

Certura Internship Program (Database Handling)

Task No: 02

Submitted by:

Huzaifa Waqar.

Submitted to:

Certura.

✓ TOOLS USED

1. **MySQL Server**
 - Database engine.
 2. **MySQL Workbench**
 - GUI tool for writing queries and managing the database.
-

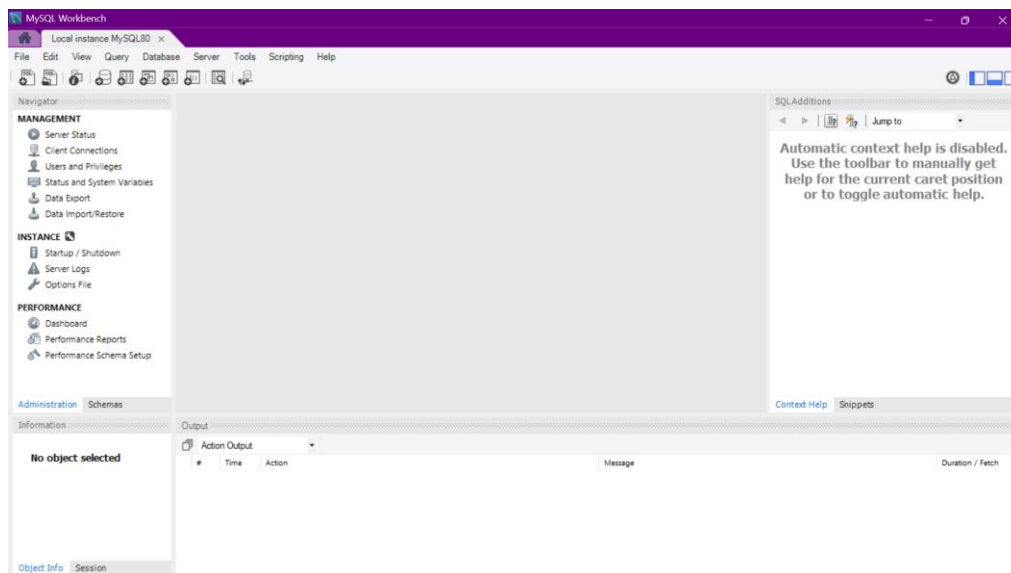
□ TASK: 2

Goal: Practice JOIN operations on multiple tables.
We'll go step-by-step, practically solving it like you are doing it live.

🔗 How to Solve the Task (Detailed Steps)

Step 1: Open MySQL Workbench and Connect

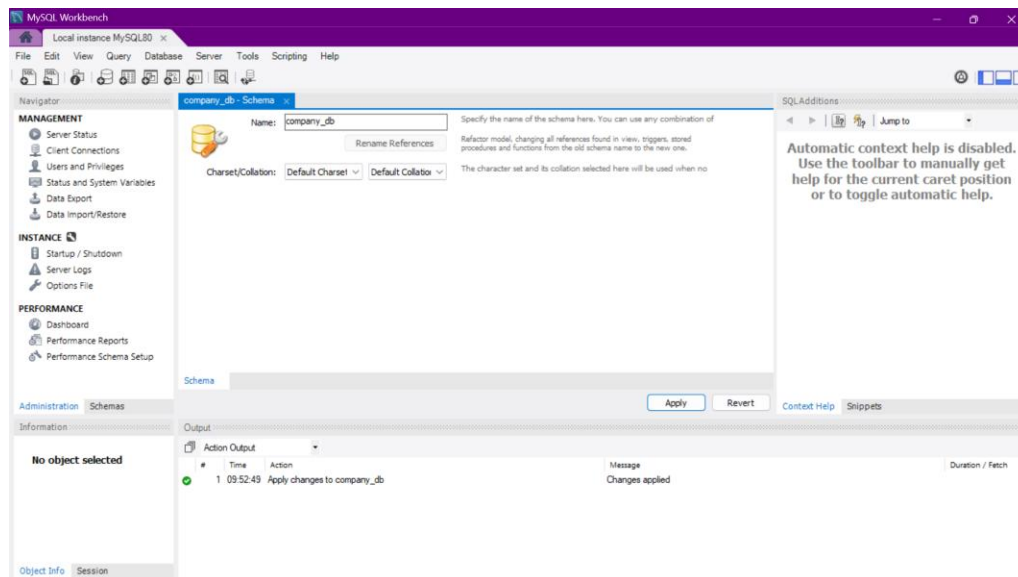
- Open MySQL Workbench.
- Connect to your local MySQL server (using the saved connection).
- You'll reach the SQL Editor screen.



