UNIVERSITY INSTITUTE of COMPUTING
Asia's Fastest Growing University

NAAC GRADE A+ ACCREDITED UNIVERSITY

Worksheet-2

Name: TANUJ JOSHI
UID: 25MCA20081
Branch: MCA General
Section/Group: 25MCA_KAR-1
Semester: II
Date of Performance: 22-01-26
Subject Name: Technical Training
Subject Code: 25CAP-652

Aim of the Session:

The aim of this practical session is to implement and analyze SQL SELECT queries in PostgreSQL using filtering, sorting, grouping, and aggregation techniques.
This practical helps in understanding how data can be efficiently retrieved and analyzed from a relational database for reporting and real-world business scenarios.

Objective of the Session:

The objectives of this practical session are:

- To retrieve specific records from a table using filtering conditions
- To sort query results using single and multiple attributes
- To perform aggregation using GROUP BY clause
- To apply conditions on aggregated data using HAVING clause
- To understand the difference between row-level filtering and group-level filtering
- To gain confidence in writing SQL queries commonly asked in placement interviews

Practical / Experiment Steps:

Step 1: Database and Table Preparation

- Create a database for the experiment
- Create a table representing customer orders
- Insert sample records to perform meaningful analysis

Step 2: Filtering Data Using Conditions

- Retrieve records based on specific conditions such as higher-priced orders
- Observe how filtering reduces unnecessary data

Step 3: Sorting Query Results

- Sort records using ascending and descending order
- Apply sorting on multiple columns

Step 4: Grouping Data for Aggregation

- Group records based on a common attribute
- Calculate aggregate values such as total sales

Step 5: Applying Conditions on Aggregated Data

- Apply conditions after grouping using aggregate functions

Step 6: Conceptual Understanding of Filtering vs Aggregation Conditions

- Analyze incorrect and correct placement of conditions
- Understand SQL execution order

Procedure of the Practical:

1. Start the system and log in to the computer.
2. Open the PostgreSQL client tool (psql / pgAdmin).
3. Create a table to store customer order details.

```
CREATE TABLE customer_orders (
    order_id SERIAL PRIMARY KEY,
    customer_name VARCHAR(50),
    product VARCHAR(50),
    quantity INT,
    price NUMERIC(10,2),
    order_date DATE
);
```

4. Insert sample records into the table.

```
INSERT INTO customer_orders (customer_name, product, quantity, price, order_date)
VALUES
('Amit', 'Laptop', 1, 55000, '2025-01-10'),
('Riya', 'Mobile', 2, 30000, '2025-01-12'),
('Kunal', 'Laptop', 1, 60000, '2025-01-15'),
('Sneha', 'Tablet', 3, 45000, '2025-01-18'),
('Rahul', 'Mobile', 1, 20000, '2025-01-20'),
('Anita', 'Laptop', 2, 110000, '2025-01-22');
```

Data Output | Messages | Notifications

	order_id [PK] integer	customer_name character varying (50)	product character varying (50)	quantity integer	price numeric (10,2)	order_date date
1	1	Amit	Laptop	1	55000.00	2025-01-10
2	2	Riya	Mobile	2	30000.00	2025-01-12
3	3	Kunal	Laptop	1	60000.00	2025-01-15
4	4	Sneha	Tablet	3	45000.00	2025-01-18
5	5	Rahul	Mobile	1	20000.00	2025-01-20
6	6	Anita	Laptop	2	110000.00	2025-01-22

5. Filtering records using WHERE clause (orders with price greater than 30000).

```
SELECT *
FROM customer_orders
WHERE price > 30000;
```

Data Output | Messages | Notifications

	order_id [PK] integer	customer_name character varying (50)	product character varying (50)	quantity integer	price numeric (10,2)	order_date date
1	1	Amit	Laptop	1	55000.00	2025-01-10
2	3	Kunal	Laptop	1	60000.00	2025-01-15
3	4	Sneha	Tablet	3	45000.00	2025-01-18
4	6	Anita	Laptop	2	110000.00	2025-01-22

6. Sorting results in ascending order of price.

```
SELECT customer_name, product, price
FROM customer_orders
ORDER BY price ASC;
```

Data Output | Messages | Notifications

	customer_name character varying (50)	product character varying (50)	price numeric (10,2)
1	Rahul	Mobile	20000.00
2	Riya	Mobile	30000.00
3	Sneha	Tablet	45000.00
4	Amit	Laptop	55000.00
5	Kunal	Laptop	60000.00
6	Anita	Laptop	110000.00

7. Sorting results in descending order of price.

```
SELECT customer_name, product, price
FROM customer_orders
ORDER BY price DESC;
```

Data Output | Messages | Notifications

	customer_name character varying (50)	product character varying (50)	price numeric (10,2)
1	Anita	Laptop	110000.00
2	Kunal	Laptop	60000.00
3	Amit	Laptop	55000.00
4	Sneha	Tablet	45000.00
5	Riya	Mobile	30000.00
6	Rahul	Mobile	20000.00

8. Sorting using multiple columns.

```
SELECT customer_name, product, price
FROM customer_orders
```

ORDER BY product ASC, price DESC;

Data Output | Messages | Notifications

	customer_name character varying (50)	product character varying (50)	price numeric (10,2)
1	Anita	Laptop	110000.00
2	Kunal	Laptop	60000.00
3	Amit	Laptop	55000.00
4	Riya	Mobile	30000.00
5	Rahul	Mobile	20000.00
6	Sneha	Tablet	45000.00

9. Grouping data and calculating total sales per product.

```
SELECT product, SUM(price) AS total_sales
FROM customer_orders
GROUP BY product;
```

Data Output | Messages | Notifications

	product character varying (50)	total_sales numeric
1	Mobile	50000.00
2	Tablet	45000.00
3	Laptop	225000.00

10. Applying condition on aggregated data using HAVING.

```
SELECT product, SUM(price) AS total_sales
FROM customer_orders
GROUP BY product
HAVING SUM(price) > 50000;
```

Data Output | Messages | Notifications

	product character varying (50)	total_sales numeric
1	Laptop	225000.00

11. Verify the output after execution.
12. Note down the results obtained.
13. Save the work and take screenshots for record.

I/O Analysis:

Input Provided

- SQL queries using SELECT, WHERE, ORDER BY, GROUP BY, and HAVING
- Sample customer order data inserted into the table

Output Generated

- Filtered records based on price conditions
- Sorted output in ascending and descending order
- Aggregated results showing total sales per product
- Refined grouped results based on aggregate conditions

Learning Outcomes:

- How to retrieve relevant data using filtering conditions
- How sorting improves readability and reporting of query results
- How to group data and perform aggregation for analytical purposes
- The difference between WHERE and HAVING clauses
- The correct execution order of SQL queries
- Practical exposure to real-world SQL queries used in interviews and industry