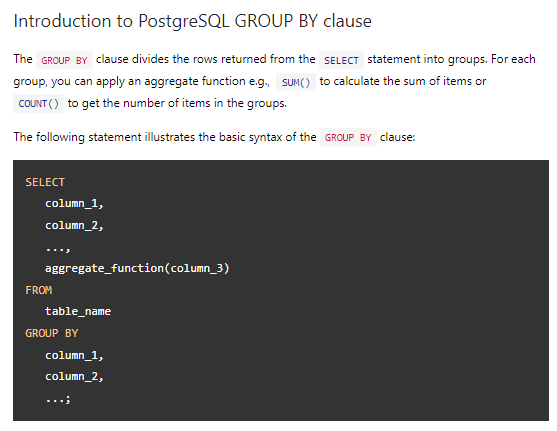
**Section 4. Grouping Data**

Introduction to PostgreSQL GROUP BY clause

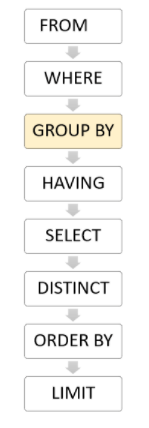
The GROUP BY clause divides the rows returned from the [SELECT](https://www.postgresqltutorial.com/postgresql-select/) statement into groups. For each group, you can apply an aggregate function e.g., [SUM ()](https://www.postgresqltutorial.com/postgresql-sum-function/) to calculate the sum of items or [COUNT ()](https://www.postgresqltutorial.com/postgresql-count-function/) to get the number of items in the groups.

The following statement illustrates the basic syntax of the GROUP BY clause:

In this syntax:

* First, select the columns that you want to group e.g., column1 and column2, and column that you want to apply an aggregate function (column3).
* Second, list the columns that you want to group in the GROUP BY clause.

The statement clause divides the rows by the values of the columns specified in the GROUP BY clause and calculates a value for each group.

It’s possible to use other clauses of the SELECT statement with the GROUP BY clause.

PostgreSQL evaluates the GROUP BY clause after the FROM and [WHERE](https://www.postgresqltutorial.com/postgresql-where/) clauses and before the [HAVING](https://www.postgresqltutorial.com/postgresql-having/) [SELECT](https://www.postgresqltutorial.com/postgresql-select/), [DISTINCT](https://www.postgresqltutorial.com/postgresql-select-distinct/), [ORDER BY](https://www.postgresqltutorial.com/postgresql-order-by/)and [LIMIT](https://www.postgresqltutorial.com/postgresql-limit/) clauses.

## PostgreSQL GROUP BY clause

### 1) Using PostgreSQL GROUP BY without an aggregate function example

You can use the GROUP BY clause without applying an aggregate function. The following query gets data from the payment table and groups the result by customer id.

