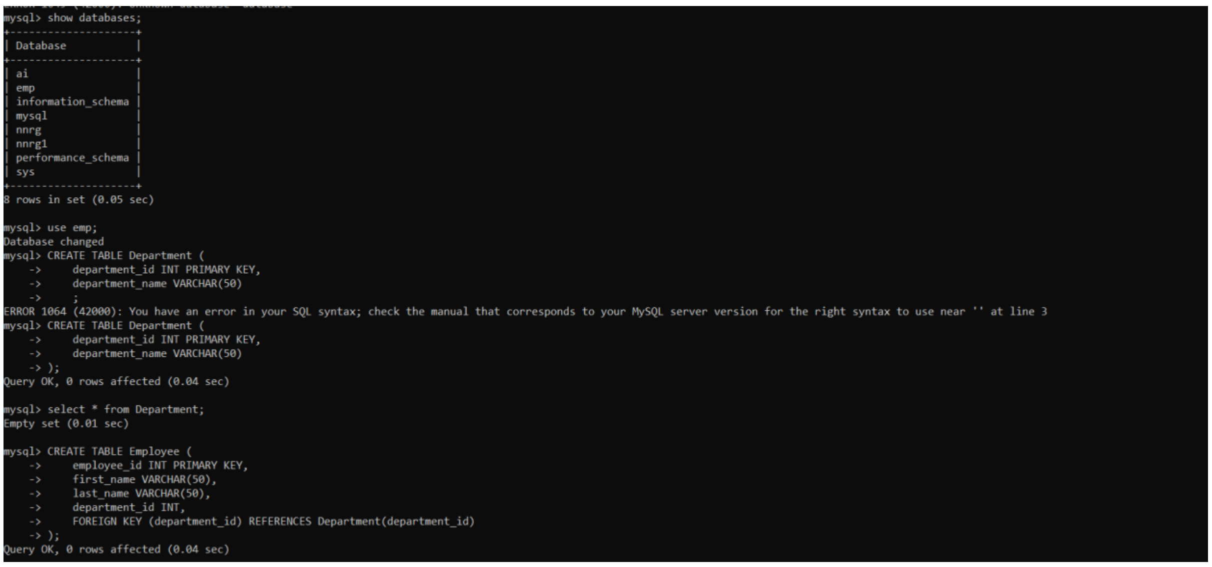
1.creating database and tables:



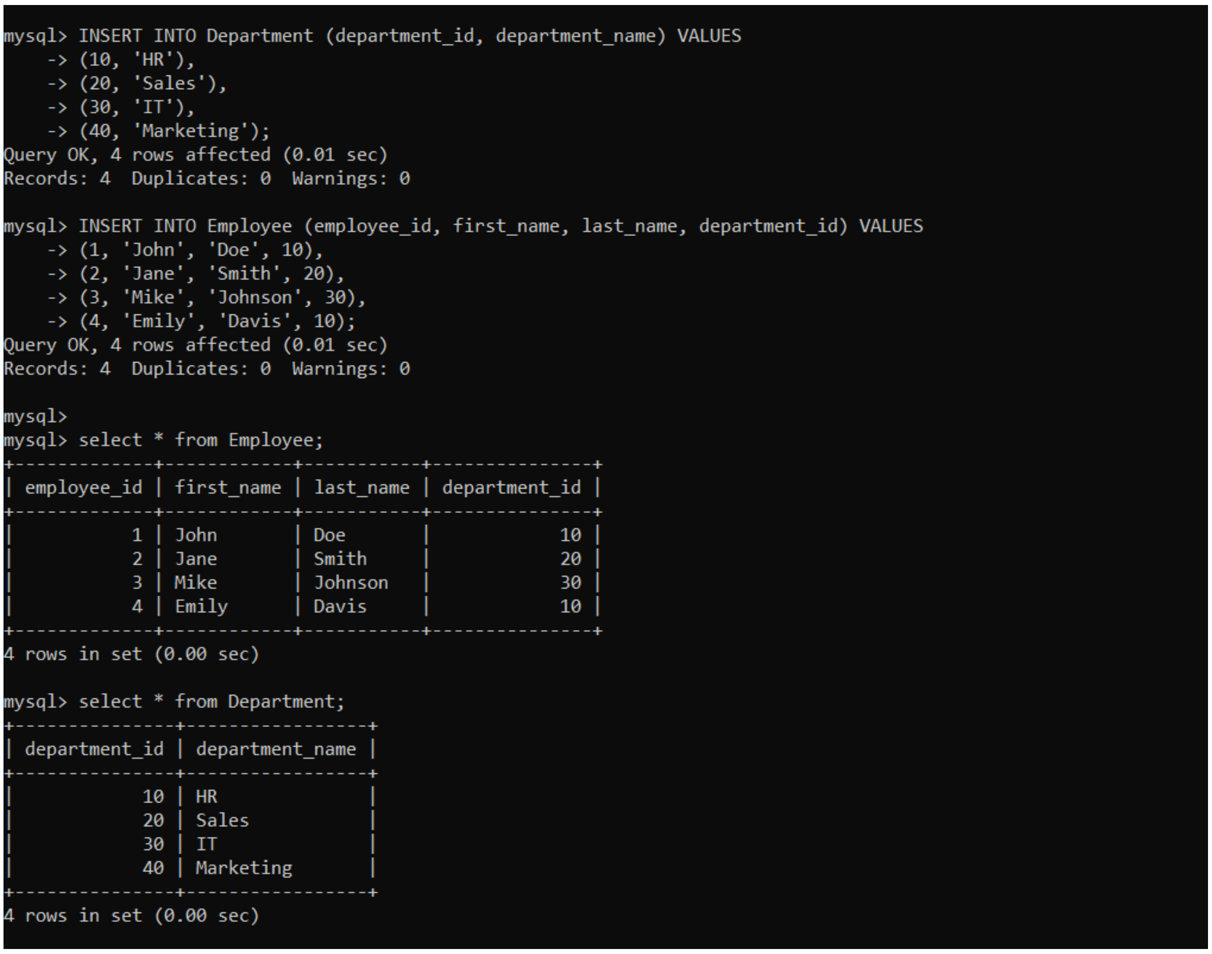
```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| ai                 |
| emp                |
| information_schema |
| mysql              |
| nnrg               |
| nnrg1              |
| performance_schema |
| sys                |
+--------------------+
8 rows in set (0.05 sec)

mysql> use emp;
Database changed
mysql> CREATE TABLE Department (
    ->     department_id INT PRIMARY KEY,
    ->     department_name VARCHAR(50)
    ->     ;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '' at line 3
mysql> CREATE TABLE Department (
    ->     department_id INT PRIMARY KEY,
    ->     department_name VARCHAR(50)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> select * from Department;
Empty set (0.01 sec)

mysql> CREATE TABLE Employee (
    ->     employee_id INT PRIMARY KEY,
    ->     first_name VARCHAR(50),
    ->     last_name VARCHAR(50),
    ->     department_id INT,
    ->     FOREIGN KEY (department_id) REFERENCES Department(department_id)
    -> );
Query OK, 0 rows affected (0.04 sec)
```

2. Inserting values tables:



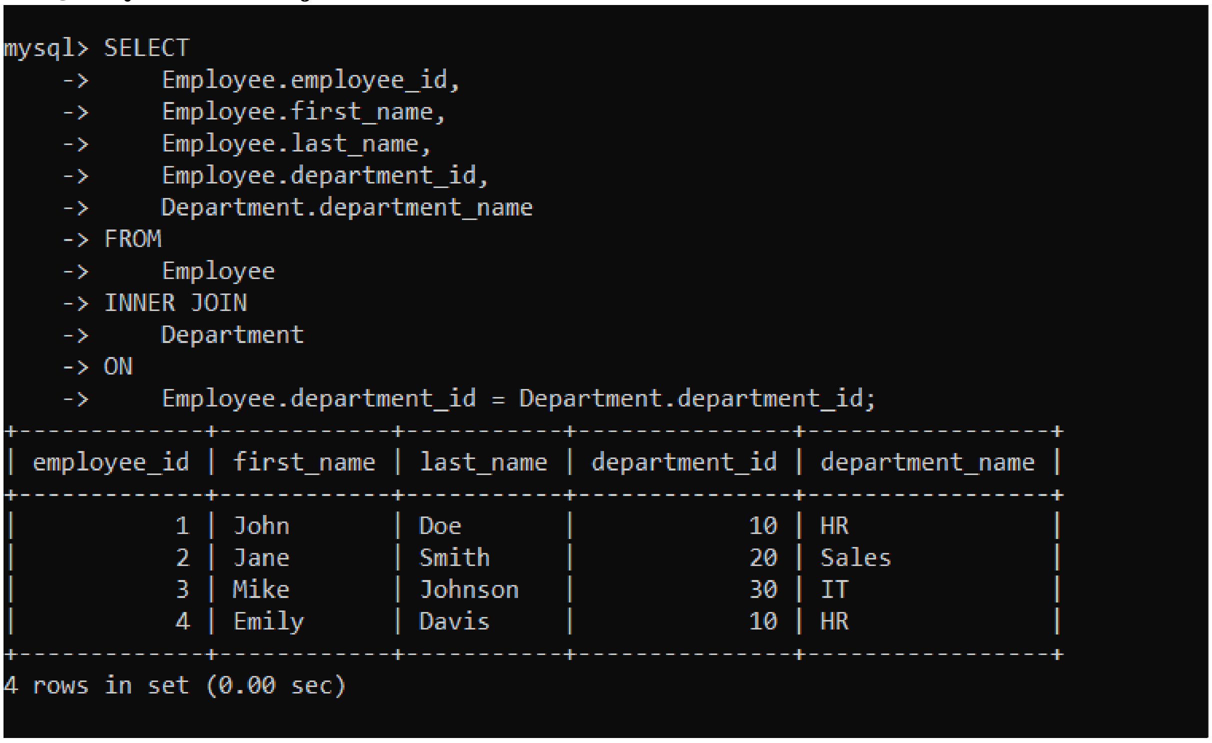
```
mysql> INSERT INTO Department (department_id, department_name) VALUES
    -> (10, 'HR'),
    -> (20, 'Sales'),
    -> (30, 'IT'),
    -> (40, 'Marketing');
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> INSERT INTO Employee (employee_id, first_name, last_name, department_id) VALUES
    -> (1, 'John', 'Doe', 10),
    -> (2, 'Jane', 'Smith', 20),
    -> (3, 'Mike', 'Johnson', 30),
    -> (4, 'Emily', 'Davis', 10);
Query OK, 4 rows affected (0.01 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql>
mysql> select * from Employee;
+-------------+------------+-----------+---------------+
| employee_id | first_name | last_name | department_id |
+-------------+------------+-----------+---------------+
|           1 | John       | Doe       |            10 |
|           2 | Jane       | Smith     |            20 |
|           3 | Mike       | Johnson   |            30 |
|           4 | Emily      | Davis     |            10 |
+-------------+------------+-----------+---------------+
4 rows in set (0.00 sec)

mysql> select * from Department;
+---------------+-----------------+
| department_id | department_name |
+---------------+-----------------+
|            10 | HR              |
|            20 | Sales           |
|            30 | IT              |
|            40 | Marketing       |
+---------------+-----------------+
4 rows in set (0.00 sec)
```

