27 CASE STUDY - COMPANY

Problem Statement

A company needs to maintain records of its employees and the projects they are assigned to. The company has several departments, and each employee belongs to a specific department. Projects are managed by the employees, and the company needs to track which employee is responsible for each project.

The company wants to store the following information in a database:

- Employee details including their department
- Project details including which employee is managing each project
- The ability to retrieve and analyze employee and project data efficiently

Database Schema

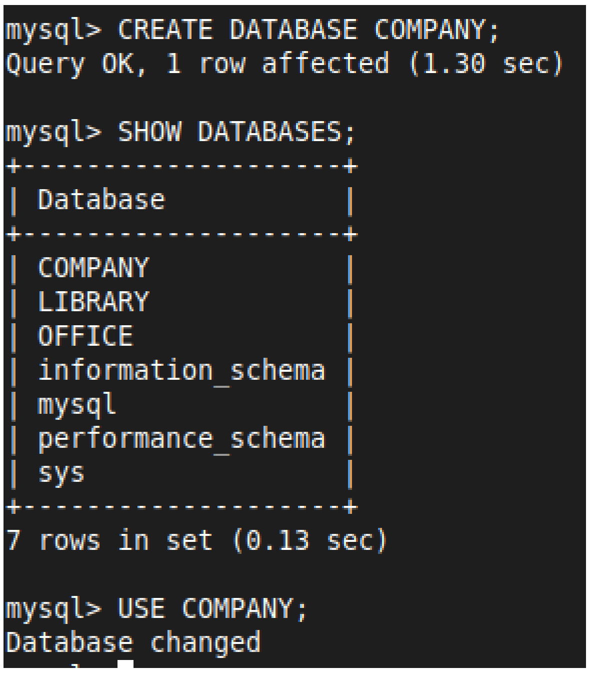
Entity	Attribute	Primary Key	Foreign Key
EMPLOYEES	employee_id, name, department	employee_id	
PROJECTS	project_id, project_name, employee_id	project_id	employee_id

SQL - DDL

1. Create and Use Database

```sql
CREATE DATABASE COMPANY;
```

```sql
USE COMPANY;
```

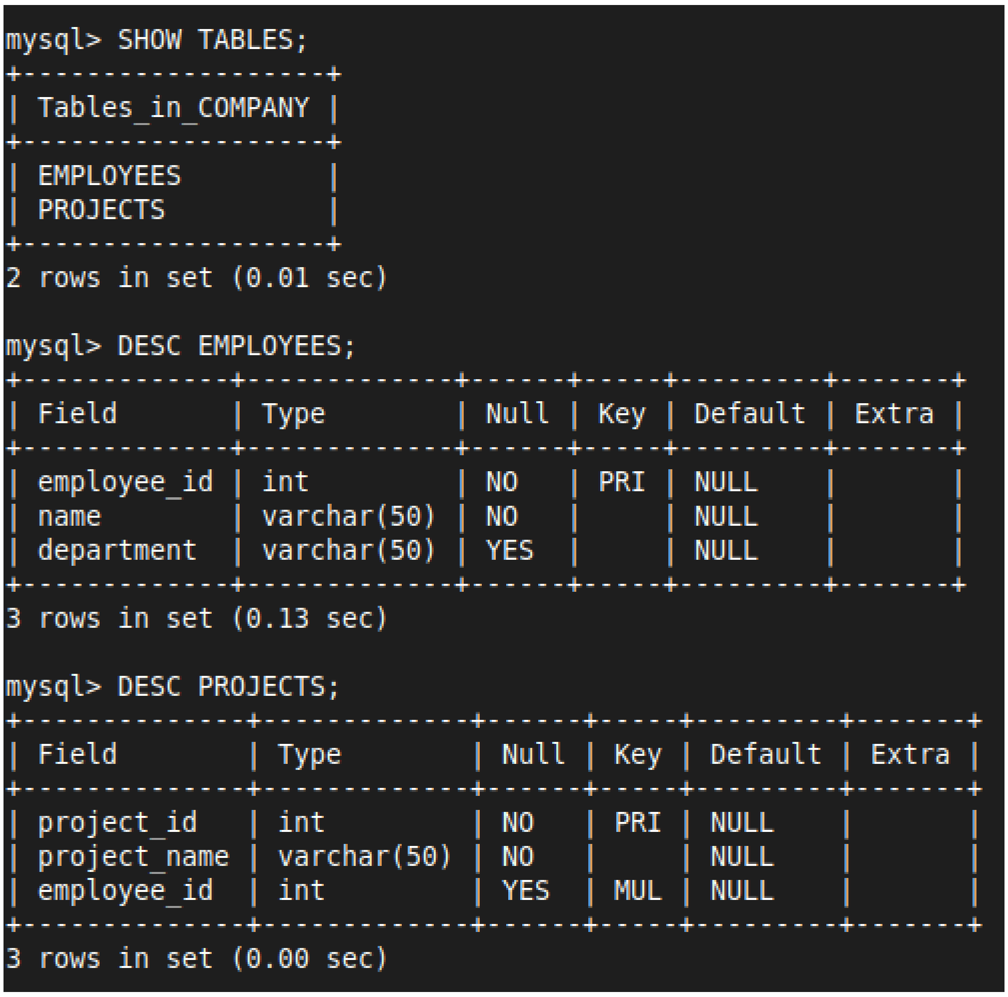


```
mysql> CREATE DATABASE COMPANY;
Query OK, 1 row affected (1.30 sec)

mysql> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| COMPANY            |
| LIBRARY            |
| OFFICE             |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
7 rows in set (0.13 sec)

mysql> USE COMPANY;
Database changed
```

