Lesson 05 Demo 08

Working with Related Tables

Objective: To demonstrate the process of managing and querying related tables in MySQL, emphasizing the importance of relational database principles

Tools required: MySQL

Prerequisites: None

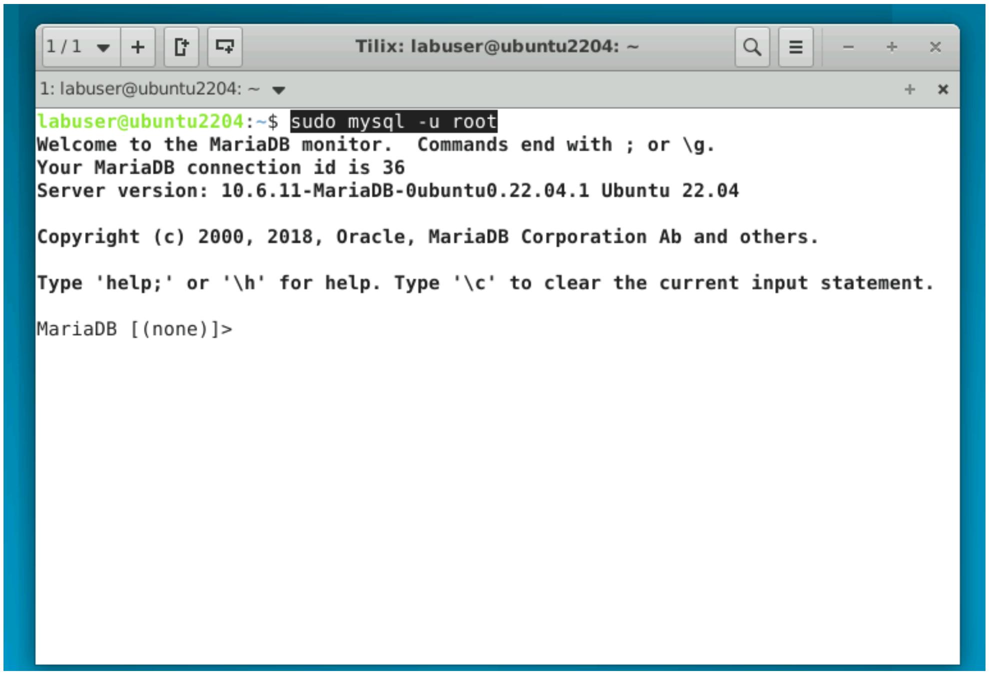
Steps to be followed:

1. Set up a database and table
2. Run a query for related tables

Step 1: Set up a database and table

1.1 Open a terminal window and access MySQL as a root user:

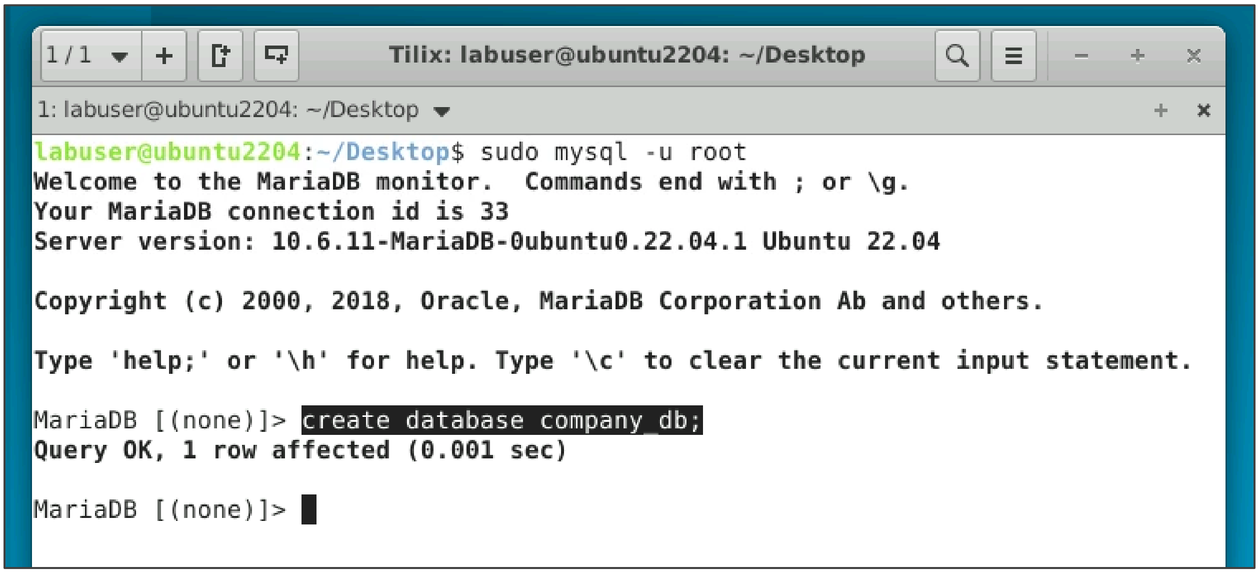
sudo mysql -u root



```
labuser@ubuntu2204:~$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 36
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>
```

