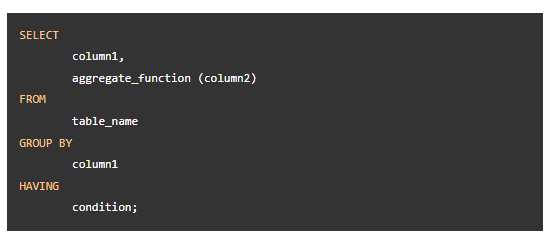
**Introduction to PostgreSQL HAVING clause**

The HAVING clause specifies a search condition for a group or an aggregate. The HAVING clause is often used with the GROUP BY clause to filter groups or aggregates based on a specified condition.

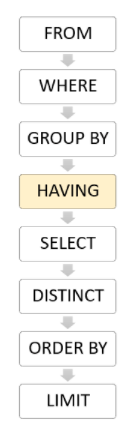
The following statement illustrates the basic syntax of the HAVING clause:



In this syntax, the group by clause returns rows grouped by the column1. The HAVING clause specifies a condition to filter the groups.

It’s possible to add other clauses of the SELECT statement such as JOIN, LIMIT, and FETCH etc.

PostgreSQL evaluates the HAVING clause after the FROM, WHERE, GROUP BY, and before the SELECT, DISTINCT, ORDER BY and LIMIT clauses.



Since the HAVING clause is evaluated before the SELECT clause, you cannot use column aliases in the HAVING clause. Because at the time of evaluating the HAVING clause, the column aliases specified in the SELECT clause are not available.

HAVING vs. WHERE

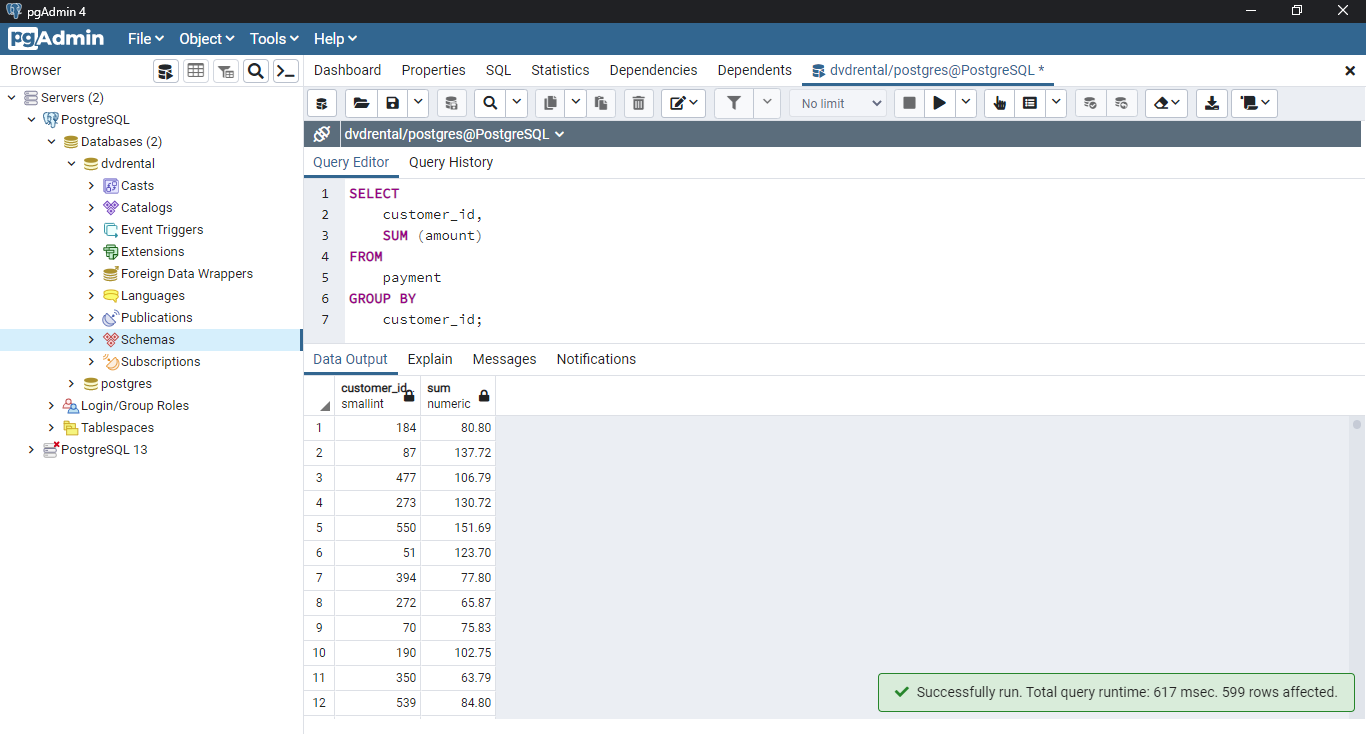
The WHERE clause allows you to filter rows based on a specified condition. However, the HAVING clause allows you to filter groups of rows according to a specified condition.

In other words, the WHERE clause is applied to rows while the HAVING clause is applied to groups of rows.

