Introduction to SQL Server HAVING clause

The HAVING clause is often used with the [GROUP BY](https://www.sqlservertutorial.net/sql-server-basics/sql-server-group-by/) clause to filter groups based on a

specified list of conditions. The following illustrates the HAVING clause syntax:

SELECT column\_name(s)  
FROM table\_name  
WHERE condition  
GROUP BY column\_name(s)HAVING conditionORDER BY column\_name(s);

**Key Points**

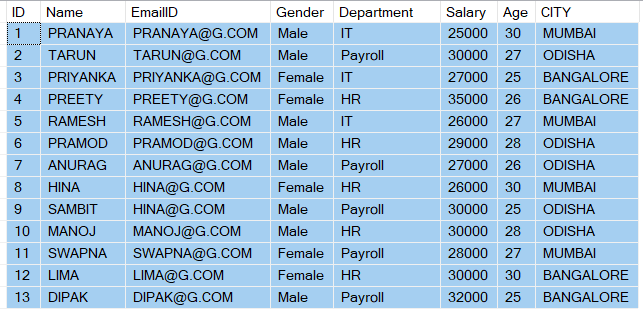
* HAVING Clause can only be used with a SELECT Statement.
* HAVING Clause is used to filter records from the groups. This means it is used to filter groups.
* HAVING Clause implements in column operations.
* HAVING Clause is used after GROUP BY Clause.
* HAVING Clause can have aggregate functions.
* The HAVING clause is slower than the WHERE clause and should be avoided whenever possible.

##### **Parameters or Arguments of Having Clause in SQL Server:**

1. **aggregate\_function:**It can be any of the aggregate functions such as SUM, COUNT, MIN, MAX, or AVG.
2. **expression1, expression2, expression\_n:**The expressions which are not encapsulated within an aggregate function must be included in the GROUP BY clause.
3. **Where Conditions:**It is optional. If you want to retrieve selected records based on some conditions then you need to specify the conditions using the Where clause in SQL Server.
4. **HAVING having\_condition:**The Having Clause Condition is used to add a

##### **Database tables used in this demo:**

We are going to use the following Employee table in this demo.



**Please use the below script to create and populate the Employee table with some dummy data.**

-- Create Person table

CREATE TABLE Employee

**(**

ID INT PRIMARY KEY IDENTITY**(**1,1**)**,

Name VARCHAR**(**100**)**,

EmailID VARCHAR**(**100**)**,

Gender VARCHAR**(**100**)**,

Department VARCHAR**(**100**)**,

Salary INT,

Age INT,

CITY VARCHAR**(**100**)**

**)**

GO

--Insert some test data **into** Person table

INSERT INTO Employee VALUES**(**'PRANAYA','PRANAYA@G.COM','Male', 'IT', 25000, 30,'MUMBAI'**)**

INSERT INTO Employee VALUES**(**'TARUN','TARUN@G.COM','Male', 'Payroll', 30000, 27,'ODISHA'**)**

INSERT INTO Employee VALUES**(**'PRIYANKA','PRIYANKA@G.COM','Female', 'IT', 27000, 25,'BANGALORE'**)**

INSERT INTO Employee VALUES**(**'PREETY','PREETY@G.COM','Female', 'HR', 35000, 26,'BANGALORE'**)**

INSERT INTO Employee VALUES**(**'RAMESH','RAMESH@G.COM','Male','IT', 26000, 27,'MUMBAI'**)**

INSERT INTO Employee VALUES**(**'PRAMOD','PRAMOD@G.COM','Male','HR', 29000, 28,'ODISHA'**)**

INSERT INTO Employee VALUES**(**'ANURAG','ANURAG@G.COM','Male', 'Payroll', 27000, 26,'ODISHA'**)**

INSERT INTO Employee VALUES**(**'HINA','HINA@G.COM','Female','HR', 26000, 30,'MUMBAI'**)**