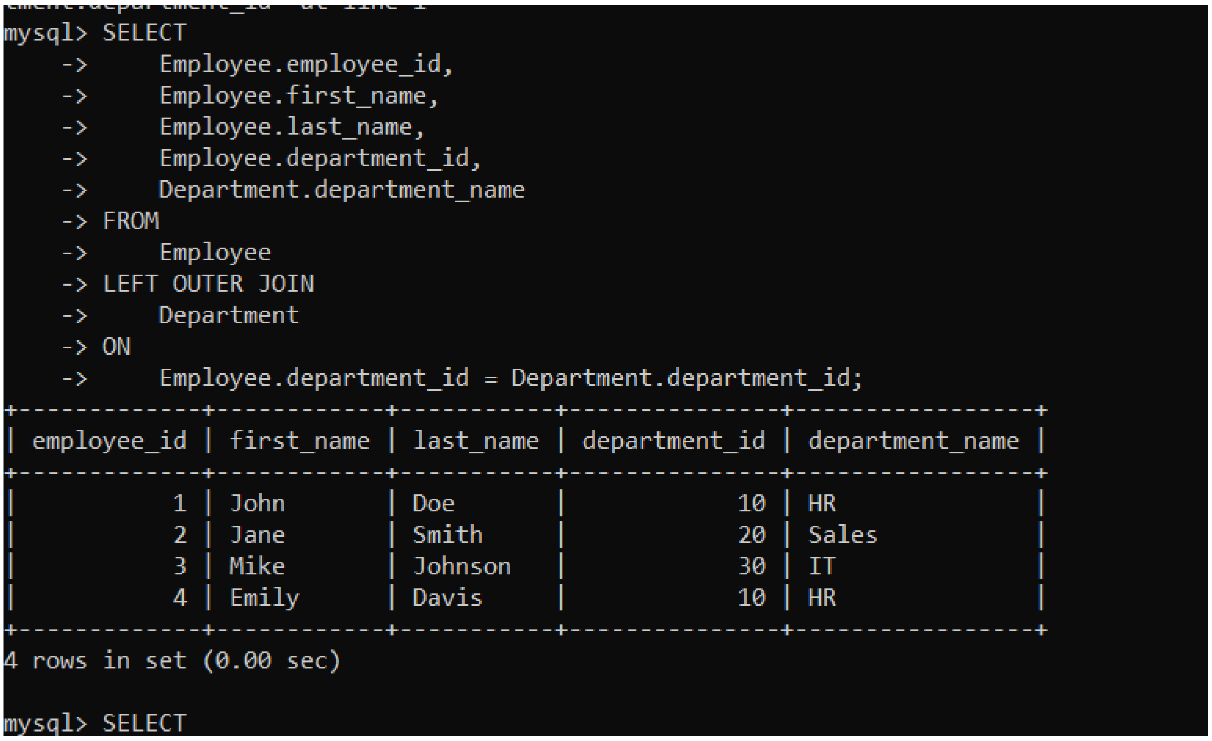
3. Query for inner join :-



```
mysql> SELECT
    ->     Employee.employee_id,
    ->     Employee.first_name,
    ->     Employee.last_name,
    ->     Employee.department_id,
    ->     Department.department_name
    -> FROM
    ->     Employee
    -> INNER JOIN
    ->     Department
    -> ON
    ->     Employee.department_id = Department.department_id;
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|           1 | John       | Doe       |            10 | HR              |
|           2 | Jane       | Smith     |            20 | Sales           |
|           3 | Mike       | Johnson   |            30 | IT              |
|           4 | Emily      | Davis     |            10 | HR              |
+-------------+------------+-----------+---------------+-----------------+
4 rows in set (0.00 sec)
```

This query will return only those rows where there is a matching department_id in both the Employee and Department tables. If an employee's department_id does not have a corresponding department_id in the Department table, that employee's data will not be included in the result set. Similarly, departments without any employees will not appear in the result set.

4. Query for left outer join :-

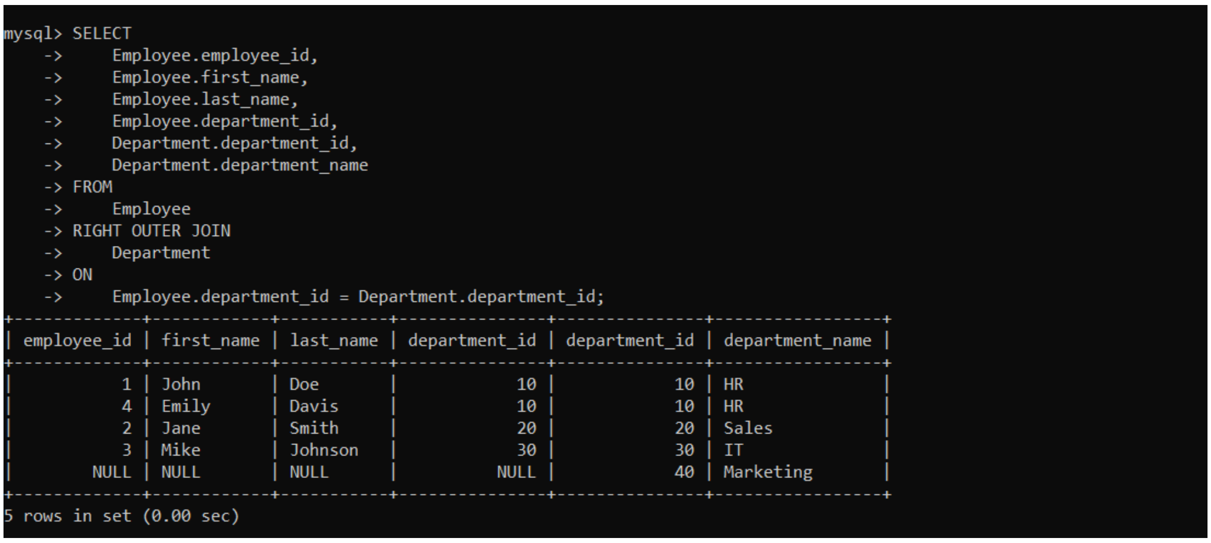


```
tment.department_id' at line 1
mysql> SELECT
    ->     Employee.employee_id,
    ->     Employee.first_name,
    ->     Employee.last_name,
    ->     Employee.department_id,
    ->     Department.department_name
    -> FROM
    ->     Employee
    -> LEFT OUTER JOIN
    ->     Department
    -> ON
    ->     Employee.department_id = Department.department_id;
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|           1 | John       | Doe       |            10 | HR              |
|           2 | Jane       | Smith     |            20 | Sales           |
|           3 | Mike       | Johnson   |            30 | IT              |
|           4 | Emily      | Davis     |            10 | HR              |
+-------------+------------+-----------+---------------+-----------------+
4 rows in set (0.00 sec)

mysql> SELECT
```