1.2 Create a new database named **company_db**:

create database company_db;

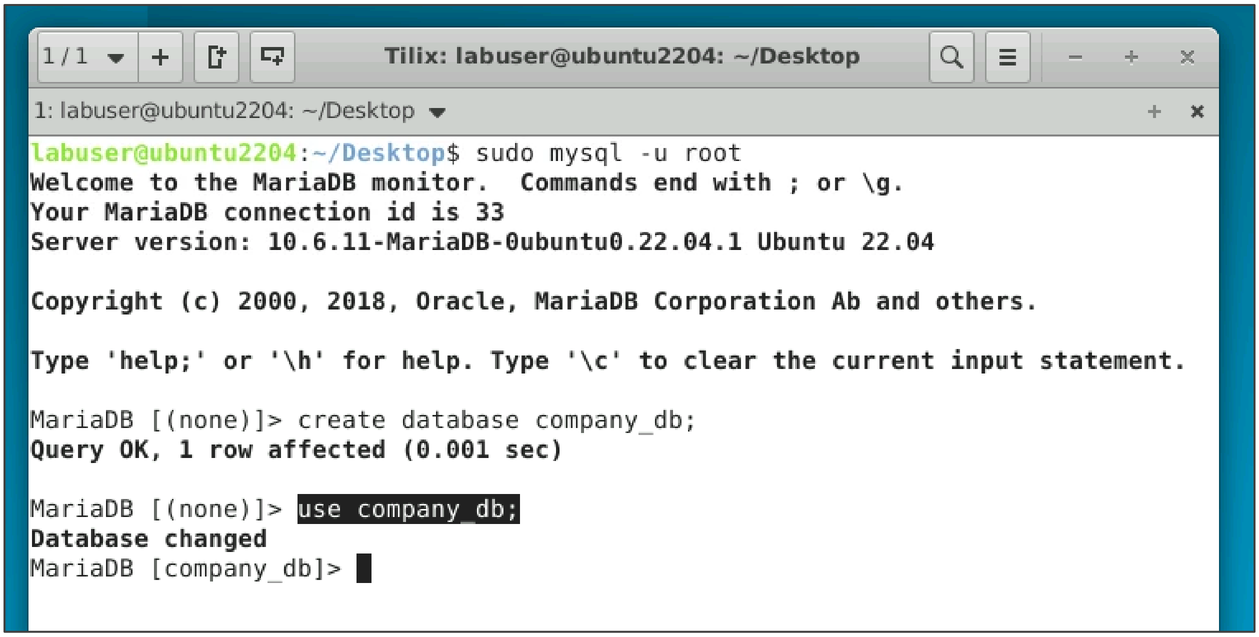


1 / 1
Tilix: labuser@ubuntu2204: ~/Desktop
1: labuser@ubuntu2204: ~/Desktop

```
labuser@ubuntu2204:~/Desktop$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database company_db;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> 
```

1.3 Select the **company_db** database:

use company_db;

