

SQL Comparison operator examples

In this reading you'll learn more about SQL comparison operators. You've learned about comparison operators in SQL that are used to form conditions for filtering data. Comparison operators are useful when you want to write conditions in the WHERE clause of a SELECT statement to filter out records from a table. The main objective of this reading is to present some more examples of how those comparison operators can be used, along with more advanced scenarios.

Refreshing SQL comparison operators

As you've already learned, all the comparison operators that are used with other programming languages also work with SQL.

These comparison operators are:

Operator	What it does
=	Checks for equality
<> or !=	Checks for not inequality
>	Check if something is greater than
>=	Check if something is greater than or equal
<	Check if something is less than
<=	Check if something is less than or equal

You've seen examples of how to use these comparison operators already but let's examine a few more scenarios here.

Using the equality operator

You can use the = operator to test for equality in a query. It compares the equality of two expressions. The equal operator is used in the WHERE clause condition of a SELECT statement.

Let's consider this example table and its data.

employee_ID	employee_name	salary	hours	allowance	tax
1	alex	24000	10	1000	1000
2	John	55000	11	3000	2000
3	James	52000	7	3000	2000
4	Sam	24000	11	1000	1000

If you want to retrieve the data for the employee whose ID value is 1, you can use this SELECT statement.

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

This statement filters the record of the employee whose ID column has a value that is equal to 1. The result is as follows:

employee_ID	employee_name	salary	hours	allowance	tax
1	Alex	24000	10	1000	1000

That was an example of using the equality operator with a numeric data typed column. Now let's review an example of using the equality operator with a text-based data typed column, the employee_name.

In this example, you need to retrieve the data for the employee whose name is James. You can use the equal operator in the WHERE clause condition.

```
1  SELECT * FROM employee WHERE employee_name = 'James';
```

The important thing to note here is that the value or literal, in this case James, is surrounded by single quotation marks.

Using the inequality operator

The inequality operator does the opposite of what the equal operator does. It compares two non-null expressions and returns true if the value of the left expression is not equal to the right one. If not, it returns the value of false. There are two ways in SQL in which it can be used, <> or != and both methods result in the same outcome.

For example, let's say you want to determine which employee receives a salary that does not equate to 24000. You can use the following SQL statement:

```
1  SELECT *
2
3  FROM employee
4
5  WHERE salary <> 24000;
```

The expression here is salary <> 24000. This expression checks the inequality of the given value in the salary column and filters out only those records. The result here are as follows:

employee_ID	employee_name	salary	hours	allowance	tax
2	John	55000	11	3000	2000
3	James	52000	7	3000	2000

You can run the same query just by replacing <> with !=. It should behave in the same way and give the same result.

Using the greater than operator

This comparison operator compares two non-null expressions and returns true if the left operand is greater than the right operand. If not, the result is false. Let's say you want to find out which employees are earning a salary of over 50000. You would write the SQL SELECT query as follows using the greater than comparison operator.

```
1  SELECT *
2
3  FROM employee
4
5  WHERE salary > 50000;
```

The expression here is salary > 50000. This expression checks if the value of the salary column is greater than 50000. If so, it includes those employees in the results. Accordingly, the result is:

employee_ID	employee_name	salary	hours	allowance	tax
2	John	55000	11	3000	2000
3	James	52000	7	3000	2000

Using the greater than or equal operator

The greater than or equal operator (>=) compares two non-null expressions. The result is true if the left expression is a value that is greater than the value of the right expression. This time let's say you want to see who pays a tax amount of 2000 or more. This is the query that will give the desired result using the greater than or equal operator:

```
1  SELECT *
2
3  FROM employee
4
5  WHERE tax >= 1000;
```

The expression here is `tax >= 1000` and it checks if the tax column value is greater than or equal to 1000. If it finds any matching records, these are added to the result set.
The result in this case is as follows:

employee_ID	employee_name	salary	hours	allowance	tax
1	Alex	24000	10	1000	1000
2	John	55000	11	3000	2000
3	James	52000	7	3000	2000
4	Sam	24000	11	1000	1000

All 4 records are included in the result because the SQL query matches the tax column and picks the rows that have a value of 1000 or more.

Using the less than operator

The `<` operator in SQL can be used to test for an expression which is less than. That is, it compares two non-null expressions. The result is true if the left operand evaluates to a value that is lower than the value of the right operand. If not, the result is false.

For example, let's say you want to determine which employees get an allowance below 2500.

The following SQL query can be used to retrieve the result.

```
1  SELECT *
2
3  FROM employee
4
5  WHERE allowance < 2500;
```

The `allowance < 2500` expression checks the values of the allowance column to determine which ones have a value of less than 2500.

The result is as follows:

employee_ID	employee_name	salary	hours	allowance	tax
1	alex	24000	10	1000	1000
4	Sam	24000	11	1000	1000

Using the less than or equal operator

In SQL, the `<=` operator tests for an expression less than or equal to. That is, it compares two non-null expressions and returns true if the left expression has a value less than or equal to the value of the right expression. If not, it returns false. You'd use it, for example, if you want to determine which employees have worked for less than or equal to 10 hours. The following syntax is an example of how you would use the less than or equal operator in the WHERE clause of the SELECT statement:

```
1  SELECT *
2
3  FROM employee
4
5  WHERE hours <= 10;
```

The expression `hours <= 10` checks the values of the hours column to see if there are any records with a value of 10 less than that. If it finds one, it adds it to the result.

Accordingly, the result of this query is as follows:

employee_ID	employee_name	salary	hours	allowance	tax
1	Alex	24000	10	1000	1000
3	James	52000	7	3000	2000

In this reading you learned about SQL comparison operators and became more familiar with how to apply it in different scenarios.