Step 2: Create a New Schema (Database)

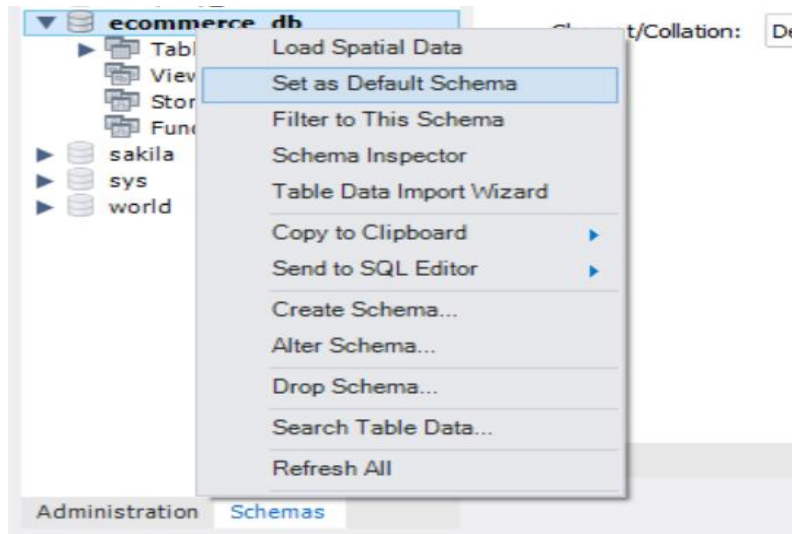
- Click on the "Schemas" tab (left-side).
- Right-click → "Create Schema".
- Name it something like: **company_db**.
- Click **Apply** → **Apply** → **Finish**.

✓ Your database (**company_db**) is now created.



Step 3: Create Tables

☞ Set your schema as default: • Right-click on **company_db** → "Set as Default Schema".



Now create two tables: employees and departments.

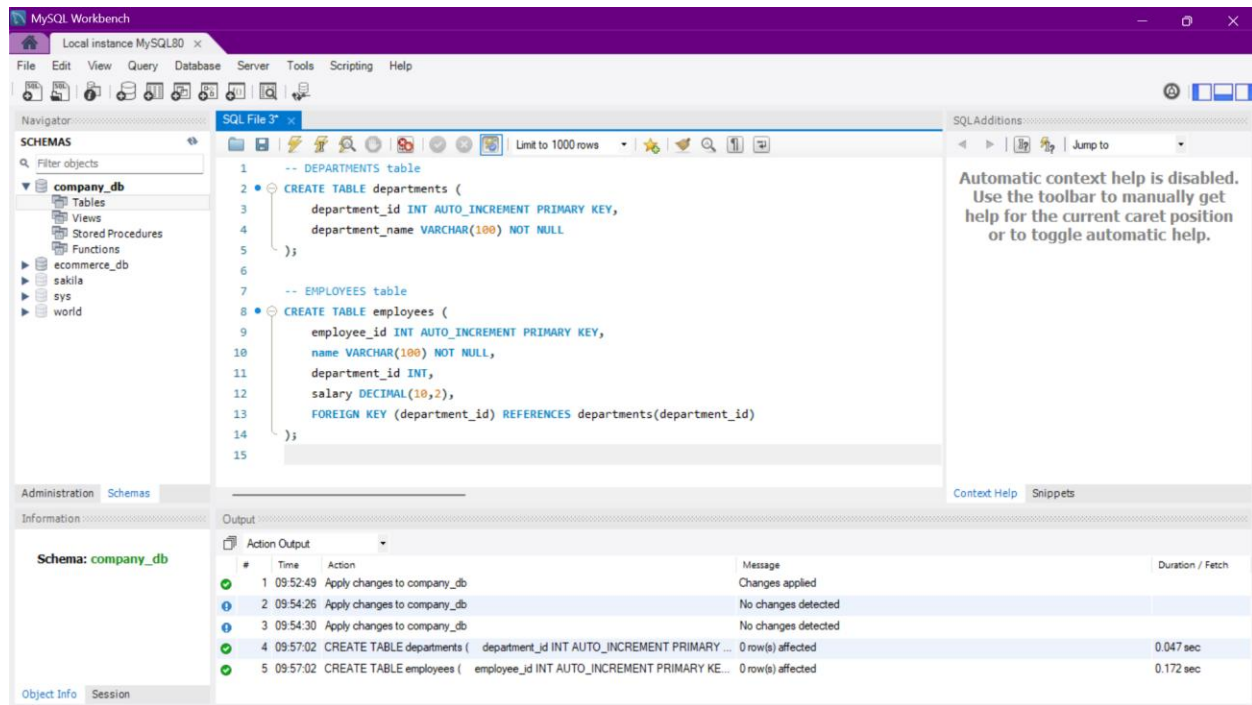
Paste this SQL code:

```
-- Drop existing tables if they exist to avoid errors
DROP TABLE IF EXISTS employees;
DROP TABLE IF EXISTS departments;

-- DEPARTMENTS table creation
CREATE TABLE departments (
    department_id INT AUTO_INCREMENT PRIMARY KEY,
    department_name VARCHAR(100) NOT NULL
);

-- EMPLOYEES table creation
CREATE TABLE employees (
    employee_id INT AUTO_INCREMENT PRIMARY KEY,
    name VARCHAR(100) NOT NULL,
    department_id INT NULL, -- Allow NULL values for department_id
    salary DECIMAL(10,2),
    FOREIGN KEY (department_id) REFERENCES departments(department_id)
);
```

✓ This will create two tables with a **foreign key** relationship between them.



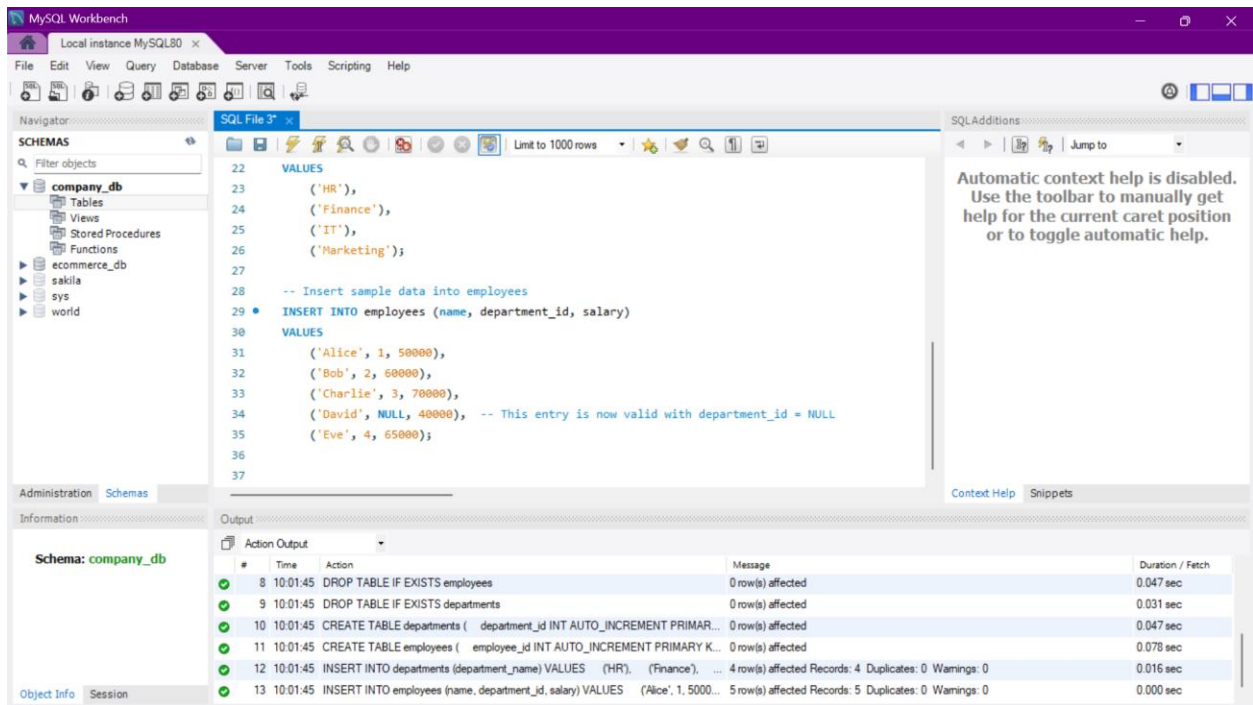
Step 4: Insert Sample Data

Insert some basic data to work with:

```
-- Insert sample data into departments
INSERT INTO departments (department_name)
VALUES
    ('HR'),
    ('Finance'),
    ('IT'),
    ('Marketing');

-- Insert sample data into employees
INSERT INTO employees (name, department_id, salary)
VALUES
    ('Alice', 1, 50000),
    ('Bob', 2, 60000),
    ('Charlie', 3, 70000),
    ('David', NULL, 40000), -- This entry is now valid with department_id =
NULL
    ('Eve', 4, 65000);
```

✓ This will populate your tables with sample data.



