Worksheet No- 2

Name – Shubham Agarwal **UID –** 25MCI10091

Branch – MCA AI&ML **Section –** MAM-1(A)

Semester – 2nd **Date of performance** – 12th Jan,2026

Subject - Technical Training **Subject Code** - 25CAP-652

Aim / Overview of the Practical

To understand and implement the CASE statement in PostgreSQL for conditional data aggregation and analysis.

Theory

The CASE statement in PostgreSQL is used to perform conditional logic similar to if-else conditions. It is commonly used in SELECT queries to apply conditions while retrieving or aggregating data.

Tools Used

- PostgreSQL
- PostgreSQL (Database Server)
- pgAdmin (Graphical User Interface)
- SQL Queries (DDL, DML, DCL commands)

Objectives

- 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

Practical / Experiment Steps

Step 1: Database and Table Preparation

```
CREATE TABLE Students
(     student_id INT,     name
VARCHAR(50),     city
VARCHAR(50),
percentage DECIMAL(5,2)
);
INSERT INTO Students VALUES
(1, 'Amit', 'Delhi', 96.5),
(2, 'Riya', 'Mumbai', 94.2),
(3, 'Rahul', 'Delhi', 97.8),
(4, 'Sneha', 'Mumbai', 98.1),
(5, 'Ankit', 'Chandigarh', 95.6),
(6, 'Pooja', 'Delhi', 93.4), (7,
'Karan', 'Chandigarh', 96.2);
select * from students;
```

	student_id integer	name character varying (50)	city character varying (50)	percentage numeric (5,2)
1	1	Amit	Delhi	96.50
2	2	Riya	Mumbai	94.20
3	3	Rahul	Delhi	97.80
4	4	Sneha	Mumbai	98.10
5	5	Ankit	Chandigarh	95.60
6	6	Pooja	Delhi	93.40
7	7	Karan	Chandigarh	96.20
8	1	Amit	Delhi	96.50
9	2	Riya	Mumbai	94.20
10	3	Rahul	Delhi	97.80
11	4	Sneha	Mumbai	98.10
12	5	Ankit	Chandigarh	95.60
13	6	Pooja	Delhi	93.40
14	7	Karan	Chandigarh	96.20

2.Query Without CASE Statement

```
SELECT city, COUNT(*) AS student_count
FROM Students
WHERE percentage > 95
GROUP BY city;
```

	city character varying (50)	studet_count bigint
1	Delhi	4
2	Mumbai	2
3	Chandigarh	4

3.Query Using CASE Statement

```
SELECT city,
SUM(CASE WHEN percentage > 95 THEN 1 ELSE 0 END) AS student_count
FROM Students
GROUP BY city;
```

	city character varying (50)	student_counts bigint
1	Mumbai	2
2	Delhi	4
3	Chandigarh	4

4.Average Calculation Using CASE

```
SELECT city,
CAST(
AVG(CASE WHEN percentage > 95 THEN percentage ELSE NULL END)
AS DECIMAL(5,2)) AS student_avg
FROM Students
GROUP BY city
ORDER BY student_avg DESC;
```

	city character varying (50)	student_avg numeric (5,2)
1	Mumbai	98.10
2	Delhi	97.15
3	Chandigarh	95.90

5.Learning Outcome

1. Apply the CASE statement in PostgreSQL to perform conditional logic within SELECT queries for data analysis.
2. Differentiate between conditional filtering using WHERE clause and conditional aggregation using CASE in GROUP BY queries.
3. Compute conditional counts and averages by combining CASE statements with aggregate functions such as COUNT() and AVG().
4. Analyse student performance data city-wise, improving understanding of real-world data summarization techniques in PostgreSQL.
5. Implement efficient SQL queries for decision-based reporting, which is useful in academic and industry-level database applications.