BETWEEN to Filter Data

by Sophia

WHAT'S COVERED

In this lesson, you will compose a SELECT statement that uses BETWEEN to search for a range of numerical values in a data set. Specifically, this lesson will cover:

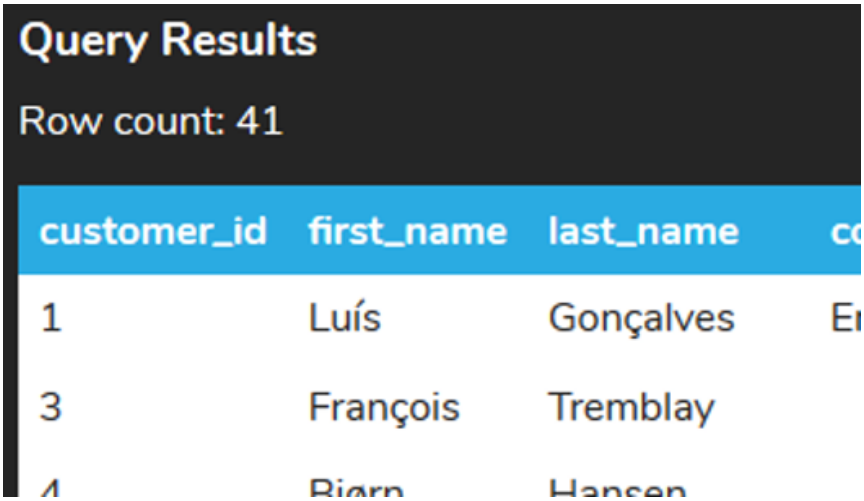
1. BETWEEN Operator

The BETWEEN operator enables us to check if an attribute is within a range of values. Like the game “Pick a number between 1 to 10,” you can use BETWEEN to find a range of values between X and Y in a column. The values defined in the BETWEEN operator include all the values BETWEEN what is being searched for, including the beginning and ending values. If a number is between 1 and 4, for example, then both 1 and 4 are included.

It is important to note that we always need to specify the smaller value first. For example, if we have the following statement:

```
SELECT *
FROM customer
WHERE support_rep_id BETWEEN 1 AND 4;
```

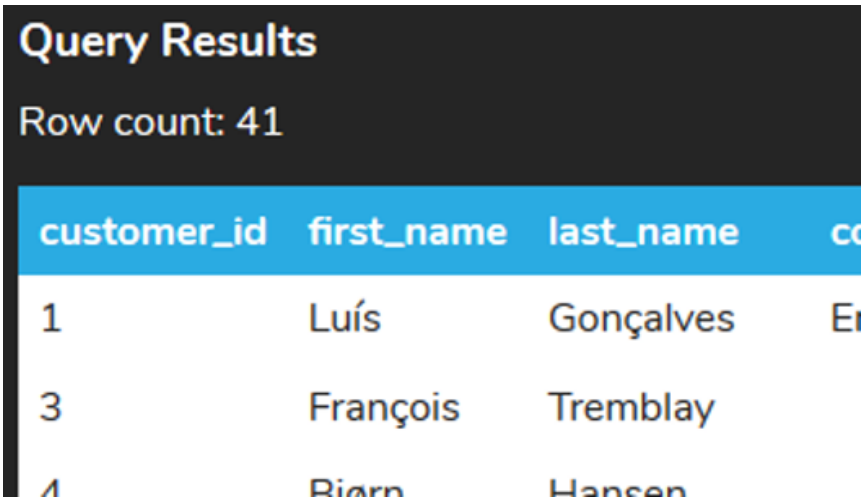
You should see a return result set with 41 rows.



This command can be interpreted as follows:

```
SELECT *
FROM customer
WHERE support_rep_id >= 1
AND support_rep_id <= 4;
```

This command would also return 41 rows.



However, if we have the larger number first:

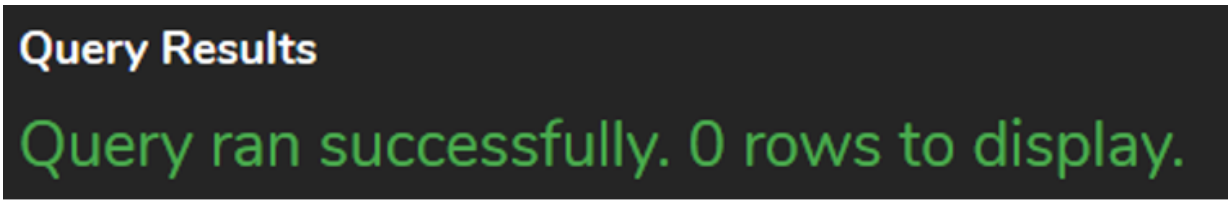
```
SELECT *
FROM customer
WHERE support_rep_id BETWEEN 4 AND 1
```

It would try to run it as the following:

```
SELECT *
FROM customer
```

```
WHERE support_rep_id >= 4
AND support_rep_id <= 1;
```

Of course, this would not work. The support_rep_id could not be greater than or equal to 4 at the same time as the support_rep_id is less than or equal to 1. Therefore, no rows could match the criteria, which is why we would get the following result:



2. Using BETWEEN on Dates

You can also use BETWEEN for dates when comparing a range of dates. For example, if we wanted to search for invoices that had the invoice_date in March 2009, we could do the following:

```
SELECT *
FROM invoice
WHERE invoice_date BETWEEN '2009-03-01' AND '2009-03-31';
```

Query Results

Row count: 7

invoice_id	customer_id	invoice_date	billing_address	billing_city	billing_state	billing_country	billing_postal_code	total
14	17	2009-03-04T00:00:00.000Z	1 Microsoft Way	Redmond	WA	USA	98052-8300	2
15	19	2009-03-04T00:00:00.000Z	1 Infinite Loop	Cupertino	CA	USA	95014	2
16	21	2009-03-05T00:00:00.000Z	801 W 4th Street	Reno	NV	USA	89503	4
17	25	2009-03-06T00:00:00.000Z	319 N. Frances Street	Madison	WI	USA	53703	6
18	31	2009-03-09T00:00:00.000Z	194A Chain Lake Drive	Halifax	NS	Canada	B3S 1C5	9
19	40	2009-03-14T00:00:00.000Z	8, Rue Hanovre	Paris		France	75002	14
20	54	2009-03-22T00:00:00.000Z	110 Raeburn Pl	Edinburgh		United Kingdom	EH4 1HH	1

3. Adding NOT

You can also use the NOT operator to return the opposite result set. Using the example from the previous lesson, if we wanted to get the tracks that had the genre_id NOT being between 10–20, we could write our query like this:

```
SELECT *
FROM track
WHERE genre_id NOT BETWEEN 10 AND 20;
```

This would include all tracks between 1–9 and 21–25 (along with any other genre_id that may be added to the table beyond 25).

Query Results

Row count: 3107

track_id	name	album_id	media_type_id	genre_id
1	For Those About To Rock (We Salute You)	1	1	1
2	Balls to the Wall	2	2	1
3	Fast As a Shark	3	2	1
4	Restless and Wild	3	2	1
5	Princess of the Dawn	3	2	1
6	Put The Finger On You	1	1	1
7	Let's Get It Up	1	1	1
8	Inject The Venom	1	1	1
9	Snowballed	1	1	1
10	Evil Walks	1	1	1
11	C.O.D.	1	1	1
12	Breaking The Rules	1	1	1
13	Night Of The Long Knives	1	1	1

We could do the same thing for dates to query invoices not between 2010-01-01 and 2010-12-31.

```
>SELECT *
FROM invoice
WHERE invoice_date NOT BETWEEN '2010-01-01' AND '2010-12-31';
```

This could also be written as:

```
SELECT *
FROM invoice
WHERE invoice_date < '2010-01-01'
OR invoice_date > '2010-12-31';
```

Notice that unlike the BETWEEN statement, this example excludes the values due to the NOT.

WATCH

TRY IT

Your turn! Open the SQL tool by clicking on the LAUNCH DATABASE button below. Then, enter in one of the examples above and see how it works. Next, try your own choices for which columns you want the query to provide.

SUMMARY

In this lesson, you learned that in PostgreSQL, the **BETWEEN operator** filters query results based on a range of values. You can check whether a given column value falls within that range by specifying a lower and upper bound, including both values. BETWEEN allows range-based conditions to be expressed in a concise and readable manner, avoiding the need for multiple comparison operators. You

also learned that an example of this is using **BETWEEN on dates**. A range of criteria can be used to filter records, such as dates, numerical values, or character strings, to simplify data querying. By using the BETWEEN operator, you will be able to extract data that falls within a defined range based on the lower and upper bounds. Finally, you learned that **adding the NOT operator** can help further isolate the data you are looking to display.

Source: THIS TUTORIAL WAS AUTHORED BY DR. VINCENT TRAN, PHD (2020) AND Faithe Wempen (2024) FOR SOPHIA LEARNING. PLEASE SEE OUR **TERMS OF USE**.