LAB-5 Nested and Join Queries

With the ability to nest a query, we can combine queries to get the desired result in a single uber query rather than executing each constituent query individually. Nested queries are generally slower but more readable and expressive than equivalent join queries. Also, in some situations nested queries are the only way to retrieve desired information from a database.

A MySQL subquery is a query nested within another query such as SELECT, INSERT, UPDATE and DELETE. Also, a subquery can be nested within another subquery.
A MySQL subquery is called an inner query while the query that contains the subquery is called an outer query. A subquery can be used anywhere that expression is used and must be closed in parentheses.

MySQL JOINS are used with SELECT statement. It is used to retrieve data from multiple tables. It is performed whenever you need to fetch records from two or more tables.
There are three types of MySQL joins:

- MySQL INNER JOIN (or sometimes called simple join)
- MySQL LEFT OUTER JOIN (or sometimes called LEFT JOIN)
- MySQL RIGHT OUTER JOIN (or sometimes called RIGHT JOIN)

MySQL Inner JOIN (Simple Join)
The MySQL Inner Join is used to return all rows from multiple tables where the join condition is satisfied. It is the most common type of join.
Syntax:

1. **SELECT** columns
2. **FROM** table1
3. **INNER** JOIN table2
4. **ON** table1.**column** = table2.**column**;

MySQL Left Outer Join
The LEFT OUTER JOIN returns all rows from the left hand table specified in the ON condition and only those rows from the other table where the join condition is fulfilled.
Syntax:

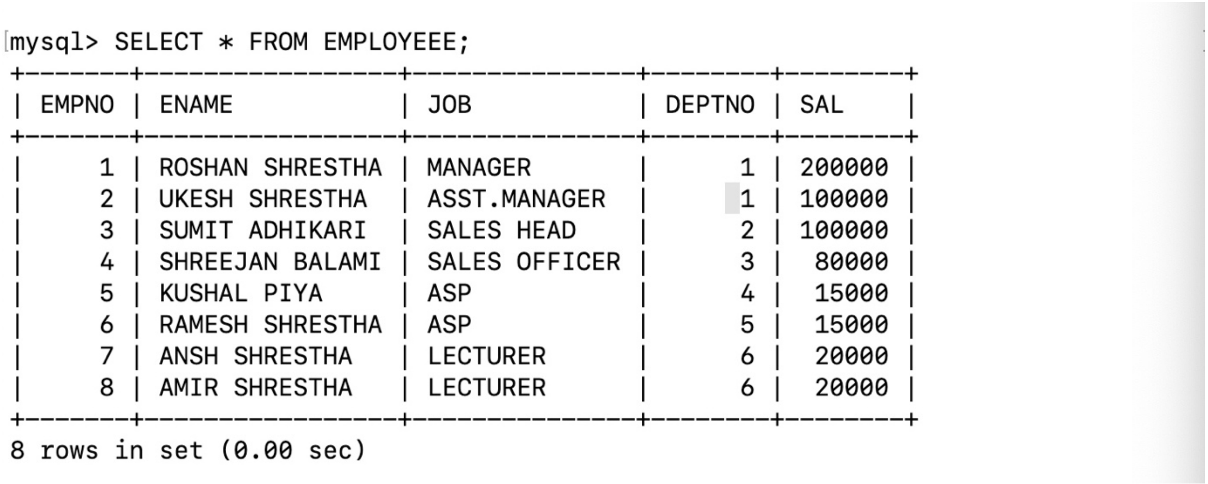
1. **SELECT** columns
2. **FROM** table1
3. LEFT [OUTER] JOIN table2
4. **ON** table1.**column** = table2.**column**;

MySQL Right Outer Join
The MySQL Right Outer Join returns all rows from the RIGHT-hand table specified in the ON condition and only those rows from the other table where he join condition is fulfilled.
Syntax:

1. **SELECT** columns
2. **FROM** table1
3. RIGHT [OUTER] JOIN table2
4. **ON** table1.**column** = table2.**column**;