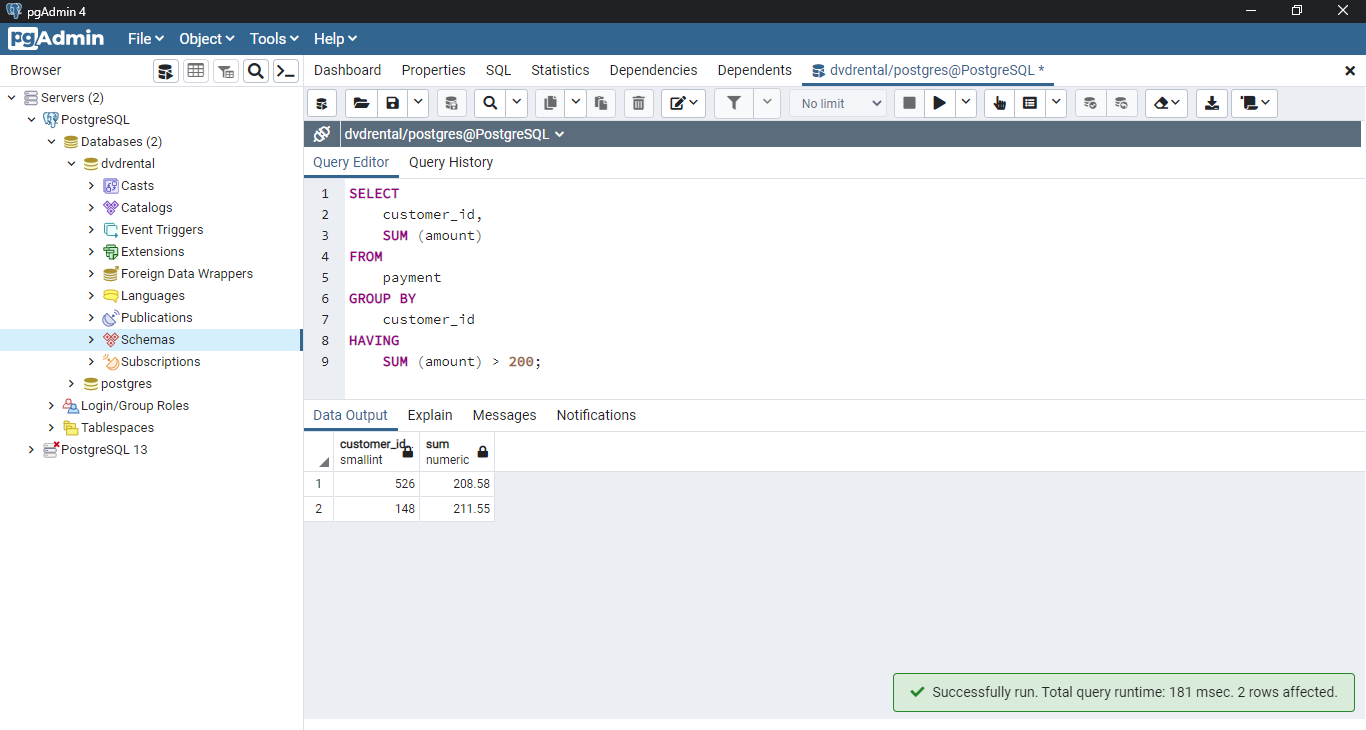
**PostgreSQL HAVING clause examples**

1) Using PostgreSQL HAVING clause with SUM function example

The following query uses the GROUP BY clause with the SUM () function to find the total amount of each customer:

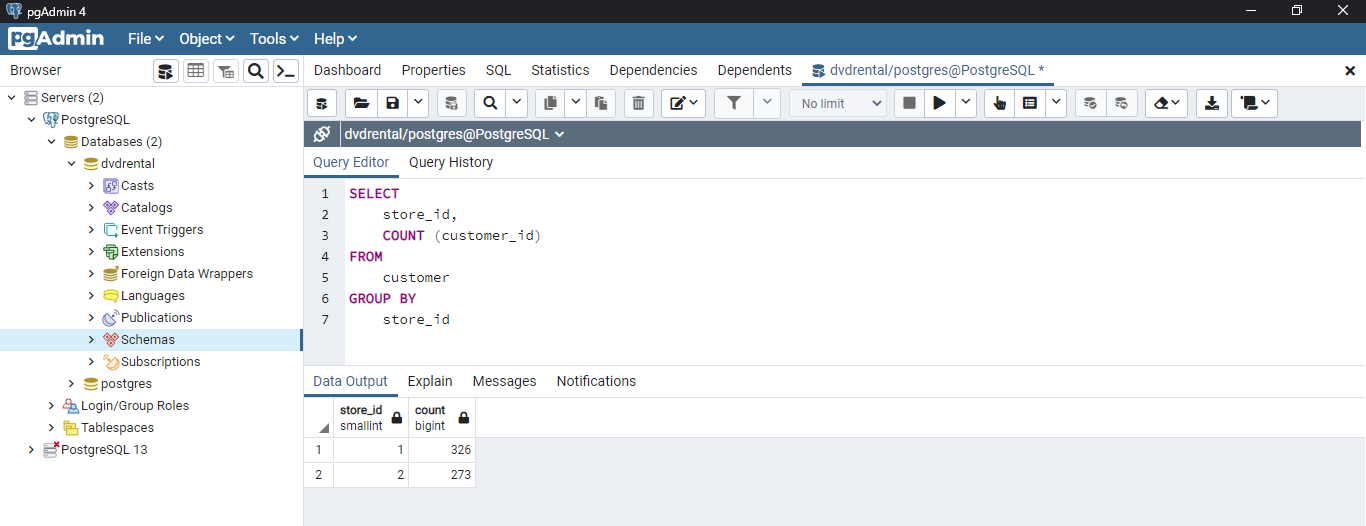


The following statement adds the HAVING clause to select the only customers who have been spending more than 200:



2) PostgreSQL HAVING clause with COUNT example

The following query uses the GROUP BY clause to find the number of customers per store:



The following statement adds the HAVING clause to select the store that has more than 300 customers:

