**Setting up a Relational Database**

**Objective:** by the end of this activity, you will be able to configure MySQL Server and use Visual Studio Code to work with the database.

**Step 1: Set Up the Database Connection in VS Code**

Ensure your connection to the MySQL server is active and ready for running SQL commands.

**Steps:**

1. Open Visual Studio Code.
2. Click on the MySQL Shell icon in the sidebar to access the extension.
3. Connect to your MySQL server:
   * Click New Connection to open a connection to MySQL
   * If prompted, enter your MySQL server details (e.g., host: localhost, user: root, password: password).
4. Confirm the connection by running a simple query. In the Editor, create the following SQL statement SHOW DATABASES;
5. Execute the query by selecting the Execute (lightning bolt) button above the code window.

**Step 2: Configure the Database**

You will create a database and configure basic settings for a relational database.

**Steps:**

1. In the Editor, create a new database by executing the following SQL command: CREATE DATABASE CompanyDB;
2. Set the new database as the default for your session: USE CompanyDB;
3. Create a table named Employees with the following structure:
   * EmployeeID (Primary Key, INT, Auto Increment)
   * FirstName (VARCHAR, 50)
   * LastName (VARCHAR, 50)
   * Department (VARCHAR, 50)

**Step З: Configure User Accounts and Permissions**

Secure your database by adding user accounts and setting permissions.

**Steps:**

1. Create a new user for accessing the database: CREATE USER 'manager'@'localhost' IDENTIFIED BY 'StrongPassword123';
2. Grant this user permissions for the CompanyDB database: GRANT ALL PRIVILEGES ON CompanyDB.\* TO 'manager'@'localhost';
3. Test the new user by connecting to the database using this account**.**

**Step 4: Test and Verify the Setup**

Ensure that the database and user account are functioning as expected.

**Steps:**

1. Insert a sample record into the Employees table: INSERT INTO Employees (FirstName, LastName, Department)  VALUES ('John', 'Doe', 'HR');
2. Query the table to confirm the record was inserted: SELECT \* FROM Employees;
3. Verify that the user manager can access and query the database by logging in and running the same query.

**Test.sql:**

-- Step 1: Create Database

**DROP** **DATABASE** **IF** **EXISTS** CompanyDB;

**CREATE** **DATABASE** CompanyDB;

USE CompanyDB;

-- Step 2: Create Table

**DROP** **TABLE** **IF** **EXISTS** Employees;

**CREATE** **TABLE** Employees (

EmployeeID INT AUTO\_INCREMENT **PRIMARY** **KEY**,

FirstName VARCHAR(**50**),

LastName VARCHAR(**50**),

Department VARCHAR(**50**)

);

-- Step 3: Insert Records

**INSERT** **INTO** Employees (FirstName, LastName, Department)

**VALUES**

('John', 'Doe', 'HR'),

('Ivan', 'Petrov', 'IT'),

('Anna', 'Smirnova', 'Finance');

-- Step 4: Verify

**SELECT** **DATABASE**();

**SHOW** TABLES;

**DESCRIBE** Employees;

**SELECT** \* **FROM** Employees;