Practical Number: 7

Title of the Exercise : Querying the database based on join opertation

a)Simple join and Self join b)Outer join and Inner join

Date of the Exercise :

OBJECTIVE (AIM) OF THE EXPERIMENT

To perform nested Queries and joining Queries using DML command.

b) Procedure for doing the experiment:

Step no.	Details of the step
1	**Relating Data through Join Concept** The purpose of a join concept is to combine data spread across tables. A join is actually performed by the „where" clause which combines specified rows of tables. Syntax; select columns from table1, table2 where logical expression; **Types of Joins** 1. Simple Join 2. Self Join 3. Outer Join 4. Inner Join
2	1. Simple Join **a) Equi-join:** A join, which is based on equalities, is called equi-join. **b) Non Equi-join:** It specifies the relationship between **Table Aliases** Table aliases are used to make multiple table queries shorted and more readable. We give an alias name to the table in the „from" clause and use it instead of the name throughout the query.
3	**Self join:** Joining of a table to itself is known as self-join. It joins one row in a table to another. It can compare each row of the table to itself and also with other rows of the same table.
	Outer Join: It extends the result of a simple join. An outer join returns all the rows returned by simple join as well as those rows from one table that do not match any row from the table. The symbol (+) represents outer join. **Inner join:** Inner join returns the matching rows from the tables that are being joined

c) Simple Join

a) Equi-join

Example: select * from item, cust where item.id=cust.id;

b) Non Equi-join

Example: select * from item, cust where item.id<cust.id;

Self join

Example: select * from emp x ,emp y where x.salary >= (select avg(salary) from x.emp where x. deptno =y.deptno);

Outer Join

Example: select ename, job, dname from emp, dept where emp.deptno (+) = dept.deptno;

d) Queries:

e) Use select from clause.

1. Use like operator to match job and in select clause to get the result.

Consider the following Tables:
EMPLOYEE(Emp_id, EMP_name,Job_name,Manager_id,Hire_date,Salary,Deptno)
DEPARTMENT(Deptno, Dname, MGRSSN)
PROJECT(Pname,Pno,Plocation,Deptno)

emp_id	emp_name	job_name	manager_id	hire_date	salary	E_Bonus	dep_no
68319	KAYLING	PRESIDENT		1991-11-18	6000.00	300.00	1001
66928	BLAZE	MANAGER	68319	1991-05-01	2750.00	200.00	3001
67832	CLARE	MANAGER	68319	1991-06-09	2550.00	200.00	1001
65646	JONAS	MANAGER	68319	1991-04-02	2957.00	200.00	2001
67858	SCARLET	ANALYST	65646	1997-04-19	3100.00	250.00	2001
69062	FRANK	ANALYST	65646	1991-12-03	3100.00	250.00	2001
63679	SANDRINE	CLERK	69062	1990-12-18	900.00	150.00	2001
64989	ADELYN	SALESMAN	66928	1991-02-20	1700.00	180.00	3001
65271	WADE	SALESMAN	66928	1991-02-22	1350.00	180.00	3001
66564	MADDEN	SALESMAN	66928	1991-09-28	1350.00	180.00	3001
68454	TUCKER	SALESMAN	66928	1991-09-08	1600.00	180.00	3001
68736	ADNRES	CLERK	67858	1997-05-23	1200.00	150.00	2001
69000	JULIUS	CLERK	66928	1991-12-03	1050.00	150.00	3001
69324	MARKER	CLERK	67832	1992-01-23	1400.00	150.00	1001

Department Table

deptno	dname	Citylocation	dCountry
1001	Accounting	New York	United States of America,
2001	Research	Dallas	United States
3001	Sales	Chicago	United States of America
4001	Marketing	Los Angeles	United States

Project Table

Pno	Pname	PCitylocation	Dept No
111	P_1	New York	1001
112	P_2	Dallas	1001
113	P_3	Chicago	2001
114	P_4	Denmark	2001
115	P_5	Paris	3001
116	P_6	Chicago	3001
117	P_7	Paris	4001

Write a query for the following:-

Q.1 Display the max salaries for each designation ordered in descending order

```
mysql> use employee_data;
Database changed
mysql> SELECT job_name, MAX(salary) AS max_salary FROM employee_160 GROUP BY job_name ORDER BY max_salary DESC;
+-----------+------------+
| job_name  | max_salary |
+-----------+------------+
| PRESIDENT |    6000.00 |
| ANALYST   |    3100.00 |
| MANAGER   |    2957.00 |
| SALESMAN  |    1700.00 |
| CLERK     |    1400.00 |
+-----------+------------+
5 rows in set (0.00 sec)
```

.

Q.2 Display the employees where salary is more than their manager.

```
mysql> SELECT e1.Emp_id, e1.Emp_name, e1.Salary AS Employee_Salary, e2.Emp_id AS Manager_id, e2.Salary AS Manager_Salary FROM employee_160 e1 LEFT JOIN employee_160 e2 ON e1.Manager_id = e2.Emp_id WHERE e1.Salar
y > e2.Salary;
+--------+----------+-----------------+------------+----------------+
| Emp_id | Emp_name | Employee_Salary | Manager_id | Manager_Salary |
+--------+----------+-----------------+------------+----------------+
|  69062 | FRANK    |         3100.00 |      65646 |        2957.00 |
|  67858 | SCARLET  |         3100.00 |      65646 |        2957.00 |
+--------+----------+-----------------+------------+----------------+
2 rows in set (0.00 sec)
```