## 

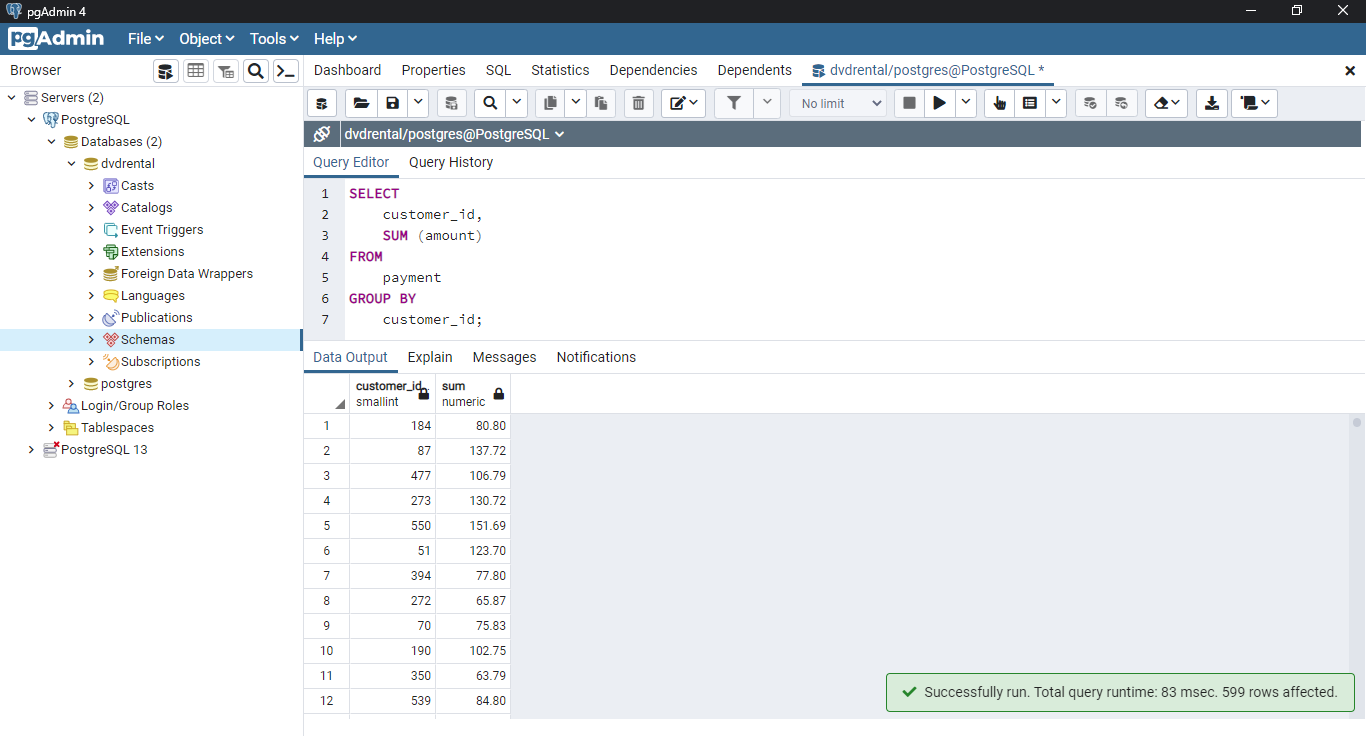
## In this case, the GROUP BY works like the DISTINCT clause that removes duplicate rows from the result set.

### 2) Using PostgreSQL GROUP BY with SUM () function example

The GROUP BY clause is useful when it is used in conjunction with an [aggregate function](https://www.postgresqltutorial.com/postgresql-aggregate-functions/).

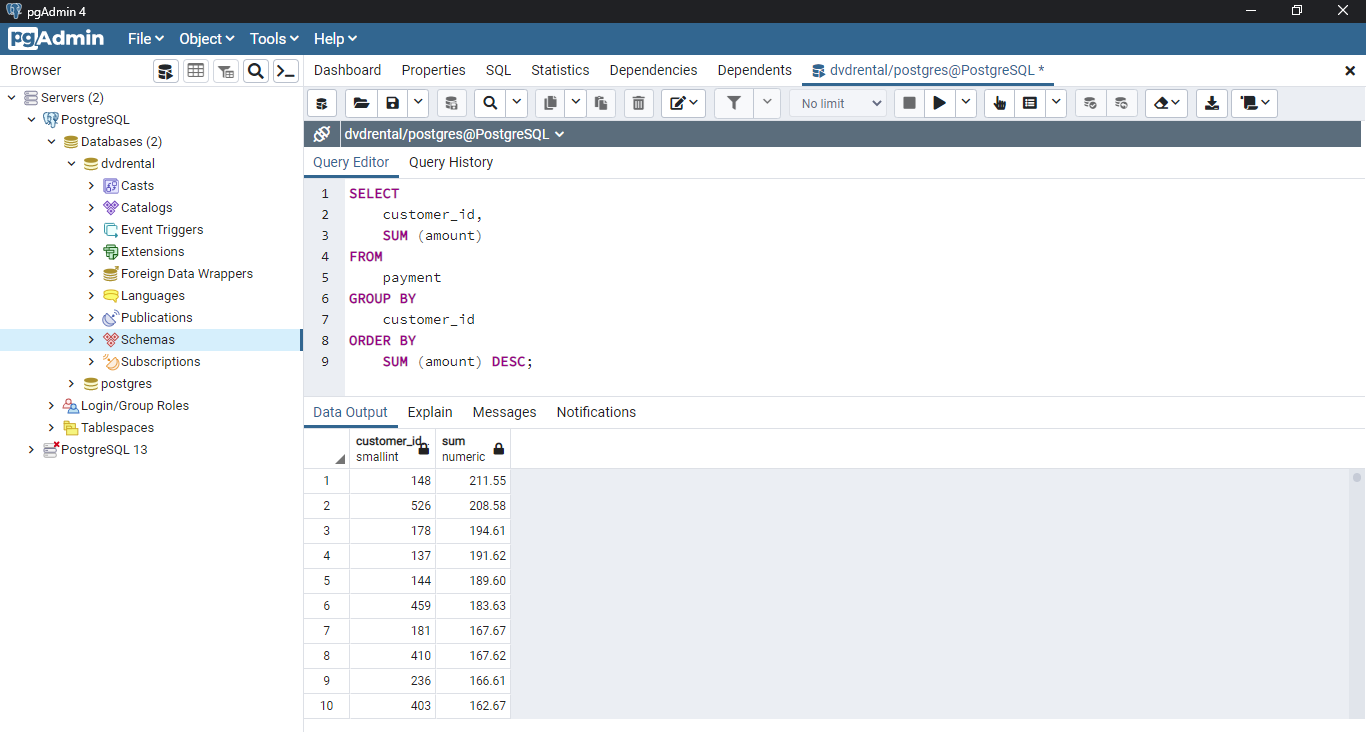
For example, to select the total amount that each customer has been paid, you use the GROUP BY clause to divide the rows in the payment table into groups grouped by customer id. For each group, you calculate the total amounts using the [SUM ()](https://www.postgresqltutorial.com/postgresql-sum-function/) function.

The following query uses the GROUP BY clause to get total amount that each customer has been paid:

****

The GROUP BY clause sorts the result set by customer id and adds up the amount that belongs to the same customer. Whenever the customer\_id changes, it adds the row to the returned result set.

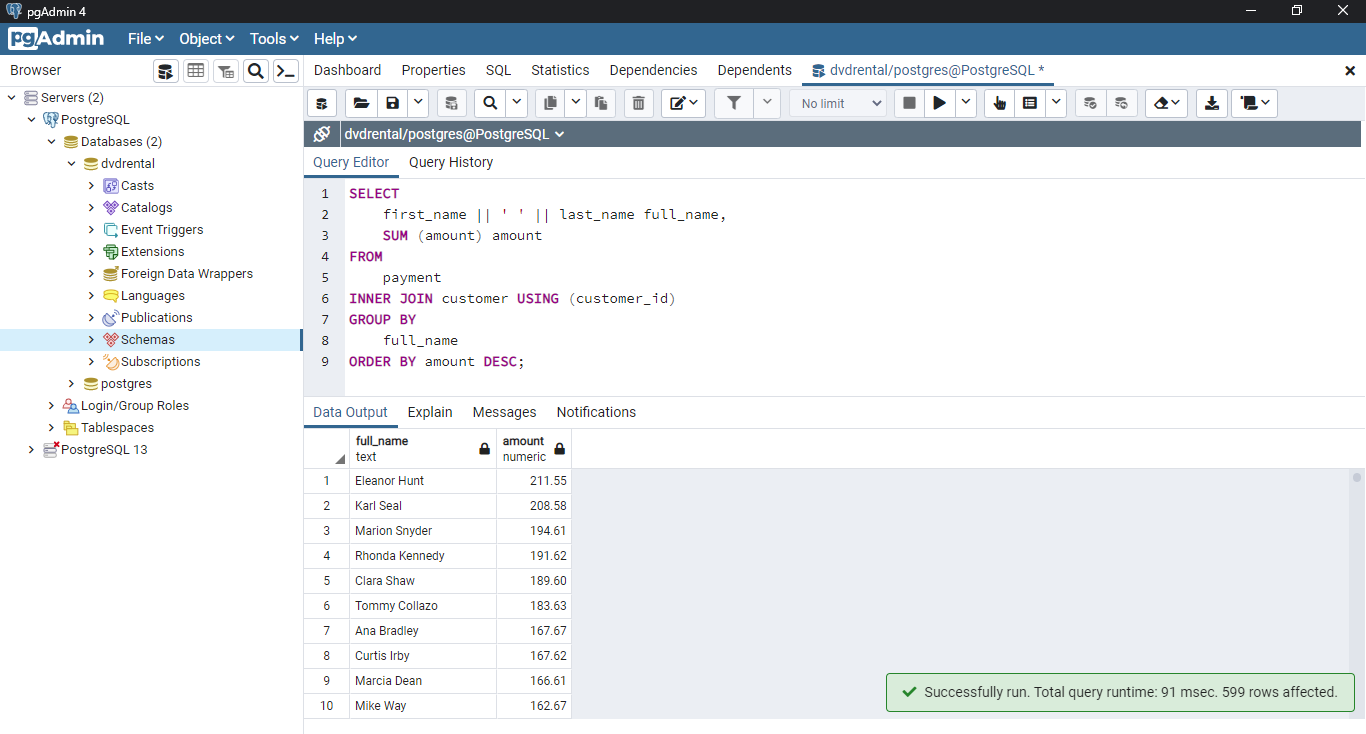
The following statement uses the ORDER BY clause with GROUP BY clause to sort the groups:



3) Using PostgreSQL GROUP BY clause with the JOIN clause

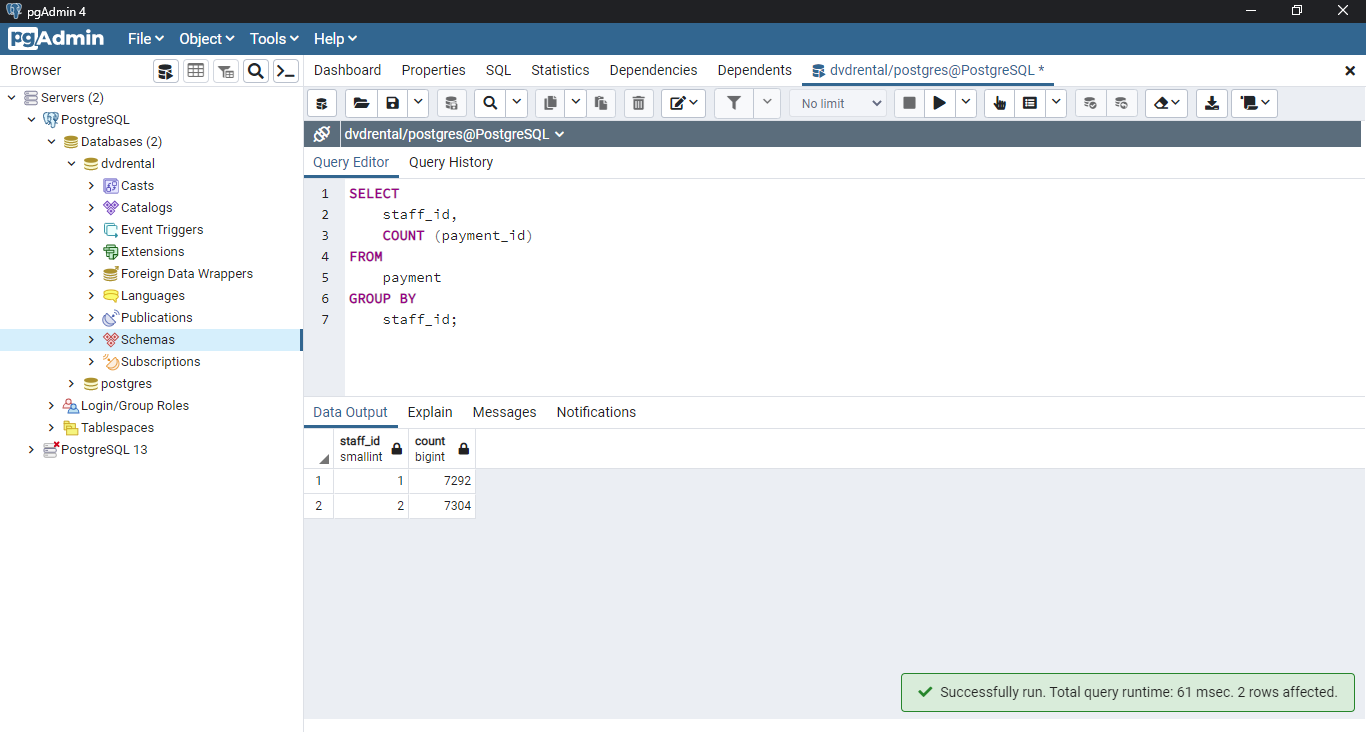
The following statement uses the GROUP BY clause with the INNER JOIN clause the get the total amount paid by each customer.

Unlike the previous example, this query joins the payment table with the customer table and group customers by their names.



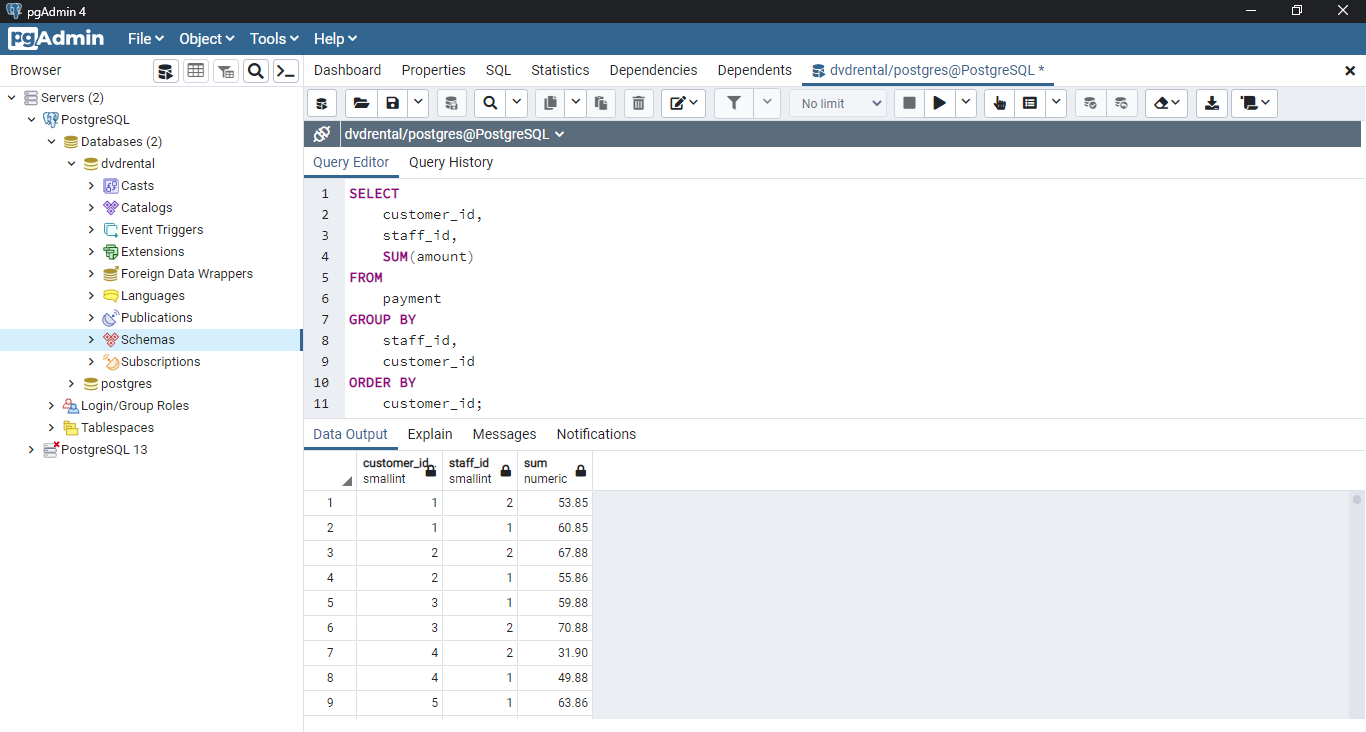
4) Using PostgreSQL GROUP BY with COUNT () function example