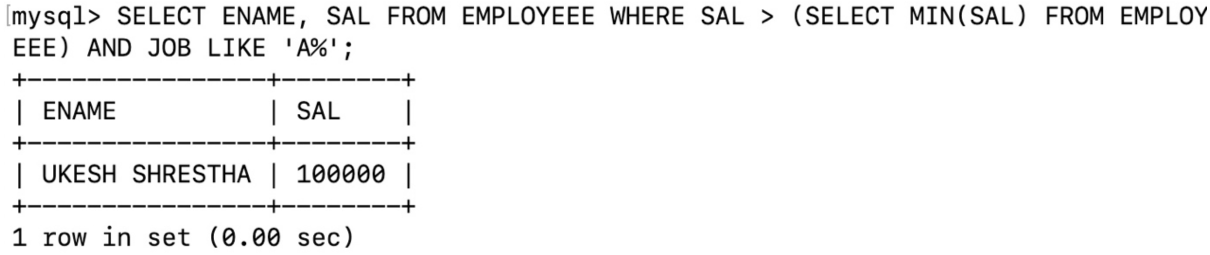
a. **Display all employee names and salary whose salary is greater than minimum salary of the company and job title starts with ‘A’.**

SELECT * FROM EMPLOYEEE;



```
[mysql> SELECT * FROM EMPLOYEEE;
+-------+-----------------+---------------+--------+--------+
| EMPNO | ENAME           | JOB           | DEPTNO | SAL    |
+-------+-----------------+---------------+--------+--------+
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |
|     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |
|     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |
|     7 | ANSH SHRESTHA   | LECTURER      |      6 |  20000 |
|     8 | AMIR SHRESTHA   | LECTURER      |      6 |  20000 |
+-------+-----------------+---------------+--------+--------+
8 rows in set (0.00 sec)
```

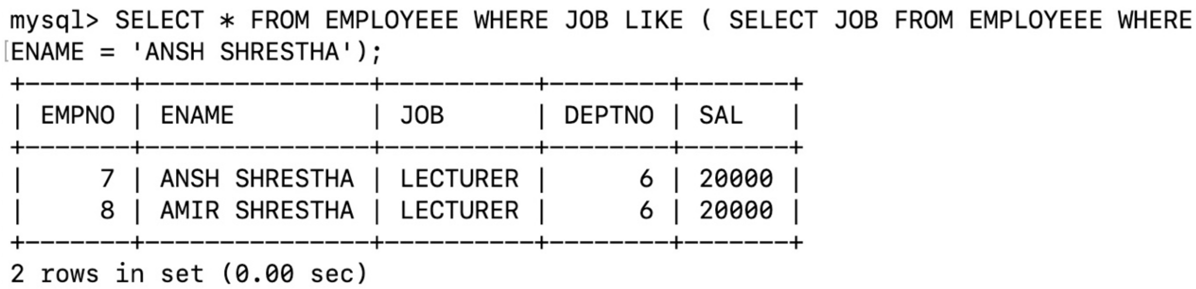
SELECT ENAME, SAL FROM EMPLOYEE WHERE SAL > (SELECT MIN(SAL) FROM EMPLOYEEE) AND JOB LIKE 'A%';



```
[mysql> SELECT ENAME, SAL FROM EMPLOYEEE WHERE SAL > (SELECT MIN(SAL) FROM EMPLOY
EEE) AND JOB LIKE 'A%';
+----------------+--------+
| ENAME          | SAL    |
+----------------+--------+
| UKESH SHRESTHA | 100000 |
+----------------+--------+
1 row in set (0.00 sec)
```

b. **Write a query to find all the employees who work in the same job as Ansh.**

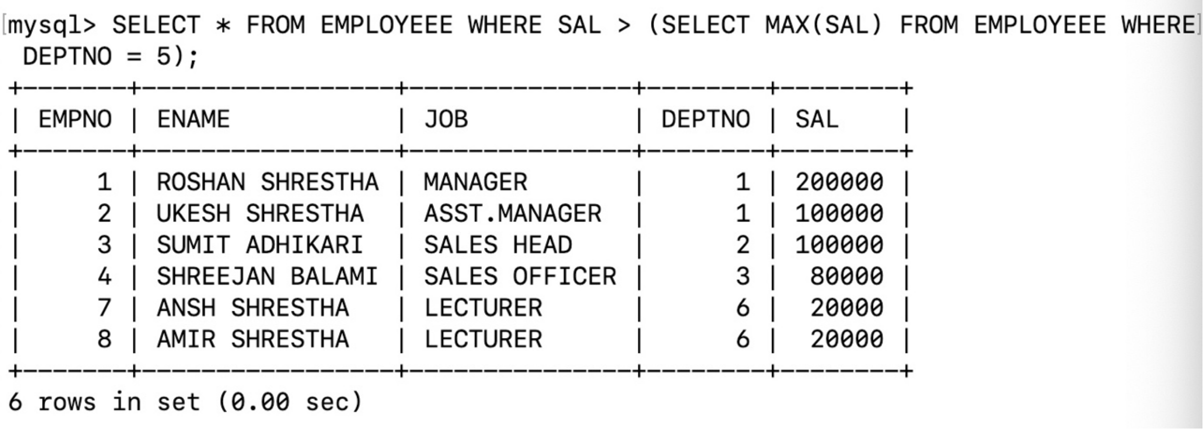
SELECT * FROM EMPLOYEEE WHERE JOB LIKE (SELECT JOB FROM EMPLOYEEE WHERE ENAME = 'ANSH SHRESTHA');



```
mysql> SELECT * FROM EMPLOYEEE WHERE JOB LIKE ( SELECT JOB FROM EMPLOYEEE WHERE
[ENAME = 'ANSH SHRESTHA');
+-------+---------------+----------+--------+-------+
| EMPNO | ENAME         | JOB      | DEPTNO | SAL   |
+-------+---------------+----------+--------+-------+
|     7 | ANSH SHRESTHA | LECTURER |      6 | 20000 |
|     8 | AMIR SHRESTHA | LECTURER |      6 | 20000 |
+-------+---------------+----------+--------+-------+
2 rows in set (0.00 sec)
```

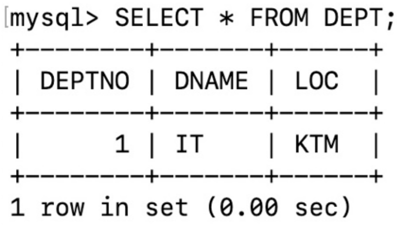
c. **Write a query to display information about employees who earn more than any employee in dept 5.**

SELECT * FROM EMPLOYEEE WHERE SAL > (SELECT MAX(SAL) FROM EMPLOYEEE WHERE DEPTNO = 5);



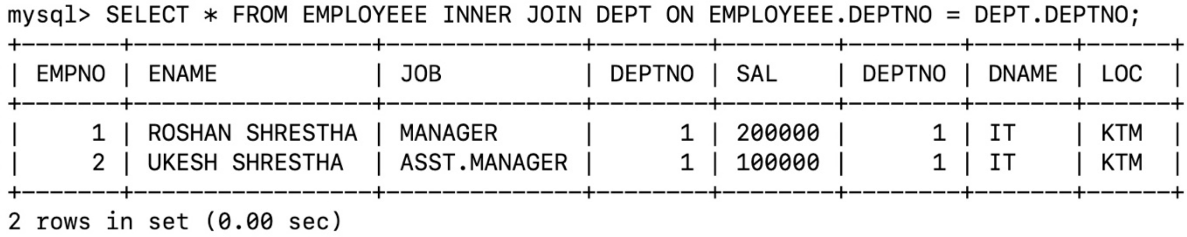
```
[mysql> SELECT * FROM EMPLOYEEE WHERE SAL > (SELECT MAX(SAL) FROM EMPLOYEEE WHERE]
 DEPTNO = 5);
+-------+-----------------+---------------+--------+--------+
| EMPNO | ENAME           | JOB           | DEPTNO | SAL    |
+-------+-----------------+---------------+--------+--------+
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |
|     7 | ANSH SHRESTHA   | LECTURER      |      6 |  20000 |
|     8 | AMIR SHRESTHA   | LECTURER      |      6 |  20000 |
+-------+-----------------+---------------+--------+--------+
6 rows in set (0.00 sec)
```

d. **Display the employee details, departments that the departments are same in both the employeee and dept.**



```
[mysql> SELECT * FROM DEPT;
+--------+-------+-----+
| DEPTNO | DNAME | LOC |
+--------+-------+-----+
|      1 | IT    | KTM |
+--------+-------+-----+
1 row in set (0.00 sec)
```

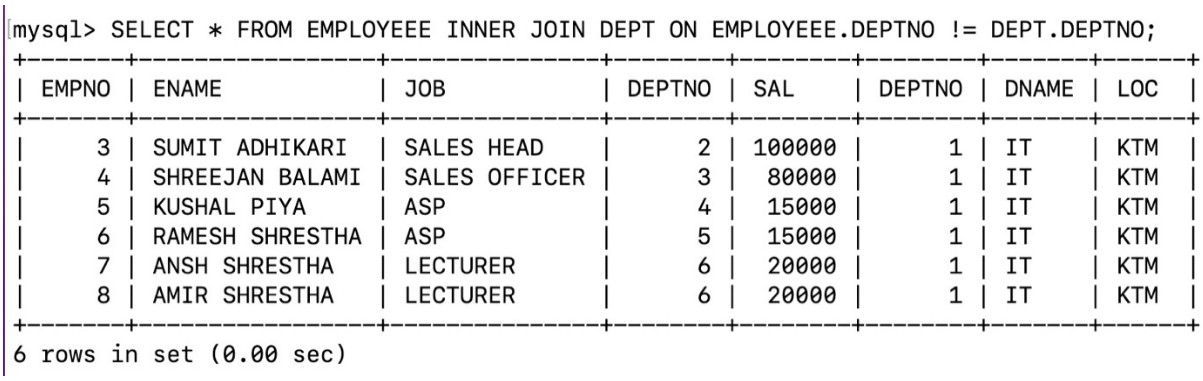
SELECT * FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO = DEPT.DEPTNO;



```
mysql> SELECT * FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO = DEPT.DEPTNO;
+-------+-----------------+--------------+--------+--------+--------+-------+-----+
| EMPNO | ENAME           | JOB          | DEPTNO | SAL    | DEPTNO | DNAME | LOC |
+-------+-----------------+--------------+--------+--------+--------+-------+-----+
|     1 | ROSHAN SHRESTHA | MANAGER      |      1 | 200000 |      1 | IT    | KTM |
|     2 | UKESH SHRESTHA  | ASST.MANAGER |      1 | 100000 |      1 | IT    | KTM |
+-------+-----------------+--------------+--------+--------+--------+-------+-----+
2 rows in set (0.00 sec)
```

e. Display the employee details, departments that the departments are not same in both emp and dept.

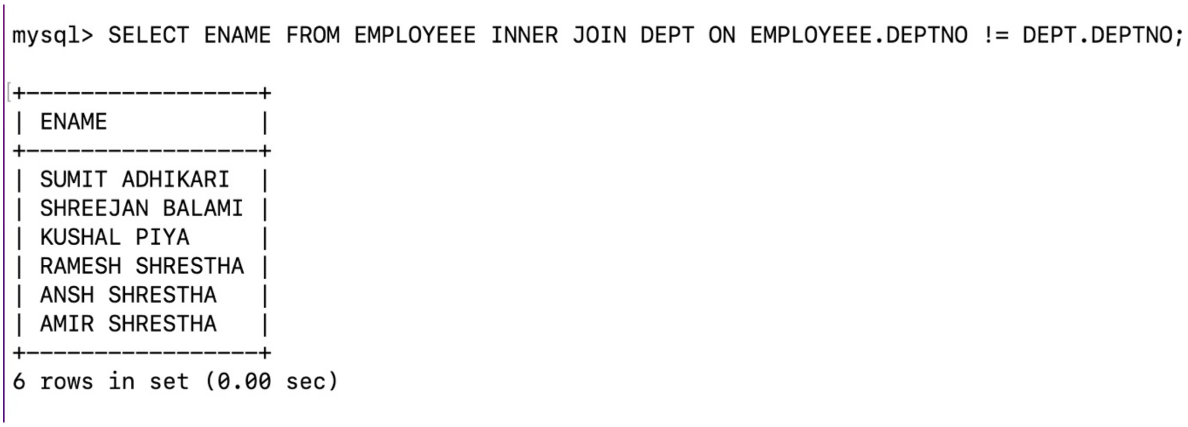
SELECT * FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO != DEPT.DEPTNO;



```
mysql> SELECT * FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO != DEPT.DEPTNO;
+-------+-----------------+---------------+--------+--------+--------+-------+------+
| EMPNO | ENAME           | JOB           | DEPTNO | SAL    | DEPTNO | DNAME | LOC  |
+-------+-----------------+---------------+--------+--------+--------+-------+------+
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |      1 | IT    | KTM  |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |      1 | IT    | KTM  |
|     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |      1 | IT    | KTM  |
|     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |      1 | IT    | KTM  |
|     7 | ANSH SHRESTHA   | LECTURER      |      6 |  20000 |      1 | IT    | KTM  |
|     8 | AMIR SHRESTHA   | LECTURER      |      6 |  20000 |      1 | IT    | KTM  |
+-------+-----------------+---------------+--------+--------+--------+-------+------+
6 rows in set (0.00 sec)
```

f. Write a query to display their employee names.

SELECT ENAME FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO != DEPT.DEPTNO;

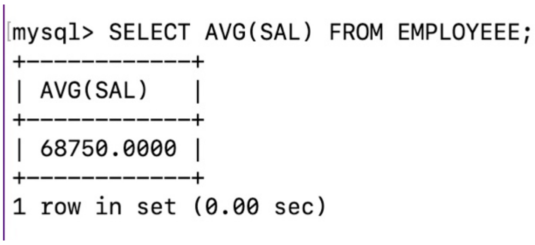


```
mysql> SELECT ENAME FROM EMPLOYEEE INNER JOIN DEPT ON EMPLOYEEE.DEPTNO != DEPT.DEPTNO;

+-----------------+
| ENAME           |
+-----------------+
| SUMIT ADHIKARI  |
| SHREEJAN BALAMI |
| KUSHAL PIYA     |
| RAMESH SHRESTHA |
| ANSH SHRESTHA   |
| AMIR SHRESTHA   |
+-----------------+
6 rows in set (0.00 sec)
```

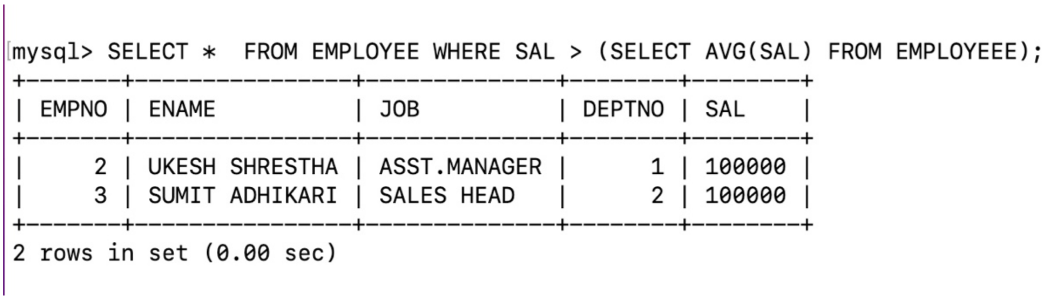
g. Display the details of those who draw the salary greater than the average salary

SELECT AVG(SAL) FROM EMPLOYEEE;



```
mysql> SELECT AVG(SAL) FROM EMPLOYEEE;
+------------+
| AVG(SAL)   |
+------------+
| 68750.0000 |
+------------+
1 row in set (0.00 sec)
```

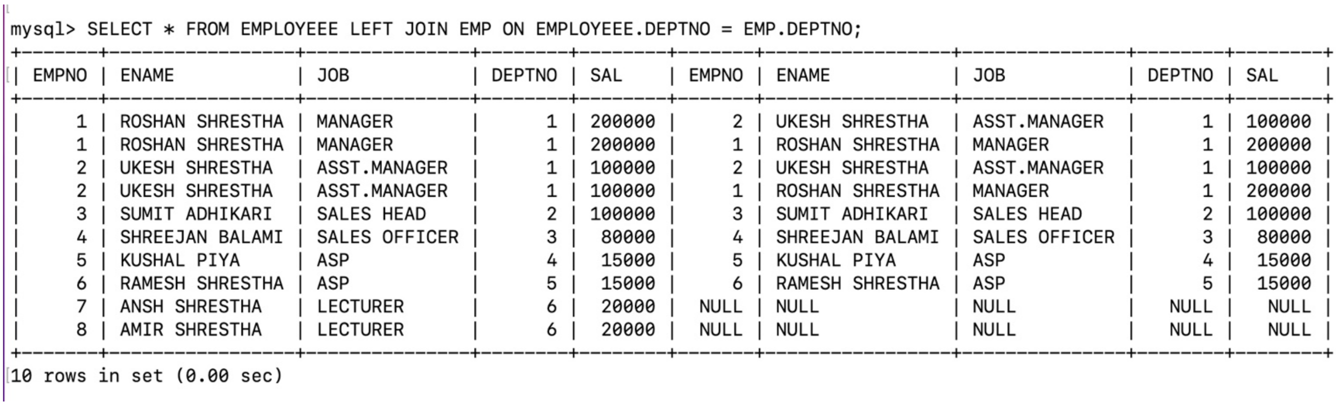
SELECT * FROM EMPLOYEE WHERE SAL > (SELECT AVG(SAL) FROM EMPLOYEEE);



```
[mysql> SELECT *  FROM EMPLOYEE WHERE SAL > (SELECT AVG(SAL) FROM EMPLOYEEE);
+-------+----------------+--------------+--------+--------+
| EMPNO | ENAME          | JOB          | DEPTNO | SAL    |
+-------+----------------+--------------+--------+--------+
|     2 | UKESH SHRESTHA | ASST.MANAGER |      1 | 100000 |
|     3 | SUMIT ADHIKARI | SALES HEAD   |      2 | 100000 |
+-------+----------------+--------------+--------+--------+
2 rows in set (0.00 sec)
```

h. Write a query to perform left outer join.

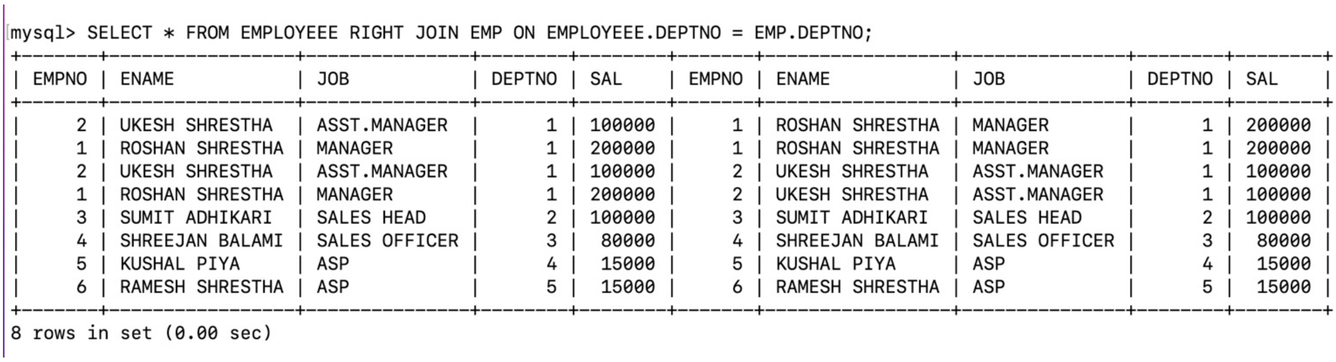
SELECT * FROM EMPLOYEEE LEFT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO;



```
mysql> SELECT * FROM EMPLOYEEE LEFT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO;
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
[| EMPNO | ENAME           | JOB           | DEPTNO | SAL    | EMPNO | ENAME           | JOB           | DEPTNO | SAL    |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |
|     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |
|     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |
|     7 | ANSH SHRESTHA   | LECTURER      |      6 |  20000 |  NULL | NULL            | NULL          |   NULL |   NULL |
|     8 | AMIR SHRESTHA   | LECTURER      |      6 |  20000 |  NULL | NULL            | NULL          |   NULL |   NULL |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
[10 rows in set (0.00 sec)
```

i. Write a query to perform right outer join.