2. Creating Tables

```sql
CREATE TABLE EMPLOYEES (
    employee_id INT PRIMARY KEY,
    name VARCHAR(50) NOT NULL,
    department VARCHAR(50)
);
```

```sql
CREATE TABLE PROJECTS (
    project_id INT PRIMARY KEY,
    project_name VARCHAR(50) NOT NULL,
    employee_id INT,
    FOREIGN KEY (employee_id) REFERENCES EMPLOYEES(employee_id) ON DELETE
CASCADE ON UPDATE CASCADE
);
```

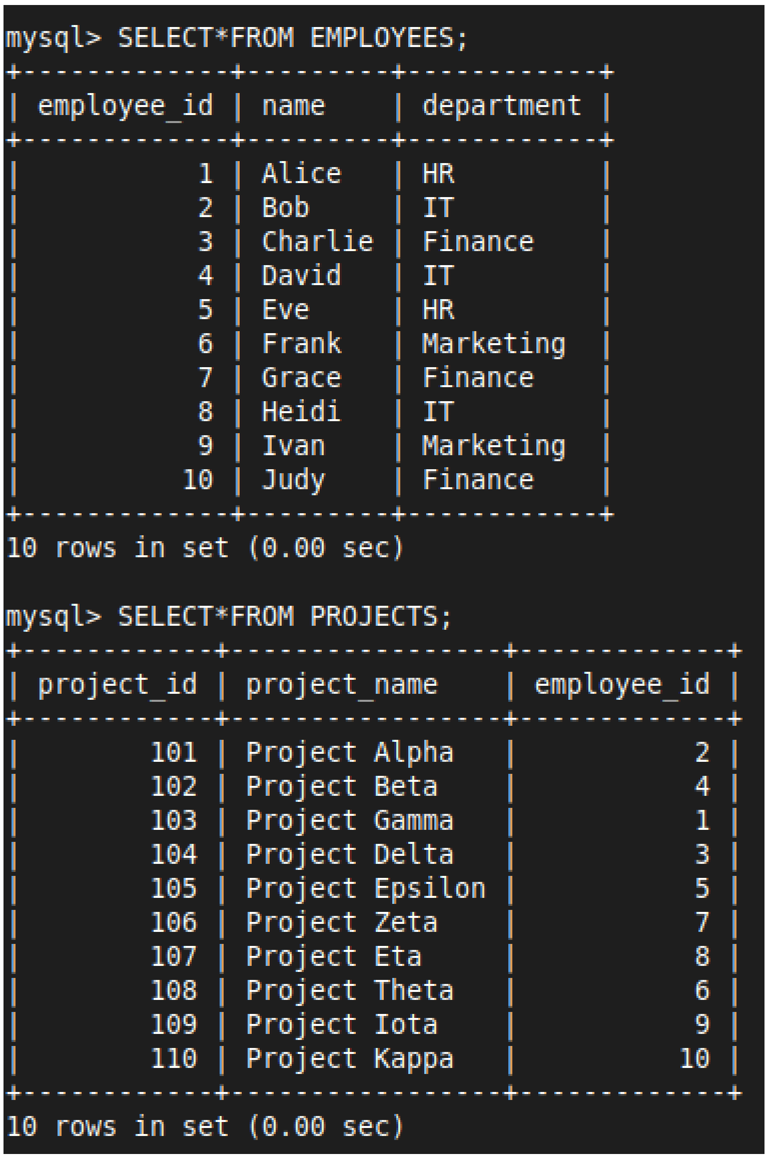


```
mysql> SHOW TABLES;
+-------------------+
| Tables_in_COMPANY |
+-------------------+
| EMPLOYEES         |
| PROJECTS          |
+-------------------+
2 rows in set (0.01 sec)

mysql> DESC EMPLOYEES;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| employee_id | int         | NO   | PRI | NULL    |       |
| name        | varchar(50) | NO   |     | NULL    |       |
| department  | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
3 rows in set (0.13 sec)

mysql> DESC PROJECTS;
+--------------+-------------+------+-----+---------+-------+
| Field        | Type        | Null | Key | Default | Extra |
+--------------+-------------+------+-----+---------+-------+
| project_id   | int         | NO   | PRI | NULL    |       |
| project_name | varchar(50) | NO   |     | NULL    |       |
| employee_id  | int         | YES  | MUL | NULL    |       |
+--------------+-------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```

3. Inserting Values

```sql
INSERT INTO EMPLOYEES (employee_id, name, department) VALUES
(1, 'Alice', 'HR'),
(2, 'Bob', 'IT'),
(3, 'Charlie', 'Finance'),
(4, 'David', 'IT'),
(5, 'Eve', 'HR'),
(6, 'Frank', 'Marketing'),
(7, 'Grace', 'Finance'),
(8, 'Heidi', 'IT'),
(9, 'Ivan', 'Marketing'),
(10, 'Judy', 'Finance');
```

```sql
INSERT INTO PROJECTS (project_id, project_name, employee_id) VALUES
(101, 'Project Alpha', 2),
(102, 'Project Beta', 4),
(103, 'Project Gamma', 1),
(104, 'Project Delta', 3),
(105, 'Project Epsilon', 5),
(106, 'Project Zeta', 7),
(107, 'Project Eta', 8),
(108, 'Project Theta', 6),
(109, 'Project Iota', 9),
(110, 'Project Kappa', 10);
```



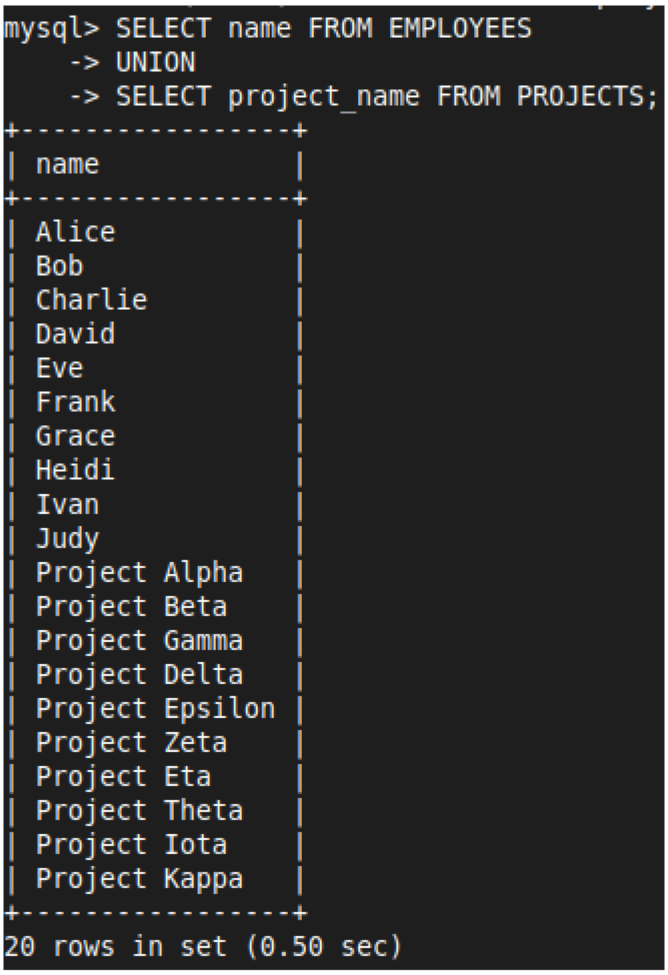
```
mysql> SELECT*FROM EMPLOYEES;
+-------------+---------+------------+
| employee_id | name    | department |
+-------------+---------+------------+
|           1 | Alice   | HR         |
|           2 | Bob     | IT         |
|           3 | Charlie | Finance    |
|           4 | David   | IT         |
|           5 | Eve     | HR         |
|           6 | Frank   | Marketing  |
|           7 | Grace   | Finance    |
|           8 | Heidi   | IT         |
|           9 | Ivan    | Marketing  |
|          10 | Judy    | Finance    |
+-------------+---------+------------+
10 rows in set (0.00 sec)