INSERT INTO Employee VALUES**(**'SAMBIT','HINA@G.COM','Male','Payroll', 30000, 25,'ODISHA'**)**

INSERT INTO Employee VALUES**(**'MANOJ','MANOJ@G.COM','Male','HR', 30000, 28,'ODISHA'**)**

INSERT INTO Employee VALUES**(**'SWAPNA','SWAPNA@G.COM','Female', 'Payroll', 28000, 27,'MUMBAI'**)**

INSERT INTO Employee VALUES**(**'LIMA','LIMA@G.COM','Female','HR', 30000, 30,'BANGALORE'**)**

INSERT INTO Employee VALUES**(**'DIPAK','DIPAK@G.COM','Male','Payroll', 32000, 25,'BANGALORE'**)**

GO

##### **Filtering Groups in SQL Server:**

The Where clause in SQL Server is used to filter the rows before aggregation, whereas the Having clause in SQL Server is used to filter the groups that mean after aggregations. The following 2 queries produce the same result.

###### **Filtering rows using WHERE clause**

SELECT City, SUM**(**Salary**)** **as** TotalSalary

FROM Employee

WHERE City = 'MUMBAI'

GROUP BY City

###### **Filtering groups using the HAVING clause, after all, aggregations take place.**

SELECT City, SUM**(**Salary**)** **as** TotalSalary

FROM Employee

GROUP BY City

HAVING City = 'MUMBAI'

From the performance point of view, we cannot say or we cannot give a guarantee that one method is more efficient than the other one. The SQL Server optimizer checks or analyzes each statement and then selects an efficient way of executing the query. So, try to eliminate the records as soon as possible which you don’t want in your result set.

###### **It is also possible to combine WHERE and HAVING**

SELECT City, SUM**(**Salary**)** **as** TotalSalary

FROM Employee

WHERE Gender = 'Male'

GROUP BY City

HAVING City = 'MUMBAI'

##### **Having Clause Using SUM Function:**

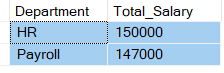
The below example uses the SUM function to return the name of the department and the total Salary (associated with the department). The Having Clause will filter the results so that only departments with a total Salary greater than 140000 will be returned.

SELECT Department, SUM**(**Salary**)** AS Total\_Salary

FROM Employee

GROUP BY Department

HAVING SUM**(**Salary**)** **>** 140000;



##### **Having Clause Using COUNT Function:**

The below example uses the COUNT function to return the city and the number of employees (residing in that city). The Having Clause will filter the results so that only cities with more than 4 employees will be returned.

SELECT City, COUNT**(**\***)** AS 'Number of employees'

FROM Employee

GROUP BY City

HAVING COUNT**(**\***)** **>** 4;

Having Clause Using COUNT Function

##### **Having Clause Using MIN Function:**

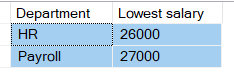
The below example uses the MIN function to return the name of each department and the minimum salary in the department. The Having Clause will return only those departments where the minimum salary is greater than 25000.

SELECT Department, MIN**(**Salary**)** AS 'Lowest salary'

FROM Employee

GROUP BY Department

HAVING MIN**(**Salary**)** **>** 25000;



##### **Having Clause Using MAX Function:**

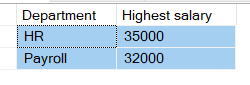
The below example uses the MAX function to return the name of each department and the maximum salary in each department. The Having Clause will return only those departments where the maximum salary is greater than 27000.

SELECT Department, MAX**(**Salary**)** AS 'Highest salary'

FROM Employee

GROUP BY Department

HAVING MAX**(**Salary**)** **>** 27000;



##### **Having Clause Using AVG Function:**

The below example uses the AVG function to return the name of each department and the Average salary of each department. Having clause will return only those departments where the average salary is greater than 27000.

SELECT Department, AVG**(**Salary**)** AS 'Average salary'

FROM Employee

GROUP BY Department

HAVING AVG**(**Salary**)** **>** 27000;

