

Group B : Assignment 7**1) Perform All types of Join:****a. Cross Join**

```
MySQL> SELECT EMP.EMPLOYEE_ID AS E_ID, L.FIRST_NAME, LOAN_AMOUNT FROM
EMPLOYEE EMP CROSS JOIN LOAN L;
```

E_ID	FIRST_NAME	LOAN_AMOUNT
7369	Jon	50000
7369	Kevin	10000
7369	Leslie	100000
7499	Jon	50000
7499	Kevin	10000
7499	Leslie	100000
7505	Jon	50000
7505	Kevin	10000
7505	Leslie	100000
7506	Jon	50000
7506	Kevin	10000
7506	Leslie	100000
7507	Jon	50000
7507	Kevin	10000
7507	Leslie	100000
7521	Jon	50000
7521	Kevin	10000
7521	Leslie	100000

```
18 rows in set (0.003 sec)Query OK, 1 row affected (0.003 sec)
```

b. Left Outer Join

```
MySQL> SELECT EMP.EMPLOYEE_ID AS E_ID, L.FIRST_NAME, LOAN_AMOUNT FROM
EMPLOYEE EMP LEFT OUTER JOIN LOAN L ON EMP.EMPLOYEE_ID =
L.EMPLOYEE_ID;
```

E_ID	FIRST_NAME	LOAN_AMOUNT
7369	Jon	50000
7499	NULL	NULL
7505	NULL	NULL
7506	NULL	NULL
7507	Leslie	100000
7521	NULL	NULL

```
6 rows in set (0.003 sec)
```

c. Right Outer Join

```
MariaDB MySQL> SELECT EMP.EMPLOYEE_ID AS E_ID, L.FIRST_NAME,
LOAN_AMOUNT FROM EMPLOYEE EMP RIGHT OUTER JOIN LOAN L ON
EMP.EMPLOYEE_ID = L.EMPLOYEE_ID;
```

E_ID	FIRST_NAME	LOAN_AMOUNT
------	------------	-------------

```

| 7369 | Jon      |      50000 |
| NULL | Kevin    |      10000 |
| 7507 | Leslie   |     100000 |
+-----+-----+-----+
3 rows in set (0.001 sec)

```

d. Inner Join

```

MySQL> SELECT EMP.EMPLOYEE_ID AS E_ID, L.FIRST_NAME, LOAN_AMOUNT FROM
EMPLOYEE EMP INNER JOIN LOAN L ON EMP.EMPLOYEE_ID = L.EMPLOYEE_ID;
+-----+-----+-----+
| E_ID | FIRST_NAME | LOAN_AMOUNT |
+-----+-----+-----+
| 7369 | Jon      |      50000 |
| 7507 | Leslie   |     100000 |
+-----+-----+-----+
2 rows in set (0.002 sec)

```

e. Equijoin

```

MySQL> SELECT E.EMPLOYEE_ID, E.FIRST_NAME, L.LOAN_AMOUNT FROM
EMPLOYEE E, LOAN L WHERE E.EMPLOYEE_ID = L.EMPLOYEE_ID;
+-----+-----+-----+
| EMPLOYEE_ID | FIRST_NAME | LOAN_AMOUNT |
+-----+-----+-----+
|          7369 | Jon      |      50000 |
|          7507 | Leslie   |     100000 |
+-----+-----+-----+
2 rows in set (0.002 sec)

```

2) Set Operations :**a. Union**

```

MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE UNION
SELECT EMPLOYEE_ID, FIRST_NAME FROM LOAN;
+-----+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+
|          7369 | Jon      |
|          7499 | Kevin    |
|          7505 | Jean     |
|          7506 | Lynn     |
|          7507 | Leslie   |
|          7521 | Cynthia  |
|          7469 | Kevin    |
+-----+-----+
7 rows in set (0.002 sec)

```

a. Intersect

```

MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE INTERSECT
SELECT EMPLOYEE_ID, FIRST_NAME FROM LOAN;
+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+

```

```
|          7369 | Jon          |
|          7507 | Leslie       |
+-----+-----+
2 rows in set (0.002 sec)
```

3) Find out no of employees working in 'Sales' department:

```
MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE E INNER
JOIN JOB J ON E.JOB_ID = J.JOB_ID AND J.FUNCTION =
'SALEPERSON';
+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+
|          7499 | Kevin      |
|          7521 | Cynthia    |
+-----+-----+
2 rows in set (0.002 sec)
```

4) Find employees who are not working in dept 10 or 30:

```
MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE E INNER
JOIN DEPARTMENT D ON E.DEPARTMENT_ID = D.DEPARTMENT_ID AND
D.DEPARTMENT_ID NOT IN (10, 30);
+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+
|          7499 | Kevin      |
|          7505 | Jean       |
|          7507 | Leslie     |
+-----+-----+
3 rows in set (0.002 sec)
```

