DATE:- 18/8/2025

TASK-3

TASK-3:- Using Clauses, operators and functions in queries perform the query processing on database for different revival results of queries using DML, DRL operations using aggregate, date, String, indent, functions, Set Clauses and operators. answer for this using employee database.

Employee data base:-

Emp_Id	Emp_Name	Dept	Salary	JoiningDate	City.
101	AliceJohnson	It	70000	2023-5-01	New York
102	Bobsmith	HR	55000	2018-3-15	chicago
103	Carol white	Finance	80000	2016-11-23	San Fransisco
104	David Brown	IT	75000	2020-7-01	New York
105	Eve Davis	Marketing	60000	2019-2-28	Los-Angeles
106	Frank Miller	IT	72000	2021-8-16	Baston
107	Grace Lee	HR	52000	2017-4-5	chicago
108	Henry wilson	Finance	81000	2015-9-30	San Fransisco
109	Isabel clark	Marketing	63000	2022-1-12	Los-Angles
110	Jack Turner	IT	68000	2024-6-19	New York.

3.1 Perform DML operations:-

a) Insert new employee

INSERT INTO Employee (Emp_ID, Emp_name, Dept, Salary, Joining Date), VALUES (111, 'Sophia Green', 'Finance', 9200.00, '2024-03-10', 'Houston');

b) Update Salary of employees in IT department by 10%.

Update Employee SET Salary = Salary * 1.10 WHERE Dept = 'IT';

c) Delete employees who joined before 2015.

DELETE FROM Employee WHERE Joining Date < '2015-01-01';

3.2 DRL queries using clauses, operators and Functions

a) Retrieve employees with Salary above average Salary.

SELECT EmpName, Salary FROM Employee WHERE Salary > (SELECT (AVG (Salary) FROM Employee);

b) Display employees with their years of service.

SELECT * FROM Employee WHERE EmpName LIKE 'A%';

c) Retrive employees whose name starts with 'A'.

SELECT * FROM Employee WHERE EmpName LIKE 'A%';

d) Retrive total Salary per department.

SELECT Dept, Sum(Salary) AS Total Salary FROM Employee GROUP BY Dept;

e) Retrive employees joined in the last 2 years.

SELECT * FROM Employee WHERE Joining Date >= DATE_SUB (CURDATE (), INTERVAL 2 YEAR);

f) use CASE operator to classify employees by Salary.

SELECT EmpName, Salary, CASE WHEN Salary >= 80000 THEN 'High Salary' WHEN Salary BETWEEN 60000 AND 79999 THEN 'Medium Salary' ELSE 'Low Salary' END AS Salary Category FROM Employee;

3.3 Set operators Examples (using two tables : Employee and New Employee)

a) Combine employees from both tables without duplicates (UNION)

SELECT EmpName FROM Employee UNION SELECT EmpName FROM NewEmployee;

b) Find employees Common in both tables (INTERSECT)

```
SELECT EmpName FROM Employee WHERE EmpName IN (SELECT
EmpName FROM New Employee);
```

c) Find employees in Employee but not in New Employee (MINUS/EXCEPT).

```
SELECT EmpName FROM Employee WHERE EmpName NOT IN
(SELECT Emp Name FROM Neco Employee);
```

3.4 Using String Functions.

a) Concatenate employee name and city

```
SELECT CONCAT (EmpName, '_', city) AS Name coith city.
FROM Employee;
```

b) Find employees coith name length greater than 6.

```
SELECT EmpName FROM Employee WHERE LENGTH (EmpName)
>6;
```

VEL TECH - CSE	
EX NO.	3
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15
SIGN WITH DATE	15/5/25

Result:- Hence, the operations using clauses, operators and functions in queries is done successfully.