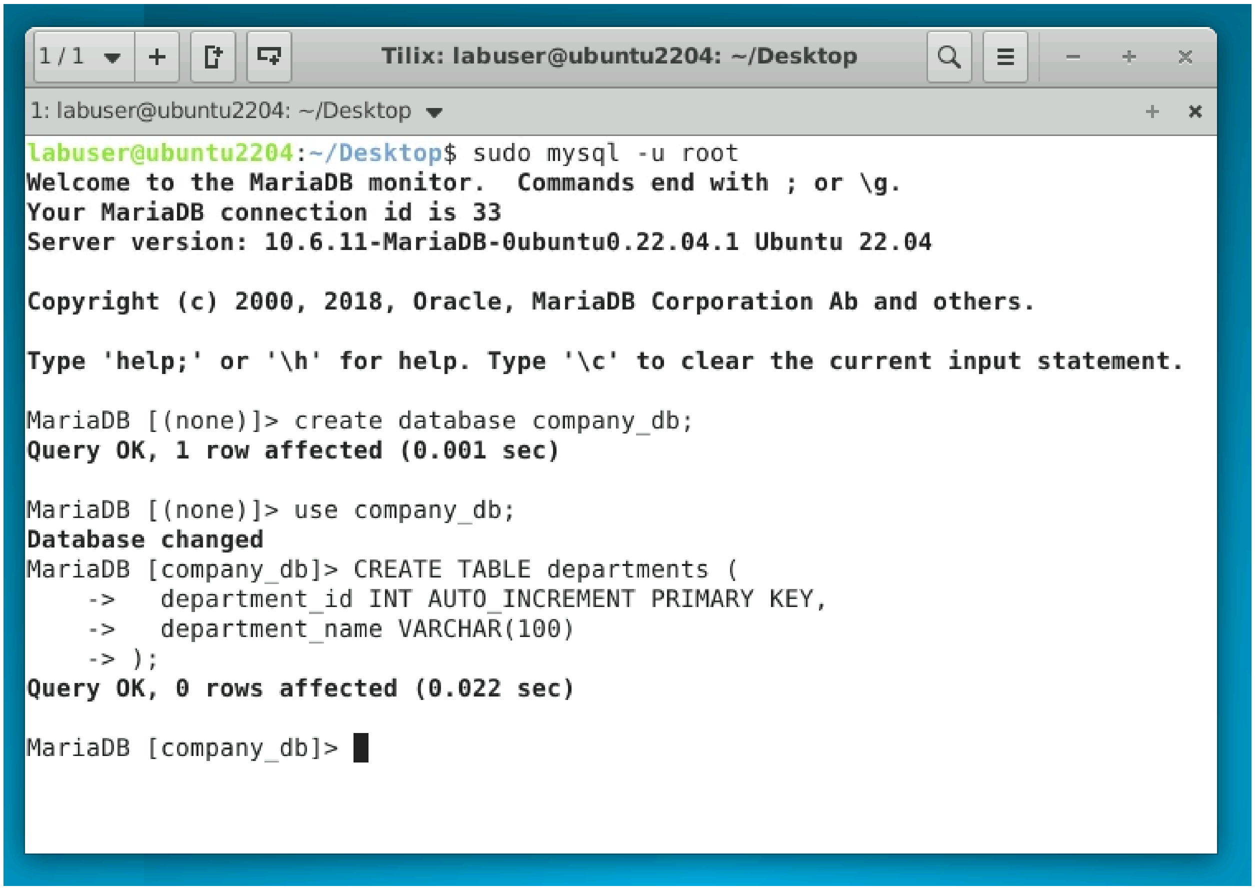


1 / 1
Tilix: labuser@ubuntu2204: ~/Desktop
1: labuser@ubuntu2204: ~/Desktop

```
labuser@ubuntu2204:~/Desktop$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database company_db;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use company_db;
Database changed
MariaDB [company_db]> 
```

1.4 Create a **departments** table with relevant fields:

```
CREATE TABLE departments (
  department_id INT AUTO_INCREMENT PRIMARY KEY,
  department_name VARCHAR(100)
);
```

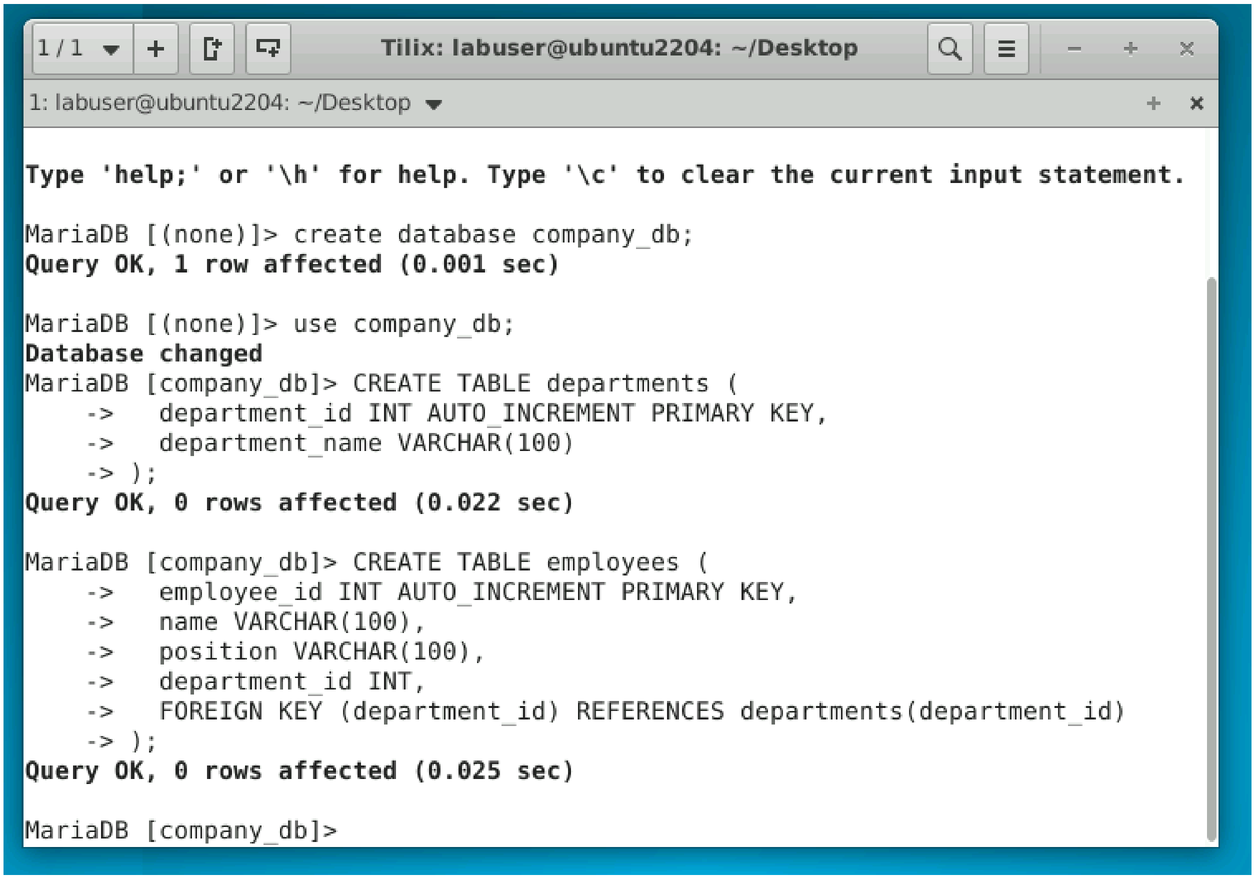


1 / 1
Tilix: labuser@ubuntu2204: ~/Desktop
1: labuser@ubuntu2204: ~/Desktop

```
labuser@ubuntu2204:~/Desktop$ sudo mysql -u root
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 33
Server version: 10.6.11-MariaDB-0ubuntu0.22.04.1 Ubuntu 22.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database company_db;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use company_db;
Database changed
MariaDB [company_db]> CREATE TABLE departments (
    ->   department_id INT AUTO_INCREMENT PRIMARY KEY,
    ->   department_name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.022 sec)

MariaDB [company_db]> 
```

