Types of SQL operators

- Arithmetic
- Comparison
- Logical
- Set

Arithmetic

Performs math operation on numerical data, we can find addition, subtraction, multiplication, division and modulus.

Comparison

Compares two different data returning a boolean value (TRUE or FALSE), checking if equal, greater or lesser.

Logical

Creates conditional expressions that returns a boolean value (TRUE or FALSE).
We can find ALL, AND, ANY, BETWEEN, EXISTS, IN, LIKE, NOT, OR, IS NULL.

Set

Combines similar type of data from tables mixing the result of queries and returning a single result.
Its operators are UNION, UNION ALL, MINUS, INTERSECT.

</>
MG
MANUEL GENTILE

Comparison Operators

EQUAL

Checks if the values of two operands are equal or not, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 = OPERAND2;
```

Comparison Operators

EQUAL EXAMPLE

```
SELECT * FROM employee WHERE 1 = 1;
```

Simple condition, returns TRUE

```
SELECT * FROM employee
WHERE last_name = ‘Gentile’;
```

Condition between a column and a value

```
SELECT * FROM employee
WHERE last_name = first_name;
```

Condition between two columns of a table

Comparison Operators

NOT EQUAL

Checks if the values of two operands are equal or not, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 != OPERAND2;
.. WHERE OPERAND1 <> OPERAND2;
```

You can use the two symbols interchangeably

Comparison Operators

NOT EQUAL EXAMPLE

```
SELECT * FROM employee WHERE 1 <> 1;
```

Simple condition, returns FALSE

```
SELECT * FROM employee
WHERE last_name != ‘Gentile’;
```

Condition between a column and a value

```
SELECT * FROM employee
WHERE last_name <> first_name;
```

Condition between two columns of a table

Comparison Operators

GREATER

Checks if the operand on the left is greater than the operand on the right, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 > OPERAND2;
```

Comparison Operators

GREATER EXAMPLE

```
SELECT * FROM employee WHERE 1 > 2;
```

Simple condition, returns FALSE

```
SELECT * FROM employee
WHERE age > 21;
```

Condition between a column and a value

```
SELECT * FROM products
WHERE max_value > min_value;
```

Condition between two columns of a table

Comparison Operators

LESS

Checks if the operand on the left is less than the operand on the right, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 < OPERAND2;
```

Comparison Operators

LESS EXAMPLE

```
SELECT * FROM employee WHERE 1 < 2;
```

Simple condition, returns TRUE

```
SELECT * FROM employee
WHERE age < 18;
```

Condition between a column and a value

```
SELECT * FROM products
WHERE max_value < min_value;
```

Condition between two columns of a table

Comparison Operators

GREATER OR EQUAL

Checks if the operand on the left is greater than or equal to the operand on the right, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 >= OPERAND2;
```

Comparison Operators

GREATER OR EQUAL EXAMPLE

```
SELECT * FROM employee WHERE 1 >= 2;
```

Simple condition, returns FALSE

```
SELECT * FROM employee
WHERE age >= 21;
```

Condition between a column and a value

```
SELECT * FROM products
WHERE max_value >= min_value;
```

Condition between two columns of a table

Comparison Operators

LESS OR EQUAL

Checks if the operand on the left is less than or equal to the operand on the right, returning a boolean (TRUE, FALSE).

```
.. WHERE OPERAND1 <= OPERAND2;
```

Comparison Operators

LESS EXAMPLE

```
SELECT * FROM employee WHERE 1 <= 2;
```

Simple condition, returns TRUE

```
SELECT * FROM employee
WHERE age <= 18;
```

Condition between a column and a value

```
SELECT * FROM products
WHERE max_value <= min_value;
```

Condition between two columns of a table

Let’s connect

If you want to learn more about the topic, connect or send me a DM.

in **Manuel Gentile**

MG
MANUEL GENTILE