**Sales Order Case Study**

# Total available Tables

**SELECT \* FROM products;**

**SELECT \* FROM customers;**

**SELECT \* FROM employees;**

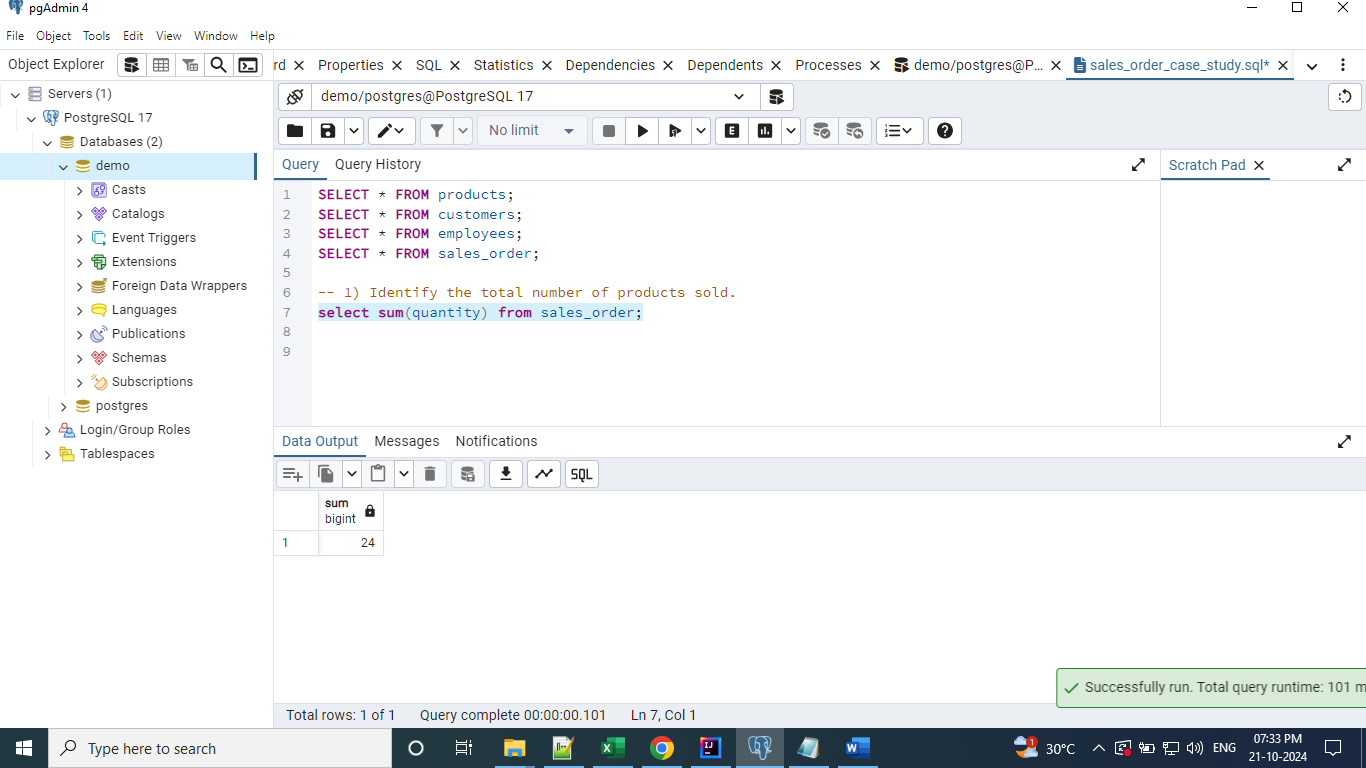
**SELECT \* FROM sales\_order;**

# Problems

## Identify the total number of products sold.

**We can get the total number of products details from the Products table. But the twist here is we need to get the total number of Products sold. The Sales\_Order is the only table which has the quantity of all products sold.**

