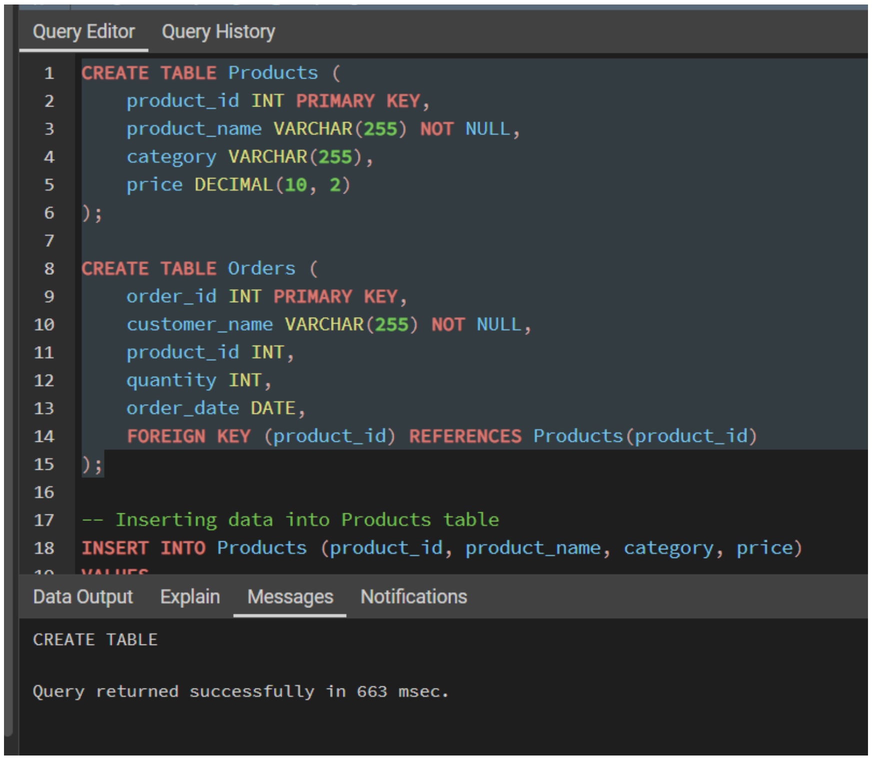
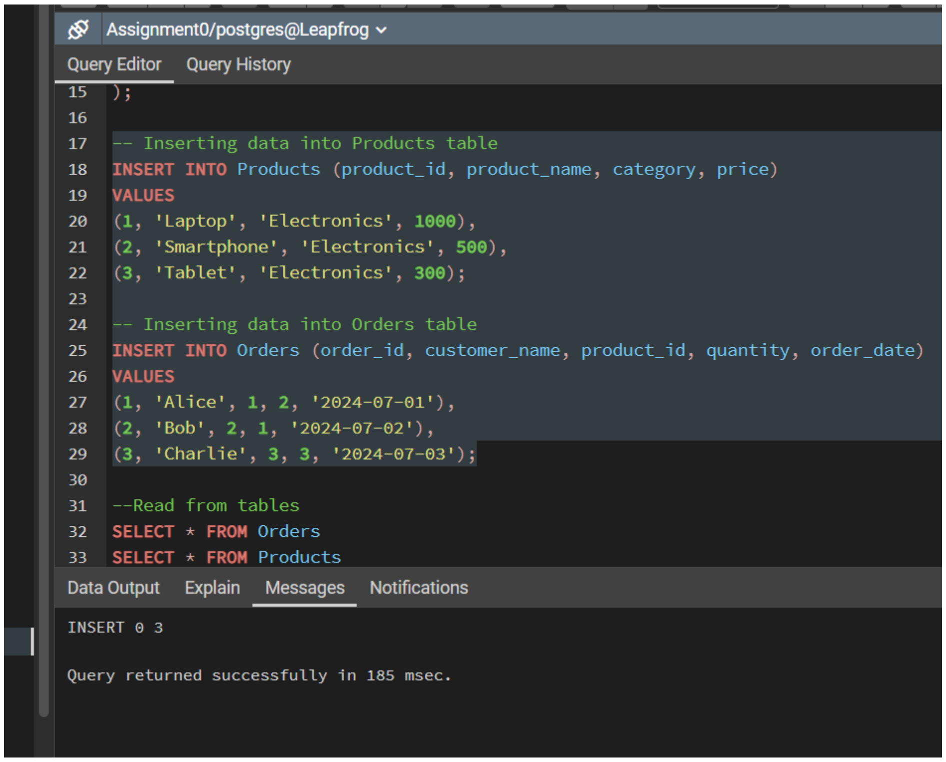
Create the tables.



```sql
CREATE TABLE Products (
    product_id INT PRIMARY KEY,
    product_name VARCHAR(255) NOT NULL,
    category VARCHAR(255),
    price DECIMAL(10, 2)
);

CREATE TABLE Orders (
    order_id INT PRIMARY KEY,
    customer_name VARCHAR(255) NOT NULL,
    product_id INT,
    quantity INT,
    order_date DATE,
    FOREIGN KEY (product_id) REFERENCES Products(product_id)
);

-- Inserting data into Products table
INSERT INTO Products (product_id, product_name, category, price)
```

```
CREATE TABLE

Query returned successfully in 663 msec.
```