This query will return all records from the Employee table, along with their corresponding department information from the Department table if available. If an employee does not have a corresponding department (i.e., there is no match in the Department table), the department-related fields (department_name) will be NULL for that employee.

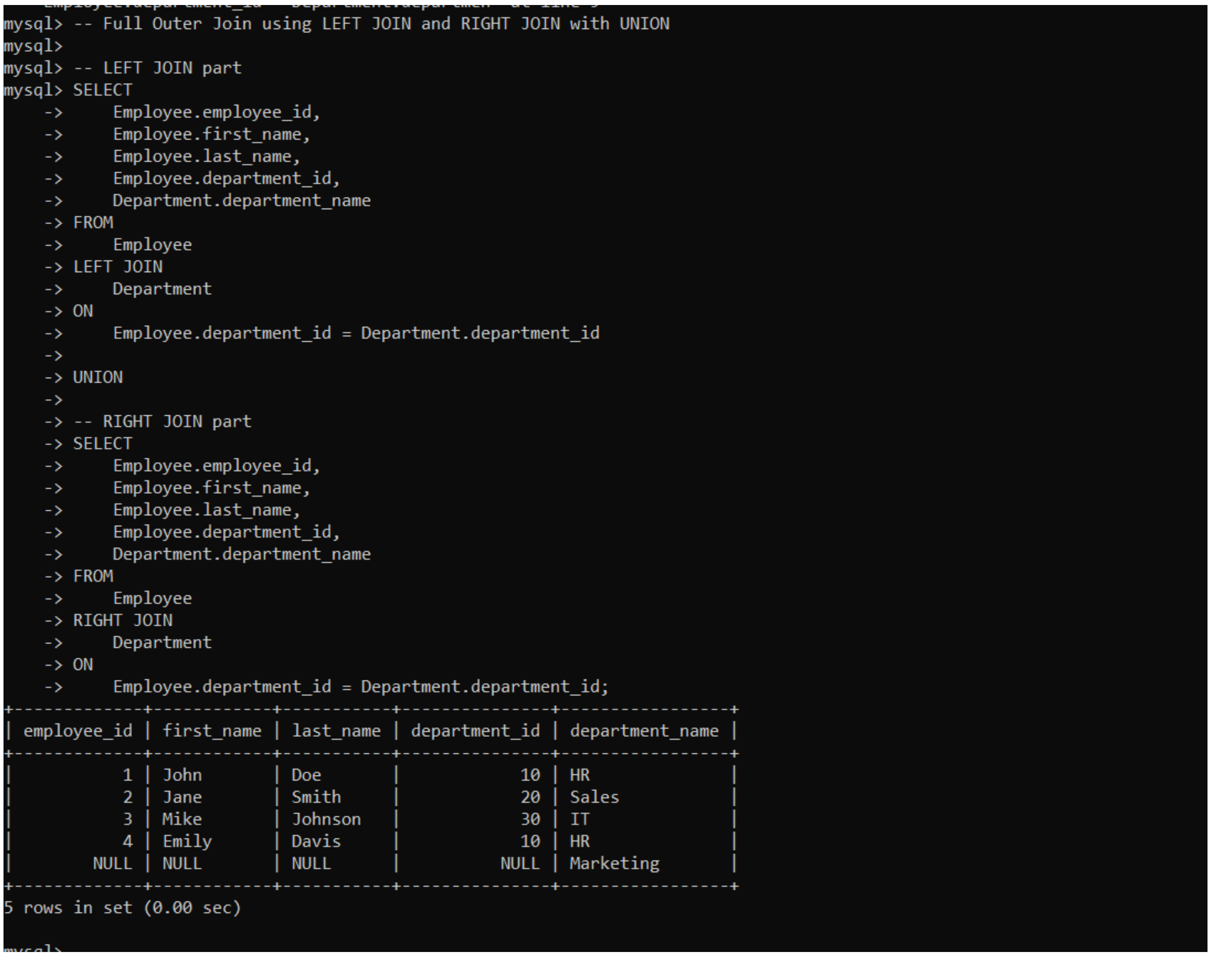
5. Query for right outer join :-



```
mysql> SELECT
    ->     Employee.employee_id,
    ->     Employee.first_name,
    ->     Employee.last_name,
    ->     Employee.department_id,
    ->     Department.department_id,
    ->     Department.department_name
    -> FROM
    ->     Employee
    -> RIGHT OUTER JOIN
    ->     Department
    -> ON
    ->     Employee.department_id = Department.department_id;
+-------------+------------+-----------+---------------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_id | department_name |
+-------------+------------+-----------+---------------+---------------+-----------------+
|           1 | John       | Doe       |            10 |            10 | HR              |
|           4 | Emily      | Davis     |            10 |            10 | HR              |
|           2 | Jane       | Smith     |            20 |            20 | Sales           |
|           3 | Mike       | Johnson   |            30 |            30 | IT              |
|        NULL | NULL       | NULL      |          NULL |            40 | Marketing       |
+-------------+------------+-----------+---------------+---------------+-----------------+
5 rows in set (0.00 sec)
```

