

## Task 3: SQL for Data Analysis

### SELECT + WHERE + ORDER BY:

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with Schemas (public) containing Tables (commerce, dot, ecommerce, orders, post\_meta, post\_metadata, posts, social\_media, users).
- Query Editor:** Displays the following SQL query:

```
1 SELECT product_name, category, price
2 FROM orders
3 WHERE price > 500
4 ORDER BY price DESC;
```
- Data Output:** Shows the results of the query in a table format:| productName | category | price |
| --- | --- | --- |
| character varying (100) | character varying (50) | numeric (10,2) |

### GROUP BY + SUM/AVG:

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with Schemas (public) containing Tables (orders, commerce, dot, ecommerce).
- Query Editor:** Displays the following SQL query:

```
1 SELECT category, AVG(price) AS avg
2 FROM orders
3 GROUP BY category;
```
- Data Output:** Shows the results of the query in a table format:| category | avg.price | total.quantity |
| --- | --- | --- |
| character varying (50) | numeric | bigint |

### JOINS (Use customers + orders tables):

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with Schemas (public) containing Tables (customers, orders).
- Query Editor:** Displays the following SQL query:

```
1 SELECT c.customer_name, o.order_id, o.price
2 FROM customers c
3 INNER JOIN orders o
4 ON c.customer_id = o.customer_id;
```
- Data Output:** Shows the results of the query in a table format:| customer\_name | order\_id | price |
| --- | --- | --- |
| character varying (50) | integer | numeric (10,2) |

## Subquery:

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with various objects like Schemas, Functions, Tables, and Views.
- SQL Editor:** Contains the following subquery:

```
1 SELECT
2   FROM orders
3   WHERE price > (
4       SELECT AVG(price) FROM orders
5   );
6
7
8
9
10
11
```

- Data Output:** Displays the results of the query, which is an empty set (0 rows).

## Create

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with various objects like Schemas, Functions, Tables, and Views.
- SQL Editor:** Contains the following SQL code to create a view:

```
1 CREATE VIEW high_value_orders AS
2 SELECT order_id, customer_id, price
3 FROM orders
4 WHERE price > 1000;
5
6
7
8
9
10
11
```

- Data Output:** Displays the results of the query, which is an empty set (0 rows).

## Query Optimization (PostgreSQL Index):

The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database schema with various objects like Publications, Schemas, Functions, Tables, and Views.
- SQL Editor:** Contains the following SQL code to create an index:

```
1 CREATE INDEX _customer_id
2 ON orders(customer_id);
3
4
5
6
7
8
9
10
```

- Data Output:** Displays the results of the query, indicating successful execution.