mysql> SELECT*FROM PROJECTS;
+------------+-----------------+-------------+
| project_id | project_name    | employee_id |
+------------+-----------------+-------------+
|        101 | Project Alpha   |           2 |
|        102 | Project Beta    |           4 |
|        103 | Project Gamma   |           1 |
|        104 | Project Delta   |           3 |
|        105 | Project Epsilon |           5 |
|        106 | Project Zeta    |           7 |
|        107 | Project Eta     |           8 |
|        108 | Project Theta   |           6 |
|        109 | Project Iota    |           9 |
|        110 | Project Kappa   |          10 |
+------------+-----------------+-------------+
10 rows in set (0.00 sec)
```

Set Operators, Nested Queries & Joins

1. Set Operators

- **UNION**

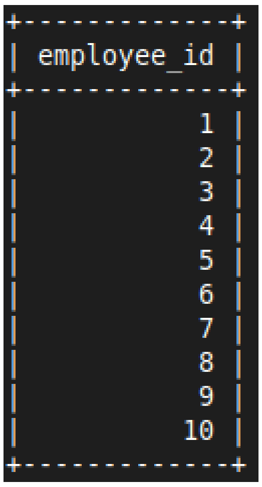
```
SELECT name FROM EMPLOYEES
UNION
SELECT project_name FROM PROJECTS;
```



```
mysql> SELECT name FROM EMPLOYEES
    -> UNION
    -> SELECT project_name FROM PROJECTS;
+-----------------+
| name            |
+-----------------+
| Alice           |
| Bob             |
| Charlie         |
| David           |
| Eve             |
| Frank           |
| Grace           |
| Heidi           |
| Ivan            |
| Judy            |
| Project Alpha   |
| Project Beta    |
| Project Gamma   |
| Project Delta   |
| Project Epsilon |
| Project Zeta    |
| Project Eta     |
| Project Theta   |
| Project Iota    |
| Project Kappa   |
+-----------------+
20 rows in set (0.50 sec)
```

- **INTERSECT**

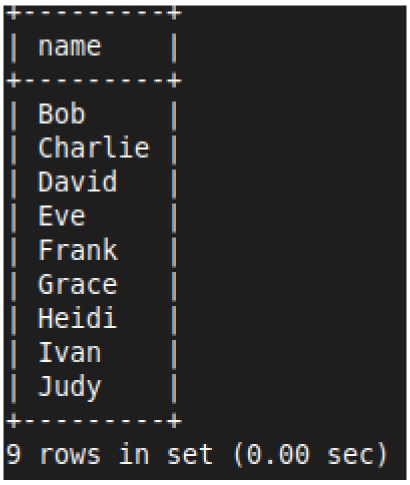
```
SELECT employee_id  FROM EMPLOYEES
INTERSECT
SELECT employee_id FROM PROJECTS;
```



```
+-------------+
| employee_id |
+-------------+
|           1 |
|           2 |
|           3 |
|           4 |
|           5 |
|           6 |
|           7 |
|           8 |
|           9 |
|          10 |
+-------------+
```

- **EXCEPT**

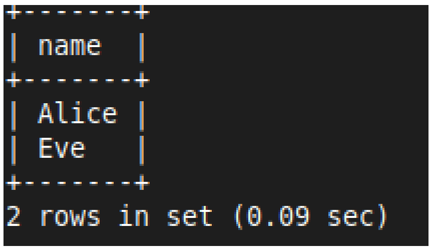
```
SELECT name FROM EMPLOYEES
EXCEPT
SELECT 'Alice' AS name FROM PROJECTS;
```