Insert data into the tables.

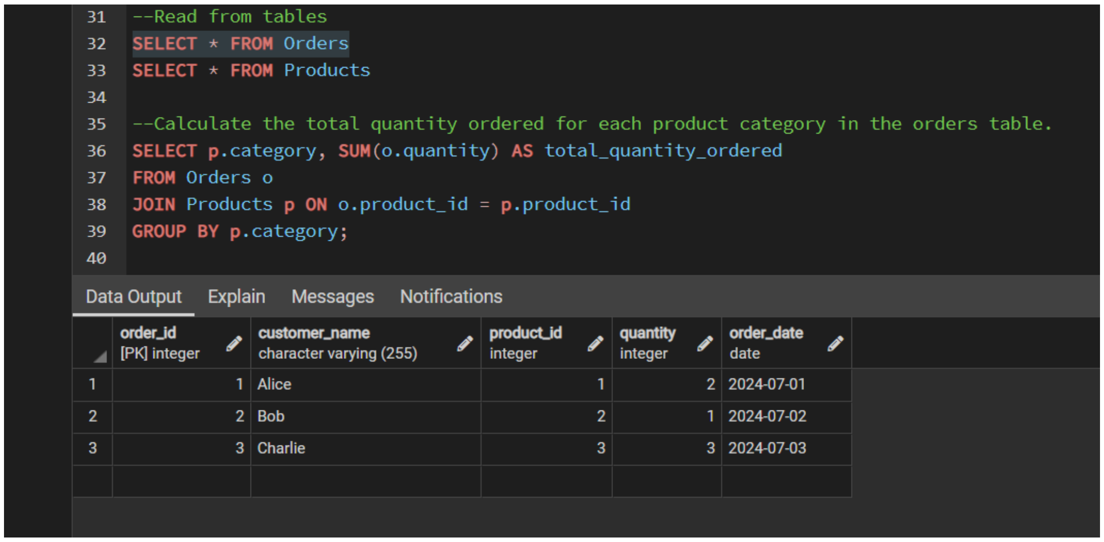


```sql
);

-- Inserting data into Products table
INSERT INTO Products (product_id, product_name, category, price)
VALUES
(1, 'Laptop', 'Electronics', 1000),
(2, 'Smartphone', 'Electronics', 500),
(3, 'Tablet', 'Electronics', 300);

-- Inserting data into Orders table
INSERT INTO Orders (order_id, customer_name, product_id, quantity, order_date)
VALUES
(1, 'Alice', 1, 2, '2024-07-01'),
(2, 'Bob', 2, 1, '2024-07-02'),
(3, 'Charlie', 3, 3, '2024-07-03');

--Read from tables
SELECT * FROM Orders
SELECT * FROM Products
```

```
INSERT 0 3

Query returned successfully in 185 msec.
```