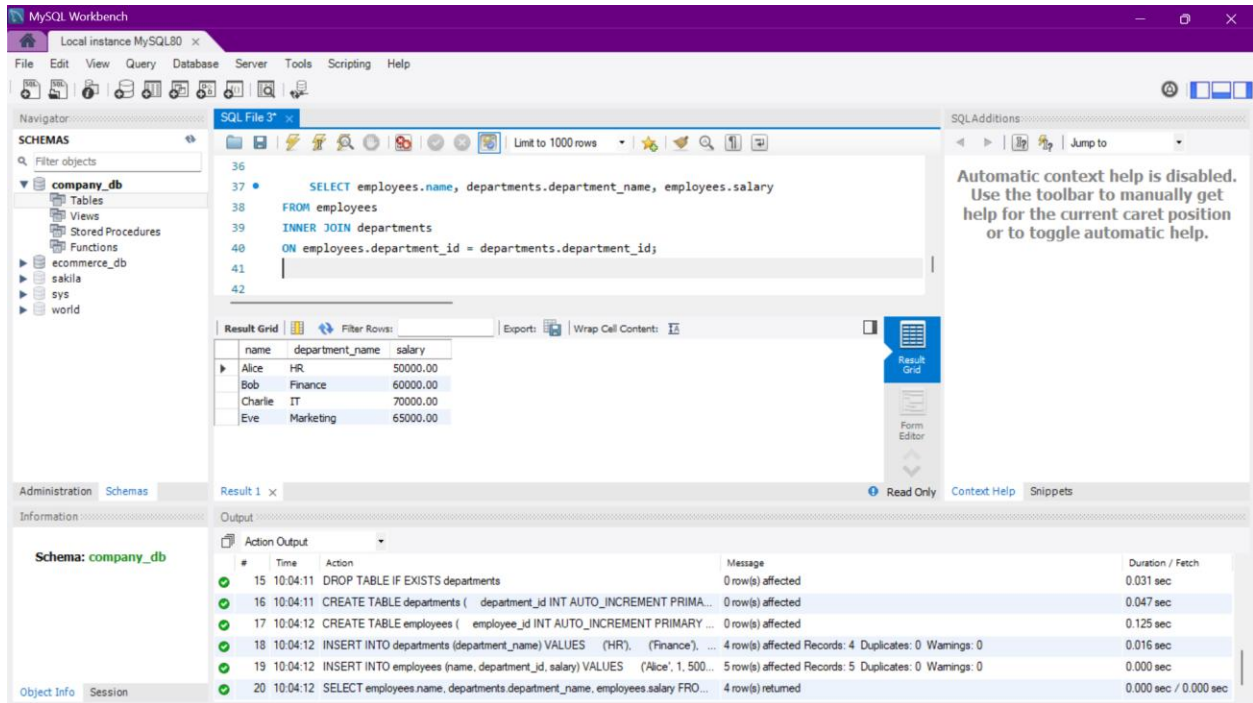
Step 5: Write Queries for INNER JOIN, LEFT JOIN, RIGHT JOIN

Now we'll perform **advanced JOINS**:

★ INNER JOIN (Fetch employees with matching departments)

```
SELECT employees.name, departments.department_name, employees.salary
FROM employees
INNER JOIN departments
ON employees.department_id = departments.department_id;
```

✓ This will show only employees who are linked to a department.



★ LEFT JOIN (Fetch all employees even if they don't belong to any department)

```

SELECT employees.name, departments.department_name, employees.salary
FROM employees
LEFT JOIN departments
ON employees.department_id = departments.department_id;

```

✓ This will show all employees, and for those without a department, NULL will appear.

