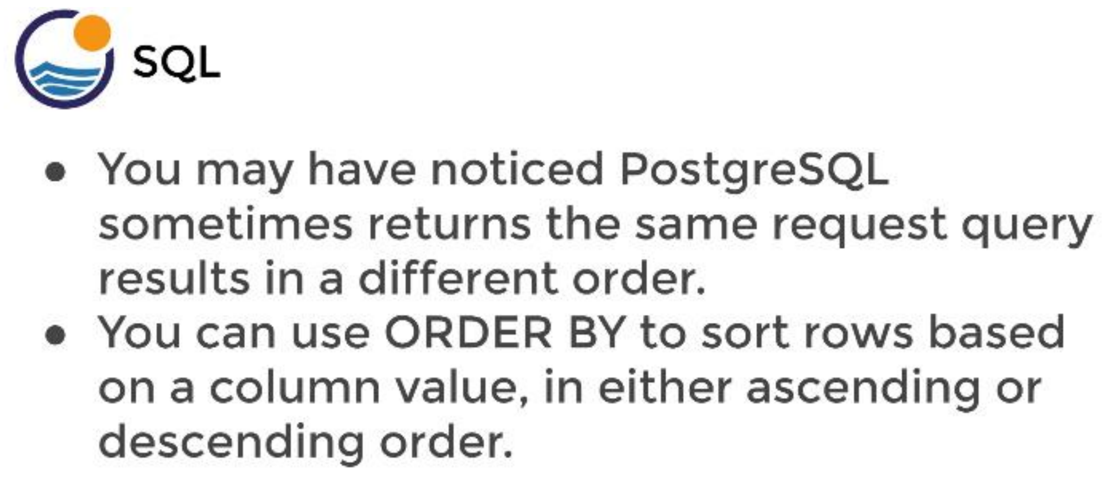
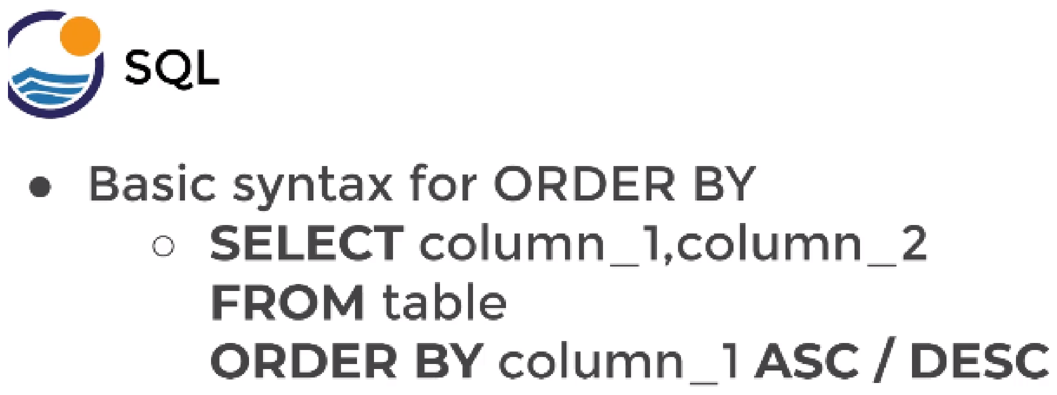
ORDER BY:



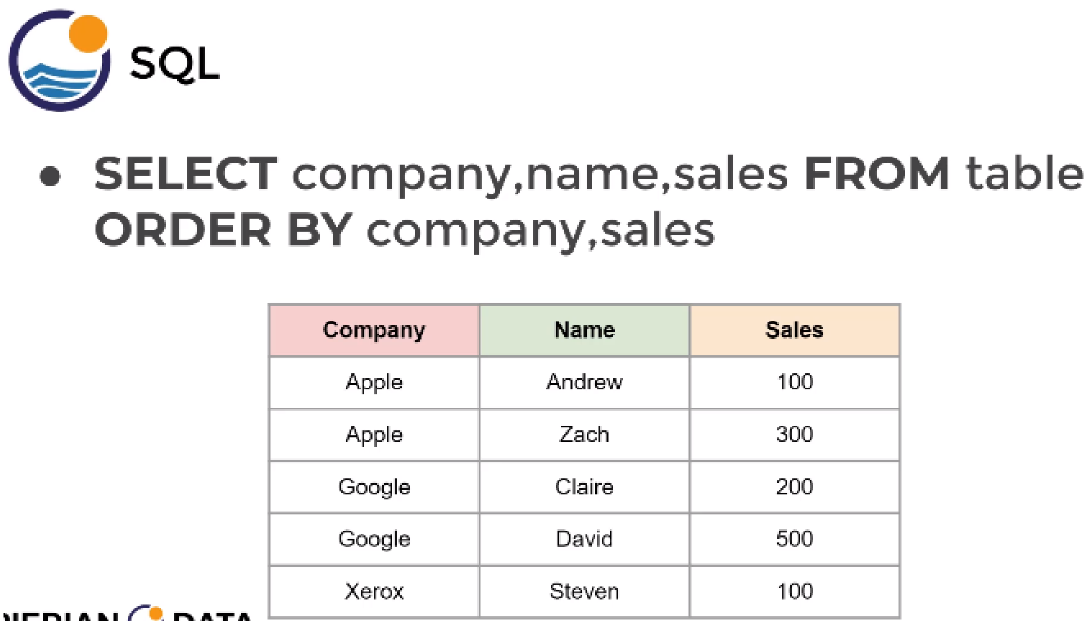
SQL

- You may have noticed PostgreSQL sometimes returns the same request query results in a different order.
- You can use ORDER BY to sort rows based on a column value, in either ascending or descending order.



SQL

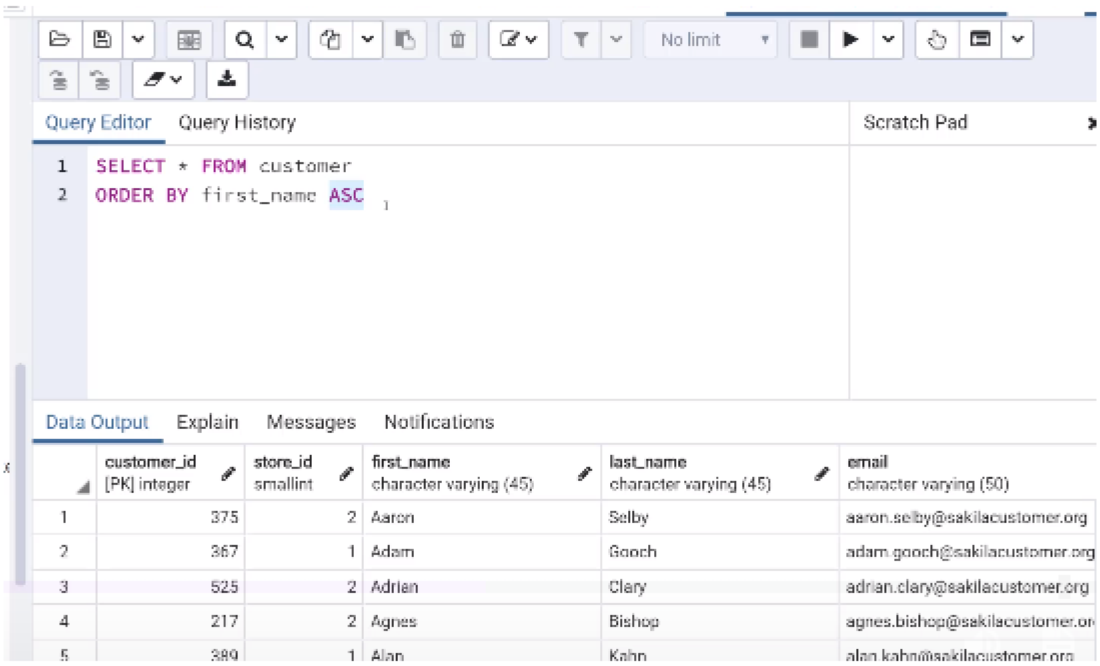
- Basic syntax for ORDER BY
  - **SELECT** column_1,column_2
    FROM table
    ORDER BY column_1 **ASC / DESC**



SQL

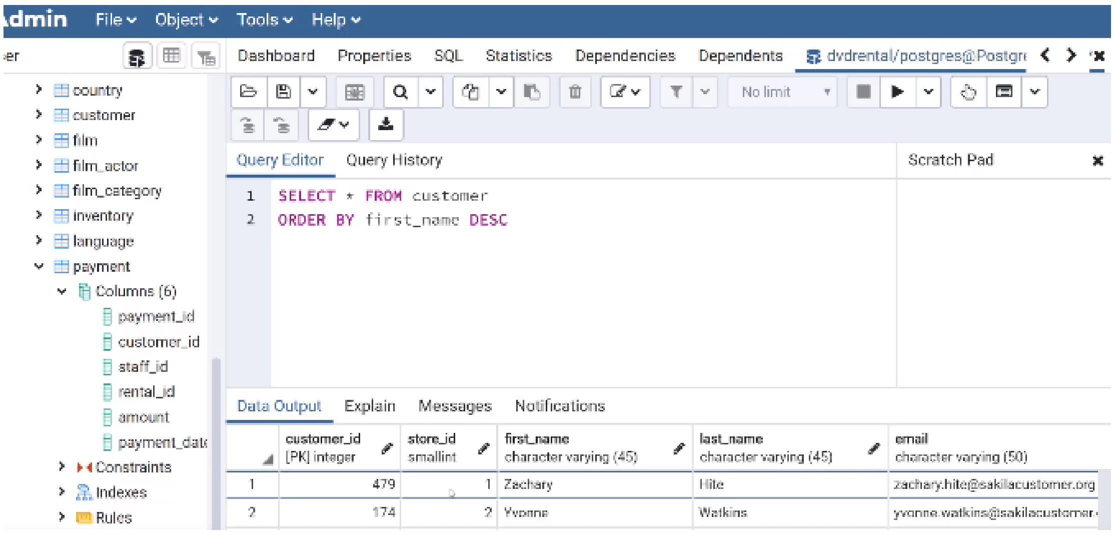
- **SELECT** company,name,sales **FROM** table
  ORDER BY company,sales

Company	Name	Sales
Apple	Andrew	100
Apple	Zach	300
Google	Claire	200
Google	David	500
Xerox	Steven	100



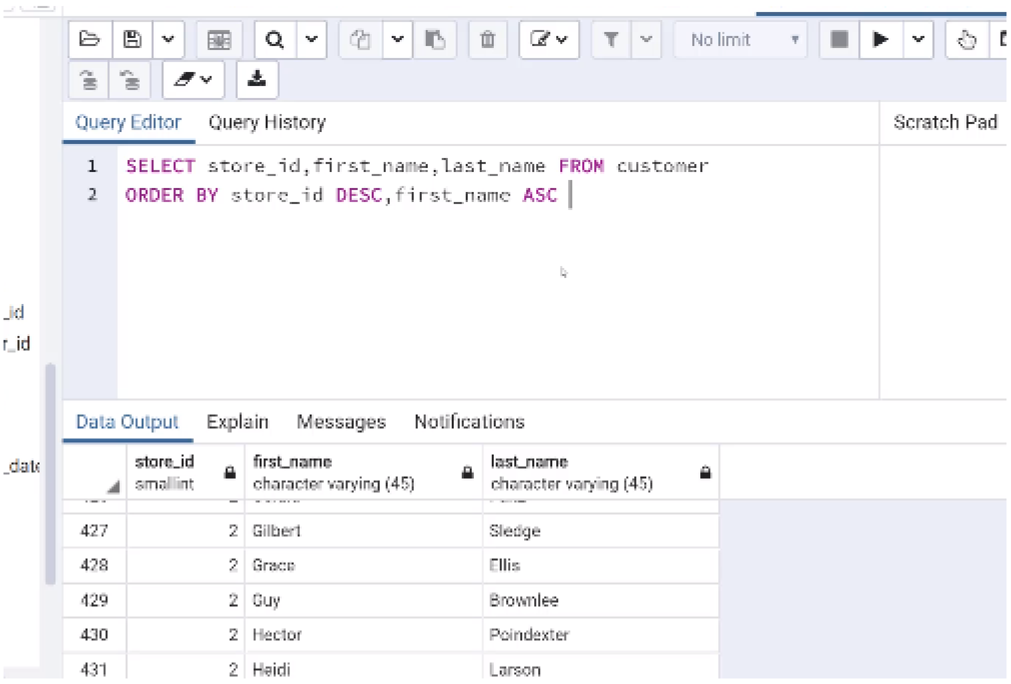
```
SELECT * FROM customer
ORDER BY first_name ASC
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)
1	375	2	Aaron	Selby	aaron.selby@sakilacustomer.org
2	367	1	Adam	Gooch	adam.gooch@sakilacustomer.org
3	525	2	Adrian	Clary	adrian.clary@sakilacustomer.org
4	217	2	Agnes	Bishop	agnes.bishop@sakilacustomer.or
5	389	1	Alan	Kahn	alan.kahn@sakilacustomer.org



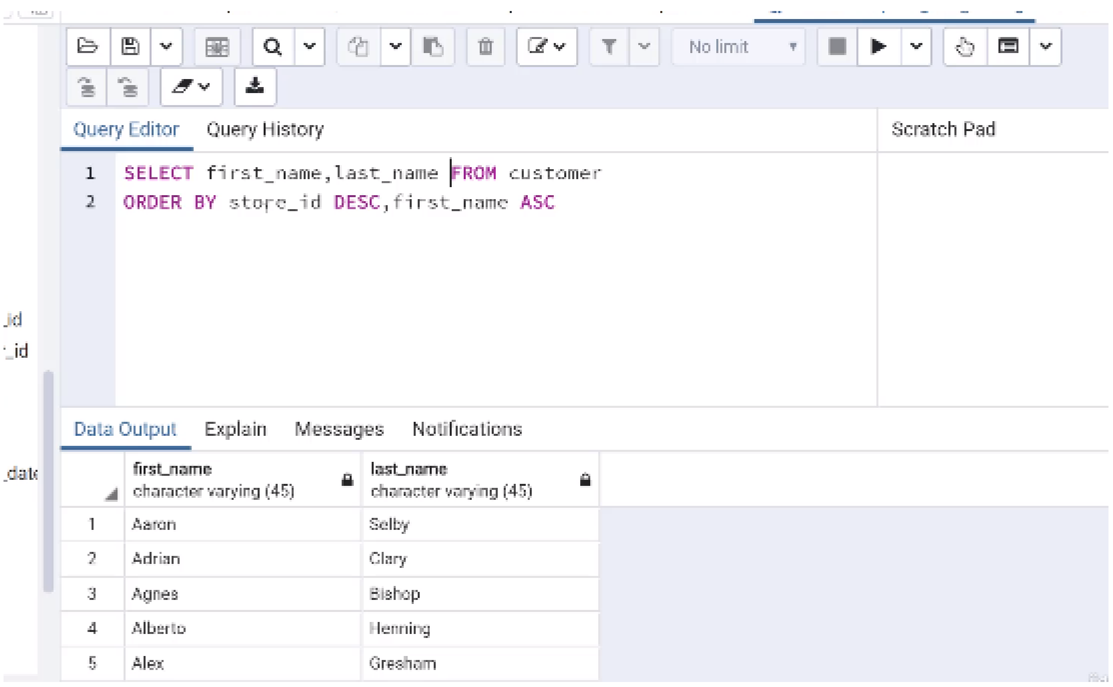
```
SELECT * FROM customer
ORDER BY first_name DESC
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)
1	479	1	Zachary	Hite	zachary.hite@sakilacustomer.org
2	174	2	Yvonne	Watkins	yvonne.watkins@sakilacustomer.



```
SELECT store_id,first_name,last_name FROM customer
ORDER BY store_id DESC,first_name ASC
```

	store_id smallint	first_name character varying (45)	last_name character varying (45)
427	2	Gilbert	Sledge
428	2	Grace	Ellis
429	2	Guy	Brownlee
430	2	Hector	Poindexter
431	2	Heidi	Larson



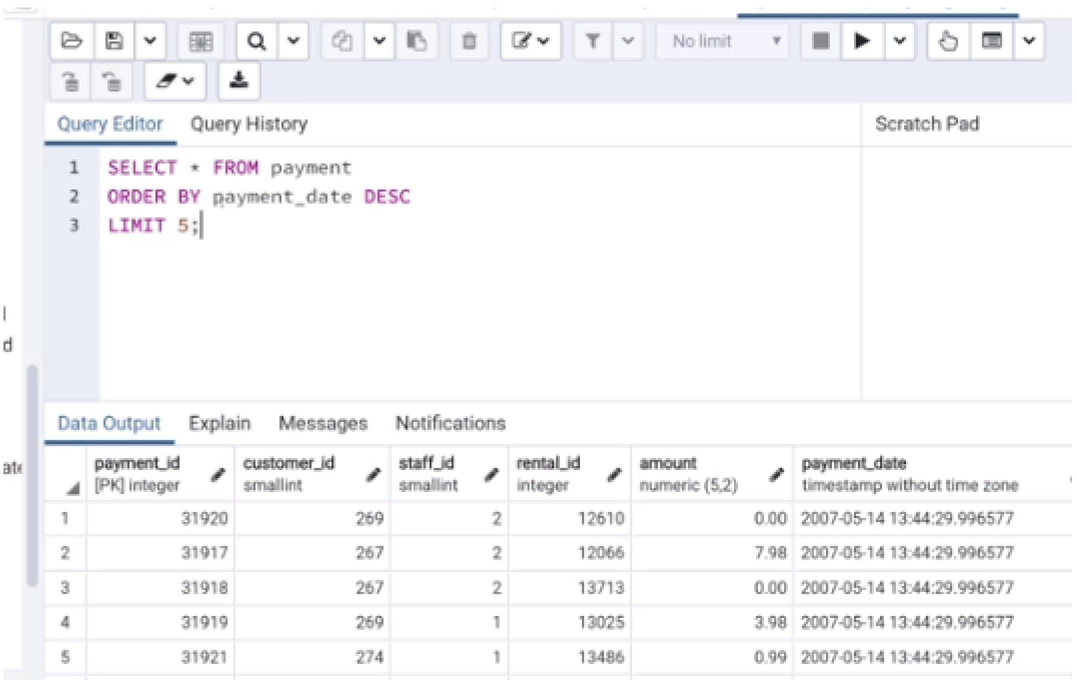
```
SELECT first_name,last_name FROM customer
ORDER BY store_id DESC,first_name ASC
```

	first_name character varying (45)	last_name character varying (45)
1	Aaron	Selby
2	Adrian	Clary
3	Agnes	Bishop
4	Alberto	Henning
5	Alex	Gresham

LIMT :

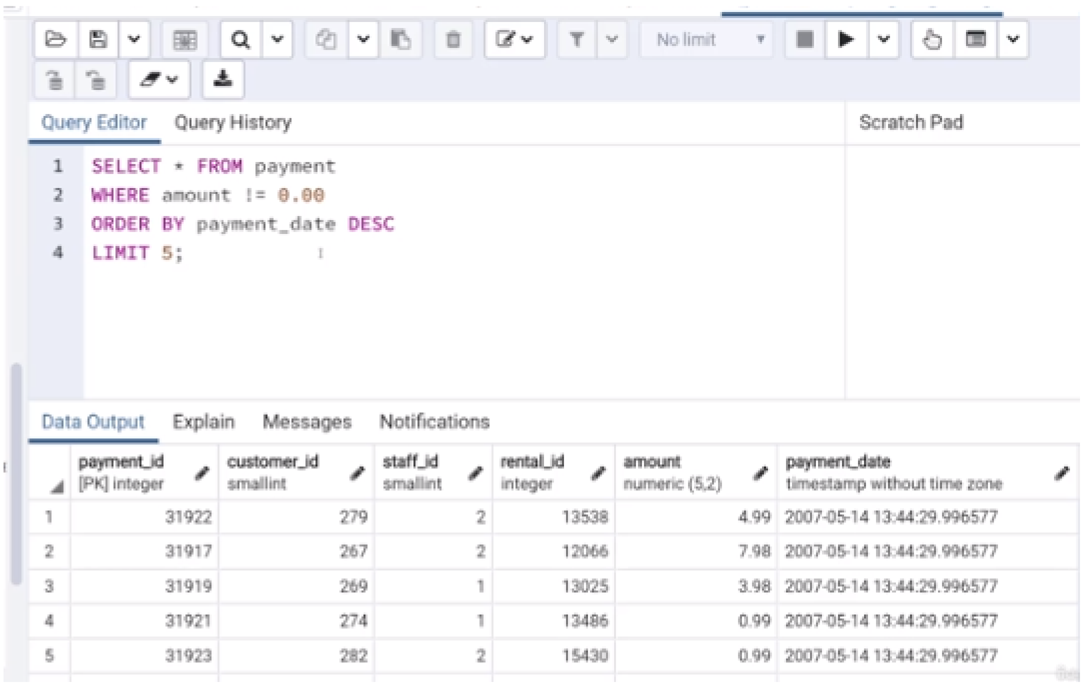
SQL

- The LIMIT command allows us to limit the number of rows returned for a query.
- Useful for not wanting to return every single row in a table, but only view the top few rows to get an idea of the table layout.
- LIMIT also becomes useful in combination with ORDER BY



```
SELECT * FROM payment
ORDER BY payment_date DESC
LIMIT 5;
```

	payment_id [PK] integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone
1	31920	269	2	12610	0.00	2007-05-14 13:44:29.996577
2	31917	267	2	12066	7.98	2007-05-14 13:44:29.996577
3	31918	267	2	13713	0.00	2007-05-14 13:44:29.996577
4	31919	269	1	13025	3.98	2007-05-14 13:44:29.996577
5	31921	274	1	13486	0.99	2007-05-14 13:44:29.996577


Query Editor | Query History | Scratch Pad

```
SELECT * FROM payment
WHERE amount != 0.00
ORDER BY payment_date DESC
LIMIT 5;
```

Data Output | Explain | Messages | Notifications

	payment_id [PK] integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone
1	31922	279	2	13538	4.99	2007-05-14 13:44:29.996577
2	31917	267	2	12066	7.98	2007-05-14 13:44:29.996577
3	31919	269	1	13025	3.98	2007-05-14 13:44:29.996577
4	31921	274	1	13486	0.99	2007-05-14 13:44:29.996577
5	31923	282	2	15430	0.99	2007-05-14 13:44:29.996577

Q & A:

SQL

- Challenge Task
  - We want to reward our first 10 paying customers.
  - What are the customer ids of the first 10 customers who created a payment?

SQL

- Solution

```
SELECT customer_id FROM payment
ORDER BY payment_date ASC
LIMIT 10;
```

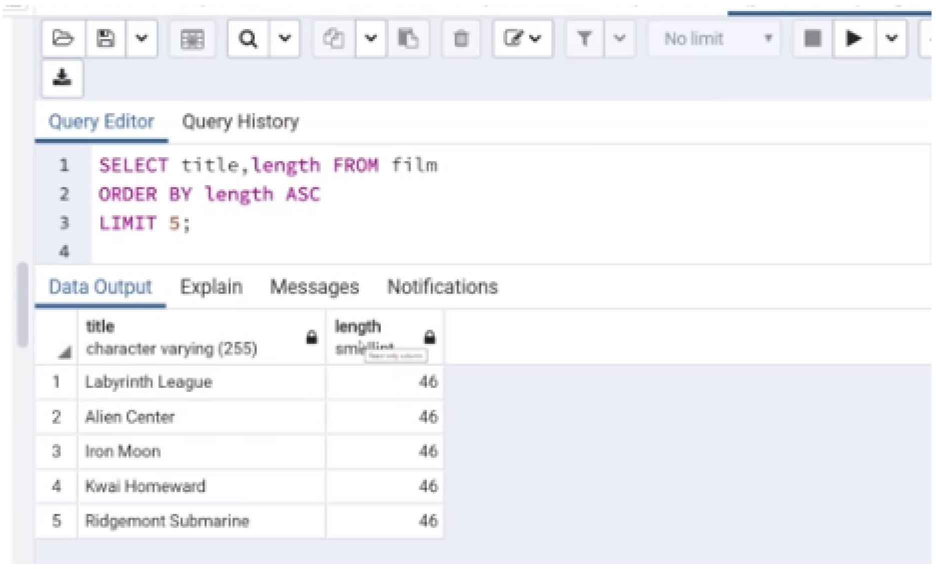
SQL

- Challenge Task
  - A customer wants to quickly rent a video to watch over their short lunch break.
  - What are the titles of the 5 shortest (in length of runtime) movies?

SQL

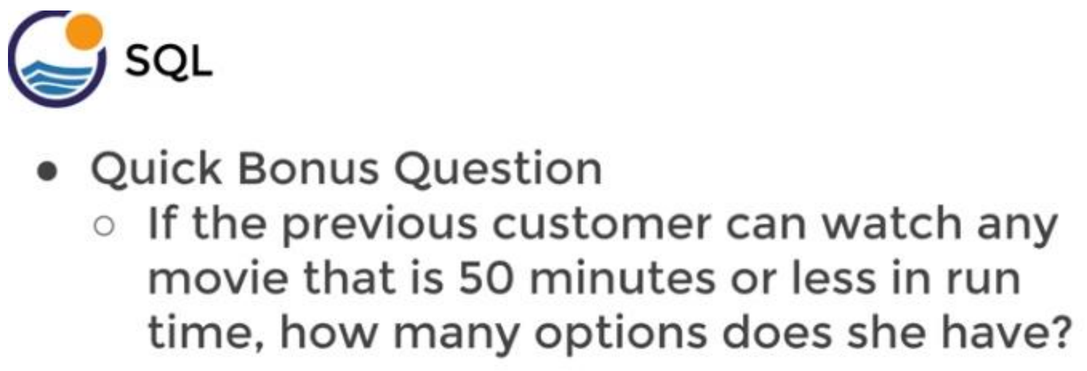
- Example Solution

  SELECT title,length FROM film
  ORDER BY length ASC
  LIMIT 5;

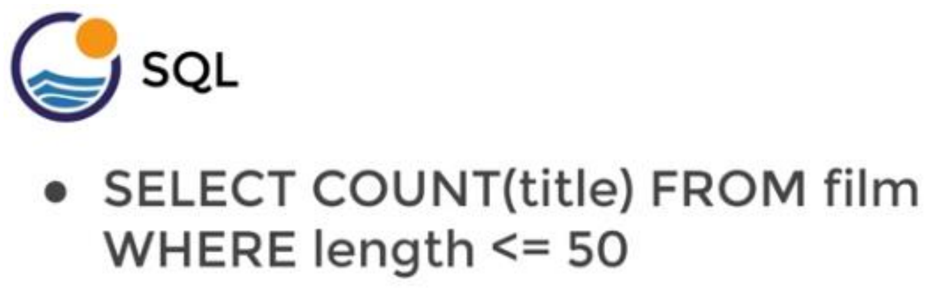


```
SELECT title,length FROM film
ORDER BY length ASC
LIMIT 5;
```

	title character varying (255)	length smallint
1	Labyrinth League	46
2	Alien Center	46
3	Iron Moon	46
4	Kwai Homeward	46
5	Ridgemont Submarine	46

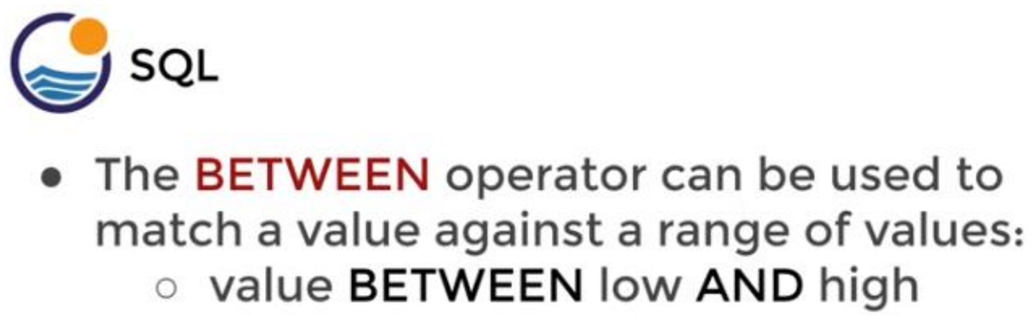

SQL

- Quick Bonus Question
  - If the previous customer can watch any movie that is 50 minutes or less in run time, how many options does she have?

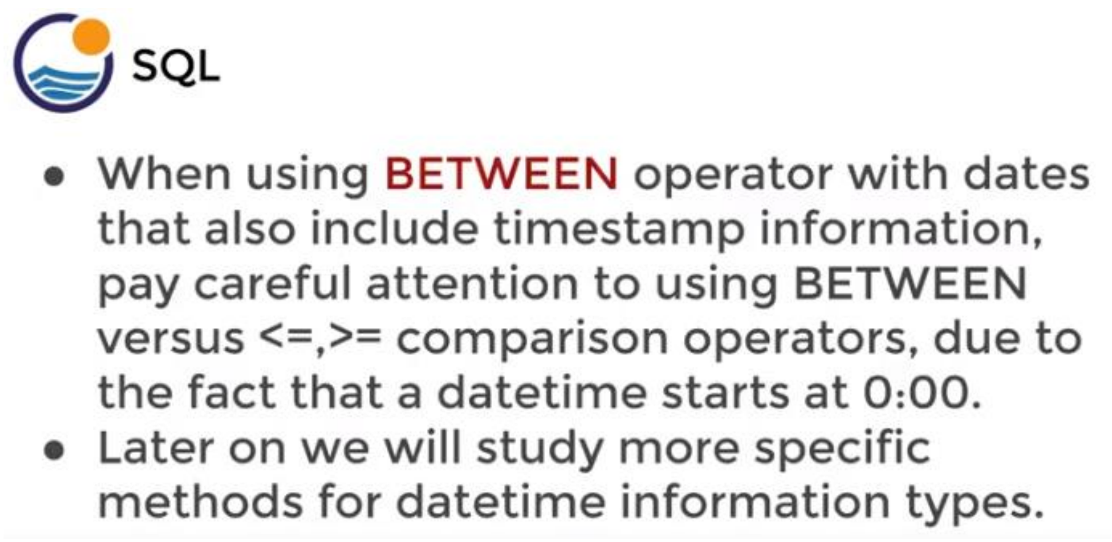

SQL

- SELECT COUNT(title) FROM film WHERE length <= 50

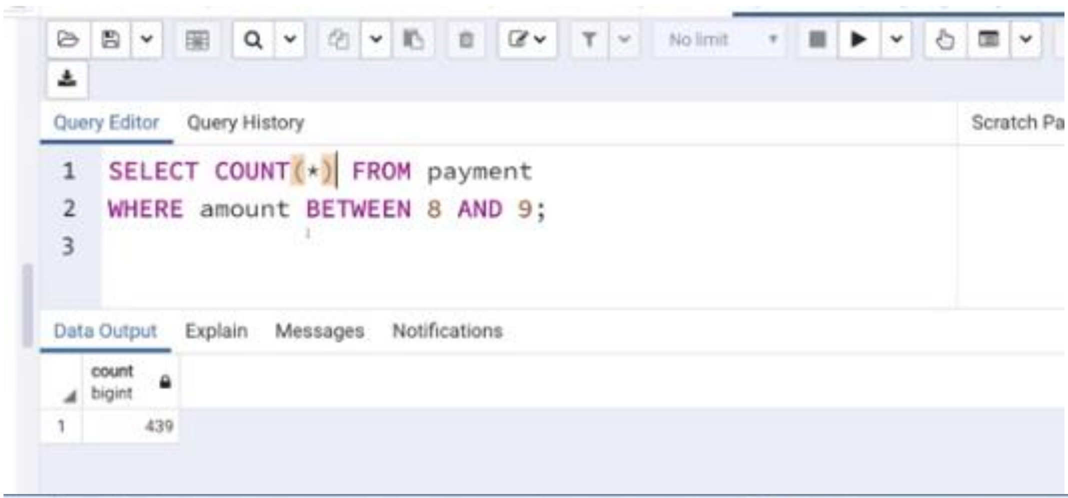
BETWEEN & NOT-BETWEEN:


SQL

- The **BETWEEN** operator can be used to match a value against a range of values:
  - value **BETWEEN** low **AND** high

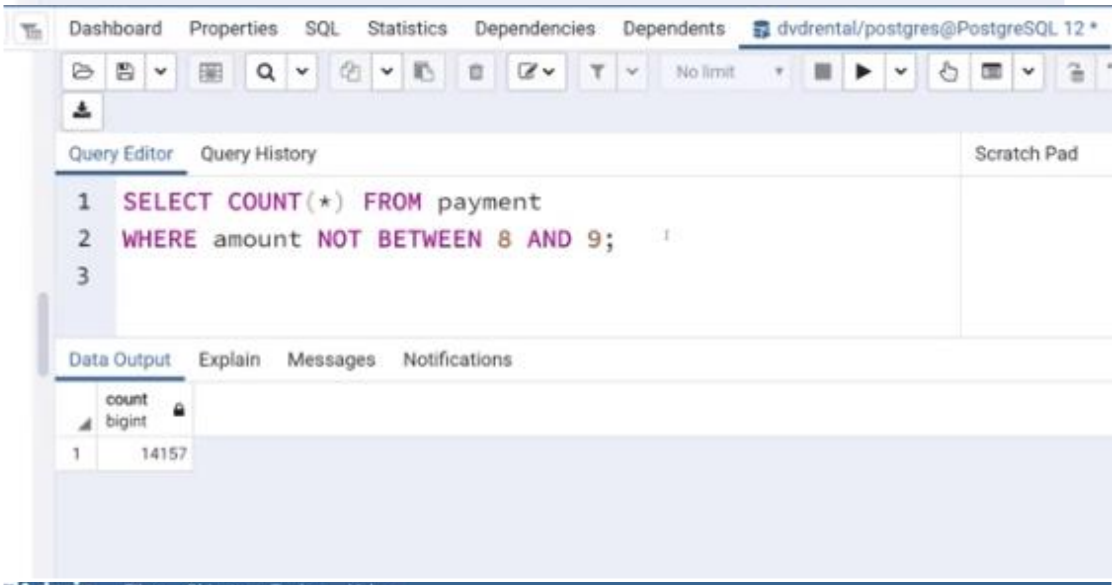

SQL

- When using **BETWEEN** operator with dates that also include timestamp information, pay careful attention to using BETWEEN versus <=,>= comparison operators, due to the fact that a datetime starts at 0:00.
- Later on we will study more specific methods for datetime information types.



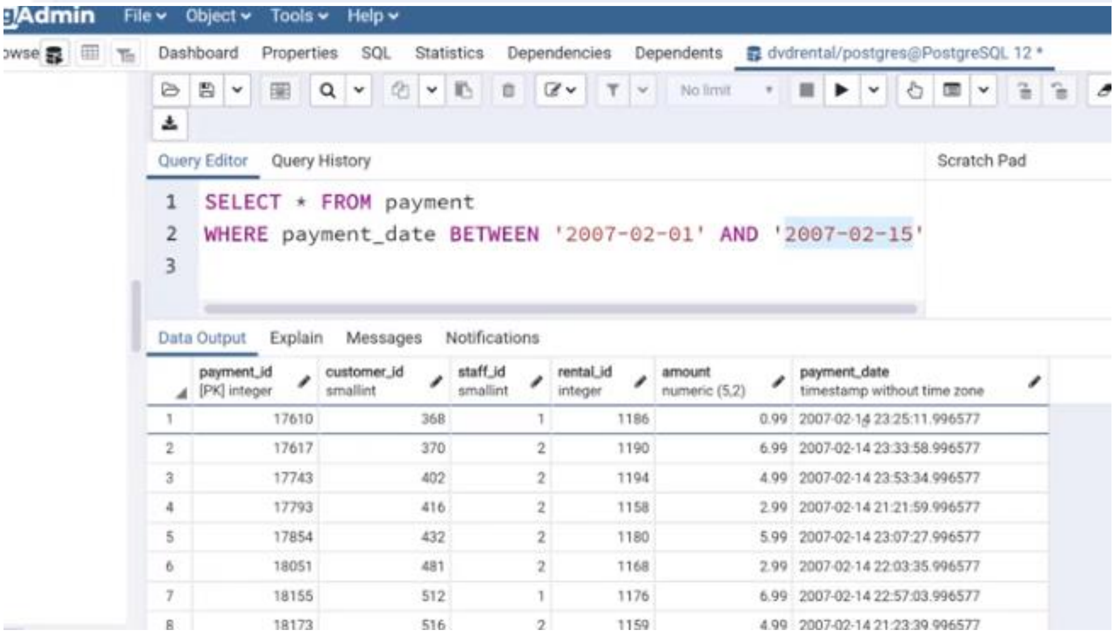
```sql
SELECT COUNT(*) FROM payment
WHERE amount BETWEEN 8 AND 9;
```

```
count
bigint
1    439
```



```sql
SELECT COUNT(*) FROM payment
WHERE amount NOT BETWEEN 8 AND 9;
```

```
count
bigint
1    14157
```



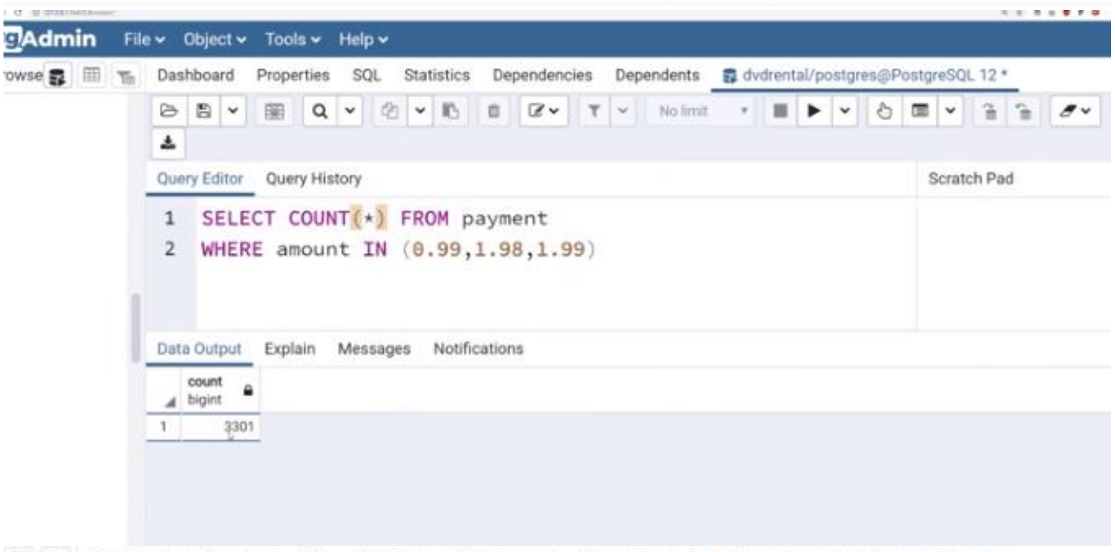
```sql
SELECT * FROM payment
WHERE payment_date BETWEEN '2007-02-01' AND '2007-02-15'
```

	payment_id [PK] integer	customer_id smallint	staff_id smallint	rental_id integer	amount numeric (5,2)	payment_date timestamp without time zone
1	17610	368	1	1186	0.99	2007-02-14 23:25:11.996577
2	17617	370	2	1190	6.99	2007-02-14 23:33:58.996577
3	17743	402	2	1194	4.99	2007-02-14 23:53:34.996577
4	17793	416	2	1158	2.99	2007-02-14 21:21:59.996577
5	17854	432	2	1180	5.99	2007-02-14 23:07:27.996577
6	18051	481	2	1168	2.99	2007-02-14 22:03:35.996577
7	18155	512	1	1176	6.99	2007-02-14 22:57:03.996577
8	18173	516	2	1159	4.99	2007-02-14 21:23:39.996577

IN & NOT-IN:

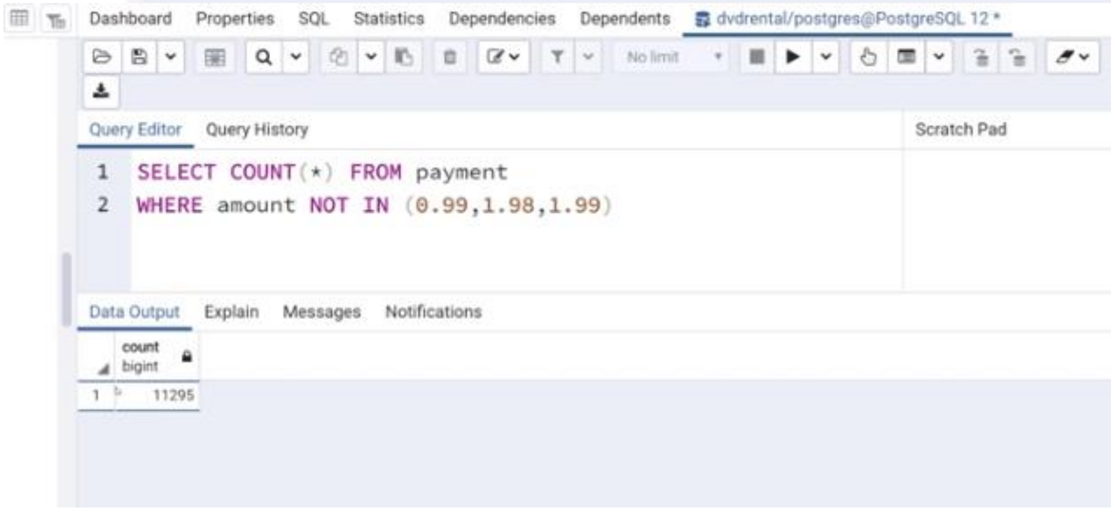
SQL

- In certain cases you want to check for multiple possible value options, for example, if a user's name shows up **IN** a list of known names.
- We can use the **IN** operator to create a condition that checks to see if a value in included in a list of multiple options.



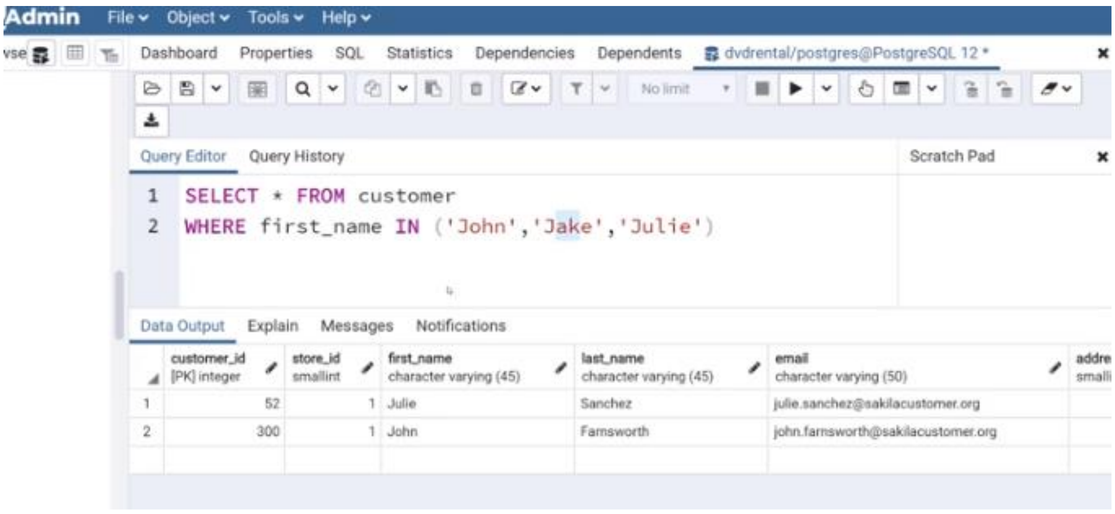
```
SELECT COUNT(*) FROM payment
WHERE amount IN (0.99,1.98,1.99)
```

```
  count
  bigint
1   3301
```



```
SELECT COUNT(*) FROM payment
WHERE amount NOT IN (0.99,1.98,1.99)
```

```
  count
  bigint
1   11295
```



```sql
SELECT * FROM customer
WHERE first_name IN ('John','Jake','Julie')
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	addre smalli
1	52	1	Julie	Sanchez	julie.sanchez@sakilacustomer.org	
2	300	1	John	Farnsworth	john.farnsworth@sakilacustomer.org	

LIKE & NOT LIKE (ILIKE):

SQL

- We've already been able to perform direct comparisons against strings, such as:
  - WHERE first_name= 'John'
- But what if we want to match against a general pattern in a string?
  - All emails ending in '@gmail.com'
  - All names that begin with an 'A'

SQL

- The LIKE operator allows us to perform pattern matching against string data with the use of **wildcard** characters:
  - Percent %
    - Matches any sequence of characters
  - Underscore _
    - Matches any single character

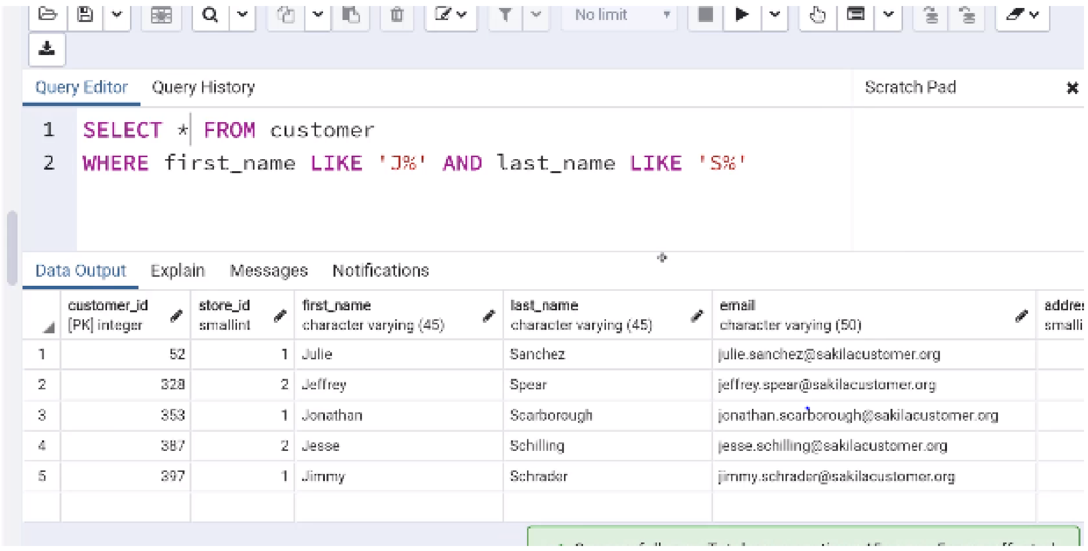
SQL

- All names that begin with an 'A'
  - WHERE name LIKE 'A%'
- All names that end with an 'a'
  - WHERE name LIKE '%a'

Notice that LIKE is case-sensitive, we can use ILIKE which is case-insensitive

SQL

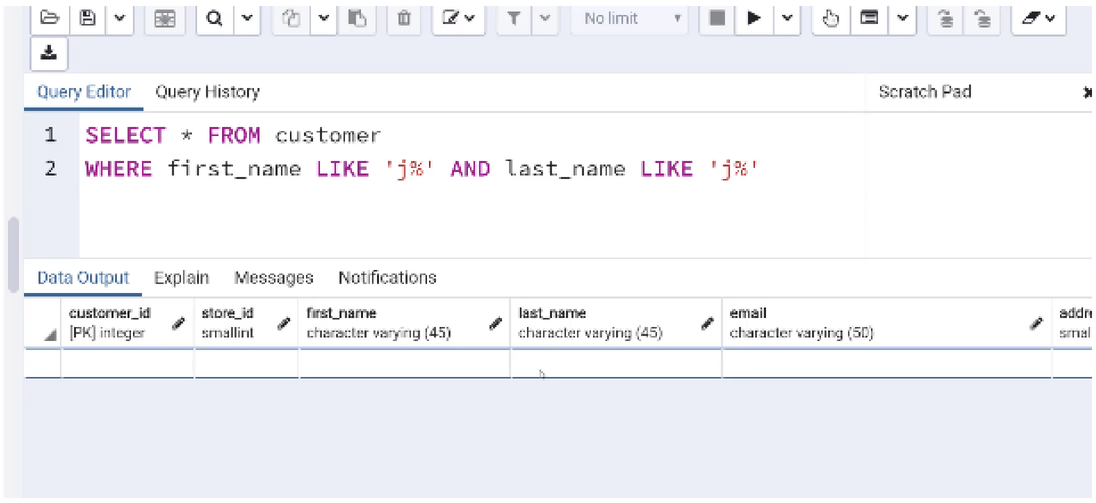
- We can also combine pattern matching operators to create more complex patterns
  - WHERE name LIKE '_her%'
    - Cheryl
    - Theresa
    - Sherri



```
SELECT * FROM customer
WHERE first_name LIKE 'J%' AND last_name LIKE 'S%'
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	addre smalli
1	52	1	Julie	Sanchez	julie.sanchez@sakilacustomer.org	
2	328	2	Jeffrey	Spear	jeffrey.spear@sakilacustomer.org	
3	353	1	Jonathan	Scarborough	jonathan.scarborough@sakilacustomer.org	
4	387	2	Jesse	Schilling	jesse.schilling@sakilacustomer.org	
5	397	1	Jimmy	Schrader	jimmy.schrader@sakilacustomer.org	

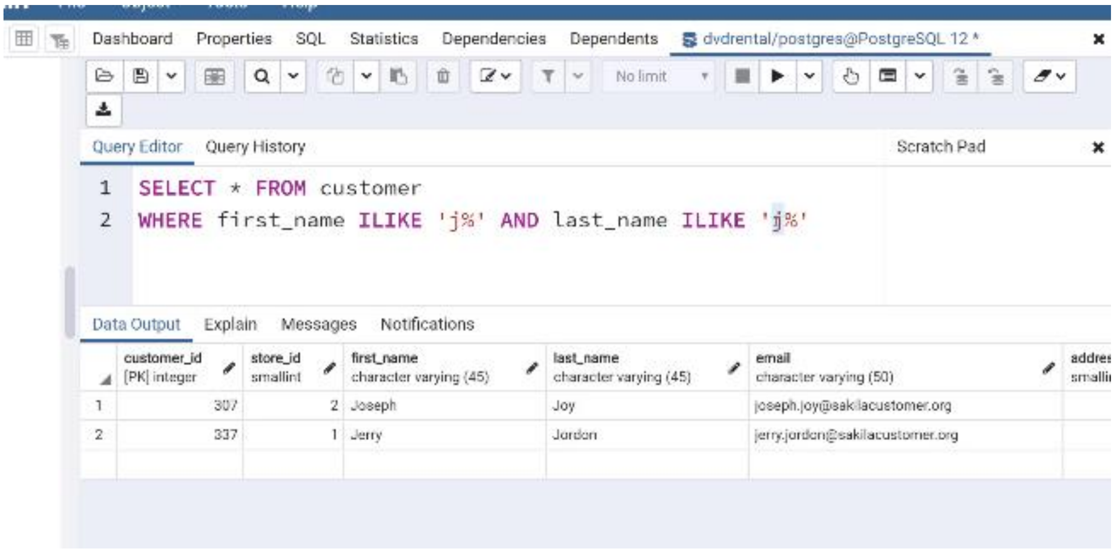
CASE SENSITIVE :



```sql
SELECT * FROM customer
WHERE first_name LIKE 'j%' AND last_name LIKE 'j%'
```

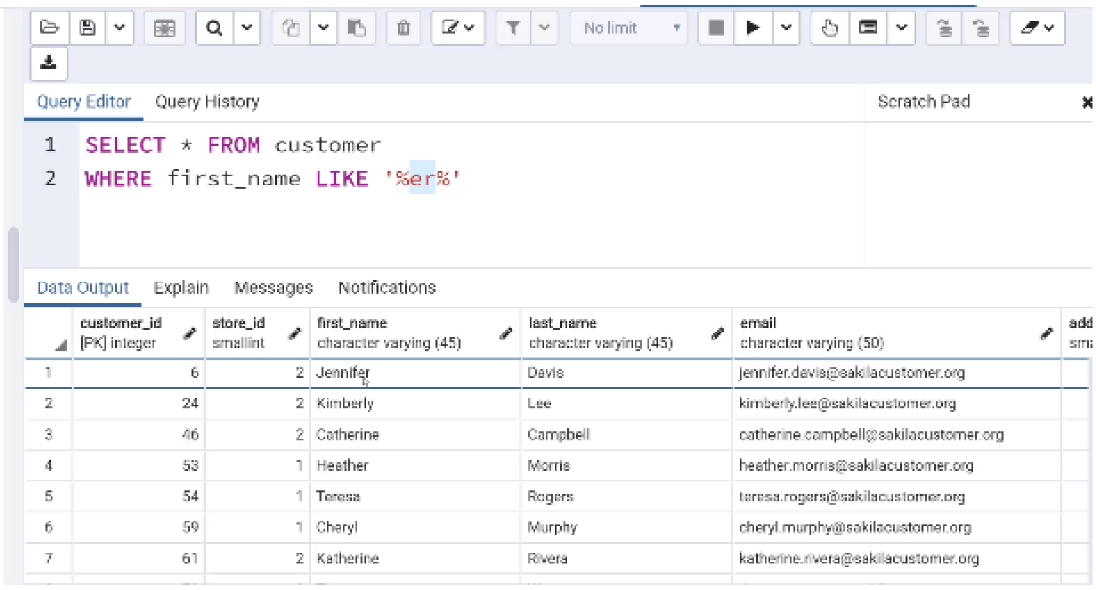
	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	addr smal

IN-SENSITIVE:



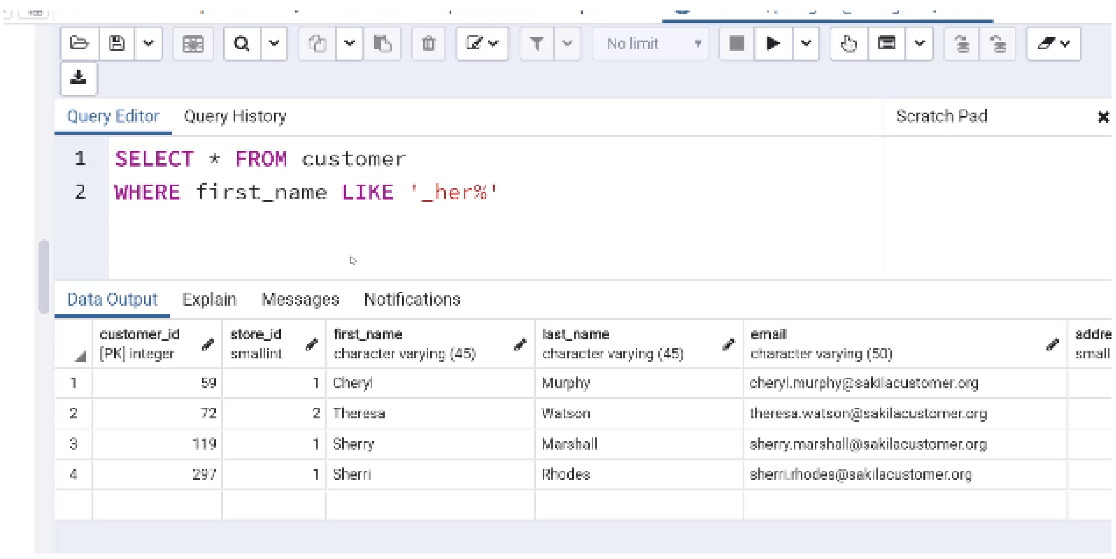
```sql
SELECT * FROM customer
WHERE first_name ILIKE 'j%' AND last_name ILIKE 'j%'
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	addres smalli
1	307	2	Joseph	Joy	joseph.joy@sakilacustomer.org	
2	337	1	Jerry	Jordon	jerry.jordon@sakilacustomer.org	



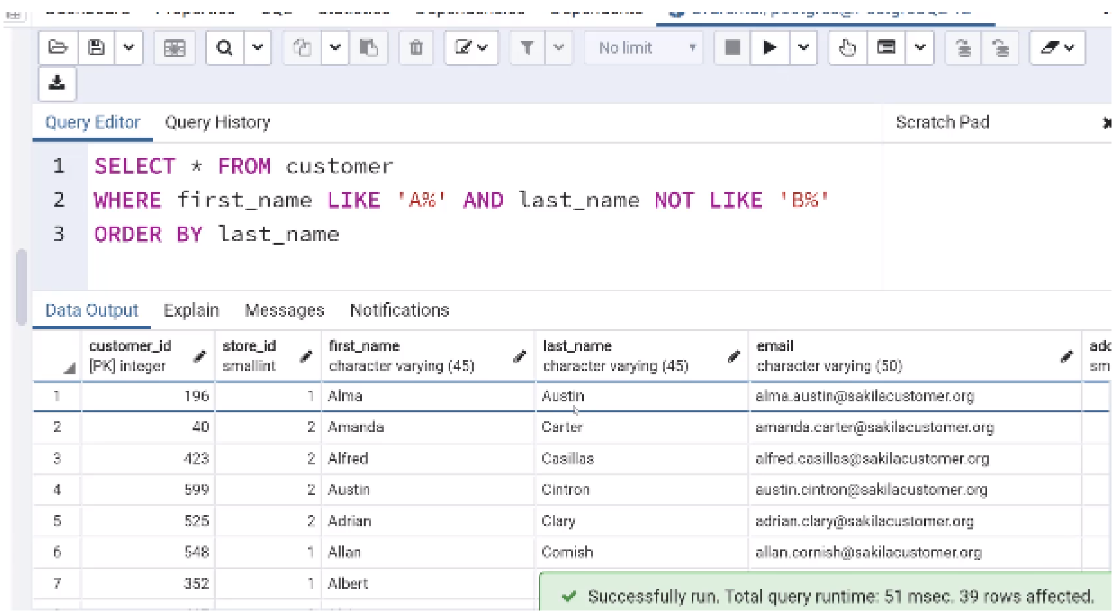
```sql
SELECT * FROM customer
WHERE first_name LIKE '%er%'
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	add sma
1	6	2	Jennifer	Davis	jennifer.davis@sakilacustomer.org	
2	24	2	Kimberly	Lee	kimberly.lee@sakilacustomer.org	
3	46	2	Catherine	Campbell	catherine.campbell@sakilacustomer.org	
4	53	1	Heather	Morris	heather.morris@sakilacustomer.org	
5	54	1	Teresa	Rogers	teresa.rogers@sakilacustomer.org	
6	59	1	Cheryl	Murphy	cheryl.murphy@sakilacustomer.org	
7	61	2	Katherine	Rivera	katherine.rivera@sakilacustomer.org	



```sql
SELECT * FROM customer
WHERE first_name LIKE '_her%'
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	addre small
1	59	1	Cheryl	Murphy	cheryl.murphy@sakilacustomer.org	
2	72	2	Theresa	Watson	theresa.watson@sakilacustomer.org	
3	119	1	Sherry	Marshall	sherry.marshall@sakilacustomer.org	
4	297	1	Sherri	Rhodes	sherri.rhodes@sakilacustomer.org	



```sql
SELECT * FROM customer
WHERE first_name LIKE 'A%' AND last_name NOT LIKE 'B%'
ORDER BY last_name
```

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	add sm
1	196	1	Alma	Austin	alma.austin@sakilacustomer.org	
2	40	2	Amanda	Carter	amanda.carter@sakilacustomer.org	
3	423	2	Alfred	Casillas	alfred.casillas@sakilacustomer.org	
4	599	2	Austin	Cintron	austin.cintron@sakilacustomer.org	
5	525	2	Adrian	Clary	adrian.clary@sakilacustomer.org	
6	548	1	Allan	Cornish	allan.cornish@sakilacustomer.org	
7	352	1	Albert			

✓ Successfully run. Total query runtime: 51 msec. 39 rows affected.

Q & A:

Challenge

- How many payment transactions were greater than $5.00?

SOLUTION

```
SELECT COUNT(amount) FROM payment
WHERE amount > 5;
```

Challenge

- How many actors have a first name that starts with the letter P?

SOLUTION

```
SELECT COUNT(*) FROM actor
WHERE first_name LIKE 'P%';
```

Challenge

- How many unique districts are our customers from?

SOLUTION

```
SELECT COUNT(DISTINCT(district))
FROM address;
```

Challenge

- Retrieve the list of names for those distinct districts from the previous question.

SOLUTION

```
SELECT DISTINCT(district) FROM address;
```

Challenge

- How many films have a rating of R and a replacement cost between $5 and $15?

SOLUTION

```
SELECT COUNT(*) FROM film
WHERE rating = 'R'
AND replacement_cost BETWEEN 5 AND 15;
```

Challenge

- How many films have the word Truman somewhere in the title?

SOLUTION

```
SELECT COUNT(*) FROM film
WHERE title LIKE '%Truman%';
```