```
+---------+
| name    |
+---------+
| Bob     |
| Charlie |
| David   |
| Eve     |
| Frank   |
| Grace   |
| Heidi   |
| Ivan    |
| Judy    |
+---------+
9 rows in set (0.00 sec)
```

2. Nested Queries (Subqueries)

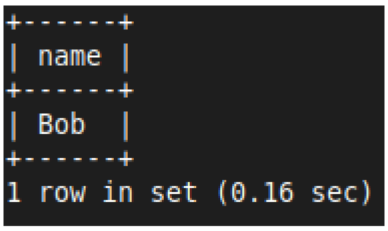
- **Subquery in a WHERE Clause**
    - This query finds all employees who work in the same department as 'Alice'

```
SELECT name FROM EMPLOYEES
WHERE department = (SELECT department FROM EMPLOYEES WHERE name =
'Alice');
```

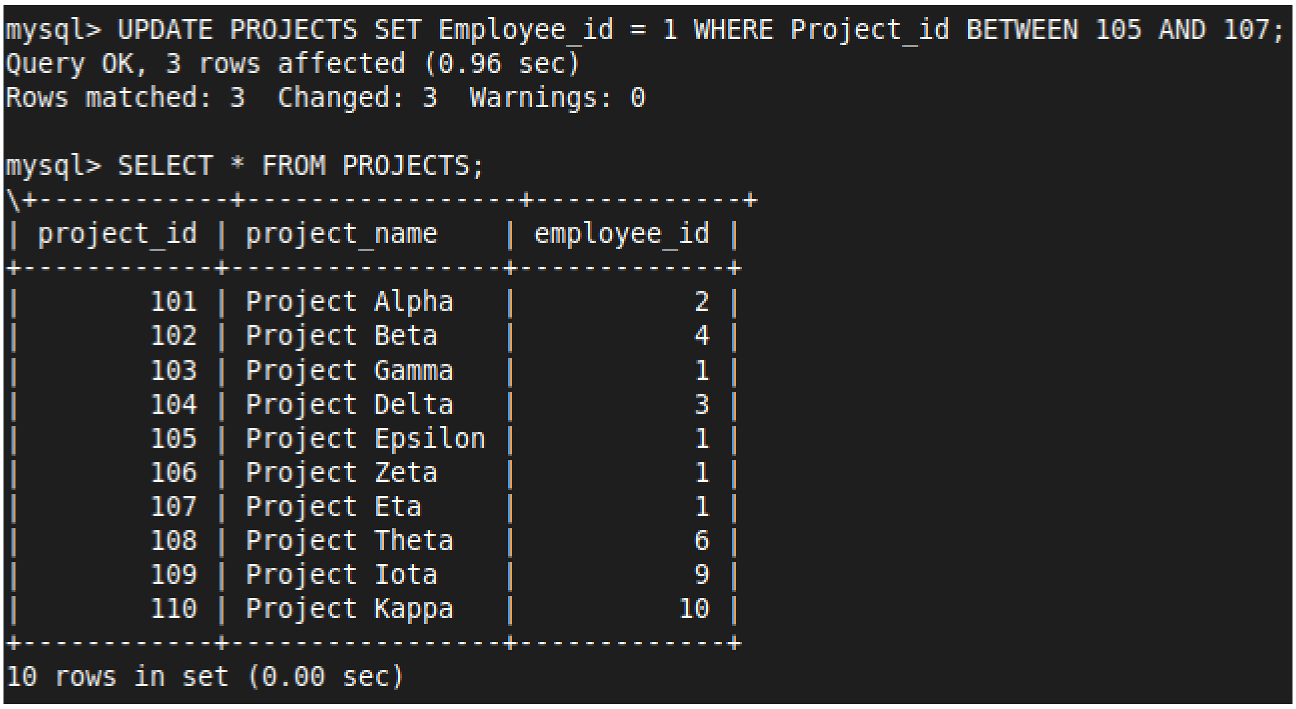

```
+-------+
| name  |
+-------+
| Alice |
| Eve   |
+-------+
2 rows in set (0.09 sec)
```

- **Subquery with `IN`**
    - This query finds the names of employees who are working on 'Project Alpha'

```
SELECT name FROM EMPLOYEES
WHERE employee_id IN
        (SELECT employee_id FROM PROJECTS WHERE project_name = 'Project
Alpha');
```


```
+------+
| name |
+------+
| Bob  |
+------+
1 row in set (0.16 sec)
```

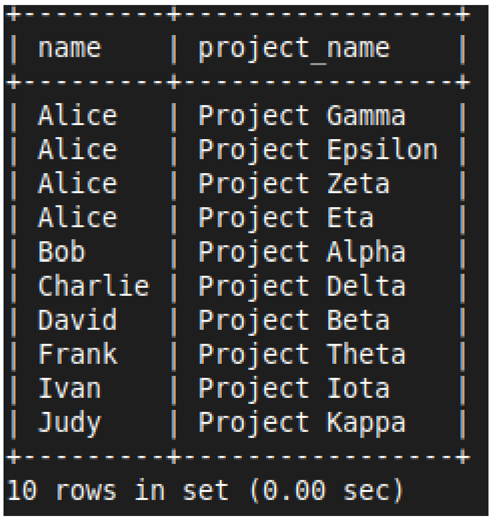
3. Joins



```
mysql> UPDATE PROJECTS SET Employee_id = 1 WHERE Project_id BETWEEN 105 AND 107;
Query OK, 3 rows affected (0.96 sec)
Rows matched: 3  Changed: 3  Warnings: 0

mysql> SELECT * FROM PROJECTS;
\+------------+-----------------+-------------+
| project_id | project_name    | employee_id |
+------------+-----------------+-------------+
|        101 | Project Alpha   |           2 |
|        102 | Project Beta    |           4 |
|        103 | Project Gamma   |           1 |
|        104 | Project Delta   |           3 |
|        105 | Project Epsilon |           1 |
|        106 | Project Zeta    |           1 |
|        107 | Project Eta     |           1 |
|        108 | Project Theta   |           6 |
|        109 | Project Iota    |           9 |
|        110 | Project Kappa   |          10 |
+------------+-----------------+-------------+
10 rows in set (0.00 sec)
```

- **INNER JOIN**
  - This query retrieves the names of employees and their respective project names

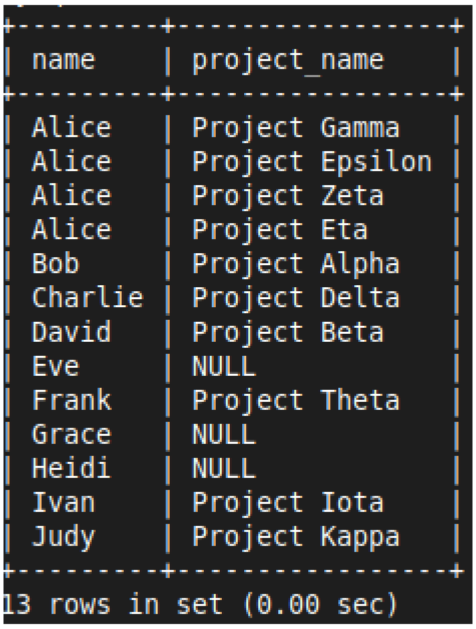
```sql
SELECT EMPLOYEES.name, PROJECTS.project_name
FROM EMPLOYEES
INNER JOIN PROJECTS ON EMPLOYEES.employee_id = PROJECTS.employee_id;
```



```
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Project Gamma   |
| Alice   | Project Epsilon |
| Alice   | Project Zeta    |
| Alice   | Project Eta     |
| Bob     | Project Alpha   |
| Charlie | Project Delta   |
| David   | Project Beta    |
| Frank   | Project Theta   |
| Ivan    | Project Iota    |
| Judy    | Project Kappa   |
+---------+-----------------+
10 rows in set (0.00 sec)
```

- **LEFT JOIN** (or **LEFT OUTER JOIN**)
  - This query retrieves all employees, even those who are not assigned to any projects

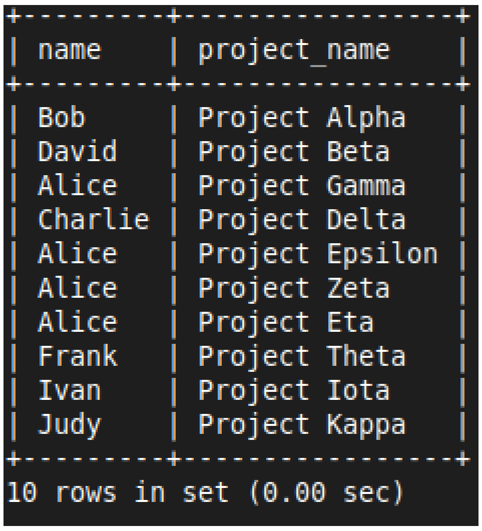
```sql
SELECT EMPLOYEES.name, PROJECTS.project_name
FROM EMPLOYEES
LEFT JOIN PROJECTS ON EMPLOYEES.employee_id = PROJECTS.employee_id;
```



