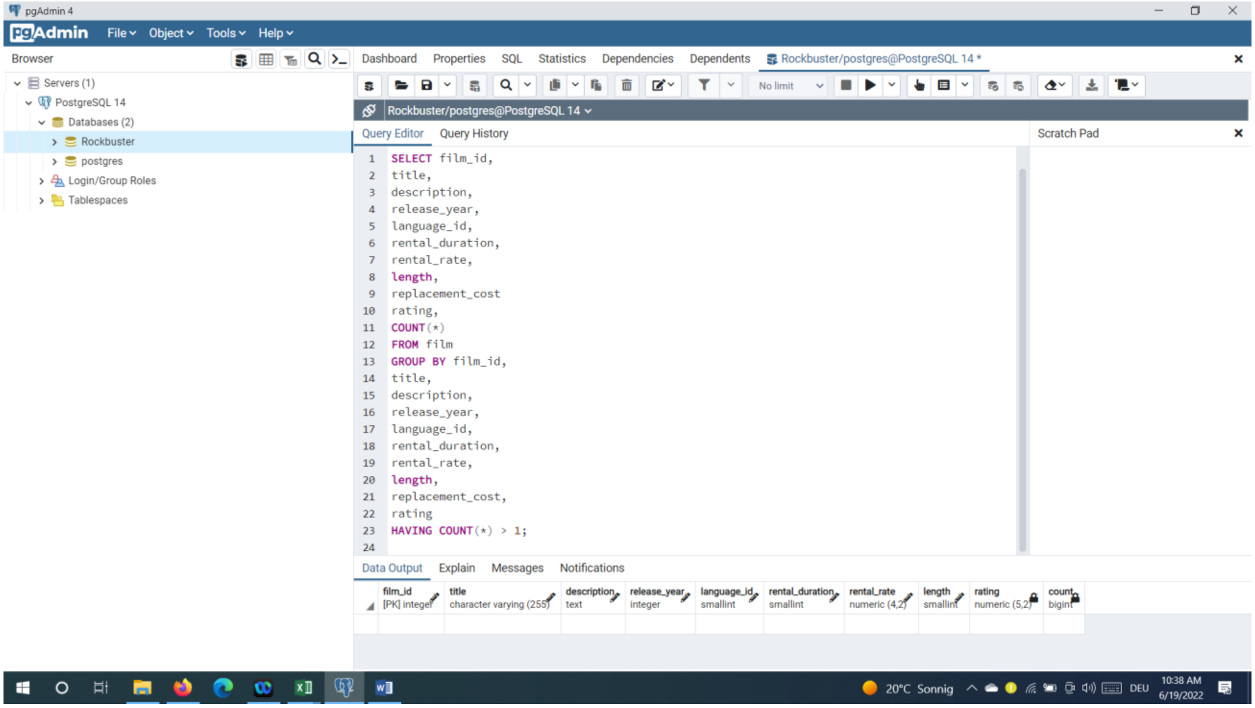
Task 3.6: Summarizing & Cleaning Data in SQL

Directions

Rockbuster's database engineers have loaded some new data into the database, and your manager has asked you to clean and profile it. Follow the instructions below to complete their request:

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).

