


## POSTGRESQL LIKE

Query Editor		Query History	Data Output	Explain	Messages
1	--use the PostgreSQL LIKE operator to match		first_name	last_name	
2	--the first name of the customer with a		character varying (45)	character varying (45)	
3	--string like this query:		1	Jennifer	Davis
4			2	Jennie	Terry
5	SELECT		3	Jenny	Castro
6	first_name,				
7	last_name				
8	FROM				
9	customer				
10	WHERE				
11	first_name LIKE 'Jen%';				

Query Editor		Query History	Data Output	Explain	Messages
1	--examples		?column?	?column?	?column?
2	SELECT		boolean	boolean	boolean
3	'foo' LIKE 'foo', -- true		1	true	true
4	'foo' LIKE 'f%', -- true				true
5	'foo' LIKE '_o_', -- true				false
6	'bar' LIKE 'b_'; -- false				

Query Editor		Query History	Data Output	Explain	Messages
1	--THE query returns customers whose		first_name	last_name	
2	--first name contains		character varying (45)	character varying (45)	
3	--er string like Jenifer, Kimberly, etc.		1	Albert	Crouse
4	SELECT		2	Alberto	Henning
5	first_name,		3	Alexander	Fennell
6	last_name		4	Amber	Dixon
7	FROM		5	Bernard	Colby
8	customer		6	Bernice	Willis
9	WHERE		7	Bertha	Ferguson
10	first_name LIKE '%er%'		8	Beverly	Brooks
11	ORDER BY		9	Catherine	Campbell
12	first_name;		10	Cheryl	Murphy


dvdrental/postgres@PostgreSQL ▾

Query Editor

Query History

```

1  --You can combine the percent ( %)
2  --with underscore ( _) to construct a pattern
3  --as the following example:
4
5  SELECT
6      first_name,
7      last_name
8  FROM
9      customer
10 WHERE
11     first_name LIKE '_her%'
12 ORDER BY
13     first_name;

```

Data Output

Explain

Messages

	first_name character varying (45)	last_name character varying (45)
1	Cheryl	Murphy
2	Sherri	Rhodes
3	Sherry	Marshall
4	Theresa	Watson

dvdrental/postgres@PostgreSQL

Query Editor
 Query History

1
 --uses the NOT LIKE operator to find customers

2
 --whose first names do not begin with Jen:

3
 SELECT

4
 first\_name,

5
 last\_name

6
 FROM

7
 customer

8
 WHERE

9
 first\_name NOT LIKE 'Jen%'

10
 ORDER BY

11
 first\_name

first\_name
 character varying (45)

last\_name
 character varying (45)

1
 Aaron
 Selby

2
 Adam
 Gooch

3
 Adrian
 Clary

4
 Agnes
 Bishop

5
 Alan
 Kahn

6
 Albert
 Crouse

7
 Alberto
 Henning

8
 Alex
 Gresham

9
 Alexander
 Fennell

10
 Alfred
 Coillac

dvdrental/postgres@PostgreSQL ▾

Query Editor

Query History

1

--PostgreSQL supports the ILIKE operator

2

--that works like the LIKE operator. In addition,

3

--'the ILIKE operator matches value

4

--case-insensitively. For example:

5

6

SELECT

7

first\_name,

8

last\_name

9

FROM

10

customer

11

WHERE

12

first\_name ILIKE 'BAR%';

Data Output

Explain

Messages

	first_name character varying (45)	last_name character varying (45)
1	Barbara	Jones
2	Barry	Lovelace