

## SQL LAB -6

(Boolean Operator, And Operator, OR operator)

**Submitted by: Thummala Pavani**

**AF ID : AF036672**

**Batch ID : ANP-C7281**

### Lab 1:

#### Database Schema

Already we have created an Employee table in day 2 lab, let's utilize this.

**Task:** Add two more columns in the Employee table named Salary and department and add data into it. Now Imagine you work for a company with various departments, and there is a need to analyze employee salaries within the IT department. Write a query to retrieve all employees from the "employee" table who have a salary greater than 50000 and are in the 'IT' department

```
mysql> -- Add Salary and Department columns to the Employee table
mysql> ALTER TABLE Employee
  -> ADD COLUMN Salary DECIMAL(10, 2),
  -> ADD COLUMN Department VARCHAR(100);
Query OK, 0 rows affected (0.05 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+-----+
| emp_id | firstname | lastname | age | email | Salary | Department |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | aadhya | singh | 20 | aadhya20@gmail.com | NULL | NULL |
| 2 | laxman | pandya | 25 | laxmanp@gmail.com | NULL | NULL |
| 3 | hrithik | roshan | 24 | hrithikro@gmail.com | NULL | NULL |
+-----+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

```
mysql> UPDATE Employee
  -> SET Salary = 60000.00, Department = 'IT'
  -> WHERE emp_id = 1;
Query OK, 1 row affected (0.01 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> UPDATE Employee
  -> SET Salary = 55000.00, Department = 'Finance'
  -> WHERE emp_id = 2;
Query OK, 1 row affected (0.04 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> UPDATE Employee
  -> SET Salary = 70000.00, Department = 'IT'
  -> WHERE emp_id = 3;
Query OK, 1 row affected (0.03 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> select * from employee;
```

emp_id	firstname	lastname	age	email	Salary	Department
1	aadhya	singh	20	aadhya20@gmail.com	60000.00	IT
2	laxman	pandya	25	laxmanp@gmail.com	55000.00	Finance
3	hrithik	roshan	24	hrithikro@gmail.com	70000.00	IT

3 rows in set (0.00 sec)

```
mysql> SELECT *  
-> FROM Employee  
-> WHERE Salary > 50000 AND Department = 'IT';
```

emp_id	firstname	lastname	age	email	Salary	Department
1	aadhya	singh	20	aadhya20@gmail.com	60000.00	IT
3	hrithik	roshan	24	hrithikro@gmail.com	70000.00	IT

2 rows in set (0.00 sec)

## Lab 2:

### Database Schema

Use our database Ecommerce to complete the task.

**Task:** Imagine you are managing an e-commerce platform, and the holiday season is approaching. To capitalize on the festive spirit and boost sales, you decide to organize a special seasonal sale featuring electronics. The goal is to offer discounts on electronics and include products with a price less than rs. 70,000 in the promotion. Write a query to find products from the "product" table that are either in the 'Electronics' category or have a price less than 70000.

```
mysql> create database ecommerce;  
Query OK, 1 row affected (0.02 sec)
```

```
mysql> use ecommerce;  
Database changed
```

```
mysql> -- Create the product table  
mysql> CREATE TABLE product (  
->     product_id INT PRIMARY KEY,  
->     product_name VARCHAR(100),  
->     category VARCHAR(100),  
->     price DECIMAL(10, 2),  
->     discount DECIMAL(5, 2) -- Discount as a percentage  
-> );  
Query OK, 0 rows affected (0.09 sec)
```

```
mysql> describe product;
```

Field	Type	Null	Key	Default	Extra
product_id	int	NO	PRI	NULL	
product_name	varchar(100)	NO		NULL	
category	varchar(100)	NO		NULL	
price	decimal(10,2)	YES		NULL	
discount	decimal(5,2)	YES		NULL	

```
5 rows in set (0.00 sec)
```

```
mysql> -- Insert sample data into the product table
```

```
mysql> INSERT INTO product (product_id, product_name, category, price, discount) VALUES  
-> (1, 'Laptop', 'Electronics', 65000.00, 10.00), -- 10% discount  
-> (2, 'Smartphone', 'Electronics', 55000.00, 15.00), -- 15% discount  
-> (3, 'Refrigerator', 'Home Appliances', 80000.00, 5.00), -- 5% discount  
-> (4, 'Television', 'Electronics', 72000.00, 20.00), -- 20% discount  
-> (5, 'Microwave Oven', 'Home Appliances', 30000.00, 12.00), -- 12% discount  
-> (6, 'Tablet', 'Electronics', 45000.00, 10.00), -- 10% discount  
-> (7, 'Washing Machine', 'Home Appliances', 50000.00, 8.00); -- 8% discount
```

```
Query OK, 7 rows affected (0.01 sec)
```

```
Records: 7 Duplicates: 0 Warnings: 0
```

```
mysql> SELECT product_id, product_name, category, price, discount  
-> FROM product  
-> WHERE category = 'Electronics' OR price < 70000;
```

product_id	product_name	category	price	discount
1	Laptop	Electronics	65000.00	10.00
2	Smartphone	Electronics	55000.00	15.00
4	Television	Electronics	72000.00	20.00
5	Microwave Oven	Home Appliances	30000.00	12.00
6	Tablet	Electronics	45000.00	10.00
7	Washing Machine	Home Appliances	50000.00	8.00

```
6 rows in set (0.00 sec)
```

### Lab 3.

**Task:** Imagine you are an HR analyst responsible for conducting a comprehensive analysis of average salaries across different departments within a company. The goal is to understand and compare the average salaries of employees in various departments. Write a query to Calculate the average salary of employee in each department from the "employee" table.

```
mysql> CREATE DATABASE CompanyHR;
```

```
Query OK, 1 row affected (0.04 sec)
```

```
mysql> USE CompanyHR;  
Database changed
```



```
mysql> CREATE TABLE Employee (
->     emp_id INT PRIMARY KEY,
->     first_name VARCHAR(50),
->     last_name VARCHAR(50),
->     age INT,
->     email VARCHAR(100),
->     Salary DECIMAL(10, 2),
->     Department VARCHAR(100)
-> );
Query OK, 0 rows affected (0.04 sec)
```

```
mysql> -- Insert sample data into the Employee table
mysql> INSERT INTO Employee (emp_id, first_name, last_name, age, email, Salary, Department) VALUES
-> (1, 'John', 'Doe', 30, 'john.doe@example.com', 60000.00, 'IT'),
-> (2, 'Jane', 'Smith', 45, 'jane.smith@example.com', 55000.00, 'Finance'),
-> (3, 'Alice', 'Johnson', 28, 'alice.johnson@example.com', 70000.00, 'IT'),
-> (4, 'Bob', 'Brown', 50, 'bob.brown@example.com', 48000.00, 'HR'),
-> (5, 'Charlie', 'Davis', 35, 'charlie.davis@example.com', 65000.00, 'IT'),
-> (6, 'Diana', 'Evans', 40, 'diana.evans@example.com', 52000.00, 'Finance'),
-> (7, 'Evan', 'Garcia', 32, 'evan.garcia@example.com', 47000.00, 'HR');
Query OK, 7 rows affected (0.01 sec)
Records: 7 Duplicates: 0 Warnings: 0
```

```
mysql> select*from employee
-> ;
```

emp_id	first_name	last_name	age	email	Salary	Department
1	John	Doe	30	john.doe@example.com	60000.00	IT
2	Jane	Smith	45	jane.smith@example.com	55000.00	Finance
3	Alice	Johnson	28	alice.johnson@example.com	70000.00	IT
4	Bob	Brown	50	bob.brown@example.com	48000.00	HR
5	Charlie	Davis	35	charlie.davis@example.com	65000.00	IT
6	Diana	Evans	40	diana.evans@example.com	52000.00	Finance
7	Evan	Garcia	32	evan.garcia@example.com	47000.00	HR

```
7 rows in set (0.00 sec)
```

```
mysql> SELECT Department, AVG(Salary) AS AverageSalary
-> FROM Employee
-> GROUP BY Department;
```

Department	AverageSalary
IT	65000.000000
Finance	53500.000000
HR	47500.000000

```
3 rows in set (0.00 sec)
```

## ChatGPT Exercise

Using ChatGPT generates SQL queries of the below problem .

**Scenario 1:** Determine the average age of employees in each department from the "employees" table. We have an "Employee" table with the following columns: employee\_id, employee\_name, department, and salary and you want to find the average salary for each department. Generate the chatGPT prompt for the above scenario.

```
SELECT department, AVG(age) AS average_age
FROM Employee
GROUP BY department;
```

**ChatGPT Prompt:** "Generate an SQL query to determine the average age of employees in each department from the 'Employee' table. The table has columns: 'employee\_id' (integer), 'employee\_name' (string), 'department' (string), and 'salary' (decimal). You need to calculate the average age for each department."