1.5 Create an **employees** table with relevant fields and a foreign key to **departments**:

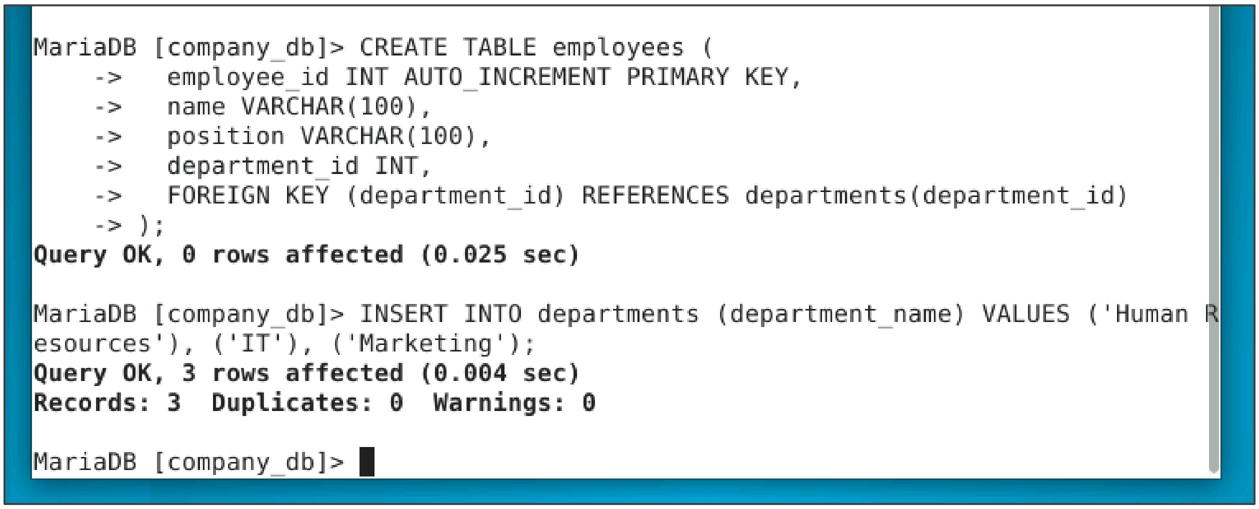
```
CREATE TABLE employees (
  employee_id INT AUTO_INCREMENT PRIMARY KEY,
  name VARCHAR(100),
  position VARCHAR(100),
  department_id INT,
  FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```


1 / 1
Tilix: labuser@ubuntu2204: ~/Desktop
1: labuser@ubuntu2204: ~/Desktop

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database company_db;
Query OK, 1 row affected (0.001 sec)

MariaDB [(none)]> use company_db;
Database changed
MariaDB [company_db]> CREATE TABLE departments (
    ->   department_id INT AUTO_INCREMENT PRIMARY KEY,
    ->   department_name VARCHAR(100)
    -> );
Query OK, 0 rows affected (0.022 sec)

MariaDB [company_db]> CREATE TABLE employees (
    ->   employee_id INT AUTO_INCREMENT PRIMARY KEY,
    ->   name VARCHAR(100),
    ->   position VARCHAR(100),
    ->   department_id INT,
    ->   FOREIGN KEY (department_id) REFERENCES departments(department_id)
    -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [company_db]>
```