5) Employees list in order with descending salary:

```
MySQL> SELECT EMPLOYEE_ID, LAST_NAME FROM EMPLOYEE ORDER BY
SALARY DESC;
+-----+-----+
| EMPLOYEE_ID | LAST_NAME |
+-----+-----+
|          7505 | Doyle     |
|          7506 | Dennis    |
|          7507 | Baker     |
|          7499 | Allen     |
|          7521 | Wark      |
|          7369 | Smith     |
+-----+-----+
6 rows in set (0.002 sec)
```

6) Employees working in different department:

```
MySQL> SELECT FIRST_NAME, COUNT(DEPARTMENT_ID) AS C FROM
EMPLOYEE GROUP BY DEPARTMENT_ID HAVING C =1;
+-----+-----+
```

```

| FIRST_NAME | C |
+-----+---+
| Lynn      | 1 |
| Leslie    | 1 |
+-----+---+
2 rows in set (0.002 sec)

```

7) List department in which 4 employees are working:

```

MySQL> SELECT DEPARTMENT_ID, COUNT(DEPARTMENT_ID) AS C FROM
EMPLOYEE GROUP BY DEPARTMENT_ID HAVING C = 4;
Empty set (0.002 sec)

```

8) Display employee whose salary is maximum:

```

MySQL> SELECT * FROM EMPLOYEE WHERE SALARY = (SELECT MAX(SALARY) FROM EMPLOYEE);
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| EMPLOYEE_ID | LAST_NAME | FIRST_NAME | MIDDLE_NAME | JOB_ID | MANAGER_ID | HIRE_DATE | SALARY | DEPARTMENT_ID |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 7505 | Doyle | Jean | K | 671 | 7839 | 1985-04-04 | 2850 | 20 |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.002 sec)

```

9) Update employee salary by 10% whose job is clerk:

```

MySQL> UPDATE EMPLOYEE, JOB SET SALARY = (SALARY +
(SALARY/100)*10) WHERE EMPLOYEE.JOB_ID = JOB.JOB_ID AND
JOB.FUNCTION = 'Clerk';
Query OK, 1 row affected (0.087 sec)
Rows matched: 1 Changed: 1 Warnings: 0

```

10) Delete employees who are not working in accounting:

```

MySQL> DELETE FROM EMPLOYEE WHERE DEPARTMENT_ID NOT IN (SELECT
DEPARTMENT_ID FROM DEPARTMENT WHERE NAME = 'ACCOUNTING');
Query OK, 4 rows affected (0.091 sec)

```

11) Department in which no any employee works:

```

MySQL> SELECT NAME FROM DEPARTMENT WHERE DEPARTMENT_ID NOT IN
(SELECT DEPARTMENT_ID FROM EMPLOYEE);
+-----+
| NAME |
+-----+
| Research |
| Sale |
| Operation |

```

12) Show department wise salary:

```

MySQL> SELECT DEPARTMENT_ID, MAX(SALARY), MIN(SALARY),
AVG(SALARY) FROM EMPLOYEE GROUP BY DEPARTMENT_ID;
+-----+-----+-----+-----+
| DEPARTMENT_ID | MAX(SALARY) | MIN(SALARY) | AVG(SALARY) |
+-----+-----+-----+-----+
| 10 | 1250 | 800 | 1025 |
| 20 | 2850 | 1600 | 2225 |

```

```

|          30 |          2750 |          2750 |          2750 |
|          40 |          2200 |          2200 |          2200 |
+-----+-----+-----+-----+
3 rows in set (0.002 sec)

```

13) Find out employees who joined in year 1985:

```

MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE WHERE
YEAR(HIRE_DATE) = '1985';
+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+
|          7505 | Jean       |
|          7507 | Leslie     |
|          7521 | Cynthia    |
+-----+-----+
3 rows in set (0.015 sec)

```

14) Find out employees working in New York:

```

MySQL> SELECT EMPLOYEE_ID, FIRST_NAME FROM EMPLOYEE WHERE
DEPARTMENT_ID = (SELECT DEPARTMENT_ID FROM DEPARTMENT WHERE
LOCATION_ID = (SELECT LOCATION_ID FROM LOCATION WHERE
REGIONAL_GROUP = 'New York'));
+-----+-----+
| EMPLOYEE_ID | FIRST_NAME |
+-----+-----+
|          7369 | Jon        |
|          7521 | Cynthia    |
+-----+-----+
2 rows in set (0.002 sec)

```

15) Find out employees with dept name:

```

MySQL> SELECT E.FIRST_NAME, E.DEPARTMENT_ID, D.NAME FROM
EMPLOYEE E INNER JOIN DEPARTMENT D ON E.DEPARTMENT_ID =
D.DEPARTMENT_ID;
+-----+-----+-----+
| FIRST_NAME | DEPARTMENT_ID | NAME          |
+-----+-----+-----+
| Jon        | 10            | Accounting    |
| Cynthia    | 10            | Accounting    |
| Kevin      | 20            | Research      |
| Jean       | 20            | Research      |
| Lynn       | 30            | Sale          |
| Leslie     | 40            | Operation     |
+-----+-----+-----+
6 rows in set (0.002 sec)

```