**SQL HAVING**

To specify a condition for groups, you use the **HAVING** clause.

The HAVING clause is often used with the GROUP BY clause in the [SELECT statement](http://www.sqltutorial.org/sql-select/). If you use a **HAVING** clause without a **GROUP BY** clause, the **HAVING** clause behaves like the [**WHERE** clause](http://www.sqltutorial.org/sql-where/).

The following illustrates the syntax of the **HAVING** clause:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | **SELECT**      column1,      column2,  **AGGREGATE\_FUNCTION** (column3)  **FROM**      table1  **GROUP BY**      column1,      column2  **HAVING**      group\_condition; |

Note that the HAVING clause appears immediately after the GROUP BY clause.

## **HAVING vs. WHERE**

The [WHERE](http://www.sqltutorial.org/sql-where/) clause applies the condition to individual rows before the rows are summarized into groups by the GROUP BY clause. However, the HAVING clause applies the condition to the groups after the rows are grouped into groups.

Therefore, it is important to note that the HAVING clause is applied after whereas the WHERE clause is applied before the GROUP BY clause.

## SQL HAVING clause examples

We will take the employees and departments tables in the [sample database](http://www.sqltutorial.org/sql-sample-database/) for the demonstration.



To get the managers and their direct reports, you use the GROUP BY clause to group employees by the managers and use the [COUNT function](http://www.sqltutorial.org/sql-aggregate-functions/sql-count/) to count the direct reports.

The following query illustrates the idea:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | SELECT      manager\_id,      first\_name,      last\_name,      COUNT(employee\_id) direct\_reports  FROM      employees  WHERE      manager\_id IS NOT NULL  GROUP BY manager\_id; |

To find the headcount of each department, you group the employees by the department\_id column, and apply the [COUNT](http://www.sqltutorial.org/sql-count/) function to each group as the following query:

SELECT

    department\_id,

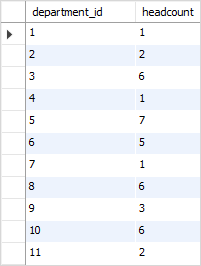
    COUNT(employee\_id) headcount

FROM

    employees

GROUP BY

    department\_id;



### **SQL GROUP BY with INNER JOIN example**

To get the department name, you [join](http://www.sqltutorial.org/sql-inner-join/) the employees table with the departments table as follows:

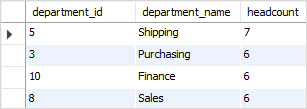
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | SELECT      e.department\_id,      department\_name,      COUNT(employee\_id) headcount  FROM      employees e  INNER JOIN departments d ON d.department\_id = e.department\_id  GROUP BY      e.department\_id; |



### SQL GROUP BY with HAVING example

To find the department whose headcount is greater than 5, you use the [HAVING](http://www.sqltutorial.org/sql-having/) clause as the following query:

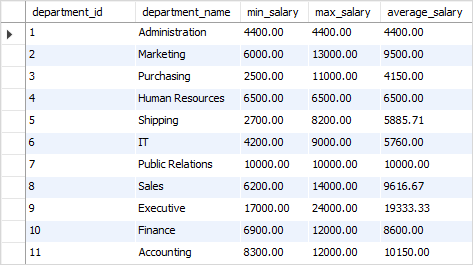
|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | SELECT      e.department\_id,      department\_name,      COUNT(employee\_id) headcount  FROM      employees e          INNER JOIN      departments d ON d.department\_id = e.department\_id  GROUP BY e.department\_id  HAVING headcount > 5  ORDER BY headcount DESC; |



### SQL GROUP BY with MIN, MAX, and AVG example

The following query returns the [minimum](http://www.sqltutorial.org/sql-aggregate-functions/sql-min/), [maximum](http://www.sqltutorial.org/sql-aggregate-functions/sql-max/), and [average](http://www.sqltutorial.org/sql-aggregate-functions/sql-avg/) salary of employees in each department.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | SELECT      e.department\_id,      department\_name,      MIN(salary) min\_salary,      MAX(salary) max\_salary,      ROUND(AVG(salary), 2) average\_salary  FROM      employees e          INNER JOIN      departments d ON d.department\_id = e.department\_id  GROUP BY e.department\_id; |



### SQL GROUP BY with SUM function example

To get the total salary per department, you apply the [SUM](http://www.sqltutorial.org/sql-sum/) function to the salary column and group employees by the department\_id column as follows:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | SELECT      e.department\_id,      department\_name,      SUM(salary) total\_salary  FROM      employees e          INNER JOIN      departments d ON d.department\_id = e.department\_id  GROUP BY e.department\_id; |

