SQL for Citizen Data Scientists

SQL HAVING Clause - Learn By Example

SETScholars & WACAMLDS

In addition to creating groups using GROUP BY clause, you can also decide which groups to include in the output and which to exclude. For example, you might want a list of jobs for which more than one employee is hired. To get this kind of data you have to filter by group and not by individual rows.

SQL provides a clause for this purpose: **the HAVING clause**.

Just as WHERE is to SELECT, so is HAVING to GROUP BY. In other words, the WHERE clause filters individual rows, and the HAVING clause filters the groups created by the GROUP BY clause. So when you use the HAVING clause, you effectively include or exclude whole groups of data from the query results.

Syntax

The syntax of HAVING is similar to WHERE; just the keyword is different. And of course HAVING is used only in conjunction with the GROUP BY clause.

```
SELECT column_name(s)
FROM table_name
WHERE condition
GROUP BY column_name(s)
HAVING condition;
```

Sample Table

To help you better understand the examples, and enable you to follow along with the tutorial, we are going to use the following sample table.

This table is part of an ‘Employee Management System’ that contains basic information about employees.

ID	Name	City	Job	Salary	HireDate
1	Bob	New York	Manager	60000	2012-03-07
2	Kim	Chicago	Manager	55000	2014-04-25
3	Eve	New York	Developer	32000	2015-03-11
4	Max	New York	Janitor	9000	2015-01-15
5	Joe	Chicago	Developer	30000	2016-10-05
6	Amy	New York	Developer	31000	2013-05-13
7	Ray	New York	Supervisor	40000	2016-01-15
8	Sam	Chicago	Janitor	10000	2012-02-10
9	Liv	Chicago	Developer	30000	2014-08-10
10	Ian	New York	Janitor	8500	2018-01-12

Using HAVING Clause

After the data is grouped, you may want to apply a filter condition to include or exclude certain groups. The HAVING clause is where you should place these types of filter conditions.

For example, the following query returns a list of jobs for which more than one employee is hired.

```
SELECT Job, COUNT(*) AS emp_count
FROM Employees
GROUP BY Job
HAVING COUNT(*) > 1;
```

Job	emp_count
Developer	4
Janitor	3
Manager	2

The first three lines of this query instructs the DBMS to group the data by 'Job' and then count the number of employees for each group. The final line adds a HAVING clause that filters out those groups and excludes jobs with only one or no employee.

Difference Between HAVING and WHERE

HAVING is very similar to WHERE. In fact, all the options and techniques you've learned so far about WHERE (including wildcards and multiple operators) can also be applied to HAVING.

The only difference is that WHERE filters rows and HAVING filters groups. This is because, WHERE acts on data before it is grouped, and HAVING acts on data after the groups have been created.

To demonstrate, let's modify the above example to put the grouping condition in the WHERE clause instead of HAVING:

```
SELECT Job, COUNT(*) AS emp_count
FROM Employees
WHERE COUNT(*) > 1
GROUP BY Job;
--An aggregate may not appear in the WHERE clause unless it is
  in a subquery contained in a HAVING clause or a select list,
  and the column being aggregated is an outer reference.
```

This query fails because the condition in the WHERE clause is evaluated before the grouping occurs. In fact, WHERE has no idea what a group is, so it can't perform any functions on groups.

Before adding a condition to a query with a GROUP BY clause, think carefully whether the condition works on raw data, in which case it belongs to the WHERE clause; and if it works on grouped data, it belongs to the HAVING clause.

Using HAVING and WHERE in One Statement

You might be wondering if there is a need to use both WHERE and HAVING clauses in one statement? Actually, yes, there is.

Suppose you want to further filter the above query so that it takes into account all the job titles except 'Janitor'. To do this, you can add a WHERE clause that filters all job titles except 'Janitor'. You can then add a HAVING clause to filter groups with only two or more employees.

```
SELECT Job, COUNT(*) AS emp_count
FROM Employees
WHERE Job != 'Janitor'
GROUP BY Job
HAVING COUNT(*) > 1;
```

Job	emp_count
Developer	4
Manager	2

Filter Groups by Aggregate Functions that don’t Appear in SELECT

You may use aggregate functions in the HAVING clause, that do not appear in the SELECT clause, as demonstrated by the following query.

```
SELECT Job, COUNT(*) AS emp_count
FROM Employees
GROUP BY Job
HAVING AVG(Salary) > 30500;
```

Job	emp_count
Developer	4
Manager	2
Supervisor	1

This query groups the data by the ‘Job’ column and then counts the number of employees for each group, but then the filter condition in the HAVING clause excludes all groups for which the average salary is less than $30,500.