```
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Project Gamma   |
| Alice   | Project Epsilon |
| Alice   | Project Zeta    |
| Alice   | Project Eta     |
| Bob     | Project Alpha   |
| Charlie | Project Delta   |
| David   | Project Beta    |
| Eve     | NULL            |
| Frank   | Project Theta   |
| Grace   | NULL            |
| Heidi   | NULL            |
| Ivan    | Project Iota    |
| Judy    | Project Kappa   |
+---------+-----------------+
13 rows in set (0.00 sec)
```

- **RIGHT JOIN** (or **RIGHT OUTER JOIN**)
  - This query retrieves all projects, including those that do not have any employees assigned

```
SELECT EMPLOYEES.name, PROJECTS.project_name
FROM EMPLOYEES
RIGHT JOIN PROJECTS ON EMPLOYEES.employee_id = PROJECTS.employee_id;
```

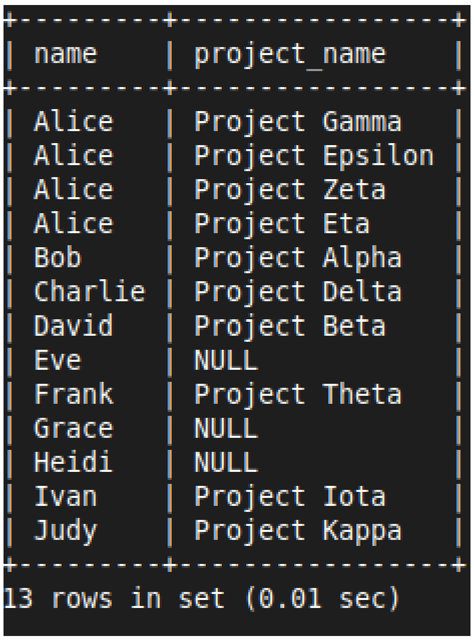


```
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Bob     | Project Alpha   |
| David   | Project Beta    |
| Alice   | Project Gamma   |
| Charlie | Project Delta   |
| Alice   | Project Epsilon |
| Alice   | Project Zeta    |
| Alice   | Project Eta     |
| Frank   | Project Theta   |
| Ivan    | Project Iota    |
| Judy    | Project Kappa   |
+---------+-----------------+
10 rows in set (0.00 sec)
```

- **FULL JOIN** (or **FULL OUTER JOIN**)
  - This query retrieves all employees and all projects, matching them where possible

```
SELECT EMPLOYEES.name, PROJECTS.project_name
FROM EMPLOYEES
LEFT JOIN PROJECTS ON EMPLOYEES.employee_id = PROJECTS.employee_id

UNION

SELECT EMPLOYEES.name, PROJECTS.project_name
```

```sql
FROM PROJECTS
RIGHT JOIN EMPLOYEES ON EMPLOYEES.employee_id = PROJECTS.employee_id;
```



```
+---------+-----------------+
| name    | project_name    |
+---------+-----------------+
| Alice   | Project Gamma   |
| Alice   | Project Epsilon |
| Alice   | Project Zeta    |
| Alice   | Project Eta     |
| Bob     | Project Alpha   |
| Charlie | Project Delta   |
| David   | Project Beta    |
| Eve     | NULL            |
| Frank   | Project Theta   |
| Grace   | NULL            |
| Heidi   | NULL            |
| Ivan    | Project Iota    |
| Judy    | Project Kappa   |
+---------+-----------------+
13 rows in set (0.01 sec)
```