



## IT 131 Advanced Database Systems Try IT!

### Introduction

Welcome to this SQL hands-on workbook! In this guide, you will learn how to write and execute SQL SELECT statements through step-by-step activities. By the end, you'll understand how to create a database, query data, and manipulate results using filtering, sorting, pattern matching, and more. Let's get started!

### Step 1: Setting Up the Environment

#### 1.1 Create the Database

To begin, we'll create a database called **sql\_practice**. This will serve as the foundation for all the activities.

```
CREATE DATABASE sql_practice;
```

#### 1.2 Create the Tables

Next, we will create two tables: Employees and Projects. These tables will store data about employees and their respective projects.

```
CREATE TABLE Employees (  
    EmployeeID INT AUTO_INCREMENT PRIMARY KEY,  
    FirstName VARCHAR(50),  
    LastName VARCHAR(50),  
    Department VARCHAR(50),  
    Salary DECIMAL(10, 2),  
    HireDate DATE  
);
```

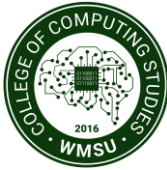
```
CREATE TABLE Projects (  
    ProjectID INT AUTO_INCREMENT PRIMARY KEY,  
    ProjectName VARCHAR(100),  
    Department VARCHAR(50),  
    StartDate DATE,  
    EndDate DATE  
);
```

#### 1.3 Insert Sample Data

Now that the tables are ready, let's populate them with sample data.

```
-- Insert sample data into Employees  
INSERT INTO Employees (FirstName, LastName, Department, Salary, HireDate) VALUES  
( 'John', 'Doe', 'IT', 60000, '2021-06-15'),  
( 'Jane', 'Smith', 'HR', 55000, '2019-03-20'),  
( 'Alice', 'Johnson', 'IT', 65000, '2022-01-10'),  
( 'Bob', 'Brown', 'Finance', 70000, '2020-11-05'),  
( 'Emma', 'Davis', 'HR', 52000, '2023-07-01');
```

```
-- Insert sample data into Projects  
INSERT INTO Projects (ProjectName, Department, StartDate, EndDate) VALUES  
( 'Website Redesign', 'IT', '2023-01-01', '2023-06-30'),  
( 'Employee Wellness Program', 'HR', '2023-04-01', '2023-12-31'),  
( 'Audit Preparation', 'Finance', '2023-02-01', '2023-05-31'),  
( 'Cloud Migration', 'IT', '2024-01-01', '2024-12-31');
```



1.4 Verify the Data

The database tables should look like the examples below once you have inserted the data:

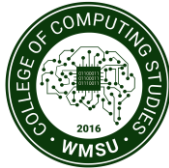
Employees Table:

EmployeeID	FirstName	LastName	Department	Salary	HireDate
1	John	Doe	IT	60000.00	2021-06-15
2	Jane	Smith	HR	55000.00	2019-03-20
3	Alice	Johnson	IT	65000.00	2022-01-10
4	Bob	Brown	Finance	70000.00	2020-11-05
5	Emma	Davis	HR	52000.00	2023-07-01

Projects Table:

ProjectID	ProjectName	Department	StartDate	EndDate
1	Website Redesign	IT	2023-01-01	2023-06-30
2	Employee Wellness Program	HR	2023-04-01	2023-12-31
3	Audit Preparation	Finance	2023-02-01	2023-05-31
4	Cloud Migration	IT	2024-01-01	2024-12-31

**Congratulations!** You have successfully created the database, tables, and populated them with sample data. Now, let’s start querying the data!



## Step 2: Basic Queries

### 2.1 Select All Data

Let's begin by retrieving all the data from the Employees table. This will display every column and row.

```
SELECT * FROM Employees;
```

### 2.2 Select Specific Columns

Sometimes, you might only need specific columns. For example, let's retrieve the FirstName, LastName, and Salary of all employees.

```
SELECT FirstName, LastName, Salary FROM Employees;
```

## Step 3: Filtering Data

### 3.1 Filter by Department

To display employees in the IT department only, use the WHERE clause.

```
SELECT * FROM Employees WHERE Department = 'IT';
```

### 3.2 Filter by Salary

To find employees earning more than 60,000, use the following query:

```
SELECT * FROM Employees WHERE Salary > 60000;
```

### 3.3 Filter by Date

To find employees hired after January 1, 2021, use a WHERE clause with a date condition.

```
SELECT * FROM Employees WHERE HireDate > '2021-01-01';
```

## Step 4: Pattern Matching

### 4.1 Find Names Starting with 'J'

To find employees whose first names start with the letter 'J', use the LIKE operator.

```
SELECT * FROM Employees WHERE FirstName LIKE 'J%';
```

### 4.2 Find Names Containing 'a'

To find employees whose names contain the letter 'a', use %a%.

```
SELECT * FROM Employees WHERE FirstName LIKE '%a%';
```

### 4.3 Find Names with a Specific Pattern

To find employees whose first names are exactly 5 characters long, use underscores (\_).

```
SELECT * FROM Employees WHERE FirstName LIKE '_____';
```



## Step 5: Use Aliases for Columns

### 5.1 Display columns with custom names using AS.

Renames FirstName to "First Name" and LastName to "Last Name" in the result set.

```
SELECT FirstName AS 'First Name', LastName AS 'Last Name' FROM Employees;
```

## Step 6: Concatenate Columns

### 6.1 Combine two columns into one for display.

Combines the first and last names into a single column labeled "Full Name."

```
SELECT CONCAT(FirstName, ' ', LastName) AS 'Full Name' FROM Employees;
```

## Step 7: Filter Data with Multiple Conditions

### 7.1 Use logical operators AND to filter results.

Displays employees in the IT department with salaries greater than 60,000.

```
SELECT * FROM Employees WHERE Department = 'IT' AND Salary > 60000;
```

### 7.2 Use logical operators like OR to filter results.

Retrieves employees in either the IT or HR departments.

```
SELECT * FROM Employees WHERE Department = 'IT' OR Department = 'HR';
```

### 7.3 Combine multiple conditions using AND/OR.

Find Employees in HR or Finance with Salaries Below 55,000

```
SELECT * FROM Employees WHERE (Department = 'HR' OR Department = 'Finance') AND Salary < 55000;
```

## Step 8: Using IN and NOT

### 8.1 Using IN

Use IN to match multiple values in a single column.

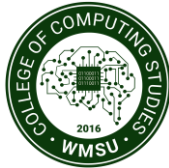
Find Employees in Specific Departments

```
SELECT * FROM Employees WHERE Department IN ('IT', 'HR');
```

Use IN to filter projects by department.

Find Projects in Specific Departments

```
SELECT * FROM Projects WHERE Department IN ('Finance', 'IT');
```



## 8.2 Using NOT

**Use NOT to exclude specific values.**

Find Employees Not in the IT Department

```
SELECT * FROM Employees WHERE Department NOT IN ('IT');
```

**Combine NOT with LIKE.**

Find Employees Whose Last Name Does Not Contain 'a'

```
SELECT * FROM Employees WHERE LastName NOT LIKE '%a%';
```

## Step 9: Using BETWEEN

**9.1 Use BETWEEN to filter by a range of values.**

Find Employees Earning Between 55,000 and 70,000

```
SELECT * FROM Employees WHERE Salary BETWEEN 55000 AND 70000;
```

**9.2 Use BETWEEN for date ranges.**

Find Projects with Dates in 2023

```
SELECT * FROM Projects WHERE StartDate BETWEEN '2023-01-01' AND '2023-12-31';
```

## Step 10: Use DISTINCT to Remove Duplicates

**10.1 Show unique departments in the Employees table.**

Displays each department only once.

```
SELECT DISTINCT Department FROM Employees;
```

## Step 11: Limit the Number of Rows Returned

**11.1 Retrieve only the first few rows of data.**

Displays the first 3 rows of the Employees table.

```
SELECT * FROM Employees LIMIT 3;
```



## Step 12: SELECT Queries Using Mathematical Operators

### 12.1 Add Values

Calculate the total salary with a fixed allowance.

```
SELECT FirstName, Salary, Salary + 5000 AS 'Total Salary' FROM Employees;
```

### 12.2 Subtract Values

Find the remaining salary after a deduction.

```
SELECT FirstName, Salary, Salary - 3000 AS 'Net Salary' FROM Employees;
```

### 12.3 Multiply Values

Calculate the annual salary of employees.

```
SELECT FirstName, Salary, Salary * 12 AS 'Annual Salary' FROM Employees;
```

### 12.4 Divide Values

Calculate the monthly salary when given an annual salary.

```
SELECT FirstName, Salary * 12 AS 'Annual Salary', (Salary * 12) / 12 AS 'Monthly Salary' FROM Employees;
```

### 12.5 Use Parentheses for Order of Operations

Calculate adjusted salary after multiple operations.

```
SELECT FirstName, Salary, (Salary + 5000) * 1.1 AS 'Adjusted Salary' FROM Employees;
```

### 12.6 Calculate Percentages

Find the percentage of an employee's salary compared to the department's total salary.

```
SELECT FirstName, Salary, (Salary / (SELECT SUM(Salary) FROM Employees WHERE Department = 'IT') * 100) AS 'Percentage' FROM Employees WHERE Department = 'IT';
```