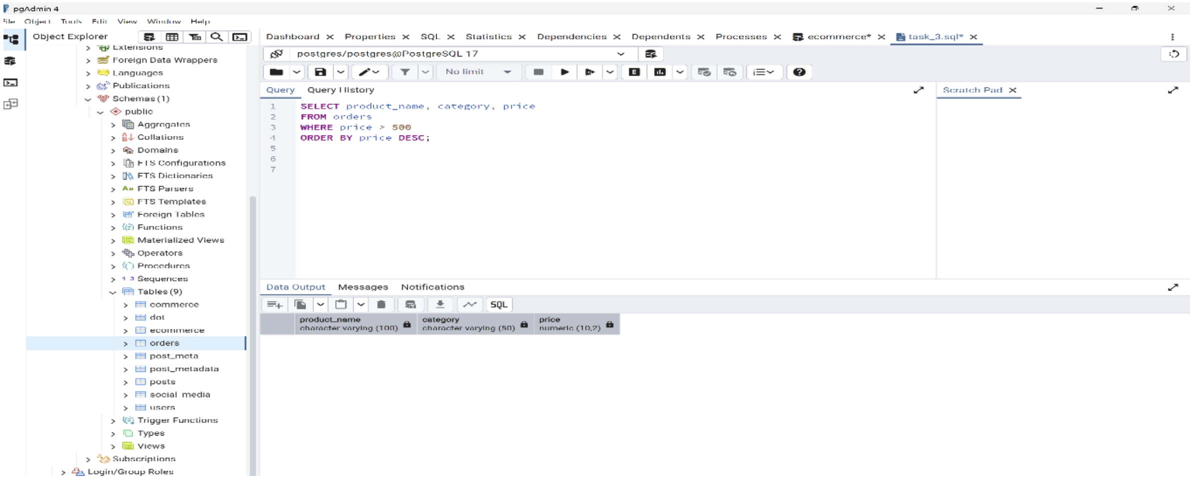
Task 3: SQL for Data Analysis

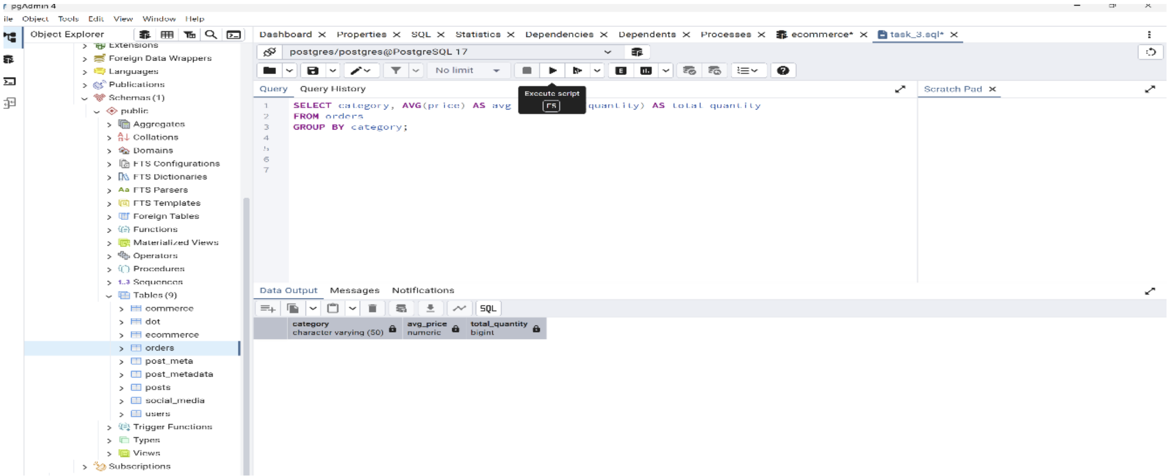
SELECT + WHERE + ORDER BY:



```sql
SELECT product_name, category, price
FROM orders
WHERE price > 500
ORDER BY price DESC;
```

product_name character varying (100)	category character varying (50)	price numeric (10,2)

