Answers 3.6

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).

Non-uniform data

```
SELECT DISTINCT rating
FROM film
GROUP BY rating
```

Non-uniform values can be updated to uniform values. Use UPDATE, SET, WHERE commands to replace non-uniform values.

Duplicate data

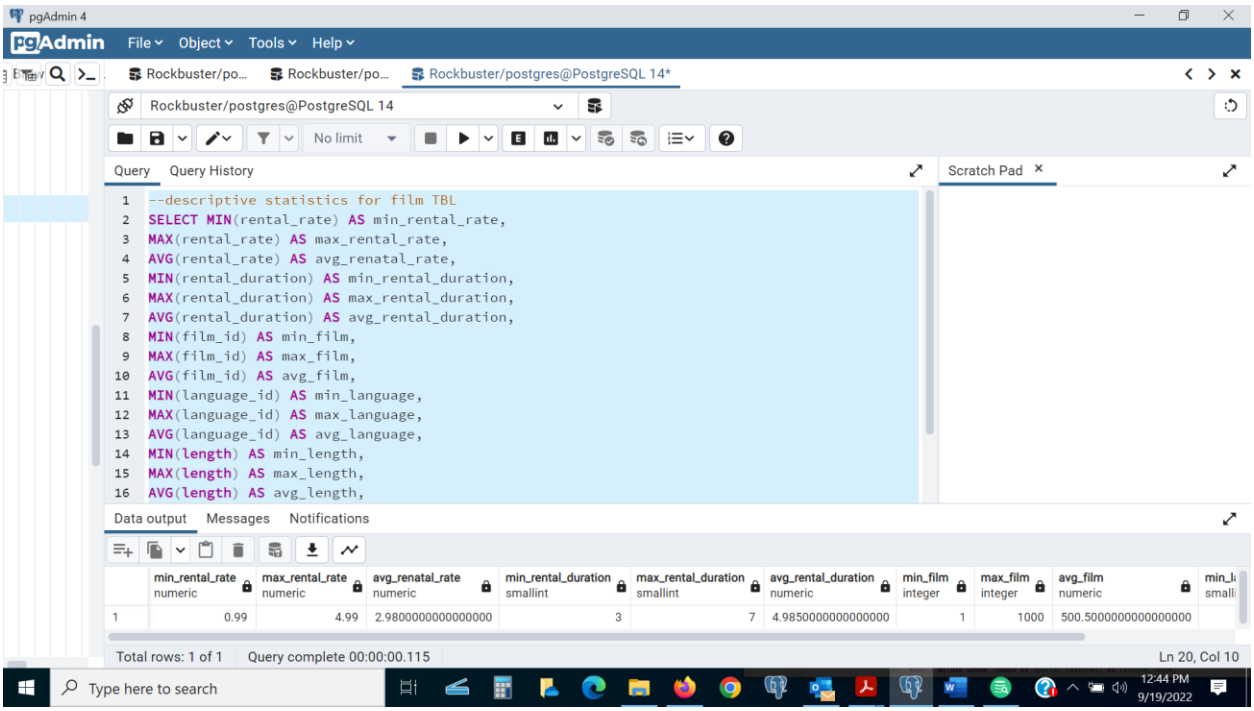
```
--looking for duplicate data in Film TBL
SELECT film_id,
title,
release_year,
rental_duration,
rental_rate,
        COUNT(*)
FROM film
GROUP BY film_id,
title,
release_year,
rental_duration,
rental_rate
HAVING COUNT(*) > 1

---- looking for duplicate values customer TBL
SELECT first_name,
            last_name
            email,
            address_id
            active,
            COUNT(*)
FROM customer
GROUP BY first_name,
      last_name,
                  email,
                  address_id,
                  active
HAVING COUNT(*) > 1
```

For Duplicate values a VIEW function can be used to create a view table to store duplicate values to be later cleaned or the DELETE command can be used to delete the duplicate values by appropriate data engineers with permissions.