1.6 Insert data into the **departments**:

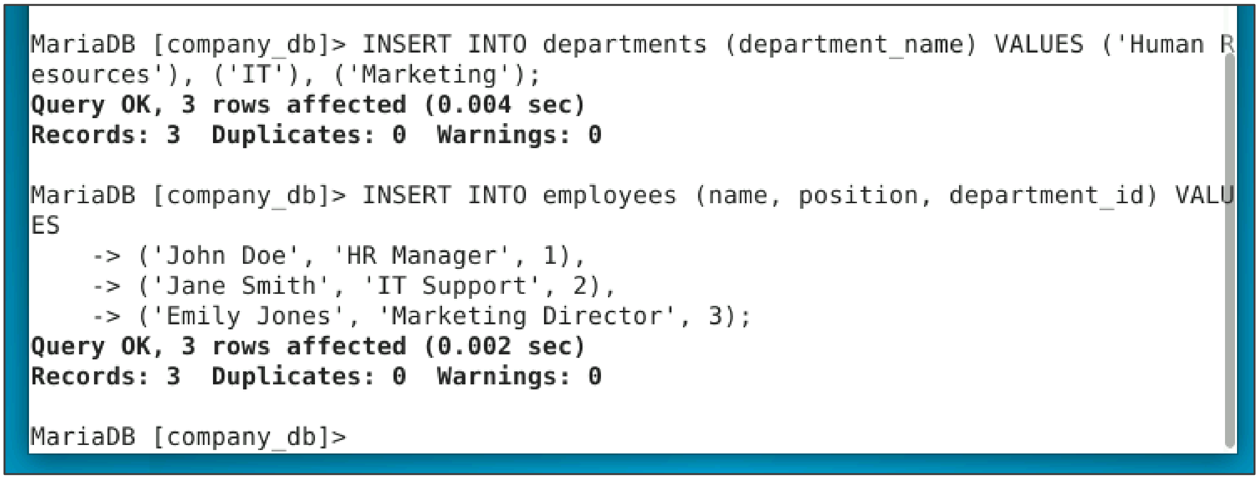
INSERT INTO departments (department_name) VALUES ('Human Resources'), ('IT'), ('Marketing');



```
MariaDB [company_db]> CREATE TABLE employees (
    ->   employee_id INT AUTO_INCREMENT PRIMARY KEY,
    ->   name VARCHAR(100),
    ->   position VARCHAR(100),
    ->   department_id INT,
    ->   FOREIGN KEY (department_id) REFERENCES departments(department_id)
    -> );
Query OK, 0 rows affected (0.025 sec)

MariaDB [company_db]> INSERT INTO departments (department_name) VALUES ('Human R
esources'), ('IT'), ('Marketing');
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [company_db]> 
```

1.7 Insert data into the **employees**:

INSERT INTO employees (name, position, department_id) VALUES
('John Doe', 'HR Manager', 1),
('Jane Smith', 'IT Support', 2),
('Emily Jones', 'Marketing Director', 3);

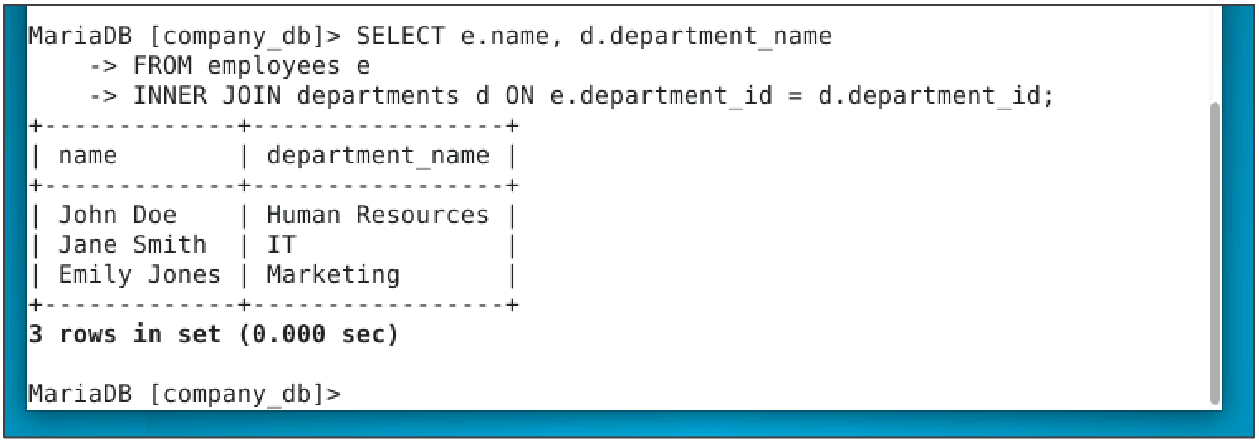


```
MariaDB [company_db]> INSERT INTO departments (department_name) VALUES ('Human R
esources'), ('IT'), ('Marketing');
Query OK, 3 rows affected (0.004 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [company_db]> INSERT INTO employees (name, position, department_id) VALU
ES
    -> ('John Doe', 'HR Manager', 1),
    -> ('Jane Smith', 'IT Support', 2),
    -> ('Emily Jones', 'Marketing Director', 3);
Query OK, 3 rows affected (0.002 sec)
Records: 3  Duplicates: 0  Warnings: 0

MariaDB [company_db]>
```

Step 2: Run a query for related tables

2.1 Display the employee names along with their department names:

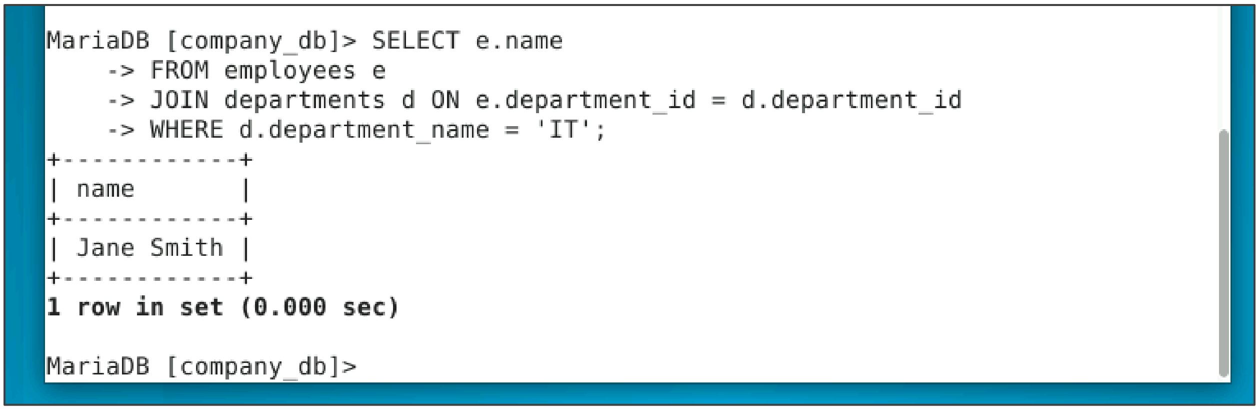
SELECT e.name, d.department_name
FROM employees e
INNER JOIN departments d ON e.department_id = d.department_id;



```
MariaDB [company_db]> SELECT e.name, d.department_name
    -> FROM employees e
    -> INNER JOIN departments d ON e.department_id = d.department_id;
+-------------+-----------------+
| name        | department_name |
+-------------+-----------------+
| John Doe    | Human Resources |
| Jane Smith  | IT              |
| Emily Jones | Marketing       |
+-------------+-----------------+
3 rows in set (0.000 sec)

MariaDB [company_db]>
```

2.2 Find all employees in the **IT** department:

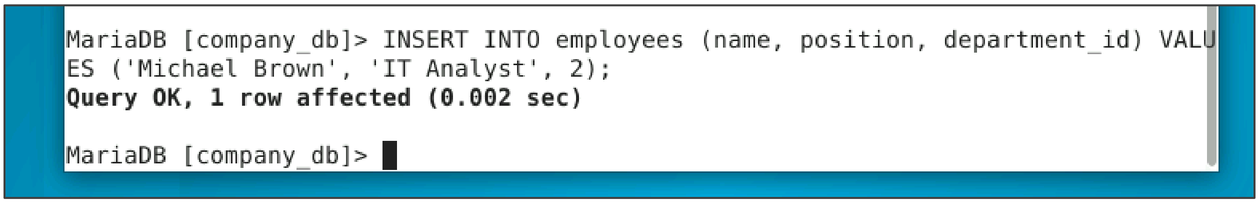
SELECT e.name
FROM employees e
JOIN departments d ON e.department_id = d.department_id
WHERE d.department_name = 'IT';



```
MariaDB [company_db]> SELECT e.name
    -> FROM employees e
    -> JOIN departments d ON e.department_id = d.department_id
    -> WHERE d.department_name = 'IT';
+------------+
| name       |
+------------+
| Jane Smith |
+------------+
1 row in set (0.000 sec)

MariaDB [company_db]>
```