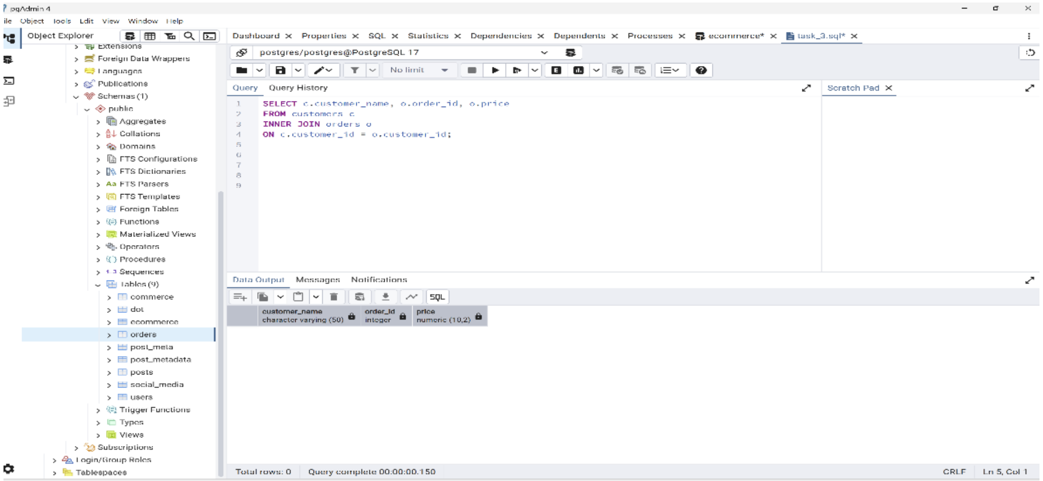
GROUP BY + SUM/AVG:



```sql
SELECT category, AVG(price) AS avg                quantity) AS total_quantity
FROM orders
GROUP BY category;
```

category character varying (50)	avg_price numeric	total_quantity bigint

JOINS (Use customers + orders tables):

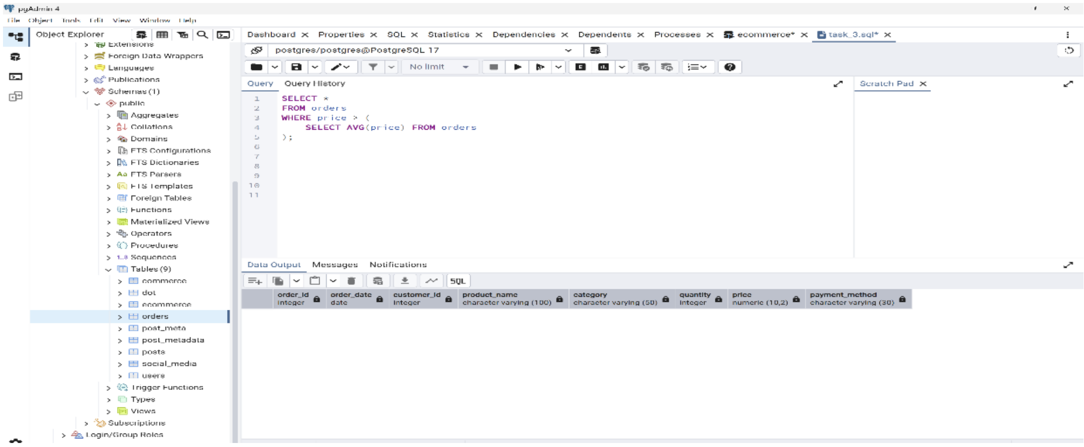


```sql
SELECT c.customer_name, o.order_id, o.price
FROM customers c
INNER JOIN orders o
ON c.customer_id = o.customer_id;
```

customer_name character varying (50)	order_id integer	price numeric (10,2)

Total rows: 0   Query complete 00:00:00.150   CRLF   Ln 5, Col 1

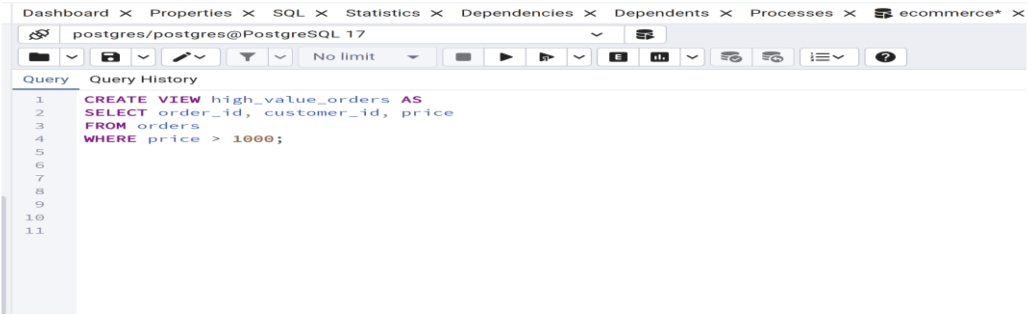
Subquery:



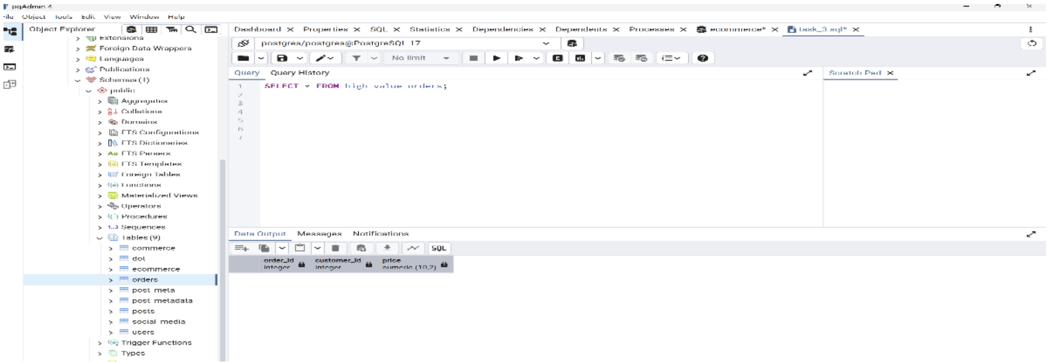
```sql
SELECT *
FROM orders
WHERE price > (
    SELECT AVG(price) FROM orders
);
```

	order_id	order_date	customer_id	product_name	category	quantity	price	payment_method
	integer	date	integer	character varying (100)	character varying (50)	integer	numeric (10,2)	character varying (30)

Create



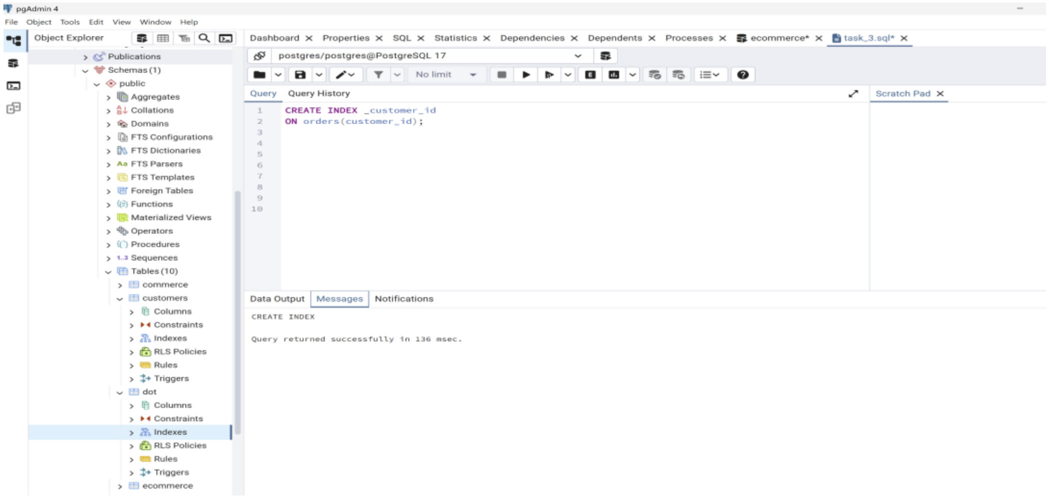
```sql
CREATE VIEW high_value_orders AS
SELECT order_id, customer_id, price
FROM orders
WHERE price > 1000;
```



```sql
SELECT * FROM high_value_orders;
```

	order_id	customer_id	price
	integer	integer	numeric (10,2)

Query Optimization (PostgreSQL Index):



```sql
CREATE INDEX _customer_id
ON orders(customer_id);
```

```
CREATE INDEX

Query returned successfully in 136 msec.
```