**Lab # 10:**

**Lab Title:**

**“Operators In MySQL”**

**Lab Objectives:**

* To understand the working of SELECT queries and use different conditions involving logical and arithmetic operators to get the user desired results through queries.

**Introduction:**

In MySQL, operators serve as powerful tools for performing a wide array of operations within SQL queries. They are instrumental in conducting calculations, making comparisons, executing logical evaluations, and performing bitwise operations on data stored within databases.

**Types of Operators in MySQL:**

1. **Arithmetic Operators:** Enable mathematical operations like addition, subtraction, multiplication, division, and modulo operations.
2. **Comparison Operators:** Allow for comparisons between values, such as equality, inequality, greater than, less than, and their combinations.
3. **Logical Operators:** Aid in constructing compound conditions by utilizing logical AND, OR, and NOT operations.
4. **Unary Operators:** Modify the sign or value of an operand (e.g., **-** for negation, **+** for a positive value).
5. **Bitwise Operators:** Facilitate operations at the bit level, such as bitwise AND, OR, XOR, and NOT.

**In-Lab Tasks:**

**Creating Table and Database:**

|  |
| --- |
| CREATE DATABASE lab\_10\_db;  USE lab\_10\_db;  CREATE TABLE customer (  cus\_id INT PRIMARY KEY,  cus\_name VARCHAR(255) NOT NULL,  cus\_city VARCHAR(255) NOT NULL,  grad INT NOT NULL,  opening\_amt DECIMAL(10,2) NOT NULL,  receive\_amt DECIMAL(10,2) NOT NULL,  payment\_amt DECIMAL(10,2) NOT NULL,  phone\_no VARCHAR(20)  );  INSERT INTO customer (cus\_id, cus\_name, cus\_city, grad, opening\_amt, receive\_amt, payment\_amt, phone\_no)  VALUES  (13, 'e Holmes', 'London', 2, 6000.00, 5000.00, 7000.00, '2'),  (12, 'Micheal', 'New York', 2, 5000.00, 5000.00, 2000.00, NULL),  (11, 'Albert', 'New york', 3, 6000.00, 7000.00, 6000.00, NULL),  (10, 'Ravindr', 'Bangalo', 3, 4000.00, 7000.00, 4000.00, NULL); |

**Task#01:**

|  |
| --- |
| SELECT cus\_id, cus\_name, (opening\_amt + receive\_amt) AS total\_amount FROM customer  WHERE opening\_amt + receive\_amt > 10000; |

**Task#02:**

|  |
| --- |
| SELECT cus\_id, (opening\_amt - payment\_amt) AS difference\_amount FROM customer; |

**Task#03:**

|  |
| --- |
| SELECT cus\_id, cus\_name, cus\_city, payment\_amt \* 2 AS doubled\_payment FROM customer; |

**Task#04:**

|  |
| --- |
| SELECT \* FROM customer  WHERE payment\_amt = 2000.00; |

**Task#05:**

|  |
| --- |
| SELECT \* FROM customer WHERE payment\_amt > 2000.00;  SELECT \* FROM customer WHERE payment\_amt < 2000.00;  SELECT \* FROM customer WHERE payment\_amt >= 2000.00;  SELECT \* FROM customer WHERE payment\_amt <= 2000.00;  SELECT \* FROM customer WHERE payment\_amt != 2000.00; |

**Task#06:**

|  |
| --- |
| SELECT cus\_id, cus\_name, cus\_city, grad FROM customer  WHERE cus\_city = 'New York' AND grad = 2; |

**Task#07:**

|  |
| --- |
| SELECT \* FROM customer WHERE grad <= 2; |

**Task#08:**

|  |
| --- |
| SELECT \* FROM customer  WHERE receive\_amt BETWEEN 5000.00 AND 10000.00; |

**Task#09:**

|  |
| --- |
| SELECT cus\_id, cus\_name, cus\_city FROM customer  WHERE cus\_name LIKE 'H%'; |

**Task#10:**

|  |
| --- |
| SELECT \* FROM customer  WHERE cus\_city = 'New York' OR cus\_city = 'London'; |

**Task#11:**

|  |
| --- |
| CREATE TABLE customer1 (  cus\_id INT PRIMARY KEY,  cus\_name VARCHAR(255) NOT NULL,  cus\_city VARCHAR(255) NOT NULL,  grad INT NOT NULL,  opening\_amt DECIMAL(10,2) NOT NULL,  receive\_amt DECIMAL(10,2) NOT NULL,  payment\_amt DECIMAL(10,2) NOT NULL,  phone\_no VARCHAR(20)  );  INSERT INTO customer1 (cus\_id, cus\_name, cus\_city, grad, opening\_amt, receive\_amt, payment\_amt, phone\_no)  VALUES  (14, 'Sophia', 'Paris', 2, 3000.00, 2000.00, 1500.00, NULL),  (15, 'Liam', 'London', 3, 7000.00, 8000.00, 5000.00, '1234567890');  SELECT cus\_name AS customer\_name FROM customer  UNION  SELECT cus\_name AS customer\_name FROM customer1; |

**Post-Lab Tasks:**

**Creating Tables:**

|  |
| --- |
| -- creating tables  CREATE TABLE Products (  ProductID INT PRIMARY KEY,  ProductName VARCHAR(255) NOT NULL,  SupplierID INT NOT NULL,  CategoryID INT NOT NULL,  Price DECIMAL(10,2) NOT NULL  );  INSERT INTO Products (ProductID, ProductName, SupplierID, CategoryID, Price)  VALUES  (1, 'Chais', 1, 1, 18.00),  (2, 'Chang', 1, 1, 19.00),  (3, 'Aniseed Syrup', 1, 2, 10.00),  (4, 'Chef Anton\'s Cajun Seasoning', 2, 2, 22.00);  CREATE TABLE OrderDetails (  OrderDetailsID INT PRIMARY KEY,  OrderID INT NOT NULL,  ProductID INT NOT NULL,  Quantity INT NOT NULL  );  INSERT INTO OrderDetails (OrderDetailsID, OrderID, ProductID, Quantity)  VALUES  (1, 10248, 1, 12),  (2, 10248, 2, 10),  (3, 10248, 3, 15),  (4, 10249, 1, 8); |

**Task#01:**

|  |
| --- |
| SELECT ProductName FROM Products  WHERE ProductID = ALL (  SELECT ProductID  FROM OrderDetails  WHERE Quantity = 12  ); |

**Task#02:**

|  |
| --- |
| SELECT ProductName FROM Products  WHERE ProductID = ANY (  SELECT ProductID  FROM OrderDetails  WHERE Quantity > 99  ); |

**Task#03:**

|  |
| --- |
| SELECT OrderID, Quantity,  CASE  WHEN Quantity > 10 THEN 'The amount is greater than 10'  WHEN Quantity = 10 THEN 'The amount is 10'  ELSE 'The amount is under 10'  END AS Profit  FROM OrderDetails; |