SQL Operators

SQL Operators perform arithmetic, comparison, and logical operations to manipulate and retrieve data from databases.

Operators in SQL

Operators in SQL are symbols that help us to perform specific mathematical and logical computations on operands. An operator can either be unary or binary.

The unary operator operates on one operand, and the binary operator operates on two operands.

Types of Operators in SQL

- Arithmetic operator
- Comparison operator
- Logical operator
- Bitwise Operators
- Compound Operators

Arithmetic Operators

Arithmetic operators in SQL are used to perform mathematical operations on numeric values in queries. Some common arithmetic operators are:

Operator	Description
+	The addition is used to perform an addition operation on the data values.
–	This operator is used for the subtraction of the data values.
/	This operator works with the ‘ALL’ keyword and it calculates division operations.
*	This operator is used for multiplying data values.
%	Modulus is used to get the remainder when data is divided by another.

OPEN MYSQL

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| appwars            |
| information_schema |
| mlm                |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0.00 sec)

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

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| appwars            |
| arthmatic          |
| information_schema |
| mlm                |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
7 rows in set (0.00 sec)

mysql> use arthmatic;
Database changed
mysql> show tables;
Empty set (0.00 sec)
```

Step 1: Create the Table

```
mysql> CREATE TABLE Employees (
    ->     EmployeeID INT PRIMARY KEY,      -- Unique identifier for each employee
    ->     Salary DECIMAL(10, 2),           -- Salary with two decimal places
    ->     Bonus DECIMAL(10, 2),            -- Bonus with two decimal places
    ->     TotalCompensation DECIMAL(10, 2) -- Total compensation (Salary + Bonus)
    -> );
Query OK, 0 rows affected (0.03 sec)

mysql> show tables;
+--------------------+
| Tables_in_arthmatic |
+--------------------+
| employees          |
+--------------------+
1 row in set (0.00 sec)
```

```
mysql> show tables;
+---------------------+
| Tables_in_arthmatic |
+---------------------+
| employees           |
+---------------------+
1 row in set (0.00 sec)

mysql> describe employees;
+-------------------+---------------+------+-----+---------+-------+
| Field             | Type          | Null | Key | Default | Extra |
+-------------------+---------------+------+-----+---------+-------+
| EmployeeID        | int           | NO   | PRI | NULL    |       |
| Salary            | decimal(10,2) | YES  |     | NULL    |       |
| Bonus             | decimal(10,2) | YES  |     | NULL    |       |
| TotalCompensation | decimal(10,2) | YES  |     | NULL    |       |
+-------------------+---------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

Note- select * from employees;

Step 2: Insert Data into the Table;

```
1 row in set (0.00 sec)

mysql> INSERT INTO Employees (EmployeeID, Salary, Bonus, TotalCompensation)
    -> VALUES
    -> (1, 50000.00, 5000.00, 50000.00 + 5000.00),
    -> (2, 60000.00, 6000.00, 60000.00 + 6000.00),
    -> (3, 55000.00, 5500.00, 55000.00 + 5500.00);
Query OK, 3 rows affected (0.01 sec)
Records: 3  Duplicates: 0  Warnings: 0

mysql> select * from employees;
+------------+----------+---------+------------------+
| EmployeeID | Salary   | Bonus   | TotalCompensation |
+------------+----------+---------+------------------+
|          1 | 50000.00 | 5000.00 |          55000.00 |
|          2 | 60000.00 | 6000.00 |          66000.00 |
|          3 | 55000.00 | 5500.00 |          60500.00 |
+------------+----------+---------+------------------+
3 rows in set (0.00 sec)
```

Insert Data Without Calculating TotalCompensation:

```
mysql> INSERT INTO Employees (EmployeeID, Salary, Bonus, TotalCompensation)
    -> VALUES
    -> (4, 70000.00, 7000.00, 0),
    -> (5, 80000.00, 8000.00, 0);
Query OK, 2 rows affected (0.00 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> select * from employees;
+------------+----------+---------+------------------+
| EmployeeID | Salary   | Bonus   | TotalCompensation |
+------------+----------+---------+------------------+
|          1 | 50000.00 | 5000.00 |          55000.00 |
|          2 | 60000.00 | 6000.00 |          66000.00 |
|          3 | 55000.00 | 5500.00 |          60500.00 |
|          4 | 70000.00 | 7000.00 |              0.00 |
|          5 | 80000.00 | 8000.00 |              0.00 |
+------------+----------+---------+------------------+
5 rows in set (0.00 sec)
```

Update the TotalCompensation Using Arithmetic:

```
mysql> UPDATE Employees
    -> SET TotalCompensation = Salary + Bonus
    -> WHERE TotalCompensation = 0;
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2  Changed: 2  Warnings: 0

mysql> select * from employees;
+------------+----------+---------+-------------------+
| EmployeeID | Salary   | Bonus   | TotalCompensation |
+------------+----------+---------+-------------------+
|          1 | 50000.00 | 5000.00 |          55000.00 |
|          2 | 60000.00 | 6000.00 |          66000.00 |
|          3 | 55000.00 | 5500.00 |          60500.00 |
|          4 | 70000.00 | 7000.00 |          77000.00 |
|          5 | 80000.00 | 8000.00 |          88000.00 |
+------------+----------+---------+-------------------+
5 rows in set (0.00 sec)
```