

TASK-3

TASK-3:- using clauses, operators & functions in queries
Perform the query processing on database for different
Result of queries using DML, DRL operations
using aggregate, date, string, indent functions set
clauses & operations. answer for this using employee
database.

Employee database:-

Em-id	Empname	Dept	Salary	Joiningdate	city
101	Alice Johnson	IT	70000.00	2023-05-01	New York
102	Bob Smith	HR	55000.00	2018-03-15	Chicago
103	Carol White	Finance	80000.00	2016-11-23	San Francisco
104	David Brown	IT	72000.00	2021-08-10	New York
105	Eve Davis	Marketing	60000.00	2019-02-28	Los Angeles
106	Frank Miller	IT	72000.00	2021-08-16	Boston
107	Grace Lee	HR	52000.00	2017-04-05	Chicago
108	Henry Wilson	Finance	81000.00	2015-09-30	San Francisco
109	Isabel Clark	Marketing	63000.00	2022-01-12	Los Angeles
110	Jack Fisher	IT	68000.00	2014-06-19	New York

3.1. Perform DML operations :-

Insert new employee:

INSERT INTO Employee(EmplD, EmpName, Dept, Salary, Joining Date, City)
VALUES(111, 'Sophia Green', 'Finance', 72000.00, '2024-08-10',
'Houston');

update salary of employee in IT department by 10%:-

UPDATE Employee

SET salary = salary * 1.10

WHERE Dept = 'IT';

Delete employee who joined before 2015:-

DELETE FROM Employee

WHERE joiningdate < '2015-01-01';

3.2 DQL queries using clauses, operators and functions:-
a. Retrieve employee with salary above avg salary:-

SELECT EmpName, salary

FROM Employee

WHERE salary > (SELECT AVG(salary) FROM Employee);

b. Display employees with their years of service:-

SELECT EmpName, TIMESTAMPDIFF(year, joiningdate,
CURDATE()) AS yearsOfService

FROM Employee;

c. Retrieve employees whose name starts with 'A':-

SELECT *

FROM Employee

WHERE EmpName LIKE 'A%';

d. Retrieve total salary per department:-

SELECT Dept, SUM(Salary) AS totalSalary

FROM Employee

GROUP BY Dept;

e. Retrieve employee joined in the last 2 years

SELECT *

FROM Employee

WHERE joiningdate >= 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 salaryCategory

FROM Employee;

3.3 set operators Examples:-

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 NewEmployee);

3.4 using string functions:-

a. concatenate CONCAT(employee name and city)

SELECT CONCAT(EmpName, ' - ', city) AS NameWithCity

FROM Employee;

b. Find employees with name length greater than 6

SELECT EmpName

FROM Employee

WHERE LENGTH(EmpName) > 6;

Result:- Hence study and execution

of DQL, DML, set operators are

successfully verified.

LEVEL 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	25/1/25