**select sum(quantity) from sales\_order;**

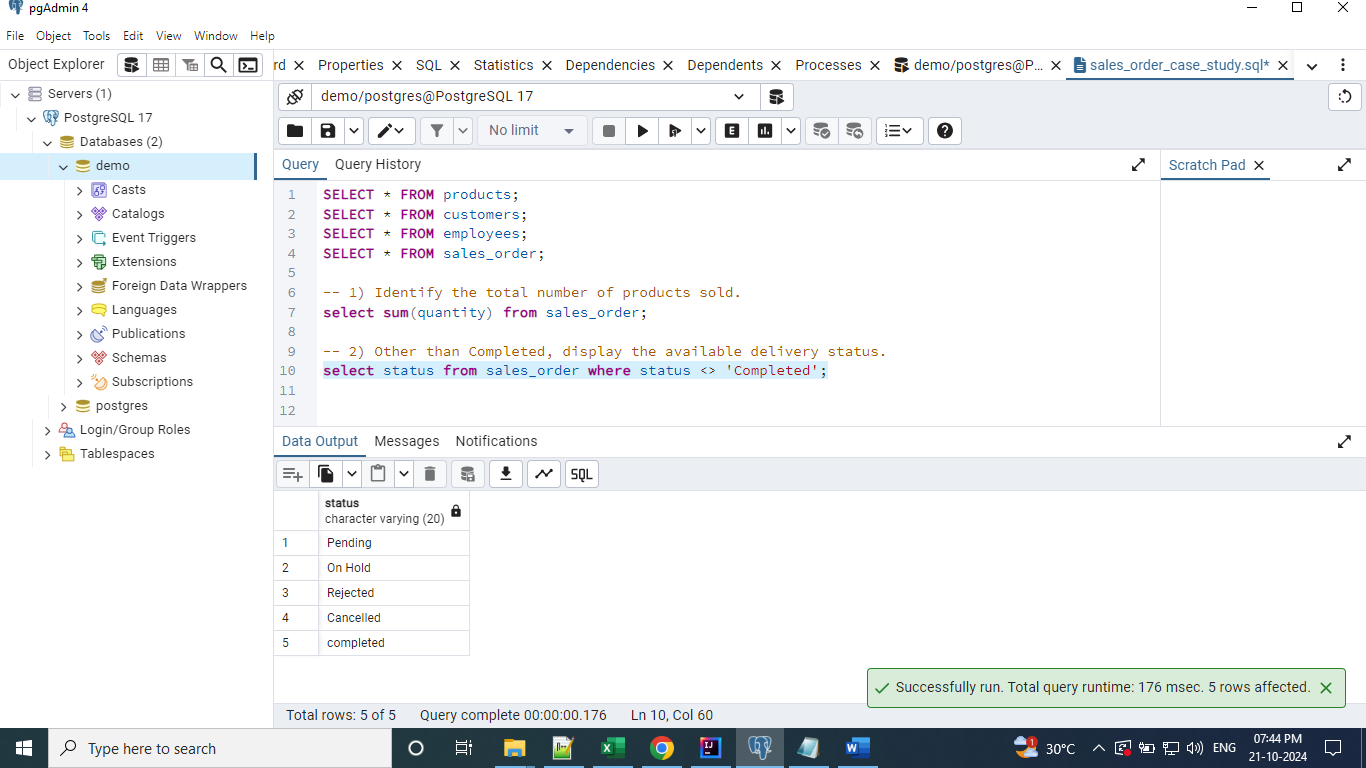


## Other than completed, display the available delivery status.

**We can get the details of order status from the Sales\_Order table.**

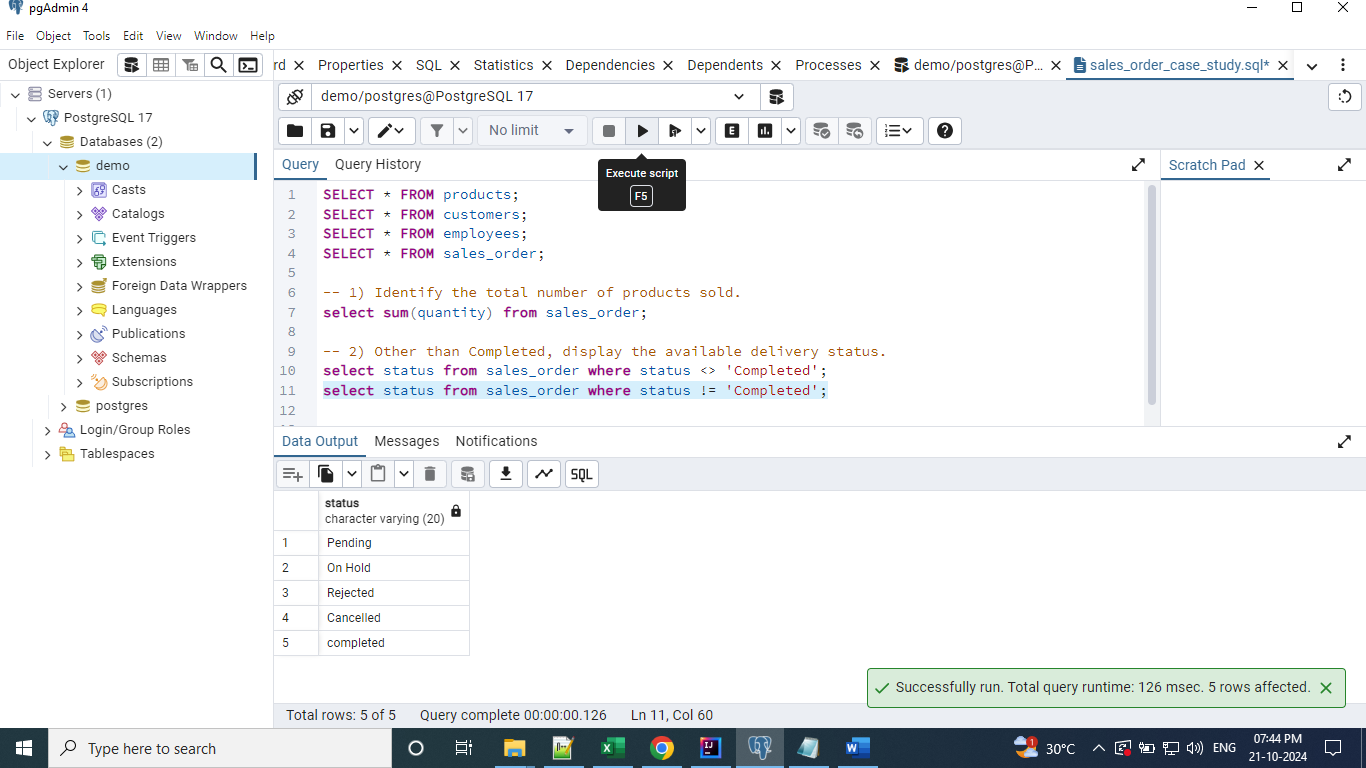
**# Approach 1**

**select status from sales\_order where status <> 'Completed';**



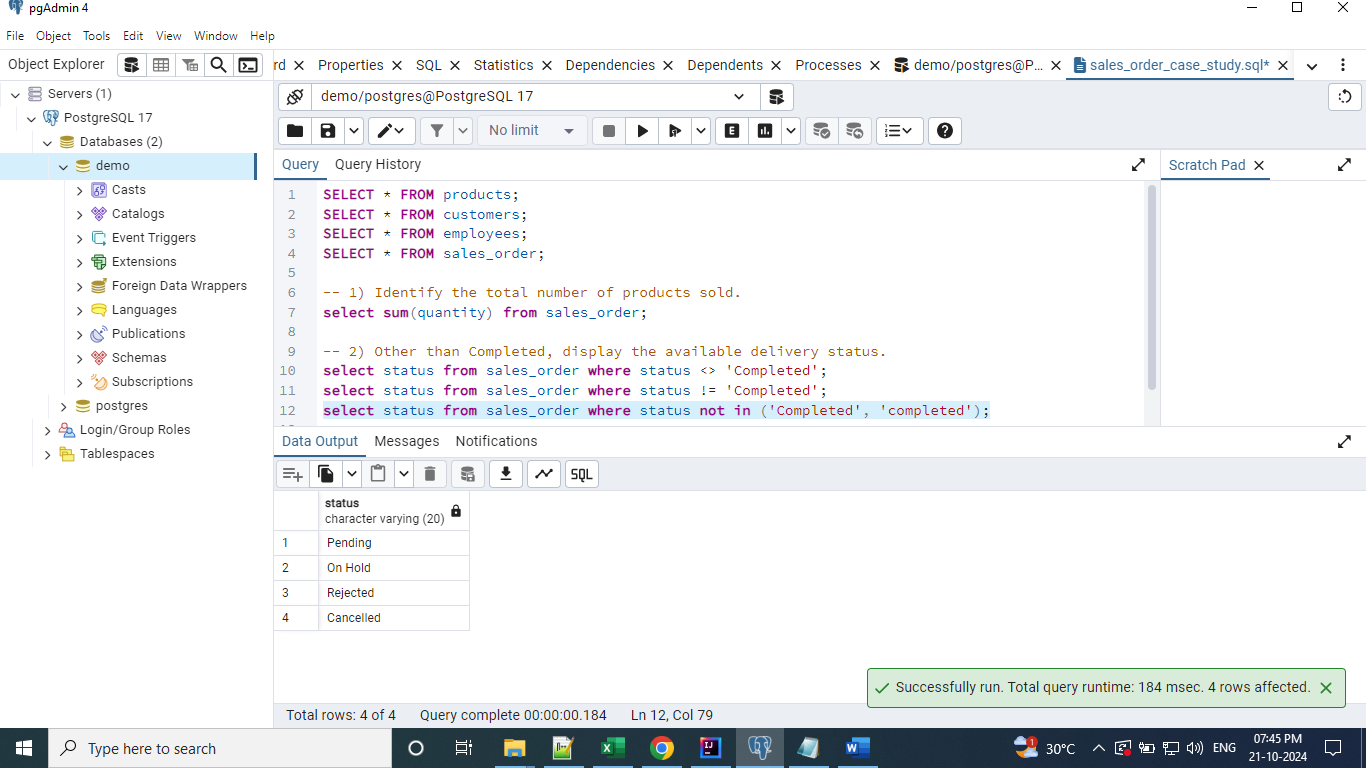
**# Approach 2**

**select status from sales\_order where status != 'Completed';**



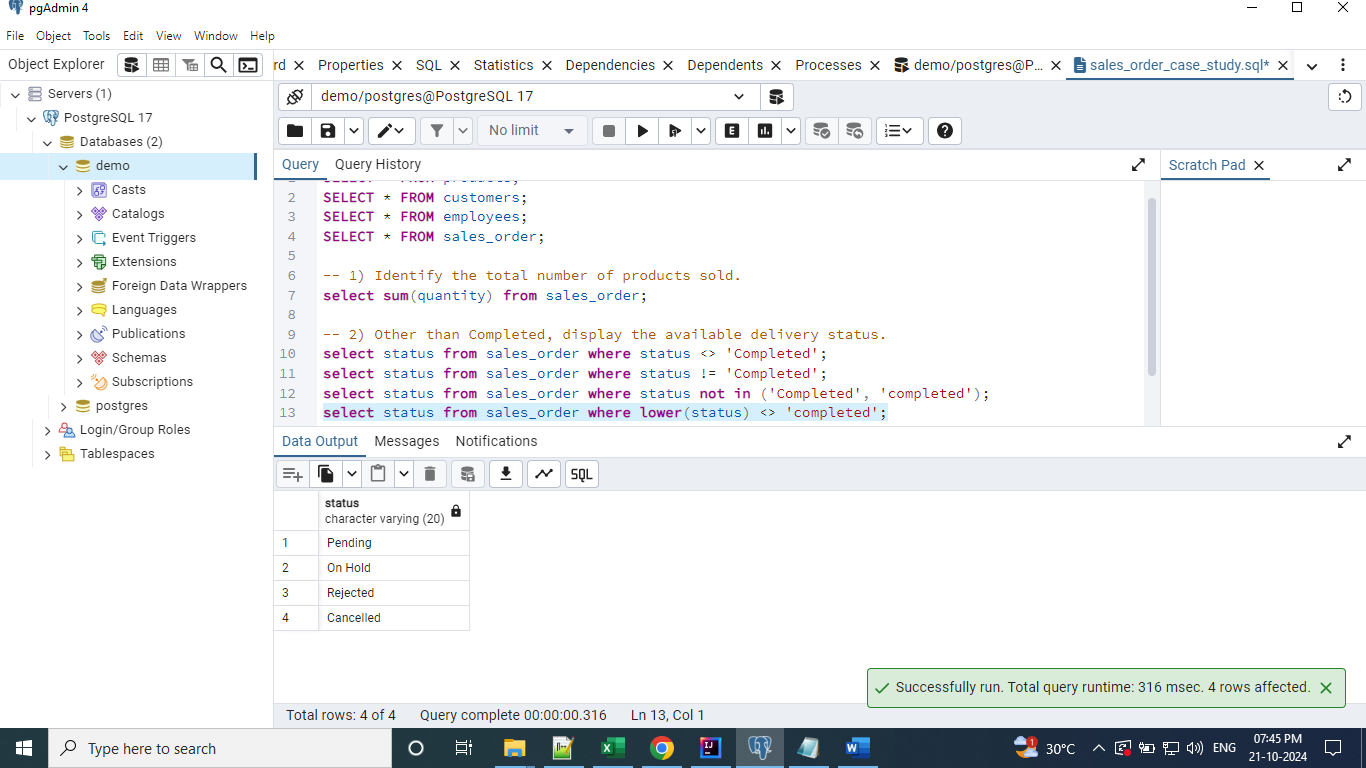
**# Approach 3**

**select status from sales\_order where status not in ('Completed', 'completed');**



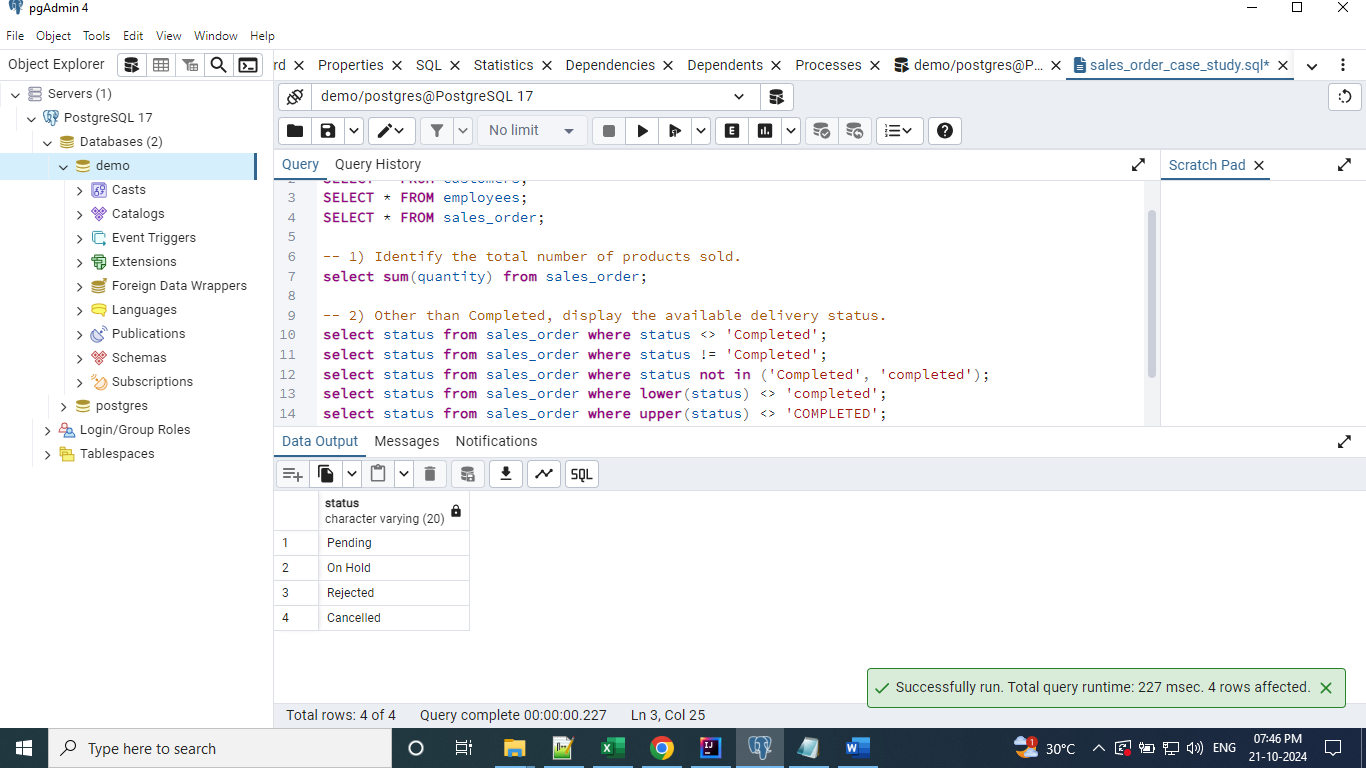
**# Approach 4**

**select status from sales\_order where lower(status) <> 'completed';**



**# Approach 5**