Q.3 Display the project details for sales department.

```
mysql> SELECT p.Pno, p.Pname, p.PCityLocation, p.DeptNo FROM project_160 p JOIN department_160 d ON p.DeptNo = d.deptno WHERE d.dname = 'Sales';
+-----+-------+---------------+--------+
| Pno | Pname | PCityLocation | DeptNo |
+-----+-------+---------------+--------+
| 115 | P_5   | Paris         |   3001 |
| 116 | P_6   | Chicago       |   3001 |
+-----+-------+---------------+--------+
2 rows in set (0.00 sec)
```

Q.4 Display the name and salaries of employees working in department at location Chicago.

```
mysql> SELECT employee_101.Emp_name, employee_160.Salary FROM employee_160 JOIN department_160 ON employee_160.dep_no = department_160.deptno WHERE department_160.Citylocation LIKE '%Chicago%';
+----------+---------+
| Emp_name | Salary  |
+----------+---------+
| BLAZE    | 2750.00 |
| ADELYN   | 1700.00 |
| WADE     | 1350.00 |
| MADDEN   | 1350.00 |
| TUCKER   | 1600.00 |
| JULIUS   | 1050.00 |
+----------+---------+
6 rows in set (0.00 sec)
```

Q.5 Find the project location for employees working in department Research.

```
mysql> SELECT e.Emp_id, e.Emp_name, d.dname AS Department, p.Pname AS Project, p.PCityLocation AS ProjectLocation FROM employee_160 e JOIN department_160 d ON e.dep_no = d.deptno JOIN project_160 p ON d.deptno =
 p.DeptNo WHERE d.dname = 'Research';
+--------+----------+------------+---------+----------------+
| Emp_id | Emp_name | Department | Project | ProjectLocation |
+--------+----------+------------+---------+----------------+
|  65646 | JONAS    | Research   | P_4     | Denmark        |
|  65646 | JONAS    | Research   | P_3     | Chicago        |
|  67858 | SCARLET  | Research   | P_4     | Denmark        |
|  67858 | SCARLET  | Research   | P_3     | Chicago        |
|  69062 | FRANK    | Research   | P_4     | Denmark        |
|  69062 | FRANK    | Research   | P_3     | Chicago        |
|  63679 | SANDRINE | Research   | P_4     | Denmark        |
|  63679 | SANDRINE | Research   | P_3     | Chicago        |
|  68736 | ADNRES   | Research   | P_4     | Denmark        |
|  68736 | ADNRES   | Research   | P_3     | Chicago        |
+--------+----------+------------+---------+----------------+
10 rows in set (0.00 sec)
```

Q.6 Display the names of departments having same project location.

```
mysql> SELECT d.dname AS Department_Name FROM department_160 d JOIN project_160 p ON d.deptno = p.DeptNo GROUP BY d.dname HAVING COUNT(DISTINCT p.PCityLocation) > 1;
+-----------------+
| Department_Name |
+-----------------+
| Accounting      |
| Research        |
| Sales           |
+-----------------+
3 rows in set (0.00 sec)
```

Q.7 Display the employee details who working on project p_3 and p_6.

```
mysql> SELECT e.* FROM employee_160 e JOIN project_160 p ON e.dep_no = p.DeptNo WHERE p.Pname IN ('P_3', 'P_6');
+--------+----------+----------+------------+------------+---------+---------+--------+
| Emp_id | Emp_name | Job_name | Manager_id | hire_date  | Salary  | E_bonus | dep_no |
+--------+----------+----------+------------+------------+---------+---------+--------+
|  66928 | BLAZE    | MANAGER  |      68319 | 1991-05-01 | 2750.00 |  200.00 |   3001 |
|  65646 | JONAS    | MANAGER  |      68319 | 1991-04-02 | 2957.00 |  200.00 |   2001 |
|  67858 | SCARLET  | ANALYST  |      65646 | 1997-04-19 | 3100.00 |  250.00 |   2001 |
|  69062 | FRANK    | ANALYST  |      65646 | 1991-12-03 | 3100.00 |  250.00 |   2001 |
|  63679 | SANDRINE | CLERK    |      69062 | 1990-12-18 |  900.00 |  150.00 |   2001 |
|  64989 | ADELYN   | SALESMAN |      66928 | 1991-02-20 | 1700.00 |  180.00 |   3001 |
|  65271 | WADE     | SALESMAN |      66928 | 1991-02-22 | 1350.00 |  180.00 |   3001 |
|  66564 | MADDEN   | SALESMAN |      66928 | 1991-09-28 | 1350.00 |  180.00 |   3001 |
|  68454 | TUCKER   | SALESMAN |      66928 | 1991-09-08 | 1600.00 |  180.00 |   3001 |
|  68736 | ADNRES   | CLERK    |      67858 | 1997-05-23 | 1200.00 |  150.00 |   2001 |
|  69000 | JULIUS   | CLERK    |      66928 | 1991-12-03 | 1050.00 |  150.00 |   3001 |
+--------+----------+----------+------------+------------+---------+---------+--------+
11 rows in set (0.00 sec)
```

Q.8Display the department names handling more than one project.

```
mysql> SELECT d.dname AS DepartmentName FROM department_160 d JOIN project_160 p ON d.deptno = p.DeptNo GROUP BY d.dname HAVING COUNT(p.Pno) > 1;
+----------------+
| DepartmentName |
+----------------+
| Accounting     |
| Research       |
| Sales          |
+----------------+
3 rows in set (0.00 sec)
```