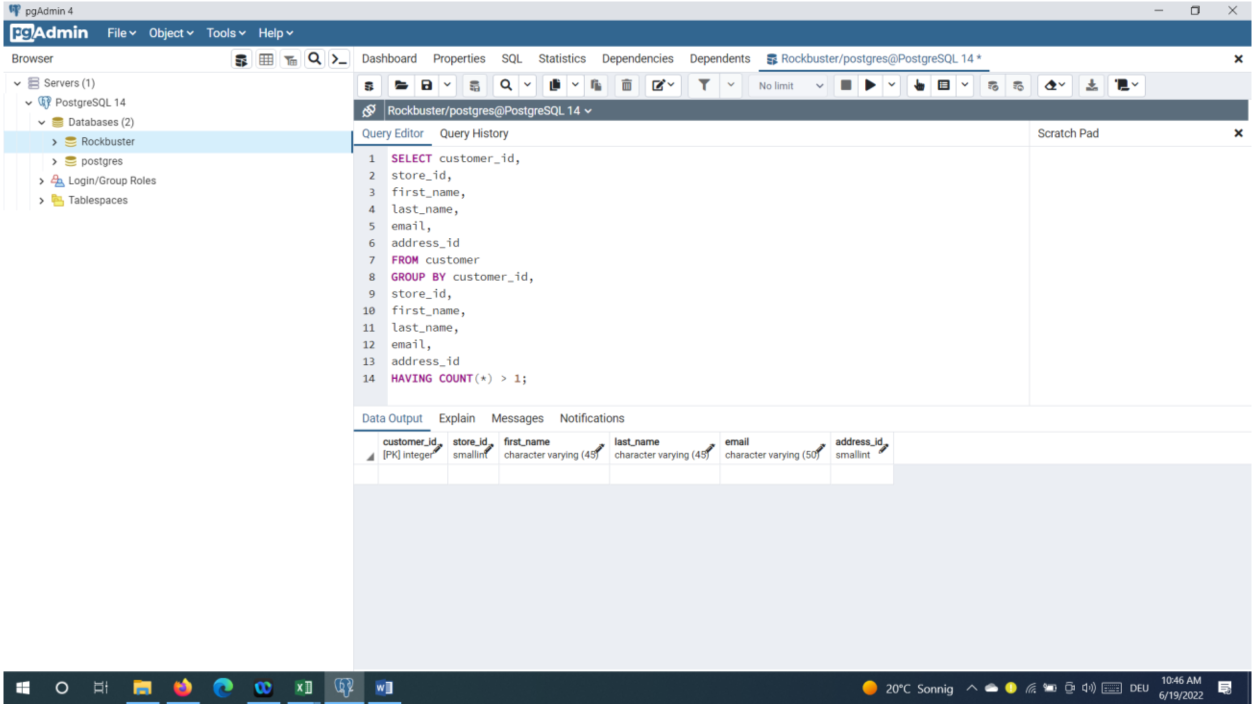
   a) Checking for duplicate value from the film table



```
SELECT film_id,
title,
description,
release_year,
language_id,
rental_duration,
rental_rate,
length,
replacement_cost
rating,
COUNT(*)
FROM film
GROUP BY film_id,
title,
description,
release_year,
language_id,
rental_duration,
rental_rate,
length,
replacement_cost,
rating
HAVING COUNT(*) > 1;
```

film_id [PK] integer	title character varying (255)	description text	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	length smallint	rating numeric (5,2)	count bigint

b) Checking for duplicate value from the customer table



```
SELECT customer_id,
store_id,
first_name,
last_name,
email,
address_id
FROM customer
GROUP BY customer_id,
store_id,
first_name,
last_name,
email,
address_id
HAVING COUNT(*) > 1;
```

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

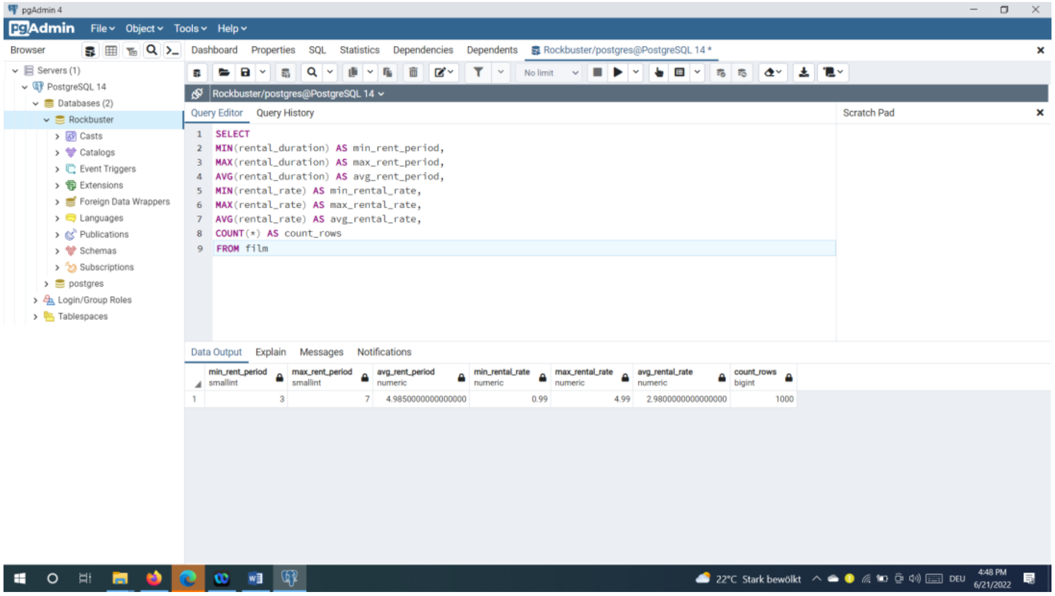
There are no duplicates in the above made tables. In case that there were any, I can:

- Create a virtual table, known as a “view,” where you select only unique records.

- Delete the duplicate record from the table or view

2. **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.

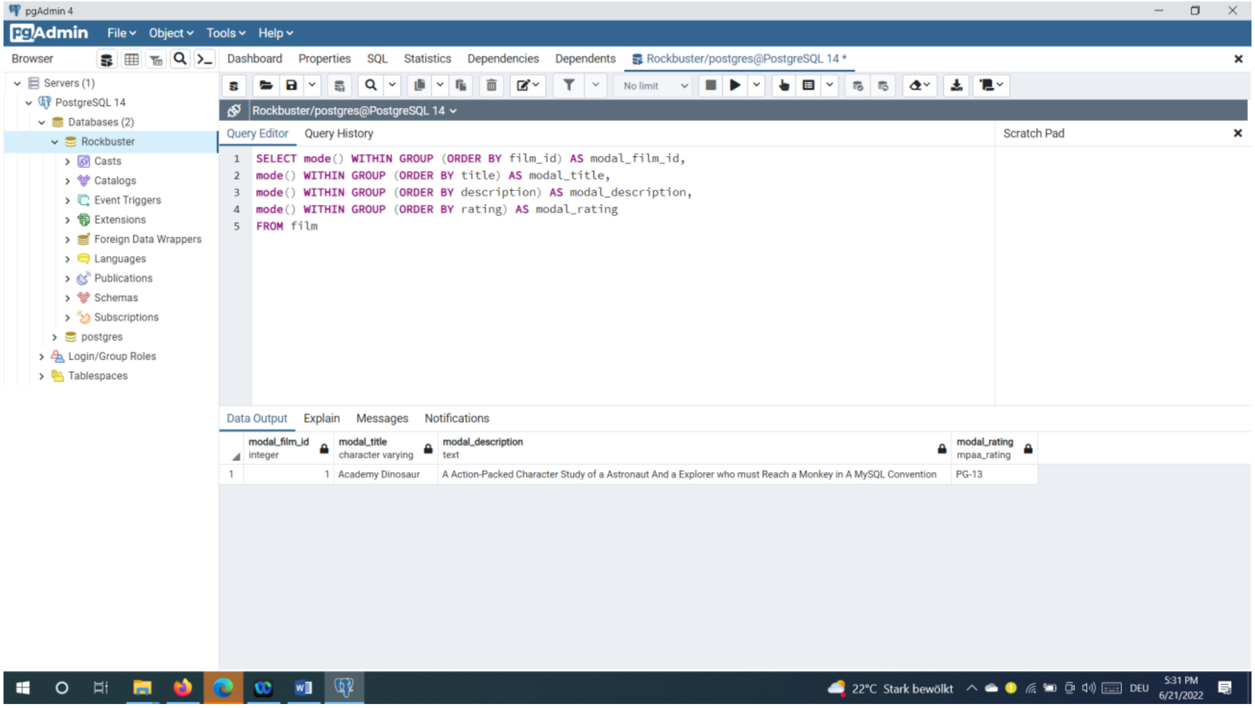
a) Summary for numeric columns in film table



```
SELECT
MIN(rental_duration) AS min_rent_period,
MAX(rental_duration) AS max_rent_period,
AVG(rental_duration) AS avg_rent_period,
MIN(rental_rate) AS min_rental_rate,
MAX(rental_rate) AS max_rental_rate,
AVG(rental_rate) AS avg_rental_rate,
COUNT(*) AS count_rows
FROM film
```

	min_rent_period smallint	max_rent_period smallint	avg_rent_period numeric	min_rental_rate numeric	max_rental_rate numeric	avg_rental_rate numeric	count_rows bigint
1	3	7	4.9850000000000000	0.99	4.99	2.9800000000000000	1000

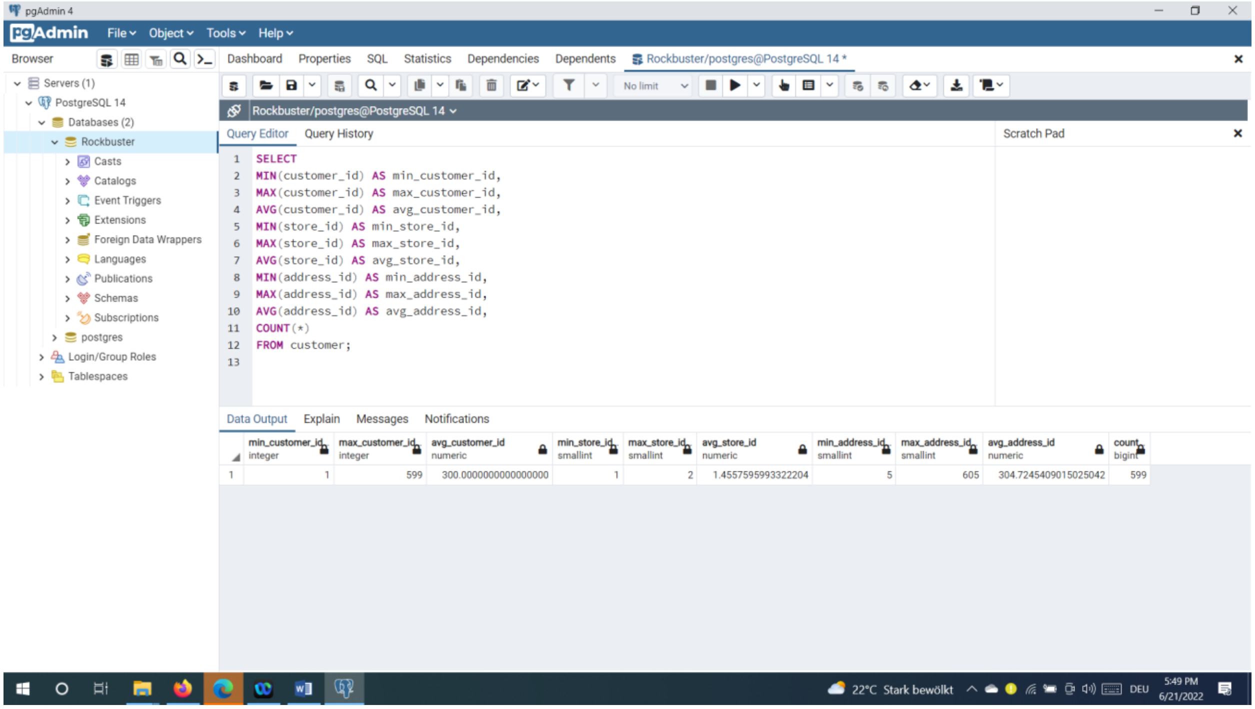
b) Summary for numeric columns in film table



```
SELECT mode() WITHIN GROUP (ORDER BY film_id) AS modal_film_id,
mode() WITHIN GROUP (ORDER BY title) AS modal_title,
mode() WITHIN GROUP (ORDER BY description) AS modal_description,
mode() WITHIN GROUP (ORDER BY rating) AS modal_rating
FROM film
```

	modal_film_id integer	modal_title character varying	modal_description text	modal_rating mpaa_rating
1	1	Academy Dinosaur	A Action-Packed Character Study of a Astronaut And a Explorer who must Reach a Monkey in A MySQL Convention	PG-13

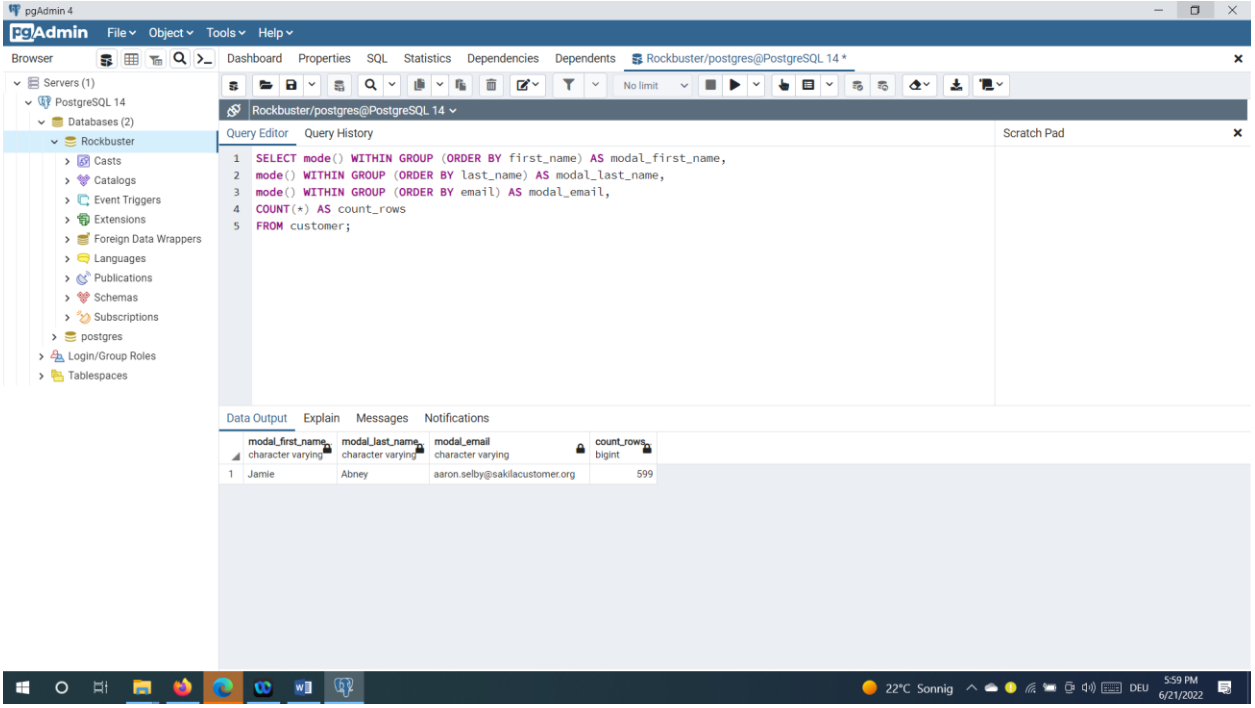
c) Summary for numeric columns in customer table



```sql
SELECT
MIN(customer_id) AS min_customer_id,
MAX(customer_id) AS max_customer_id,
AVG(customer_id) AS avg_customer_id,
MIN(store_id) AS min_store_id,
MAX(store_id) AS max_store_id,
AVG(store_id) AS avg_store_id,
MIN(address_id) AS min_address_id,
MAX(address_id) AS max_address_id,
AVG(address_id) AS avg_address_id,
COUNT(*)
FROM customer;
```

	min_customer_id integer	max_customer_id integer	avg_customer_id numeric	min_store_id smallint	max_store_id smallint	avg_store_id numeric	min_address_id smallint	max_address_id smallint	avg_address_id numeric	count bigint
1	1	599	300.0000000000000000	1	2	1.4557595993322204	5	605	304.7245409015025042	599

d) Summary for non-numeric columns in customer table



```sql
SELECT mode() WITHIN GROUP (ORDER BY first_name) AS modal_first_name,
mode() WITHIN GROUP (ORDER BY last_name) AS modal_last_name,
mode() WITHIN GROUP (ORDER BY email) AS modal_email,
COUNT(*) AS count_rows
FROM customer;
```

	modal_first_name character varying	modal_last_name character varying	modal_email character varying	count_rows bigint
1	Jamie	Abney	aaron.selby@sakilacustomer.org	599

3. 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.

Excel is a good tool in analyzing the smaller size of the data. On the other side SQL is more efficient and faster when it comes working with the bigger size of the data.