The Department table is on the left side.RIGHT OUTER JOIN specifies that all rows from the right table (Employee table) will be included in the result set, along with matched rows from the left table (Department table).If there's no match in the left table (Department), NULL values will be included for columns selected from the left table.

6. Query for full outer join :-



```
mysql> -- Full Outer Join using LEFT JOIN and RIGHT JOIN with UNION
mysql>
mysql> -- LEFT JOIN part
mysql> SELECT
    ->     Employee.employee_id,
    ->     Employee.first_name,
    ->     Employee.last_name,
    ->     Employee.department_id,
    ->     Department.department_name
    -> FROM
    ->     Employee
    -> LEFT JOIN
    ->     Department
    -> ON
    ->     Employee.department_id = Department.department_id
    ->
    -> UNION
    ->
    -> -- RIGHT JOIN part
    -> SELECT
    ->     Employee.employee_id,
    ->     Employee.first_name,
    ->     Employee.last_name,
    ->     Employee.department_id,
    ->     Department.department_name
    -> FROM
    ->     Employee
    -> RIGHT JOIN
    ->     Department
    -> ON
    ->     Employee.department_id = Department.department_id;
+-------------+------------+-----------+---------------+-----------------+
| employee_id | first_name | last_name | department_id | department_name |
+-------------+------------+-----------+---------------+-----------------+
|           1 | John       | Doe       |            10 | HR              |
|           2 | Jane       | Smith     |            20 | Sales           |
|           3 | Mike       | Johnson   |            30 | IT              |
|           4 | Emily      | Davis     |            10 | HR              |
|        NULL | NULL       | NULL      |          NULL | Marketing       |
+-------------+------------+-----------+---------------+-----------------+
5 rows in set (0.00 sec)

mysql>
```

Rows 1, 2, 3, and 4 correspond to matching rows from both tables.Row 5 contains department information (department_id = 40) from the Department table where there's no corresponding employee, so employee-related columns (employee_id, first_name, last_name) are NULL.If there were employees without corresponding departments, they would also be included with department-related columns being NULL.This way, a FULL OUTER JOIN ensures that all data from both tables is included in the result set, making it useful for combining data from disparate sources.