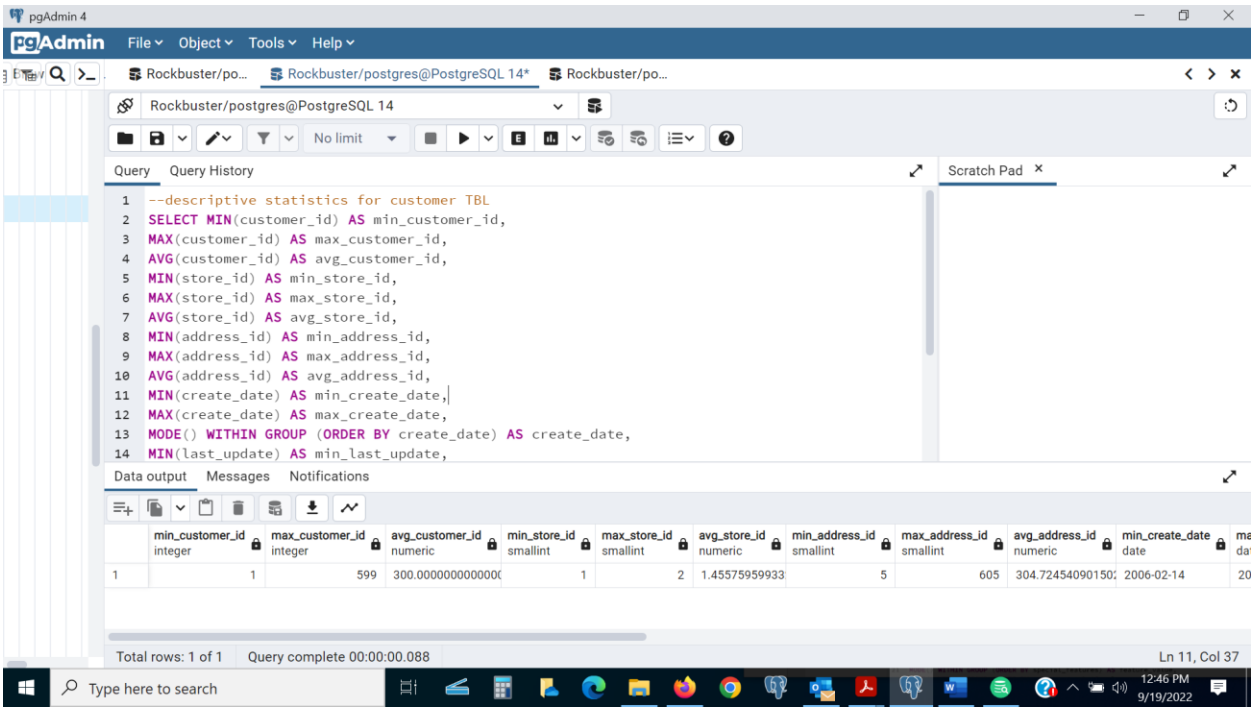
2. **Summarize your data:** Use SQL to calculate descriptive statistics for both the **film table and the customer table**. For numerical columns, this means finding the minimum, maximum, and average values. For non-numerical columns, calculate the mode value. Copy-paste your SQL queries and their outputs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

```
--descriptive statistics for film TBL
SELECT MIN(rental_rate) AS min_rental_rate,
MAX(rental_rate) AS max_rental_rate,
AVG(rental_rate) AS avg_renatal_rate,
MIN(rental_duration) AS min_rental_duration,
MAX(rental_duration) AS max_rental_duration,
AVG(rental_duration) AS avg_rental_duration,
MIN(film_id) AS min_film,
MAX(film_id) AS max_film,
AVG(film_id) AS avg_film,
MIN(language_id) AS min_language,
MAX(language_id) AS max_language,
AVG(language_id) AS avg_language,
MIN(length) AS min_length,
MAX(length) AS max_length,
AVG(length) AS avg_length,
MIN(replacement_cost) AS min_replacement_cost,
MAX(replacement_cost) AS max_replacement_cost,
AVG(replacement_cost) AS avg_replacement_cost
FROM film
```



--descriptive statistics for customer TBL

```
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,
MIN(create_date) AS min_create_date,
MAX(create_date) AS max_create_date,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MIN(last_update) AS min_last_update,
MAX(last_update) AS max_last_update,
MODE() WITHIN GROUP (ORDER BY last_update) AS last_update,
MODE() WITHIN GROUP (ORDER BY first_name) AS first_name,
MODE() WITHIN GROUP (ORDER BY last_name) AS last_name,
MODE() WITHIN GROUP (ORDER BY email) AS email,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MODE() WITHIN GROUP (ORDER BY active) AS mode_active
FROM customer
```



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.
    - Based on my prior experience with data profiling in Excel and now SQL, I feel that SQL if more effective for data profiling. SQL works well with large amounts of data where Excel does not. SQL results / output faster if using correct functions/syntax, results are returned faster and easier to modify, update and store.