2)Write a query to give inner join,left outer join,right outer join and full outer join(refer SQL_Assignments in Presentation folder)

1) INNER JOIN

An inner join returns only the rows where there is a match in both tables.

```
mysql> SELECT e.employee_id, e.first_name, e.last_name, d.department_name
    -> FROM employee e
    -> INNER JOIN edepartment d
    -> ON e.department_id = d.department_id;
+-------------+------------+-----------+-----------------+
| employee_id | first_name | last_name | department_name |
+-------------+------------+-----------+-----------------+
|           1 | John       | Doe       | HR              |
|           2 | Jane       | Smith     | Sales           |
|           3 | Mike       | Johnson   | IT              |
|           4 | Emily      | Davis     | HR              |
+-------------+------------+-----------+-----------------+
4 rows in set (0.00 sec)
```

2) LEFT OUTER JOIN

A left outer join returns all rows from the left table (employee) and the matched rows from the right table (department). If there is no match, the result is NULL on the side of the right table.

```
mysql> SELECT e.employee_id, e.first_name, e.last_name, d.department_name
    -> FROM employee e
    -> LEFT OUTER JOIN edepartment d
    -> ON e.department_id = d.department_id;
+-------------+------------+-----------+-----------------+
| employee_id | first_name | last_name | department_name |
+-------------+------------+-----------+-----------------+
|           1 | John       | Doe       | HR              |
|           2 | Jane       | Smith     | Sales           |
|           3 | Mike       | Johnson   | IT              |
|           4 | Emily      | Davis     | HR              |
+-------------+------------+-----------+-----------------+
4 rows in set (0.00 sec)
```

3) RIGHT OUTER JOIN

A right outer join returns all rows from the right table (department) and the matched rows from the left table (employee). If there is no match, the result is NULL on the side of the left table.

```
mysql> SELECT e.employee_id, e.first_name, e.last_name, d.department_name
    -> FROM employee e
    -> RIGHT OUTER JOIN edepartment d
    -> ON e.department_id = d.department_id;
+-------------+------------+-----------+-----------------+
| employee_id | first_name | last_name | department_name |
+-------------+------------+-----------+-----------------+
|           4 | Emily      | Davis     | HR              |
|           1 | John       | Doe       | HR              |
|           2 | Jane       | Smith     | Sales           |
|           3 | Mike       | Johnson   | IT              |
|        NULL | NULL       | NULL      | Marketing       |
+-------------+------------+-----------+-----------------+
5 rows in set (0.00 sec)
```

4) FULL OUTER JOIN

A full outer join returns all rows when there is a match in either left (employee) or right (department) table. If there is no match, the result is NULL from the side that does not have a match.

Note: Not all SQL databases support FULL OUTER JOIN directly. For databases that do not support it, you can use a UNION of LEFT OUTER JOIN and RIGHT OUTER JOIN to achieve the same result.

```
mysql> SELECT e.employee_id, e.first_name, e.last_name, d.department_name
    -> FROM employee e
    -> LEFT JOIN edepartment d
    -> ON e.department_id = d.department_id
    -> UNION
    -> SELECT e.employee_id, e.first_name, e.last_name, d.department_name
    -> FROM employee e
    -> RIGHT JOIN edepartment d
    -> ON e.department_id = d.department_id;
+-------------+------------+-----------+-----------------+
| employee_id | first_name | last_name | department_name |
+-------------+------------+-----------+-----------------+
|           1 | John       | Doe       | HR              |
|           2 | Jane       | Smith     | Sales           |
|           3 | Mike       | Johnson   | IT              |
|           4 | Emily      | Davis     | HR              |
|        NULL | NULL       | NULL      | Marketing       |
+-------------+------------+-----------+-----------------+
5 rows in set (0.01 sec)
```

3)Write a query to find duplicate records(refer SQL_Assignments in Presentation folder)?

Find Duplicate Records

1) Based on `first_name`

```
mysql> SELECT first_name, COUNT(*)
    -> FROM employees
    -> GROUP BY first_name
    -> HAVING COUNT(*) > 1;
+------------+----------+
| first_name | COUNT(*) |
+------------+----------+
| John       |        2 |
+------------+----------+
1 row in set (0.01 sec)
```

2) Based on `email`

```
mysql> SELECT email, COUNT(*)
    -> FROM employees
    -> GROUP BY email
    -> HAVING COUNT(*) > 1;
+----------------------+----------+
| email                | COUNT(*) |
+----------------------+----------+
| john.doe@example.com |        2 |
+----------------------+----------+
1 row in set (0.00 sec)
```

3) Based on `first_name` and `last_name`

```
mysql> SELECT first_name, last_name, COUNT(*)
    -> FROM employees
    -> GROUP BY first_name, last_name
    -> HAVING COUNT(*) > 1;
+------------+-----------+----------+
| first_name | last_name | COUNT(*) |
+------------+-----------+----------+
| John       | Doe       |        2 |
+------------+-----------+----------+
1 row in set (0.00 sec)
```

4) Based on `first_name` and `email`

```
mysql> SELECT first_name, email, COUNT(*)
    -> FROM employees
    -> GROUP BY first_name, email
    -> HAVING COUNT(*) > 1;
+------------+----------------------+----------+
| first_name | email                | COUNT(*) |
+------------+----------------------+----------+
| John       | john.doe@example.com |        2 |
+------------+----------------------+----------+
1 row in set (0.00 sec)
```