University Institute of Engineering

Department of Computer Science & Engineering

Experiment: 2

Student Name: Jagrath **UID:24BDA70365**

Branch: CSE **Section/Group: AIT-KRG-GP2**

Semester: 4th **Date of Performance:21/01/26**

Subject NameDBMS

1. **Aim of the practical:** To understand and implement SQL SELECT queries using various clauses such as WHERE, ORDER BY, GROUP BY, and HAVING to retrieve and manipulate data efficiently from relational database tables.

2. Tool Used:

- **Database Management System:**
  - **PostgreSQL**
- **Database Administration Tool:**
  - **pgAdmin**

3. **Objective:**
   - To practice writing SQL SELECT statements.
   - To apply filtering conditions using the WHERE clause.
   - To sort query results using the ORDER BY clause.
   - To group records using the GROUP BY clause.
   - To filter grouped data using the HAVING clause.
   - To analyze data using aggregate functions like COUNT(), SUM(), AVG(), MIN(), and MAX().

4. Practical / Experimental Steps

Step 1: Open **pgAdmin / SQL shell** and connect to the PostgreSQL database.

Step 2: Create the **Students** table with appropriate columns and data types.

Step 3: Insert sample student records containing id, name, city, and marks.

Step 4: Use the **COUNT()** function with **GROUP BY city** to find the number of students in each city.

Step 5: Apply **ORDER BY** on the aggregated COUNT result to sort cities based on student count.

Step 6: Use the **HAVING** clause to filter cities having at least a specified number of students.

Step 7: Use the **AVG()** function with **GROUP BY city** to calculate the average marks of students in each city.

Step 8: Execute aggregation functions such as **SUM(), MIN(), and MAX()** to analyze student marks.

Step 9: Verify the output of each query to understand grouping and aggregation behavior in SQL.

5. I / O Analysis

DATABASE DESIGN

Query to create Table Students:

```
CREATE TABLE Students (
    id NUMERIC(10,0) PRIMARY KEY,
    name VARCHAR(50),
    city VARCHAR(30),
    marks NUMERIC(10,0)
);
```

Insert same record into the table

```
INSERT INTO Students VALUES (1, 'Aman', 'Mohali', 85);
INSERT INTO Students VALUES (2, 'Rohit', 'Mohali', 78);
INSERT INTO Students VALUES (3, 'Neha', 'Mohali', 92);
INSERT INTO Students VALUES (4, 'Simran', 'Amritsar', 88);
INSERT INTO Students VALUES (5, 'Karan', 'Amritsar', 75);
```

	id [PK] numeric (10)	name character varying (50)	city character varying (30)	marks numeric (10)
1	1	Aman	Mohali	85
2	2	Rohit	Mohali	78
3	3	Neha	Mohali	92
4	4	Simran	Amritsar	88
5	5	Karan	Amritsar	75

University Institute of Engineering

Department of Computer Science & Engineering

Group By Queries:

```
SELECT CITY ,COUNT(*) AS COUNT_STUDNETS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	count_studnets bigint
1	Mohali	3
2	Amritsar	2

```
SELECT CITY ,COUNT(ID) AS COUNT_STUDNETS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	count_studnets bigint
1	Mohali	3
2	Amritsar	2

```
SELECT CITY ,COUNT(ID) AS COUNT_STUDNETS
FROM STUDENTS
GROUP BY CITY
ORDER BY COUNT_STUDNETS ASC
```

	city character varying (30)	count_studnets bigint
1	Amritsar	2
2	Mohali	3

```
SELECT CITY ,COUNT(*) AS COUNT_STUDNETS
FROM STUDENTS
GROUP BY CITY
ORDER BY COUNT(*) ASC
```

	city character varying (30)	count_studnets bigint
1	Amritsar	2
2	Mohali	3

University Institute of Engineering

Department of Computer Science & Engineering

```
SELECT CITY ,COUNT(ID) AS COUNT_STUDNETS
FROM STUDENTS
GROUP BY CITY
HAVING COUNT(ID)>=3
```

	city character varying (30)	count_studnets bigint
1	Mohali	3

```
SELECT CITY ,AVG(MARKS)::NUMERIC(10,2) AS AVERAGE_MARKS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	average_marks numeric (10,2)
1	Mohali	85.00
2	Amritsar	81.50

```
SELECT CITY ,SUM(MARKS)::NUMERIC(10,2) AS SUBMITION_MARKS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	submition_marks numeric (10,2)
1	Mohali	255.00
2	Amritsar	163.00

```
SELECT CITY ,MIN(MARKS)::NUMERIC(10,2) AS MINIMUM_MARKS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	minimum_marks numeric (10,2)
1	Mohali	78.00
2	Amritsar	75.00

```
SELECT CITY ,MAX(MARKS)::NUMERIC(10,2) AS MAXIMUM_MARKS
FROM STUDENTS
GROUP BY CITY
```

	city character varying (30)	maximum_marks numeric (10,2)
1	Mohali	92.00
2	Amritsar	88.00

6. Learning outcomes (What I have learnt):

- Understood the concept of database table creation using appropriate data types, primary keys, and constraints to maintain data integrity.
- Gained practical knowledge of SQL aggregate functions such as COUNT, SUM, AVG, MIN, and MAX along with the use of GROUP BY and HAVING clauses.
- Developed the ability to analyze and summarize data by grouping records and sorting results using ORDER BY.
- Acquired hands-on experience in executing data manipulation operations including INSERT, UPDATE, DELETE, and SELECT queries.
- Gained practical exposure to role-based access control and database security by creating roles, granting permissions, and revoking privileges in PostgreSQL.