2.3 Add a new employee to a specific department:

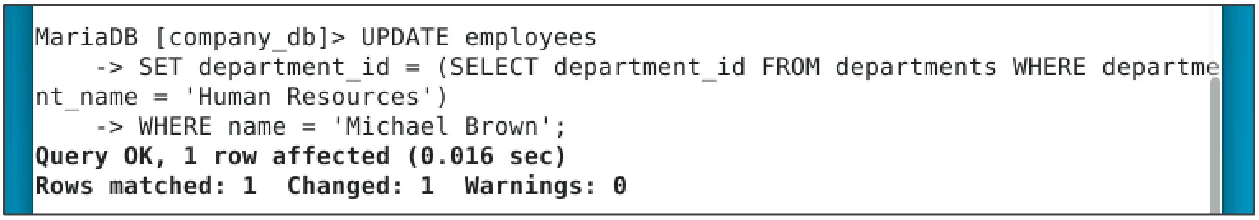
INSERT INTO employees (name, position, department_id) VALUES ('Michael Brown', 'IT Analyst', 2);



```
MariaDB [company_db]> INSERT INTO employees (name, position, department_id) VALU
ES ('Michael Brown', 'IT Analyst', 2);
Query OK, 1 row affected (0.002 sec)

MariaDB [company_db]>
```

2.4 Change an employee's department:

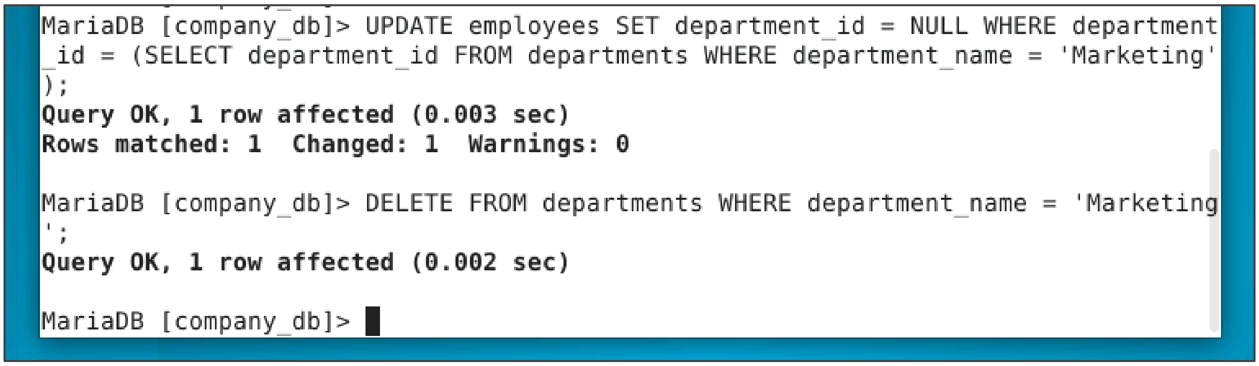
UPDATE employees
SET department_id = (SELECT department_id FROM departments WHERE department_name = 'Human Resources')
WHERE name = 'Michael Brown';



```
MariaDB [company_db]> UPDATE employees
    -> SET department_id = (SELECT department_id FROM departments WHERE departme
nt_name = 'Human Resources')
    -> WHERE name = 'Michael Brown';
Query OK, 1 row affected (0.016 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

2.5 Delete a department and handle the employee records appropriately (e.g., set their **department_id** to **NULL**):

UPDATE employees SET department_id = NULL WHERE department_id = (SELECT department_id FROM departments WHERE department_name = 'Marketing');
DELETE FROM departments WHERE department_name = 'Marketing';



```
MariaDB [company_db]> UPDATE employees SET department_id = NULL WHERE department
_id = (SELECT department_id FROM departments WHERE department_name = 'Marketing'
);
Query OK, 1 row affected (0.003 sec)
Rows matched: 1  Changed: 1  Warnings: 0

MariaDB [company_db]> DELETE FROM departments WHERE department_name = 'Marketing
';
Query OK, 1 row affected (0.002 sec)

MariaDB [company_db]>
```

By following these steps, you have successfully demonstrated how to set up related tables in a MySQL database and perform various operations such as querying, inserting, updating, and deleting data while maintaining the integrity and relationships between tables.