**SQL Operators**

What is an Operator in SQL?

An operator is a reserved word or a character used primarily in an SQL statement's WHERE clause to perform operation(s), such as comparisons and arithmetic operations.

Operators are used to specify conditions in an SQL statement and to serve as conjunctions for multiple conditions in a statement.

 Arithmetic operators

 Comparison operators

 Logical operators

 Operators used to negate conditions

mysql> select 10+20;

+-------+

| 10+20 |

+-------+

| 30 |

+-------+

1 row in set (0.00 sec)

mysql> select 10\*20;

+-------+

| 10\*20 |

+-------+

| 200 |

+-------+

1 row in set (0.00 sec)

mysql> select 10/5;

+--------+

| 10/5 |

+--------+

| 2.0000 |

+--------+

1 row in set (0.00 sec)

mysql> select 12%5;

+------+

| 12%5 |

+------+

| 2 |

+------+

1 row in set (0.00 sec)

--------------------------------------------------------------------------------------------------------------------------------------

**SQL Comparison Operators:**

|  |  |  |
| --- | --- | --- |
| **Operator** | **Description** | **Example** |
| = | Checks if the values of two operands are equal or not, if yes then condition becomes true. | (a = b) is not true. |
| != | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (a != b) is true. |
| <> | Checks if the values of two operands are equal or not, if values are not equal then condition becomes true. | (a <> b) is true. |
| > | Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true. | (a > b) is not true. |
| < | Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true. | (a < b) is true. |
| >= | Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true. | (a >= b) is not true. |

Assume variable a holds 10 and variable b holds 20, then:

Consider the CUSTOMERS table having the following records:

mysql> insert into customers(id,name,age,address,salary)

-> values(1,'ramesh',32,'ahmedabad',2000.00),

-> (2,'khilan',25,'Delhi',1500.00),

-> (3,'kaushik',23,'Kota',2000.00),

-> (4,'Chaitali',25,'Mumbai',6500.00),

-> (5,'Hardik',27,'Bhopal',8500.00),

-> (6,'Komal',22,'MP',4500.00),

-> (7,'Muffy',24,'Indore',10000.00);

Query OK, 7 rows affected (0.00 sec)

SQL> SELECT \* FROM CUSTOMERS;

+----+----------+-----+-----------+----------+

| ID | NAME | AGE | ADDRESS | SALARY |

+----+----------+-----+-----------+----------+

| 1 | Ramesh | 32 | Ahmedabad | 2000.00 |

| 2 | Khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+-----+-----------+----------+

7 rows in set (0.00 sec)

Here are simple examples showing usage of SQL Comparison Operators:

mysql> select \* from customers where salary >5000;

+----+----------+------+---------+----------+

| id | name | age | address | salary |

+----+----------+------+---------+----------+

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+---------+----------+

3 rows in set (0.00 sec)

mysql> select \* from customers where salary =2000;

+----+---------+------+-----------+---------+

| id | name | age | address | salary |

+----+---------+------+-----------+---------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

+----+---------+------+-----------+---------+

2 rows in set (0.00 sec)

mysql> select \* from customers where salary !=2000;

+----+----------+------+---------+----------+

| id | name | age | address | salary |

+----+----------+------+---------+----------+

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+---------+----------+

5 rows in set (0.00 sec)

mysql> select \* from customers where salary <>2000;

+----+----------+------+---------+----------+

| id | name | age | address | salary |

+----+----------+------+---------+----------+

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+---------+----------+

5 rows in set (0.00 sec)

mysql> select \* from customers where salary >=6500;

+----+----------+------+---------+----------+

| id | name | age | address | salary |

+----+----------+------+---------+----------+

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+---------+----------+

3 rows in set (0.00 sec)

**--------------------------------------------------------------------------------------------------------------------------------------**

**SQL Logical Operators:**

Here is a list of all the logical operators available in SQL.

|  |  |
| --- | --- |
| **Operator** | **Description** |
| ALL | The ALL operator is used to compare a value to all values in another value set. |
| AND | The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause. |
| ANY | The ANY operator is used to compare a value to any applicable value in the list according to the condition. |
| BETWEEN | The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value. |
| EXISTS | The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria. |
| IN | The IN operator is used to compare a value to a list of literal values that have been specified. |
| LIKE | The LIKE operator is used to compare a value to similar values using wildcard operators. |
| NOT | The NOT operator reverses the meaning of the logical operator with which it is used. Eg: NOT EXISTS, NOT BETWEEN, NOT IN, etc. **This is a negate operator.** |
| OR | The OR operator is used to combine multiple conditions in an SQL statement's WHERE clause. |
| IS NULL | The NULL operator is used to compare a value with a NULL value. |
| UNIQUE | The UNIQUE operator searches every row of a specified table for uniqueness (no duplicates). |

System cls; this is used for clean the screen

mysql> select \* from customers;

+----+----------+------+-----------+----------+

| id | name | age | address | salary |

+----+----------+------+-----------+----------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 2 | khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+-----------+----------+

7 rows in set (0.00 sec)

Here are simple examples showing usage of SQL Comparison Operators:

mysql> select \* from customers where age >=25 and salary>=6500;

+----+----------+------+---------+---------+

| id | name | age | address | salary |

+----+----------+------+---------+---------+

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

+----+----------+------+---------+---------+

2 rows in set (0.00 sec)

mysql> select \* from customers where age >=25 or salary>=6500;

+----+----------+------+-----------+----------+

| id | name | age | address | salary |

+----+----------+------+-----------+----------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+-----------+----------+

5 rows in set (0.00 sec)

mysql> select \* from customers where age is not null;

+----+----------+------+-----------+----------+

| id | name | age | address | salary |

+----+----------+------+-----------+----------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 2 | khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+-----------+----------+

7 rows in set (0.00 sec)

mysql> select \* from customers where name like 'Ko%';

+----+-------+------+---------+---------+

| id | name | age | address | salary |

+----+-------+------+---------+---------+

| 6 | Komal | 22 | MP | 4500.00 |

+----+-------+------+---------+---------+

1 row in set (0.00 sec)

mysql> select \* from customers where age in (25,27);

+----+----------+------+---------+---------+

| id | name | age | address | salary |

+----+----------+------+---------+---------+

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

+----+----------+------+---------+---------+

3 rows in set (0.00 sec)

mysql> select \* from customers where age between 25 and 27;

+----+----------+------+---------+---------+

| id | name | age | address | salary |

+----+----------+------+---------+---------+

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

+----+----------+------+---------+---------+

3 rows in set (0.00 sec)

mysql> select age from customers

-> where exists (select age from customers where salary>6500);

+------+

| age |

+------+

| 32 |

| 25 |

| 23 |

| 25 |

| 27 |

| 22 |

| 24 |

+------+

7 rows in set (0.00 sec)

mysql> select \* from customers

-> where age>all(select age from customers where salary>6500);

+----+--------+------+-----------+---------+

| id | name | age | address | salary |

+----+--------+------+-----------+---------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

+----+--------+------+-----------+---------+

1 row in set (0.00 sec)

mysql> select \* from customers

-> where age>any(select age from customers where salary>6500);

+----+----------+------+-----------+---------+

| id | name | age | address | salary |

+----+----------+------+-----------+---------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 2 | khilan | 25 | Delhi | 1500.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

+----+----------+------+-----------+---------+

4 rows in set (0.00 sec)

**Sql Expressions**

An expression is a combination of one or more values, operators, and SQL functions that evaluate to a value.

SQL EXPRESSIONs are like formulas and they are written in query language. You can also use them to query the database for specific set of data.

Syntax:

Consider the basic syntax of the SELECT statement as follows:

SELECT column1, column2, columnN

FROM table\_name

WHERE [CONDITION|EXPRESSION];

There are different types of SQL expressions, which are mentioned below:

**SQL -Boolean Expressions:**

SQL Boolean Expressions fetch the data on the basis of matching single value. Following is the syntax:

SELECT column1, column2, columnN

FROM table\_name

WHERE SINGLE VALUE MATCHTING EXPRESSION;

mysql> select \* from customers;

+----+----------+------+-----------+----------+

| id | name | age | address | salary |

+----+----------+------+-----------+----------+

| 1 | ramesh | 32 | ahmedabad | 2000.00 |

| 2 | khilan | 25 | Delhi | 1500.00 |

| 3 | kaushik | 23 | Kota | 2000.00 |

| 4 | Chaitali | 25 | Mumbai | 6500.00 |

| 5 | Hardik | 27 | Bhopal | 8500.00 |

| 6 | Komal | 22 | MP | 4500.00 |

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+----------+------+-----------+----------+

7 rows in set (0.00 sec)

Boolean expression

mysql> select \* from customers where salary=10000;

+----+-------+------+---------+----------+

| id | name | age | address | salary |

+----+-------+------+---------+----------+

| 7 | Muffy | 24 | Indore | 10000.00 |

+----+-------+------+---------+----------+

1 row in set (0.00 sec)

SQL -Numeric Expression:

This expression is used to perform any mathematical operation in any query. Following is the syntax:

SELECT numerical\_expression as OPERATION\_NAME

[FROM table\_name

WHERE CONDITION] ;

mysql> select(15+6) as Addition;

+----------+

| Addition |

+----------+

| 21 |

+----------+

1 row in set (0.00 sec)

There are several built-in functions like avg(), sum(), count(), etc., to perform what is known as aggregate data calculations against a table or a specific table column.

mysql> select count(\*) as Records from customers;

+---------+

| Records |

+---------+

| 7 |

+---------+

1 row in set (0.00 sec)

SQL -Date Expressions:

Date Expressions return current system date and time values:

mysql> SELECT CURRENT\_TIMESTAMP();

+---------------------+

| CURRENT\_TIMESTAMP() |

+---------------------+

| 2025-05-28 01:02:15 |

+---------------------+

1 row in set (0.00 sec)

**OR we can use**

mysql> select now();

+---------------------+

| now() |

+---------------------+

| 2025-05-28 01:03:01 |

+---------------------+

1 row in set (0.00 sec)

Another date expression is as follows:

mysql> SELECT CURDATE();

+------------+

| CURDATE() |

+------------+

| 2025-05-28 |

+------------+

1 row in set (0.00 sec)

Below was not worked for me :-

SQL> SELECT GETDATE();;

+-------------------------+

| GETDATE |

+-------------------------+

| 2009-10-22 12:07:18.140 |

+-------------------------+

1 row in set (0.00 sec)

mysql> create database testDB;

Query OK, 1 row affected (0.01 sec)

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| information\_schema |

| mysql |

| nit |

| performance\_schema |

| sys |

| testdb |

+--------------------+

6 rows in set (0.00 sec)