**select status from sales\_order where upper(status) <> 'COMPLETED';**



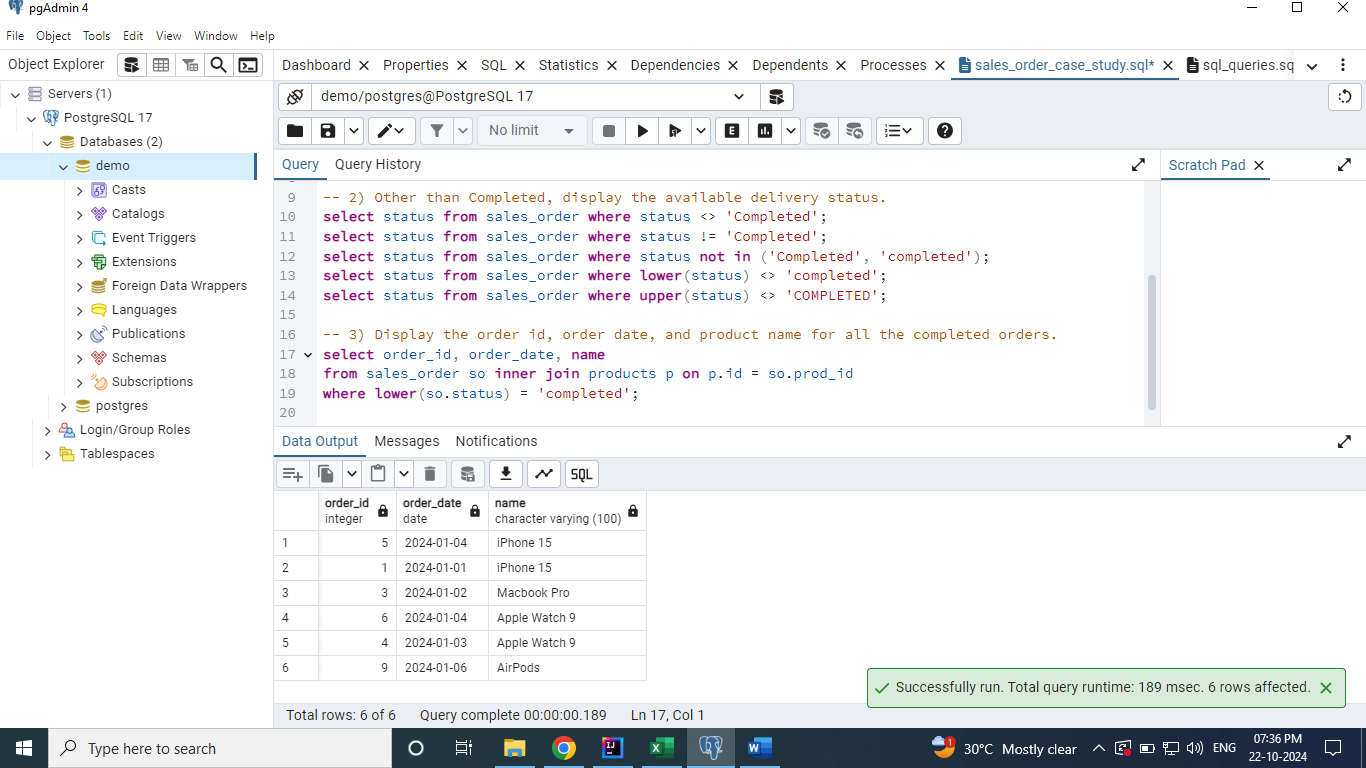
## Display the order id, order date, and product name for all the completed orders.

**Here, we have to fetch the column values from two tables, so need to use the join function. Inner Join is used here to fetch the records.**

**select order\_id, order\_date, name**

**from sales\_order so inner join products p on p.id = so.prod\_id**

**where lower(so.status) = 'completed';**



## Sort the above query to show the earliest orders at the top. Also display the customer who purchased these orders.

**Here, we need to print the customer details too. So, we have joined the customer table along with previous join and sorted using the order by clause.**