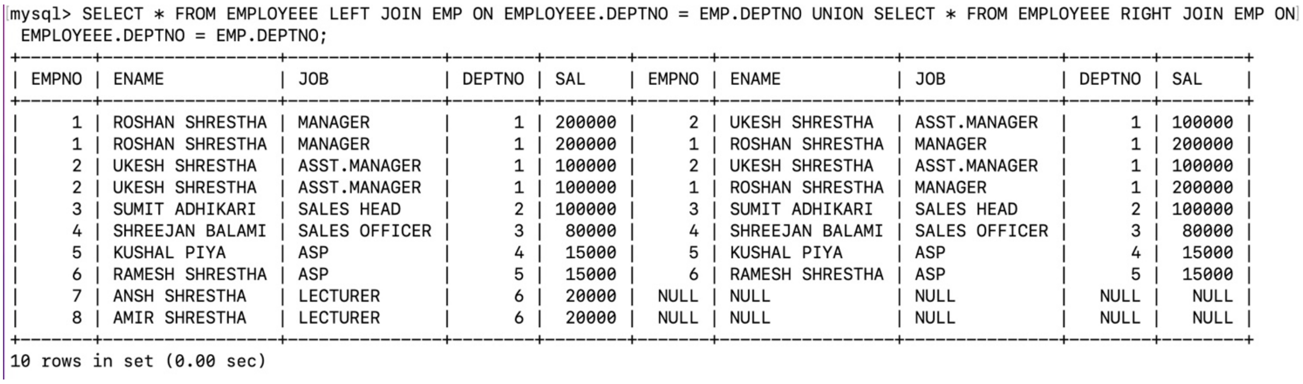
SELECT * FROM EMPLOYEEE RIGHT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO;



```
[mysql> SELECT * FROM EMPLOYEEE RIGHT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO;
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
| EMPNO | ENAME           | JOB           | DEPTNO | SAL    | EMPNO | ENAME           | JOB           | DEPTNO | SAL    |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |
|     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |
|     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
8 rows in set (0.00 sec)
```

j. Write a query to perform full outer join.

SELECT * FROM EMPLOYEEE LEFT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO **UNION** SELECT * FROM EMPLOYEEE RIGHT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO;



```
mysql> SELECT * FROM EMPLOYEEE LEFT JOIN EMP ON EMPLOYEEE.DEPTNO = EMP.DEPTNO UNION SELECT * FROM EMPLOYEEE RIGHT JOIN EMP ON
 EMPLOYEEE.DEPTNO = EMP.DEPTNO;
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
| EMPNO | ENAME           | JOB           | DEPTNO | SAL    | EMPNO | ENAME           | JOB           | DEPTNO | SAL    |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |
|     2 | UKESH SHRESTHA  | ASST.MANAGER  |      1 | 100000 |     1 | ROSHAN SHRESTHA | MANAGER       |      1 | 200000 |
|     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |     3 | SUMIT ADHIKARI  | SALES HEAD    |      2 | 100000 |
|     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |     4 | SHREEJAN BALAMI | SALES OFFICER |      3 |  80000 |
|     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |     5 | KUSHAL PIYA     | ASP           |      4 |  15000 |
|     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |     6 | RAMESH SHRESTHA | ASP           |      5 |  15000 |
|     7 | ANSH SHRESTHA   | LECTURER      |      6 |  20000 |  NULL | NULL            | NULL          |   NULL |   NULL |
|     8 | AMIR SHRESTHA   | LECTURER      |      6 |  20000 |  NULL | NULL            | NULL          |   NULL |   NULL |
+-------+-----------------+---------------+--------+--------+-------+-----------------+---------------+--------+--------+
10 rows in set (0.00 sec)
```