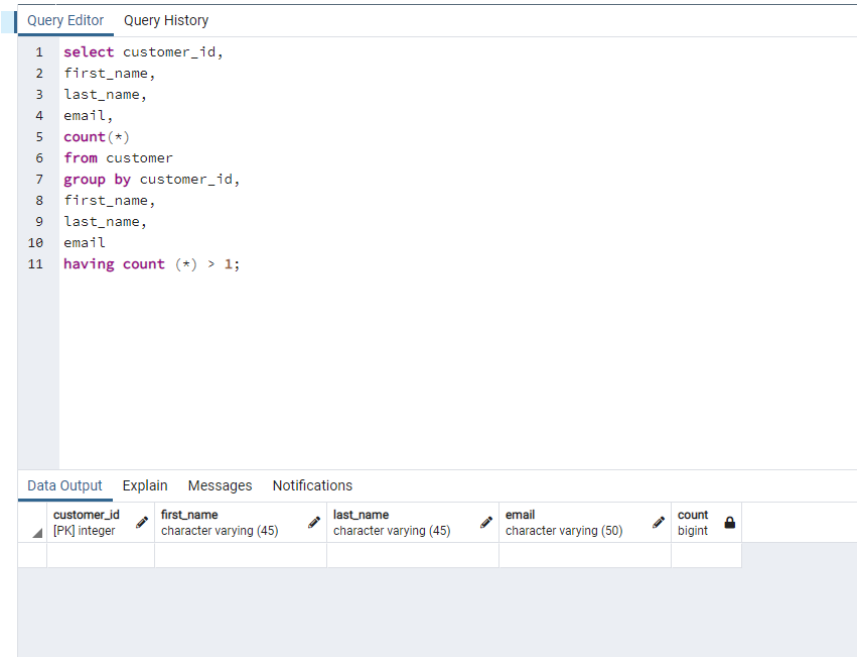
3.6 Summarizing and Cleaning Data

1. **Check for and clean dirty data:** Find out if the **film** table and the **customer** table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Create a new "Answers 3.6" document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

```
1  SELECT title,
2         release_year,
3         language_id,
4         rental_duration,
5         COUNT(*)
6  FROM film
7  GROUP BY title,
8           release_year,
9           language_id,
10          rental_duration
11 HAVING COUNT(*) >1;
```

Data Output | Explain | Messages | Notifications

title character varying (255)	release_year integer	language_id smallint	rental_duration smallint	count bigint



a. The first step is to use a general select * from [table_name] so the analyst can see all of the factors they have to check for. After, the analyst should select a few column names to check for duplicates amongst them followed by the count(*) and then the table name [select columns, count(*)]. Next, group those same columns and follow it with the 'having count (*) >1;' command. This counts any duplicates in the listed columns.
b. The above screen shots show how I did the described steps, yielding nothing in my results output. This means that, within these two tables, there are no duplicates.

2. Were there duplicates, I would utilize a command to delete the duplicates:

```
DELETE
FROM tablename
WHERE unique_id NOT IN
    (SELECT MIN(unique_id)
     FROM tablename
     GROUP BY col1,
              col2,
                col3, ...)
```

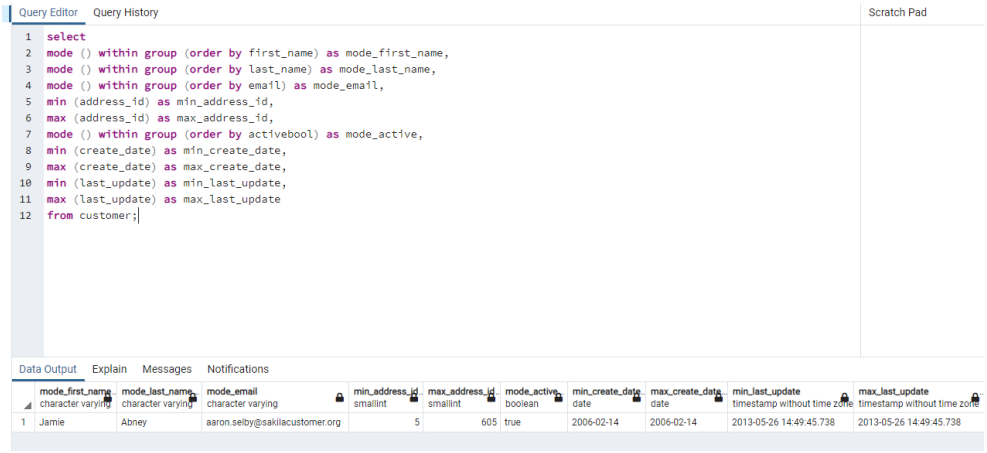
11. **Summarize your data:** Use SQL to calculate descriptive statistics for both the film table and the customer table. For numerical columns, this means finding the minimum, maximum, and average values. For non-numerical columns, calculate the mode value. Copy-paste your SQL queries and their outputs into your answers document.

Film:

```
SELECT MIN(length) AS min_length,
 MAX(length) AS max_length,
 AVG(length) AS avg_length,

MIN(rental_duration) AS min_rent_duration,
 MAX(rental_duration) AS max_rent_duration,
 AVG(rental_duration) AS avg_rent_duration,

mode() WITHIN GROUP (ORDER BY rating)
      AS modal_value,

MIN(rental_rate) AS min_rent_rate,
 MAX(rental_rate) AS max_rent_rate,
 AVG(rental_rate) AS avg_rent_rate,
 COUNT(rental_rate) AS count_rent_rate,
 COUNT(*) AS count_rows

FROM film;
```

min_length	max_length	avg_length	min_rent_duration	max_rent_duration	avg_rent_duration	modal_value	min_rent_rate	max_rent_rate	avg_rent_rate	count_rent...	count_rows
smallint	smallint	numeric	smallint	smallint	numeric	mpaa_rating	numeric	numeric	numeric	bigint	bigint
1	46	185	115.2720000000000000	3	7	4.9850000000000000	PG-13	0.99	4.99	.9800000000000000	1000	1000

Customer:



```
select
mode () within group (order by first_name) as mode_first_name,
mode () within group (order by last_name) as mode_last_name,
mode () within group (order by email) as mode_email,
min (address_id) as min_address_id,
max (address_id) as max_address_id,
mode () within group (order by activebool) as mode_active,
min (create_date) as min_create_date,
max (create_date) as max_create_date,
min (last_update) as min_last_update,
max (last_update) as max_last_update
from customer;
```

mode_first_name	mode_last_name	mode_email	min_address_id	max_address_id	mode_active	min_create_date	max_create_date	min_last_update	max_last_update
character varying	character varying	character varying	smallint	smallint	boolean	date	date	timestamp without time zone	timestamp without time zone
1 Jamie	Abney	aaron.selby@sakilacustomer.org	5	605	true	2006-02-14	2006-02-14	2013-05-26 14:49:45.738	2013-05-26 14:49:45.738

I don't see the relevance of the above statistics for a group related to something like customers. In reality, I'd just spell out the columns and compare it to a [select distinct] or utilize the ['having count (*) >1] to find duplicates. These sort of data don't really need a modal representation. Namely, there's no first name, last name, or email that should occur more than once.

12. **Reflect on your work:** Back in Achievement 1 you learned about data profiling in Excel. Based on your previous experience, which tool (Excel or SQL) do you think is more effective for data profiling, and why? Consider their respective functions, ease of use, and speed. Write a short paragraph in the running document that you have started.

    a. Excel seems faster when one is looking at things like averages, min, and max utilizing the pivot chart. One could simply print [select * from table] the whole category as a .csv. Using the min and max prompts for each category is tedious. The advantage is having a copy/paste in the sql syntax that can be modified in small ways to represent different categories; that is, if you're looking at min(), max(), avg(), in one category, simply copy and paste the syntax and edit it to target the new category/column in the table. Don't forget proper comma placement/removal when doing this.

13. Save your "Answers 3.6" document as a PDF and upload it here for your tutor to review.