**select order\_id, order\_date, p.name, c.name**

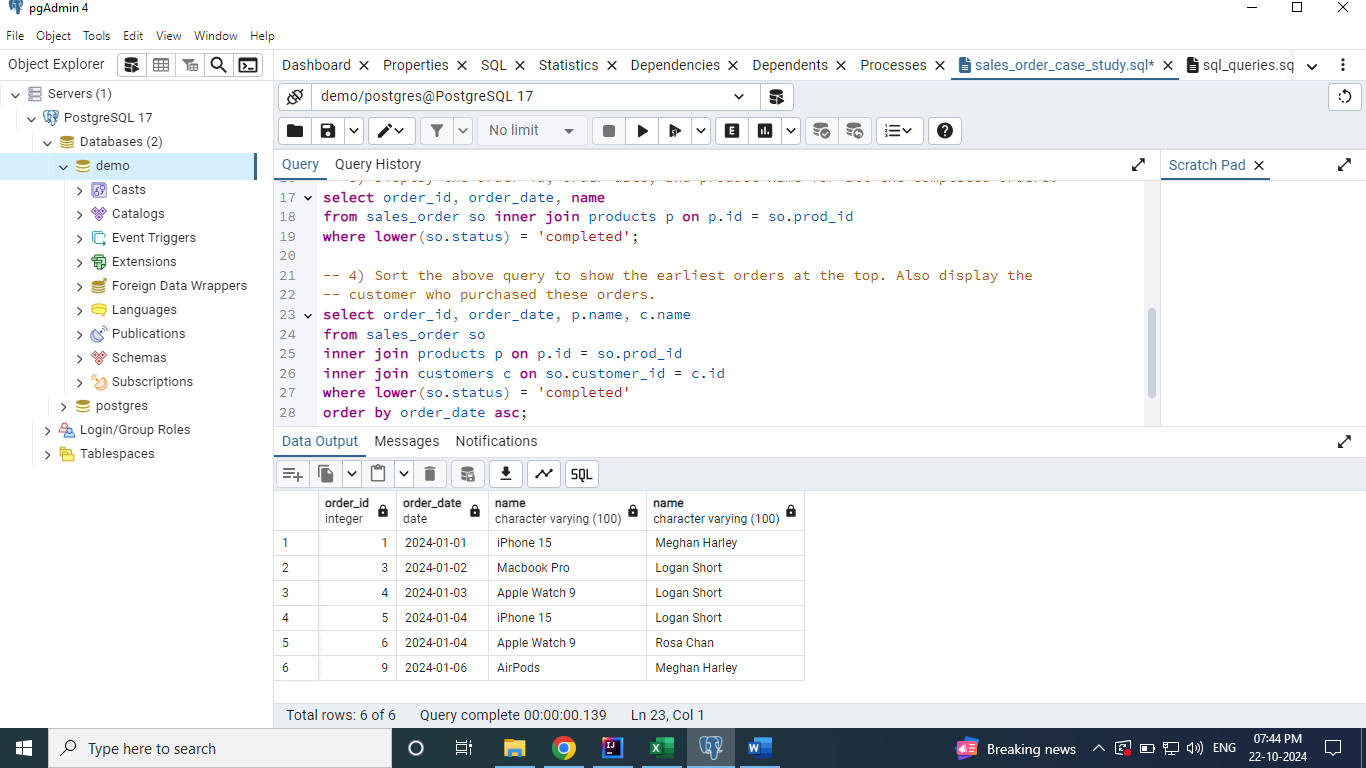
**from sales\_order so**

**inner join products p on p.id = so.prod\_id**

**inner join customers c on so.customer\_id = c.id**

**where lower(so.status) = 'completed'**

**order by order\_date asc;**



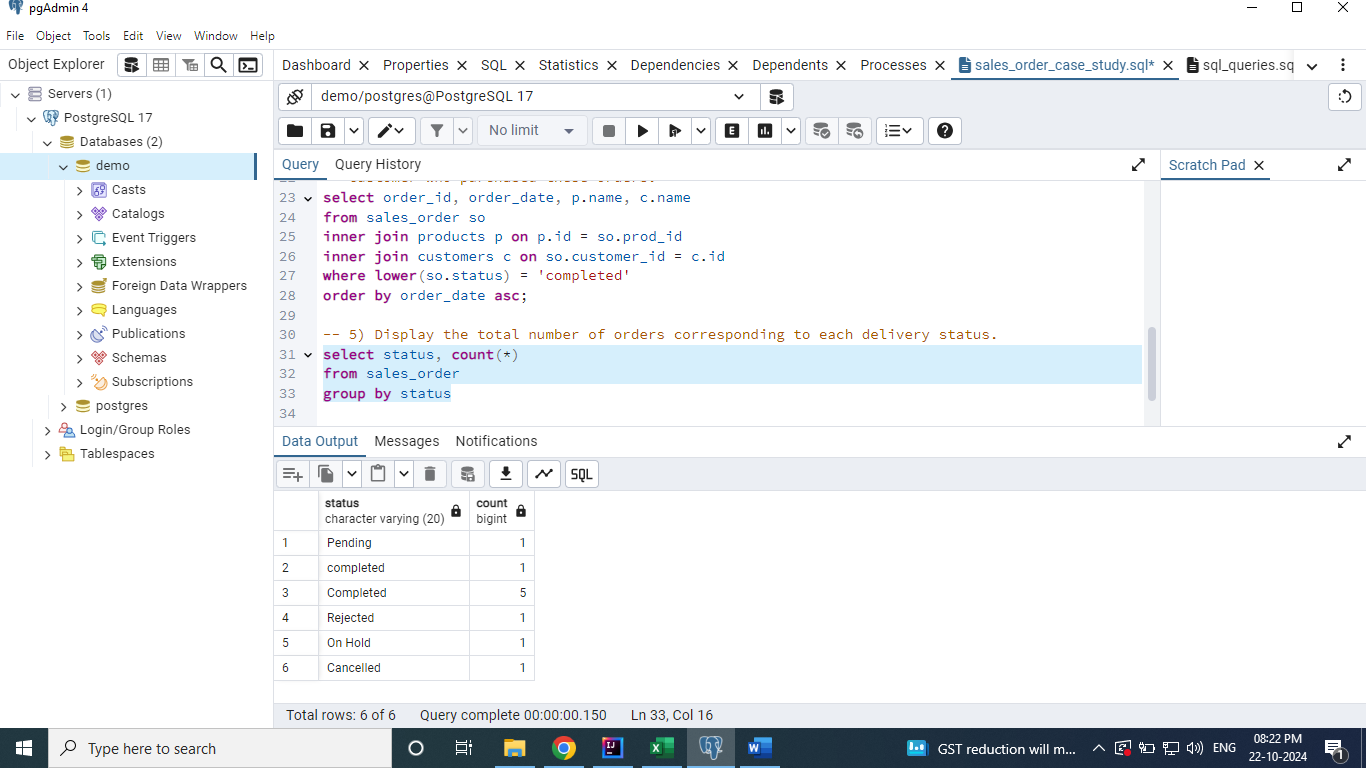
## Display the total number of orders corresponding to each delivery status.

**We can use the group by clause to group the orders based on the status.**

**select status, count(\*)**

**from sales\_order**

**group by status**



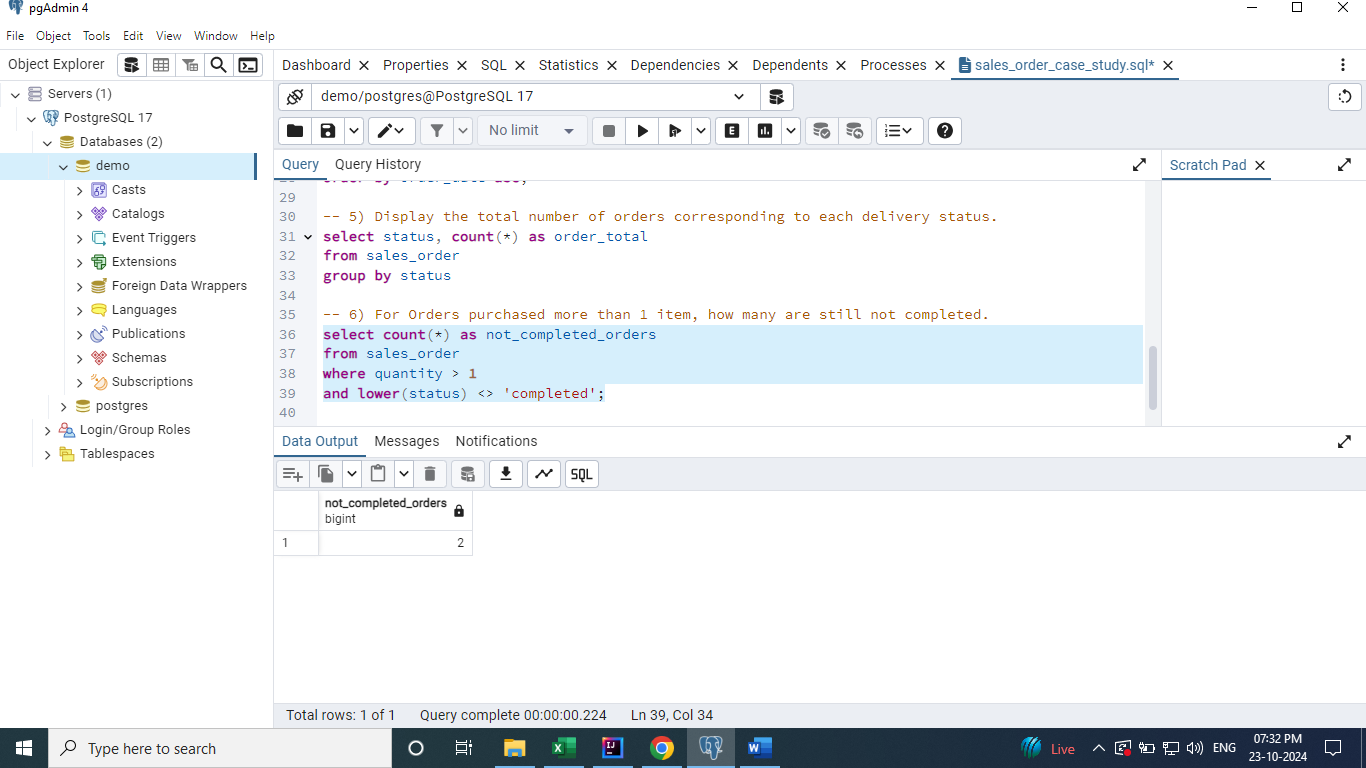
## For Orders purchased more than 1 item, how many are still not completed.

**select count(\*) as not\_completed\_orders**

**from sales\_order**

**where quantity > 1**

**and lower(status) <> 'completed';**

7. 

## 7. Find the total number of orders corresponding to each delivery status by ignoring the case in delivery status with highest number of orders should be at the top.

