Worksheet No. - 2

Student Name: Sujal Dhiman **UID: 25MCA20186**
Branch: MCA **Section/Group: 1A**
Semester: 2nd **Date of Performance: 6th Jan, 2026**
Subject Name: TECHNICAL TRAINING **Subject Code:**25CAP-652

Aim/Overview of the practical:

To implement and analyze SQL SELECT queries using filtering, sorting, grouping, and aggregation concepts in PostgreSQL for efficient data retrieval and analytical reporting.

Objective:

- To retrieve specific data using filtering conditions
- To sort query results using single and multiple attributes
- To perform aggregation using grouping techniques
- To apply conditions on aggregated data
- To understand real-world analytical queries commonly asked in placement interviews

Input/Apparatus Used:

- **PostgresSQL**
- **pgAdmin**

Procedure/Algorithm/Code :

```
CREATE TABLE Students (
    student_id INT,
    name VARCHAR(50),
    city VARCHAR(50),
    percentage DECIMAL(5,2)
);
INSERT INTO Students VALUES
(1, 'Arjun', 'Delhi', 92.4),
(2, 'Neha', 'Mumbai', 96.8),
```

```
(3, 'Sahil', 'Delhi', 98.3),
(4, 'Isha', 'Pune', 94.7),
(5, 'Rohit', 'Chandigarh', 97.1),
(6, 'Meena', 'Delhi', 95.9),
(7, 'Varun', 'Pune', 99.2);
--- Without Case Statement
SELECT CITY, COUNT(*) AS STUDET_COUNT
FROM Students
WHERE percentage > 95
GROUP BY city;
-- WITH CASE STATEMENT
SELECT CITY,
    SUM(CASE WHEN percentage > 95 THEN 1 ELSE 0 END) AS STUDENT_COUNTS
FROM Students
GROUP BY city;
-- (II)
SELECT CITY,
    AVG(CASE WHEN PERCENTAGE > 95 THEN PERCENTAGE ELSE NULL END) AS STUDENT_AVG
FROM Students
GROUP BY city
ORDER BY STUDENT_AVG DESC;
```

Output:

	city character varying (30)	studet_count bigint
1	Delhi	2
2	Mumbai	2

Without Case Statement

	city character varying (30)	student_counts bigint
1	Mumbai	2
2	Delhi	2

With Case Statement - I

	city character varying (30)	student_avg numeric
1	Mumbai	97.2500000000000000
2	Delhi	96.7500000000000000

With Case Statement - II

Learning outcomes (What I have learnt):

1. I have understood how data can be filtered to retrieve only relevant records from a database.
2. I have learned how sorting improves the readability and usefulness of query results in reports.
3. I have gained the ability to group data effectively for analytical purposes.
4. I have clearly learned to differentiate between row-level conditions and group-level conditions.
5. I have developed confidence in writing analytical SQL queries used in real-world scenarios.
6. I have become better prepared to answer SQL-based placement and interview questions related to filtering, grouping, and aggregation.