Read data from table

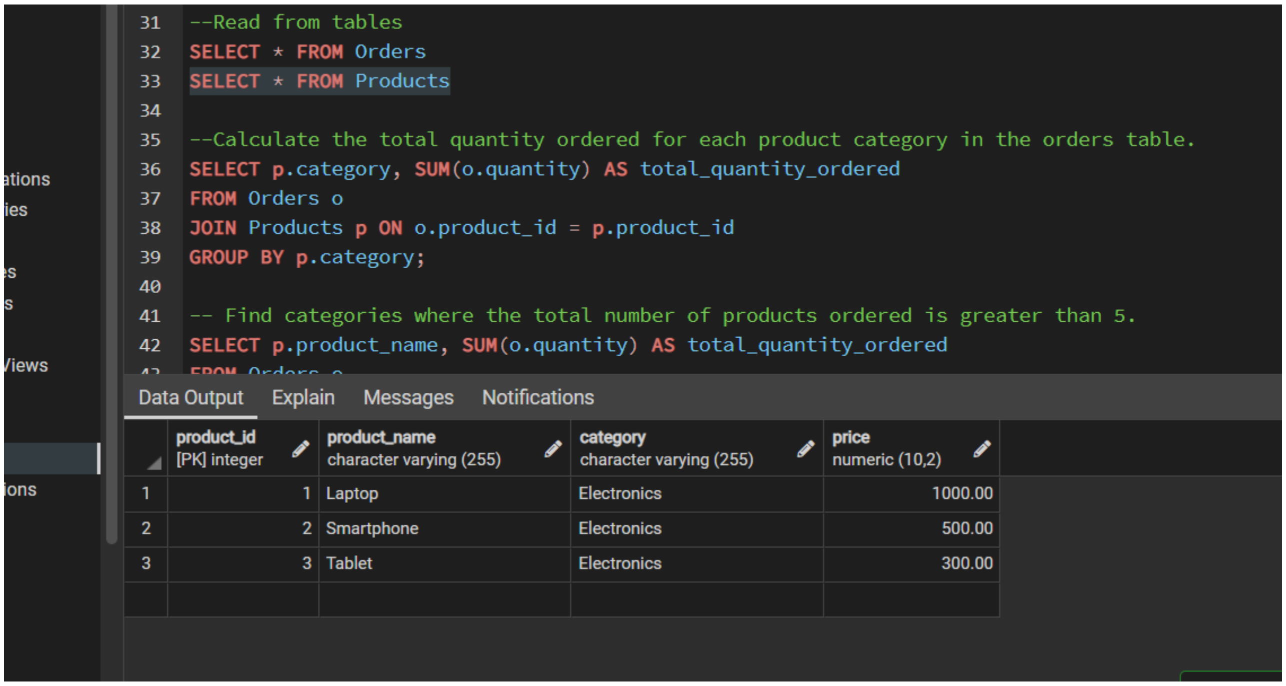
1. Orders table



```sql
--Read from tables
SELECT * FROM Orders
SELECT * FROM Products

--Calculate the total quantity ordered for each product category in the orders table.
SELECT p.category, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.category;

```

	order_id [PK] integer	customer_name character varying (255)	product_id integer	quantity integer	order_date date
1	1	Alice	1	2	2024-07-01
2	2	Bob	2	1	2024-07-02
3	3	Charlie	3	3	2024-07-03

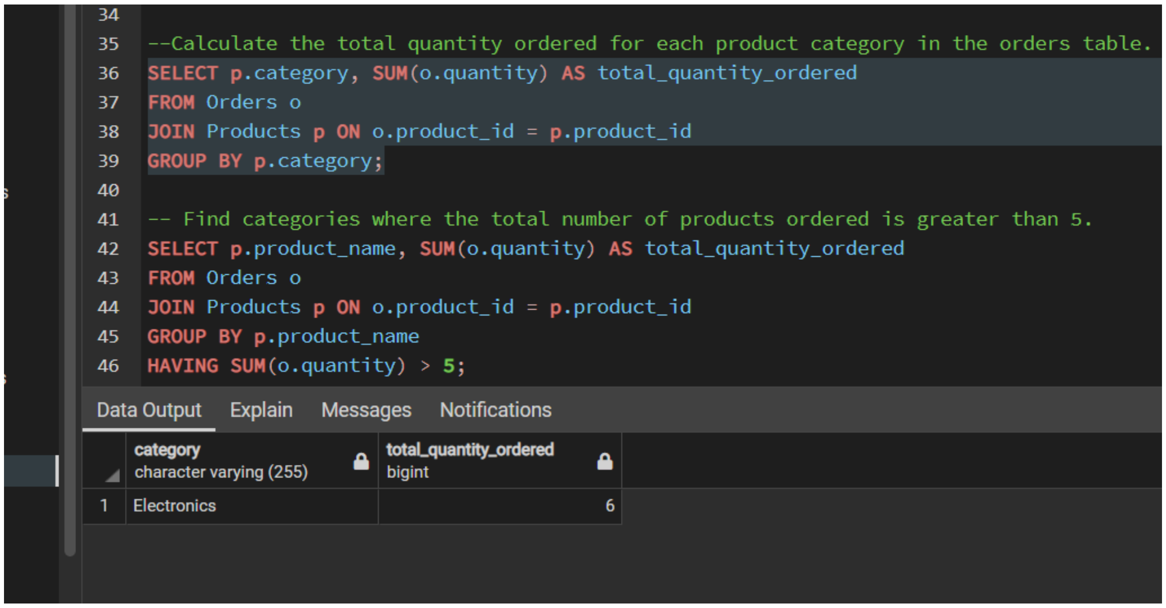
2. Products Table



```sql
--Read from tables
SELECT * FROM Orders
SELECT * FROM Products

--Calculate the total quantity ordered for each product category in the orders table.
SELECT p.category, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.category;

-- Find categories where the total number of products ordered is greater than 5.
SELECT p.product_name, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
```

