Writing SELECT Statements

Activity: Retrieving and Manipulating Data with SELECT, WHERE, and ORDER BY

This activity will guide you through setting up a sample database, writing basic SQL queries to retrieve data, and utilizing filtering and sorting functionalities. By the end, you will be able to apply SELECT statements with WHERE clauses and ORDER BY to manipulate and retrieve data effectively.

Step 1: Prepare for the Database Setup

You will create a sample database and a table to work with. This step involves setting up MySQL and populating the database with sample data.

Instructions:

1. Open Visual Studio Code and connect to your MySQL database.
2. Create a new database called EmployeeDB by running the following command: CREATE DATABASE EmployeeDB;
3. Switch to the EmployeeDB database: USE EmployeeDB;
4. Create a table named Employees with the following columns:
   - ID (INT, Primary Key, Auto Increment)
   - FirstName (VARCHAR(50))
   - LastName (VARCHAR(50))
   - Department (VARCHAR(50))
   - Salary (DECIMAL(10,2))
   - YearsExperience (INT)
5. Use the following SQL to create and populate the Employees table:

```sql
CREATE TABLE Employees (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Salary DECIMAL(10,2),
    YearsExperience INT
);

INSERT INTO Employees (FirstName, LastName, Department, Salary,
YearsExperience) VALUES
```

```
('John', 'Doe', 'HR', 60000, 10),
('Jane', 'Smith', 'Finance', 70000, 8),
('Michael', 'Brown', 'IT', 50000, 5),
('Emily', 'Davis', 'HR', 45000, 2),
('Chris', 'Wilson', 'Finance', 80000, 15);
```

Step 2: Retrieve Data Using SELECT Statements

Learn how to retrieve data using basic SELECT statements.

Instructions:

1. Write a query to retrieve all columns for all rows in the Employees table.
2. Write a query to retrieve only the FirstName and LastName of employees.
3. Write a query to retrieve unique department names using DISTINCT.

Step 3: Filter Data Using WHERE Clauses

Practice using the WHERE clause to filter data based on conditions.

Instructions:

1. Write a query to retrieve all employees from the HR department.
2. Write a query to find employees in the Finance department with a salary greater than 60,000.
3. Write a query to find employees with more than 5 years of experience and a salary less than 70,000.

Step 4: Sort Data Using ORDER BУ

Learn to organize query results using the ORDER BY clause.

Instructions:

1. Write a query to retrieve all employees sorted by their LastName in ascending order.
2. Write a query to retrieve employees from the HR department, sorted by their Salary in descending order.
3. Write a query to retrieve the top 3 highest earners across all departments.

Step 5: Combine WHERE and ORDER BY

Combine filtering and sorting to write advanced queries.

Instructions:

1. Write a query to retrieve employees from the IT department with more than 3 years of experience, sorted by YearsExperience in descending order.
2. Write a query to retrieve employees with a salary between 50,000 and 75,000, sorted by their FirstName in ascending order.

lab_select.sql:

```sql
-- Step 1. Prepare for the Database Setup
USE DbFullStackCourseraLab;

-- Drop table if it exists (clean state)
DROP TABLE IF EXISTS Employees;

-- Create Employees table
CREATE TABLE Employees (
    ID INT AUTO_INCREMENT PRIMARY KEY,
    FirstName VARCHAR(50),
    LastName VARCHAR(50),
    Department VARCHAR(50),
    Salary DECIMAL(10,2),
    YearsExperience INT
);

-- Insert sample data
INSERT INTO Employees (FirstName, LastName, Department, Salary,
YearsExperience) VALUES
('John', 'Doe', 'HR', 60000, 10),
('Jane', 'Smith', 'Finance', 70000, 8),
('Michael', 'Brown', 'IT', 50000, 5),
('Emily', 'Davis', 'HR', 45000, 2),
('Chris', 'Wilson', 'Finance', 80000, 15);

-- ============================================
-- Step 2: Retrieve Data Using SELECT Statements
-- ============================================

-- 2.1 Retrieve all columns
SELECT * FROM Employees;

-- 2.2 Retrieve only first and last names
SELECT FirstName, LastName FROM Employees;

-- 2.3 Retrieve unique department names
SELECT DISTINCT Department FROM Employees;

-- ============================================
-- Step 3: Filter Data Using WHERE Clauses
-- ============================================

-- 3.1 All employees from HR
SELECT * FROM Employees
WHERE Department = 'HR';

-- 3.2 Finance employees with salary > 60000
SELECT * FROM Employees
WHERE Department = 'Finance' AND Salary > 60000;

-- 3.3 Employees with more than 5 years experience and salary < 70000
SELECT * FROM Employees
WHERE YearsExperience > 5 AND Salary < 70000;

-- ============================================
-- Step 4: Sort Data Using ORDER BY
-- ============================================
-- 4.1 All employees sorted by last name ascending
SELECT * FROM Employees
ORDER BY LastName ASC;

-- 4.2 HR employees sorted by salary descending
SELECT * FROM Employees
WHERE Department = 'HR'
ORDER BY Salary DESC;

-- 4.3 Top 3 employees by salary
SELECT * FROM Employees
ORDER BY Salary DESC
LIMIT 3;

-- ============================================
-- Step 5: Combine WHERE and ORDER BY
-- ============================================

-- 5.1 IT employees with > 3 years experience, sorted by YearsExperience descending
SELECT * FROM Employees
WHERE Department = 'IT' AND YearsExperience > 3
ORDER BY YearsExperience DESC;

-- 5.2 Employees with salary between 50,000 and 75,000, sorted by FirstName ascending
SELECT * FROM Employees
WHERE Salary BETWEEN 50000 AND 75000
ORDER BY FirstName ASC;
```