LIKE to Search Data

by Sophia

WHAT'S COVERED

In this lesson, you will learn that sometimes you might not know exactly what should be specified in a WHERE clause. SQL has several operators that enable you to select a range of values based on wildcard characters and text strings. Specifically, this lesson will cover:

1. The LIKE Operator

You are already familiar with one operator, the * wildcard, to match everything in the table. But other operators allow you much more control over what you search for. SQL has several native operators that can bring a lot more power and control to your SQL statements.

The LIKE operator in a WHERE clause enables you to use wildcards to check if an attribute's value matches a specific string pattern. Two wildcards can be used with the LIKE operator:

- The percent sign (%) represents zero or more characters. It can be used as a wildcard to find specific items in the database. For example, if you wanted to see everyone with a country of USA or US, you could do LIKE US%, which will match USA and US in the database.
- The underscore (_) represents a single character. The underscore can be used to find more specific information in the database. For example, if you wanted to find out everyone who spent 1000 dollars in the database, you could search for LIKE '1___' (that is three underscores), and it will only show records where the value is between 1000 and 1999.

These all do somewhat the same thing. We could search for LIKE '1000', LIKE '1___', or LIKE 1%, and all of these would return similar results. Which one of these would only return 1000? The LIKE '1000' because it is looking for an exact match.

KEY CONCEPT

In PostgreSQL, the default behavior of the LIKE clause is case-sensitive. That means that, for example, 'a%' and 'A%' are treated as different patterns. This is not the case in all SQL implementations, though.

If you want to perform a case-insensitive pattern match in PostgreSQL, you can use the ILIKE clause instead of LIKE.

When a record matches the WHERE clause's criteria, it is said to **return true** because it's true that there is a match. If a record contains the string 'abc', here are some example outcomes of various LIKE operators:

- ‘abc’ would return true, as the string matches exactly.
- ‘a%’ would return true, as it looks for the letter ‘a’ and zero or more characters displayed afterward. You would see everything that starts with a (note lowercase).
- ‘_b_’ would return true, as it looks for one character, the letter b, and one more character. You would see everything that had b as a second letter.
- ‘c’ would return false, as it only looks at the letter c. There are no single letter c’s in the table, so no data would be returned to you.
- ‘_b’ would return false, as it looks for one character and then the letter b. Underscore b would return no data, as there is no data that is anything with a second letter b only.
- ‘%c’ would return true, as it looks for characters that end with the letter c. This would return data that ends in c only.

Return True

To match the criteria required by a WHERE clause.

2. The % Operator

As shown above, the % searches for zero, one, or more characters as a replacement. It is the most common operator used with the LIKE operator. If we wanted to query our customer table and list all of the customers whose name starts with the letter L, we would use the % wildcard after the letter L:

```
SELECT *
FROM customer
WHERE first_name like 'L%';
```

Query Results

Row count: 5

customer_id	first_name	last_name	company	address	city	state	country
1	Luís	Gonçalves	Embraer - Empresa Brasileira de Aeronáutica S.A.	Av. Brigadeiro Faria Lima, 2170	São José dos Campos	SP	Brazil
2	Leonie	Köhler		Theodor-Heuss-Straße 34	Stuttgart		Germany
45	Ladislav	Kovács		Erzsébet krt. 58.	Budapest		Hungary
47	Lucas	Mancini		Via Degli Scipioni, 43	Rome	RM	Italy
57	Luis	Rojas		Calle Lira, 198	Santiago		Chile

If we wanted to list all of the customers who have their email in the domain gmail.com, we would have the % wildcard operator before @gmail.com:

```
SELECT *
FROM customer
WHERE email like '%@gmail.com';
```

Query Results

Row count: 8

customer_id	first_name	last_name	company	address	city	state	country	postal_code	phone	fax	email	support_rep_id
3	François	Tremblay		1498 rue Bélanger	Montréal	QC	Canada	H2G 1A7	+1 (514) 721-4711		ftremblay@gmail.com	3
6	Helena	Holý		Ilská 31/46	Prague		Czech Republic	14300	+420 2 4177 0449		hholy@gmail.com	5
22	Heather	Leacock		120 S Orange Ave	Orlando	FL	USA	32801	+1 (407) 999-7788		hleacock@gmail.com	4
24	Frank	Ralston		162 E Superior Street	Chicago	IL	USA	60611	+1 (312) 332-3232		fralston@gmail.com	3
28	Julia	Barnett		302 S 700 E	Salt Lake City	UT	USA	84102	+1 (801) 531-7272		jubarnett@gmail.com	5
31	Martha	Silk		194A Chain Lake Drive	Halifax	NS	Canada	B3S 1C5	+1 (902) 450-0450		marthasilk@gmail.com	5
40	Dominique	Lefebvre		8, Rue Hanovre	Paris		France	75002	+33 01 47 42 71 71		dominiquelefebvre@gmail.com	4
53	Phil	Hughes		113 Lupus St	London		United Kingdom	SW1V 3EN	+44 020 7976 5722		phil.hughes@gmail.com	3

This is the most common way to use the % operator to search for content. The following lesson will dive into more complex ways to work with wildcards.

3. Comparison Operators

The underscore (_) operator matches a single wildcard character, unlike the % operator with zero or more. This allows us to query data to look for a specific data length. For example, if we wanted to look for customers that have the state starting with C and having two characters, we can do the following:

```
SELECT *
FROM customer
WHERE state LIKE 'C_';
```

Query Results

Row count: 3

customer_id	first_name	last_name	company	address	city	state	country
16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View	CA	USA
19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino	CA	USA
20	Dan	Miller		541 Del Medio Avenue	Mountain View	CA	USA

Another example is looking for customers that live in a country that has three characters and starts with the letter U. If we used the % operator, it would search for zero or more characters after U:

```
SELECT *
FROM customer
WHERE country LIKE 'U%';
```

This would return some extra rows of countries, like the United Kingdom:

52	Emma	Jones	202 Hoxton Street	London	United Kingdom
53	Phil	Hughes	113 Lupus St	London	United Kingdom
54	Steve	Murray	110 Raeburn Pl	Edinburgh	United Kingdom

If we changed the LIKE clause to use the two extra underscores to get three characters, it should look like:

```
SELECT *
FROM customer
WHERE country LIKE 'U__';
```

Query Results

Row count: 13

customer_id	first_name	last_name	company	address	city	state	country
16	Frank	Harris	Google Inc.	1600 Amphitheatre Parkway	Mountain View	CA	USA
17	Jack	Smith	Microsoft Corporation	1 Microsoft Way	Redmond	WA	USA
18	Michelle	Brooks		627 Broadway	New York	NY	USA
19	Tim	Goyer	Apple Inc.	1 Infinite Loop	Cupertino	CA	USA
20	Dan	Miller		541 Del Medio Avenue	Mountain View	CA	USA
21	Kathy	Chase		801 W 4th Street	Reno	NV	USA
22	Heather	Leacock		120 S Orange Ave	Orlando	FL	USA
23	John	Gordon		69 Salem Street	Boston	MA	USA
24	Frank	Ralston		162 E Superior Street	Chicago	IL	USA
25	Victor	Stevens		319 N. Frances Street	Madison	WI	USA
26	Richard	Cunningham		2211 W Berry Street	Fort Worth	TX	USA
27	Patrick	Gray		1033 N Park Ave	Tucson	AZ	USA
28	Julia	Barnett		302 S 700 E	Salt Lake City	UT	USA

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 the LIKE clause.

SUMMARY

In this lesson, you learned that **the LIKE operator** performs pattern matching on strings using the WHERE clause. With the LIKE operator, you can use wildcard characters to find patterns within a string column. The **percent sign (%) operator** represents sequences of characters, whereas the **comparison operator** underscore (_) represents single characters. You also learned that you are able to use these wildcard characters to create flexible and powerful queries that look for strings containing certain substrings or matching specific patterns. You can locate records using the LIKE operator when you perform text-based searches, allowing you to find partial matches within the data or specific patterns within it.

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

TERMS TO KNOW

Return True

To match the criteria required by a WHERE clause.