	Rahat Ali	98

Step 5: Execute the query

```
SELECT student, percentage  
FROM Student  
GROUP BY student, percentage  
HAVING percentage > 95;
```

Here, three students named Isha, Sumedha, and Rahat Ali scored more than 95%.

Output

## Results Messages

	student ▾	percentage ▾
1	Isha Patel	98
2	Rahat Ali	98
3	Sumedha	98

Further, we can also filter rows on multiple values using the HAVING clause. The HAVING clause also permits filtering rows using more than one aggregate condition.

### Query

```
SELECT student
FROM Student
WHERE percentage > 90
GROUP BY student, percentage
HAVING SUM(percentage) < 1000 AND AVG(percentage) > 95;
```

This query returns the students who have more percentage than 95 and the sum of percentage is less than 1000.

### Output

## Results

## Messages

	student ▼
1	Isha Patel
2	Rahat Ali
3	Sumedha

## SQL Having vs WHERE

Having	Where
In the HAVING clause it will check the condition in group of a row.	In the WHERE condition it will check or execute at each row individual.
HAVING clause can only be used with aggregate function.	The WHERE Clause cannot be used with aggregate function like Having.
Priority Wise HAVING Clause is executed after Group By.	Priority Wise WHERE is executed before Group By.