**Approach 1 [Sub Query]**

**select status,**

**case when status = 'completed'**

**then 'Completed'**

**else status**

**end as updated\_status from sales\_order;**

**select updated\_status, count(\*) as order\_total**

**from (select status,**

**case when status = 'completed'**

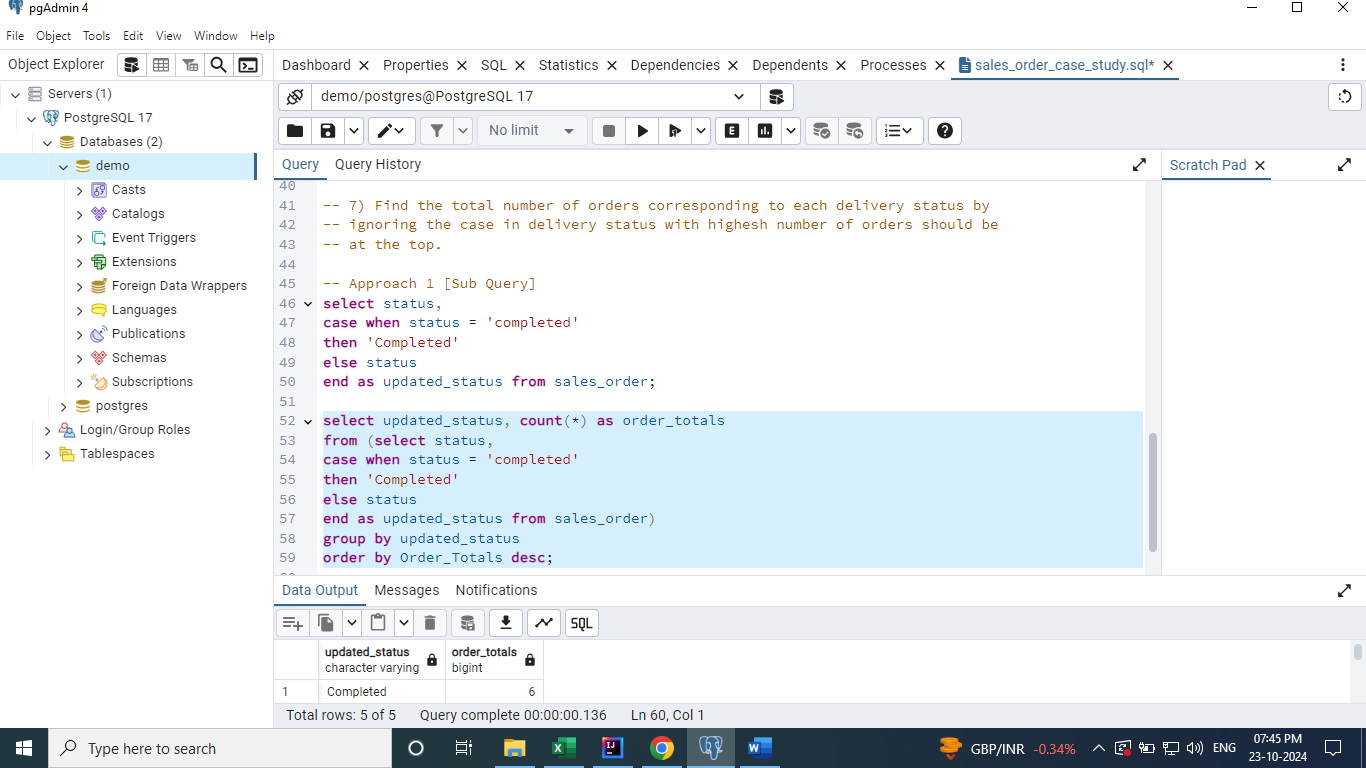
**then 'Completed'**

**else status**

**end as updated\_status from sales\_order)**

**group by updated\_status**

**order by order\_total desc;**



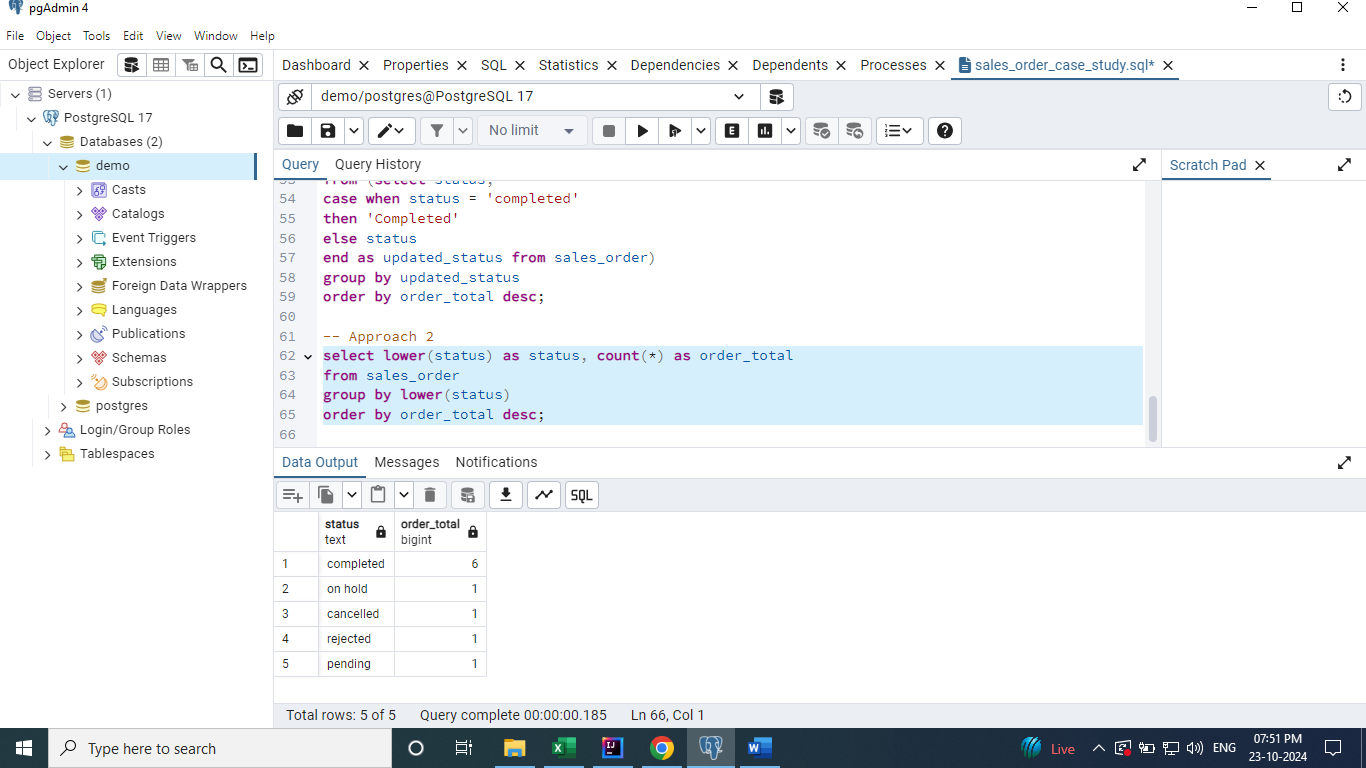
**Approach 2**

**select lower(status) as status, count(\*) as order\_total**

**from sales\_order**

**group by lower(status)**

**order by order\_total desc;**



## Write a query to identify the total products purchased by each customer.

**select c.name as customer\_name, sum(so.quantity)**

**from sales\_order so**

**join customers c on c.id = so.customer\_id**

**group by c.name;**

