

SQL OPERATORS

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

In this article, we will discuss **Operators** in SQL with examples, and understand how they work in SQL.

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

Different types of operators in SQL are:

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

SQL 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.

SQL Arithmetic Operators Example

In this example, we will retrieve all records from the "employee" table where the "emp_city" column does not start with the letter 'A'.

Query:

```
SELECT * FROM employee WHERE emp_city NOT LIKE 'A%';
```

Output:

	emp_id	emp_name	emp_city	emp_country
1	101	Utkarsh Tripathi	Varanasi	India
2	102	Abhinav Singh	Varanasi	India
3	103	Utkarsh Raghuvanshi	Varanasi	India
4	106	Ashutosh Kumar	Patna	India

SQL Comparison Operators

Comparison Operators in SQL are used to compare one expression's value to other expressions. SQL supports different types of comparison operator, which are described below:

Operator	Description
>	Greater than.
<	Less than.
>=	Greater than equal to.
<=	Less than equal to.
<>	Not equal to.

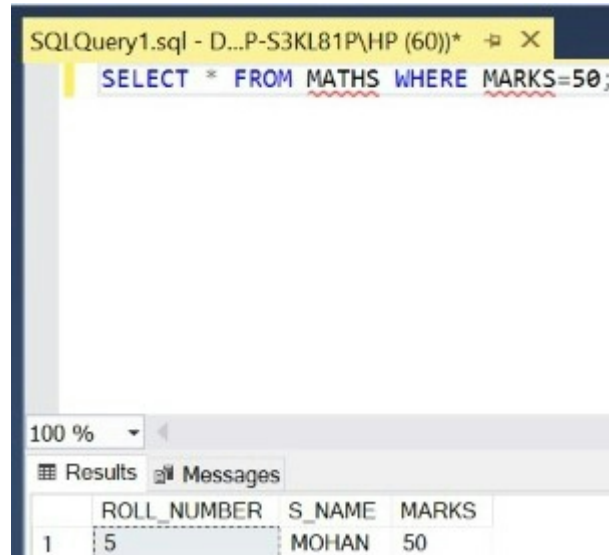
SQL Comparison Operators Example

In this example, we will retrieve all records from the “MATHS” table where the value in the “MARKS” column is equal to 50.

Query:

```
SELECT * FROM MATHS WHERE MARKS=50;
```

Output:



SQL Logical Operators

Logical Operators in SQL are used to combine or manipulate conditions in SQL queries to retrieve or manipulate data based on specified criteria..

Operator	Description
AND	Logical AND compares two Booleans as expressions and returns true when both expressions are true.
OR	Logical OR compares two Booleans as expressions and returns true when one of the expressions is true.
NOT	Not takes a single Boolean as an argument and change its value from false to true or from true to false.

SQL Logical Operators Example

In this example, retrieve all records from the “employee” table where the “emp_city” column is equal to ‘Allahabad’ and the “emp_country” column is equal to ‘India’.

Query:

```
SELECT * FROM employee WHERE emp_city =  
'Allahabad' AND emp_country = 'India';
```

Output:

Results		Messages		
	emp_id	emp_name	emp_city	emp_country
1	104	Utkarsh Singh	Allahabad	India
2	105	Sudhanshu Yadav	Allahabad	India

SQL Logical Operators

Bitwise operators in SQL are used to perform bitwise operations on binary values in SQL queries, manipulating individual bits to perform logical operations at the bit level. Some SQL Bitwise Operators are:

Operator	Description
&	Bitwise AND operator.
	Bitwise OR operator.
^	Bitwise XOR (exclusive OR) operator.
~	Bitwise NOT (complement) operator.
<<	Left shift operator.
>>	Right shift operator.

SQL Compound Operators

Compound operator in SQL are used to perform an operation and assign the result to the original value in a single line. Some Compound operators are:

Operator	Description
+=	Add and assign.
-=	Subtract and assign.
*=	Multiply and assign.
/=	Divide and assign.
%=	Modulo and assign.
&=	Bitwise AND and assign.
^=	Bitwise XOR and assign.
=	Bitwise OR and assign.

SQL Special Operators

Special operators are used in SQL queries to perform specific operations like comparing values, checking for existence, and filtering data based on certain conditions.

Operator	Description
ALL	ALL is used to select all records of a SELECT STATEMENT. It compares a value to every value in a list of results from a query. The ALL must be preceded by the comparison operators and evaluated to TRUE if the query returns no rows.
ANY	ANY compares a value to each value in a list of results from a query and evaluates to true if the result of an inner query contains at least one row.
BETWEEN	The SQL BETWEEN operator tests an expression against a range. The range consists of a beginning, followed by an AND keyword and an end expression.
IN	The IN operator checks a value within a set of values separated by commas and retrieves the rows from the table that match.
EXISTS	The EXISTS checks the existence of a result of a subquery. The EXISTS subquery tests whether a subquery fetches at least one row. When no data is returned then this operator returns 'FALSE'.
SOME	SOME operator evaluates the condition between the outer and inner tables and evaluates to true if the final result returns any one row. If not, then it evaluates to false.
UNIQUE	The UNIQUE operator searches every unique row of a specified table.

SQL Special Operator Example

In this example, we will retrieve all records from the “employee” table where the “emp_id” column has a value that falls within the range of 101 to 104 (inclusive).

Output:

```
SELECT * FROM employee WHERE emp_id BETWEEN 101 AND 104;
```

Query:

	emp_id	emp_name	emp_city	emp_country
1	101	Utkarsh Tripathi	Varanasi	India
2	102	Abhinav Singh	Varanasi	India
3	103	Utkarsh Raghuvanshi	Varanasi	India
4	104	Utkarsh Singh	Allahabad	India

Conclusion

SQL Operators are used to perform various operations on the data using SQL queries. These operators simplify arithmetic, comparison, logical, and bitwise operations on the data.

In this tutorial, we have explained SQL operators in detail. We have explained different types of operators in SQL along with their definition and examples. Using SQL operators you can efficiently perform operations on data.