Student Name: Husanpreet Kaur **UID: 24BCY70138**
Branch: AIT-CSE (Cyber Security) **Section/Group: 24AIT-KRG2**
Semester: 4 **Date of Performance:16/01/2026**
Subject Name: Database Management System **Subject Code: 24CSH-298**

WORKSHEET 2

AIM: To design and implement a sample database system using DDL, DML, and DCL commands, including database creation, data manipulation, schema modification, and role-based access control to ensure data integrity and secure, read-only access for authorized users.

S/W Requirement: Oracle Database Express Edition and pgAdmin

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

Given:

Practical / Experiment Steps

Step 1: Database and Table Preparation

- Start the PostgreSQL server.
- Open the PostgreSQL client tool.
- Create a database for the experiment.
- Prepare a sample table representing employee details containing id, name, department, salary, joining date
- Insert sufficient sample records to allow meaningful analysis.

Query:

Table Creation:

```
CREATE TABLE employee1(
   emp_id NUMERIC PRIMARY KEY,
   emp_name VARCHAR(50),
   department VARCHAR(50),
```

salary NUMERIC(10,2),

joining_date DATE[ ]

);

Insertion Of Records:

INSERT INTO employee1 VALUES (101, 'Amit', 'IT', 19000, '2020-02-12');

INSERT INTO employee1 VALUES (102, 'Priya', 'HR', 22000, '2019-03-10');

INSERT INTO employee1 VALUES (103, 'Rahul', 'Sales', 35000, '2021-07-18');

INSERT INTO employee1 VALUES (104, 'Neha', 'IT', 55000, '2018-09-22');

INSERT INTO employee1 VALUES (105, 'Rohan', 'Finance', 32000, '2022-01-05');

INSERT INTO employee1 VALUES (106, 'Sara', 'Sales', 13000, '2020-12-03');

INSERT INTO employee1 VALUES (107, 'Vikram', 'HR', 12000, '2017-04-11');

SELECT * FROM employee1;

Output:

Data Output | Messages | Notifications

Showing rows: 1 to 7 | Page No: 1 of 1

	emp_id [PK] numeric	emp_name character varying (50)	department character varying (50)	salary numeric (10,2)	joining_date date
1	101	Amit	IT	19000.00	2020-02-12
2	102	Priya	HR	22000.00	2019-03-10
3	103	Rahul	Sales	35000.00	2021-07-18
4	104	Neha	IT	55000.00	2018-09-22
5	105	Rohan	Finance	32000.00	2022-01-05
6	106	Sara	Sales	13000.00	2020-12-03
7	107	Vikram	HR	12000.00	2017-04-11

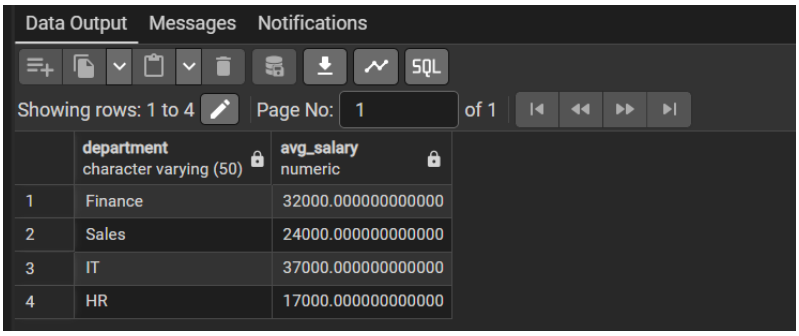
Step 2: Filtering Data Using Conditions

Query:

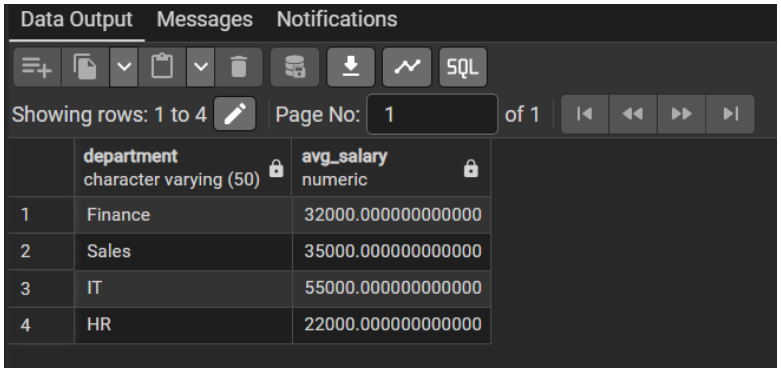
SELECT department, AVG(salary) AS avg_salary

FROM employee1

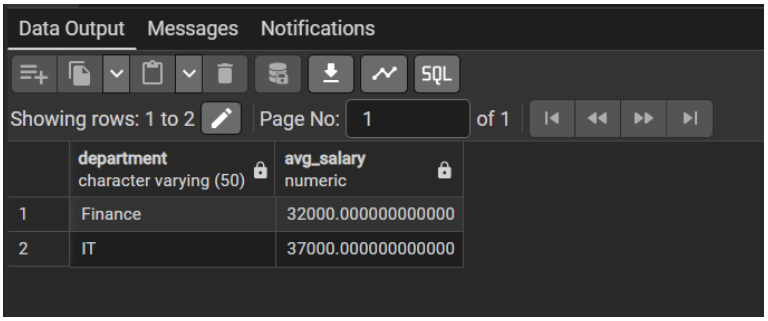
GROUP BY department;




```
SELECT department, AVG(salary) AS avg_salary
FROM employee1
WHERE salary > 20000
GROUP BY department;
```

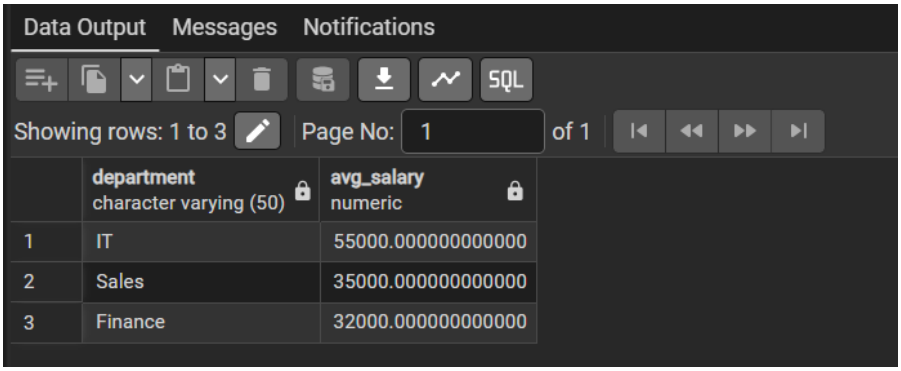



```
SELECT department, AVG(salary) AS avg_salary
FROM employee1
GROUP BY department
HAVING AVG(salary) > 30000;
```



Step 3: Sorting Query Results

```
SELECT department, AVG(salary) AS avg_salary
FROM employee1
WHERE salary > 20000
GROUP BY department
HAVING AVG(salary) > 30000
ORDER BY avg_salary DESC;
```



	department character varying (50)	avg_salary numeric
1	IT	55000.000000000000
2	Sales	35000.000000000000
3	Finance	32000.000000000000

Learning Outcomes:

- Understand how to create relational database tables using appropriate data types and constraints
- Learn to retrieve required data from a table using **row-level filtering** with the WHERE clause.
- Gain the ability to apply **column-level (group-level) filtering** using the HAVING clause.
- Clearly differentiate between **row-level filtering and group-level filtering**, and apply them correctly in real-world SQL scenarios.
- Learnt to perform grouping on the sorted data.