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```

41
42 • SELECT employees.name, departments.department_name, employees.salary
43 FROM employees
44 LEFT JOIN departments
45 ON employees.department_id = departments.department_id;
46
47

```

The Result Grid shows the following data:

name	department_name	salary
Alice	HR	50000.00
Bob	Finance	60000.00
Charlie	IT	70000.00
David	Marketing	40000.00
Eve	Marketing	65000.00

The Output tab shows the following actions:

#	Time	Action	Message	Duration / Fetch
23	10:04:45	CREATE TABLE departments (department_id INT AUTO_INCREMENT PRIMARY KEY, department_name VARCHAR(100) NOT NULL);	0 row(s) affected	0.032 sec
24	10:04:45	CREATE TABLE employees (employee_id INT AUTO_INCREMENT PRIMARY KEY, name VARCHAR(100) NOT NULL, department_id INT, salary DECIMAL(10,2));	0 row(s) affected	0.078 sec
25	10:04:45	INSERT INTO departments (department_name) VALUES ('HR'), ('Finance'), ('IT'), ('Marketing');	4 row(s) affected Records: 4 Duplicates: 0 Warnings: 0	0.016 sec
26	10:04:45	INSERT INTO employees (name, department_id, salary) VALUES ('Alice', 1, 50000), ('Bob', 2, 60000), ('Charlie', 3, 70000), ('David', 4, 40000), ('Eve', 4, 65000);	5 row(s) affected Records: 5 Duplicates: 0 Warnings: 0	0.000 sec
27	10:04:45	SELECT employees.name, departments.department_name, employees.salary FROM employees LEFT JOIN departments ON employees.department_id = departments.department_id;	4 row(s) returned	0.016 sec / 0.000 sec
28	10:04:46	SELECT employees.name, departments.department_name, employees.salary FROM employees LEFT JOIN departments ON employees.department_id = departments.department_id;	5 row(s) returned	0.000 sec / 0.000 sec

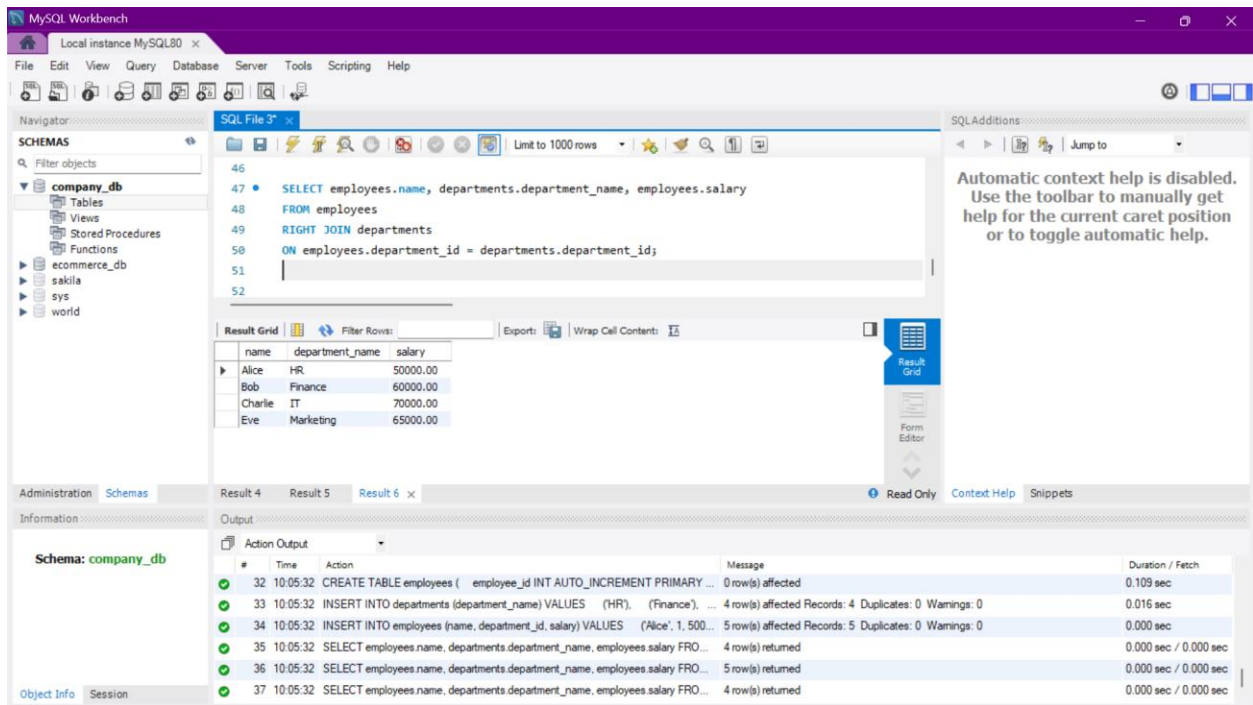
★ RIGHT JOIN (Fetch all departments even if no employee is assigned)

```

SELECT employees.name, departments.department_name, employees.salary
FROM employees
RIGHT JOIN departments
ON employees.department_id = departments.department_id;

```

✓ This will show all departments, and if a department has no employees, NULL will appear under employee name.



Step 6: Execute and Test the Queries

- Highlight each query one by one.
- Click the ⚡ Execute button.
- Check the results for accuracy.

✓ Make sure:

- INNER JOIN → Only matched records.
- LEFT JOIN → All employees shown.
- RIGHT JOIN → All departments shown.

📁 Final Summary (In Short)

Step	Task
1	Open Workbench and connect
2	Create schema (company_db)
3	Create <code>employees</code> and <code>departments</code> tables
4	Insert sample data
5	Write and execute JOIN queries
6	Verify results