To find the number of payment transactions that each staff has been processed, you group the rows in the payment table by the values in the staff\_id column and use the COUNT () function to get the number of transactions:



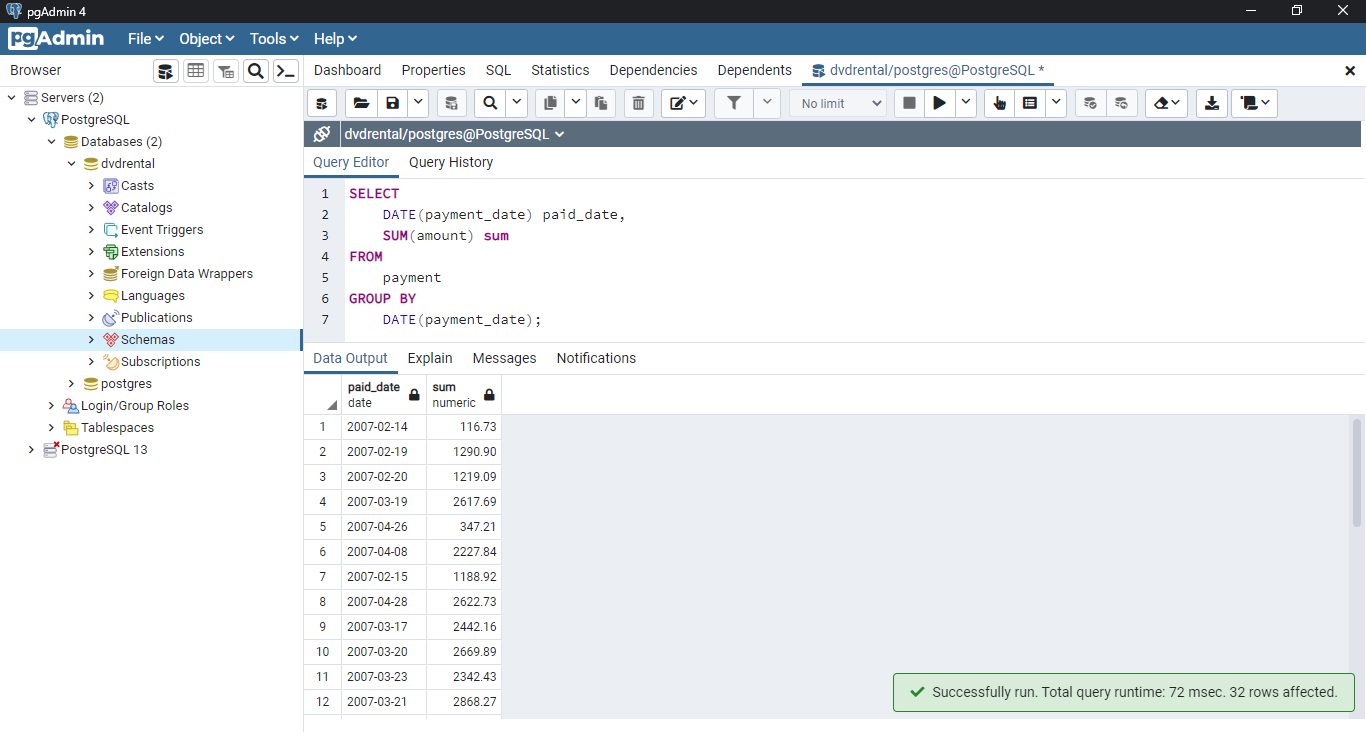
The GROUP BY clause divides the rows in the payment into groups and groups them by value in the staff\_id column. For each group, it returns the number of rows by using the COUNT () function.

5) Using PostgreSQL GROUP BY with multiple columns

The following example uses multiple columns in the GROUP BY clause:

In this example, the GROUP BY clause divides the rows in the payment table by the values in the customer\_id and staff\_id columns. For each group of (customer\_id, staff\_id), the SUM () calculates the total amount.

6) Using PostgreSQL GROUP BY clause with date column

The payment\_date is a timestamp column. To group payments by dates, you use the DATE () function to convert timestamps to dates first and then group payments by the result date: