Introduction of SQL Operators

SQL
Operators
1 Arithmetic Operators
2 Comparison Operators
3 Logical Operators
4 Compound Operators
5 Bitwise Operators
6 Unary Operator

SQL Operators are fundamental components of structured query language (SQL) that enable users to perform various operations on data within relational databases. These operators act as powerful tools for querying, filtering, and manipulating data. They encompass a wide range of functionalities, including:

1. Arithmetic Operations

2. Comparison Operations

3. Logical Operations

4. Membership Operations

5. Bitwise Operations

Arithmetic Operators in SQL:

In SQL, arithmetic operators are symbols that perform mathematical calculations on numeric data, including addition (+), subtraction (-), multiplication (*), and division(/).

1. Addition (+): Used to add two numbers together.

Example: SELECT 10 + 5; (Result: 15)

2. Subtraction (-): Used to subtract one number from another.

Example: SELECT 10 - 5; (Result: 5)

3. Multiplication (*): Used to multiply two numbers together.

Example: SELECT 10 * 5; (Result: 50)

4. Division (/): Used to divide one number by another.

Example: SELECT 10 / 5; (Result: 2)

5. Modulo (%): Returns the remainder of a division operation.

Example: SELECT 10 % 3; (Result: 1)

6. Exponentiation (^): Raises one number to the power of another (not standard in all DBMS)

Example: SELECT 5 ^ 2; (Result: 25)

Comparision Operator:

SQL Comparison Operators are used to compare two values and check if they meet the specific criteria. Some comparison operators are = Equal to, > Greater than, < Less than, etc.

1. (=) Equal: The SQL Equal Operator checks if the values of two operands are equal.

Example: SELECT * FROM MARKSHEET WHERE ROOLL_NUMBER = 1;

2. (<>) Not Equal: The SQL Not Equal Operator checks if the values of two operands are not equal.

Example: SELECT * FROM MARKSHEET WHERE AGE <>25;

3. (>=) Greater than equal: The SQL Greater Than Equals to Operator checks if the value of the left operand is greater than or equal to the value of the right operand.

Example: SELECT * FROM MARKSHEET WHERE MARKS >=400;

4. (>) Greater than: The SQL Greater Than Operator checks if the value of the left operand is greater than the value of the right operand.

Example: SELECT * FROM MARKSHEET WHERE MARKS > 350;

5. (<=) Less than equal: The SQL Less Than Equals to Operator checks if the value of the left operand is less than or equal to the value of the right operand.

Example: SELECT * FROM MARKSHEET WHERE MARKS <= 250;

6. (<) Less than: The SQL Less Than Operator checks if the value of the left operand is less than the value of the right operand.

Example: SELECT * FROM MARKSHEET WHERE MARKS < 200;

Logical Operator:

Logical operators in SQL are used to combine multiple conditions in a query to control the flow of execution. They evaluate whether these conditions are TRUE, FALSE, or NULL, assisting in refining query results effectively.

AND:

The AND operator is used to combine two or more conditions in an SQL query. It returns records only when all conditions specified in the query are true.

Example: SELECT * FROM EMPLOYEE WHERE EMP_CITY = 'ALLAHABAD' AND EMP_COUNTRY = 'INDIA';

OR: The OR operator combines multiple conditions in a SQL query and returns TRUE if at least one of the conditions is satisfied.

Example: SELECT * FROM EMPLOYEE WHERE EMP_CITY = 'VARANASI' OR EMP_COUNTRY = 'INDIA';

BETWEEN: The BETWEEN operator in SQL allows us to test if a value or expression lies within a specified range. The BETWEEN condition is inclusive, meaning it includes both the lower and in the results. This operator is particularly useful when we need to filter records based on a range of values, such as numerical ranges, dates, or even text values.

Example: SELECT * FROM EMPLOYEE WHERE EMP_ID BETWEEN 101 AND 104;

Membership Operator:

Membership operators in SQL database are used to check if a value exists within a specified set of values. They allow for efficient data filtering and categorization, making SQL queries more concise and effective.

1) IN operator: The IN operator checks if a value matches any of the values in a provided list, array, or subquery.

Example: SELECT * FROM MARKSHEET WHERE ADDRESS IN (“DELHI”, “BIHAR”, “UP”, “HARAYANA”);

2) NOT IN operator: The NOT IN operator verifies if a value does not match any of the values in the specified list or subquery.

Example: SELECT * FROM MARKSHEET WHERE ADDRESS NOT IN (“DELHI”, “BIHAR”, “UP”, “HARAYANA”);

Bitwise Operator:

Bitwise operators carry out operations on the individual bits of values.

1) Bitwise AND operator (&): The bitwise AND operator performs a bitwise AND operation between two values.

2) Bitwise OR operator (|): The bitwise OR operator performs a bitwise OR operation between two values.

3) Bitwise XOR operator (^): The bitwise XOR operator performs a bitwise exclusive OR operation between two values.

4) Bitwise NOT operator (~): The bitwise NOT operator performs a bitwise negation of a value.

5) Left shift operator (<<): The left shift operator shifts the bits of a value to the left.

6) Right shift operator (>>): The right shift operator shifts the bits of a value to the right.