Day27_SQL_Operators

June 24, 2025

SQL Operators Overview

Today, we are learning about **SQL operators**, which are used to perform operations on data in a database.
Operators are the building blocks of conditions and calculations in SQL queries.

In this notebook, we will:

- Create a sample `Students` table
- Insert sample data
- Learn and apply **three main types of SQL operators**:

1. Arithmetic Operators :

Used for basic math operations like addition, subtraction, multiplication, division, and modulus.

2. Comparison Operators :

Used to compare values — such as `=`, `<>`, `>`, `<`, `>=`, and `<=`.

3. Logical Operators :

Used to combine multiple conditions — such as `AND`, `OR`, and `NOT`.

This is a hands-on overview to understand how these operators work using real examples.
In the next lessons, we will explore more advanced filtering and pattern-matching techniques using `WHERE`, `LIKE`, `IN`, and `BETWEEN`.

1 Create Database and Table

```
-- Creating a new database
mysql> CREATE DATABASE OperatorDemo;
Query OK, 1 row affected (0.25 sec)

-- Switching to the new database
mysql> USE OperatorDemo;
Database changed

-- Creating a 'Students' table
mysql> CREATE TABLE Students (
    ->   ID INT PRIMARY KEY AUTO_INCREMENT,
    ->   Name VARCHAR(50),
    ->   Marks INT,
```

```
    ->   Age INT
    -> );
Query OK, 0 rows affected (3.20 sec)
```

---

2 Insert Sample Records

```
-- Inserting 5 records
mysql> INSERT INTO Students (Name, Marks, Age) VALUES
    -> ('Akshay', 85, 20),
    -> ('Swara', 92, 19),
    -> ('Ravi', 76, 21),
    -> ('Neha', 89, 22),
    -> ('Amit', 66, 20);
Query OK, 5 rows affected (0.59 sec)
Records: 5  Duplicates: 0  Warnings: 0
```

---

3 Arithmetic Operators

```
-- This query performs addition
mysql> SELECT 50 + 50;
+---------+
| 50 + 50 |
+---------+
|     100 |
+---------+
1 row in set (0.00 sec)

-- Subtraction operation
mysql> SELECT 100 - 80;
+----------+
| 100 - 80 |
+----------+
|       20 |
+----------+
1 row in set (0.00 sec)

-- Multiplication operation
mysql> SELECT 1000 * 3;
+----------+
| 1000 * 3 |
+----------+
|     3000 |
+----------+
1 row in set (0.00 sec)
```

```
-- Division operation
mysql> SELECT 40 / 2;
+---------+
| 40 / 2  |
+---------+
| 20.0000 |
+---------+
1 row in set (0.00 sec)

-- Modulus operation
mysql> SELECT 40 % 2;
+--------+
| 40 % 2 |
+--------+
|      0 |
+--------+
1 row in set (0.00 sec)

-- This adds 5 to each student's marks
mysql> SELECT Name, Marks, Marks + 5 AS UpdatedMarks FROM Students;
+--------+-------+--------------+
| Name   | Marks | UpdatedMarks |
+--------+-------+--------------+
| Akshay |    85 |           90 |
| Swara  |    92 |           97 |
| Ravi   |    76 |           81 |
| Neha   |    89 |           94 |
| Amit   |    66 |           71 |
+--------+-------+--------------+
5 rows in set (0.00 sec)

-- Subtract 2 from age
mysql> SELECT Name, Age, Age - 2 AS Age2YearsAgo FROM Students;
+--------+------+--------------+
| Name   | Age  | Age2YearsAgo |
+--------+------+--------------+
| Akshay |   20 |           18 |
| Swara  |   19 |           17 |
| Ravi   |   21 |           19 |
| Neha   |   22 |           20 |
| Amit   |   20 |           18 |
+--------+------+--------------+
5 rows in set (0.00 sec)
```

---

4 Comparison Operators

```
-- 3 equals 3 → true (1)
mysql> SELECT 3 = 3;
+-------+
| 3 = 3 |
+-------+
|     1 |
+-------+
1 row in set (0.00 sec)

-- 3 equals 4 → false (0)
mysql> SELECT 3 = 4;
+-------+
| 3 = 4 |
+-------+
|     0 |
+-------+
1 row in set (0.00 sec)

-- 85 > -8888 → true
mysql> SELECT 85 > -8888;
+------------+
| 85 > -8888 |
+------------+
|          1 |
+------------+
1 row in set (0.00 sec)

-- 85 < -8888 → false
mysql> SELECT 85 < -8888;
+------------+
| 85 < -8888 |
+------------+
|          0 |
+------------+
1 row in set (0.00 sec)

-- Get all students who scored exactly 85
mysql> SELECT * FROM Students WHERE Marks = 85;
+----+--------+-------+------+
| ID | Name   | Marks | Age  |
+----+--------+-------+------+
|  1 | Akshay |    85 |   20 |
+----+--------+-------+------+
1 row in set (0.00 sec)

-- Students who scored not equal to 85
```

```
mysql> SELECT * FROM Students WHERE Marks <> 85;
+----+-------+-------+------+
| ID | Name  | Marks | Age  |
+----+-------+-------+------+
|  2 | Swara |    92 |   19 |
|  3 | Ravi  |    76 |   21 |
|  4 | Neha  |    89 |   22 |
|  5 | Amit  |    66 |   20 |
+----+-------+-------+------+
4 rows in set (0.00 sec)
```

---

5 Logical Operators

```
-- Students with marks > 80 AND age < 21
mysql> SELECT * FROM Students WHERE Marks > 80 AND Age < 21;
+----+--------+-------+------+
| ID | Name   | Marks | Age  |
+----+--------+-------+------+
|  1 | Akshay |    85 |   20 |
|  2 | Swara  |    92 |   19 |
+----+--------+-------+------+
2 rows in set (0.00 sec)

-- Students with marks > 90 OR age < 20
mysql> SELECT * FROM Students WHERE Marks > 90 OR Age < 20;
+----+-------+-------+------+
| ID | Name  | Marks | Age  |
+----+-------+-------+------+
|  2 | Swara |    92 |   19 |
+----+-------+-------+------+
1 row in set (0.00 sec)

-- This is a SQL query to fetch all records where Age is NOT 20
mysql> SELECT * FROM Students WHERE NOT Age = 20;
+----+-------+-------+------+
| ID | Name  | Marks | Age  |
+----+-------+-------+------+
|  2 | Swara |    92 |   19 |
|  3 | Ravi  |    76 |   21 |
|  4 | Neha  |    89 |   22 |
+----+-------+-------+------+
3 rows in set (0.00 sec)
```

[ ]: