CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

NAAC GRADE A+
Accredited University

Worksheet No. 2

Student Name: ANINDITA DHAR **UID:** 25MCA20259
Branch: MCA (GENERAL) **Section/Group:** MCA- 1-A
Semester: II **Date of Performance:** 17/01/2026

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

1.Aim/Overview of the practical:

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

2. 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

3. Input/Apparatus Used:

- PostgreSQL
- pgAdmin

4.Procedure/Algorithm/Code:

a. Database and Table Preparation

- Start the PostgreSQL server.
- Open the PostgreSQL client tool.
- Create a database for the experiment.
- Prepare a sample table representing customer orders containing details such as customer name, product, quantity, price, and order date.
- Insert sufficient sample records to allow meaningful analysis.

b. Filtering Data Using Conditions

- Execute data retrieval operations to display only those records that satisfy specific conditions, such as higher-priced orders.
- Observe how filtering limits the number of rows returned.

c. Sorting Query Results

- Retrieve selected columns from the table and arrange the output based on numerical values such as price.
- Perform sorting using both ascending and descending order.
- Apply sorting on more than one attribute to understand priority-based ordering.

d.Grouping Data for Aggregation

- Group records based on a common attribute such as product.
- Calculate aggregate values like total sales for each group.
- Analyse how multiple rows are combined into summarised results.

5. I/O Analysis (Input / Output Analysis)

```
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);

---- A) Counting students greater than 95 percentage
---Without case statement

SELECT * from Students
WHERE percentage > 95


SELECT city, COUNT (*) AS Student_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_Count from Students
Group BY city

--- B) Finding the average using the case in each city whose percentage is greater than 95

SELECT city, AVG (CASE WHEN percentage >95 THEN percentage ELSE NULL END) as Student_Average from Students
Group BY city
ORDER BY Student_Average DESC

Output:

	student_id integer	name character varying (50)	city character varying (50)	percentage numeric (5,2)
1	1	Amit	Delhi	96.50
2	3	Rahul	Delhi	97.80
3	4	Sneha	Mumbai	98.10
4	5	Ankit	Chandigarh	95.60
5	7	Karan	Chandigarh	96.20

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

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

	city character varying (50)	student_average numeric
1	Mumbai	98.1000000000000000
2	Delhi	97.1500000000000000
3	Chandigarh	95.9000000000000000

4. Learning Outcomes:

- Students understand how data can be filtered to retrieve only relevant records from a database.
- Students learn how sorting improves the readability and usefulness of query results in reports.
- Students gain the ability to group data for analytical purposes.
- Students clearly differentiate between row-level conditions and group-level conditions.
- Students develop confidence in writing analytical SQL queries used in real-world scenarios.
- Students are better prepared to answer SQL-based placement and interview questions related to filtering, grouping, and aggregation