	product_id [PK] integer	product_name character varying (255)	category character varying (255)	price numeric (10,2)
1	1	Laptop	Electronics	1000.00
2	2	Smartphone	Electronics	500.00
3	3	Tablet	Electronics	300.00

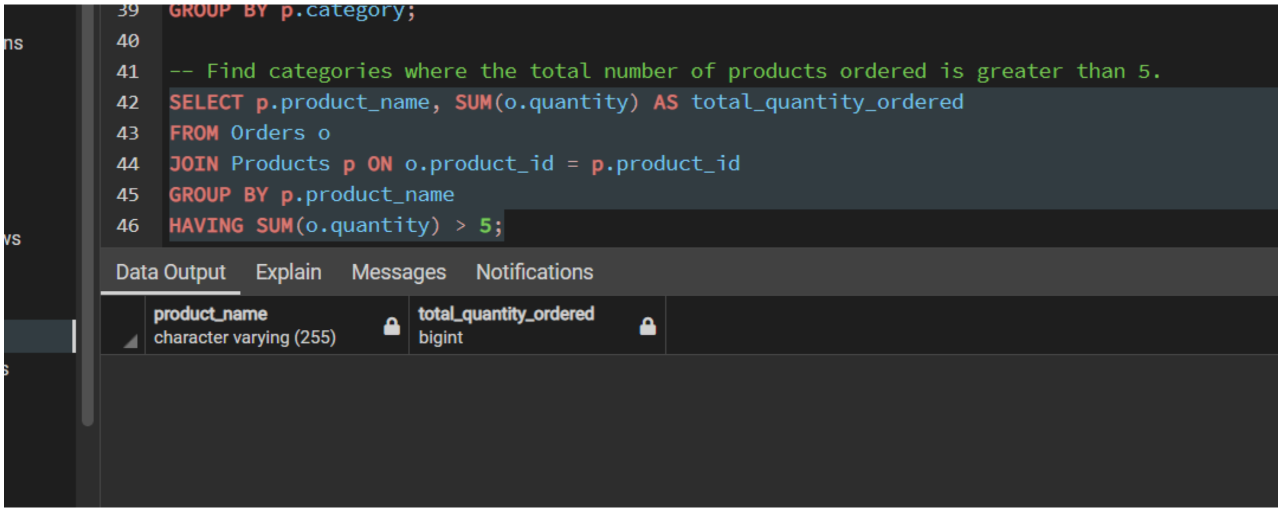
Calculate the total quantity ordered for each product category in the orders table.



```
--Calculate the total quantity ordered for each product category in the orders table.
SELECT p.category, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.category;

-- Find categories where the total number of products ordered is greater than 5.
SELECT p.product_name, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.product_name
HAVING SUM(o.quantity) > 5;
```

	category character varying (255)	total_quantity_ordered bigint
1	Electronics	6

Find categories where the total number of products ordered is greater than 5.



```
GROUP BY p.category;

-- Find categories where the total number of products ordered is greater than 5.
SELECT p.product_name, SUM(o.quantity) AS total_quantity_ordered
FROM Orders o
JOIN Products p ON o.product_id = p.product_id
GROUP BY p.product_name
HAVING SUM(o.quantity) > 5;
```

product_name character varying (255)	total_quantity_ordered bigint