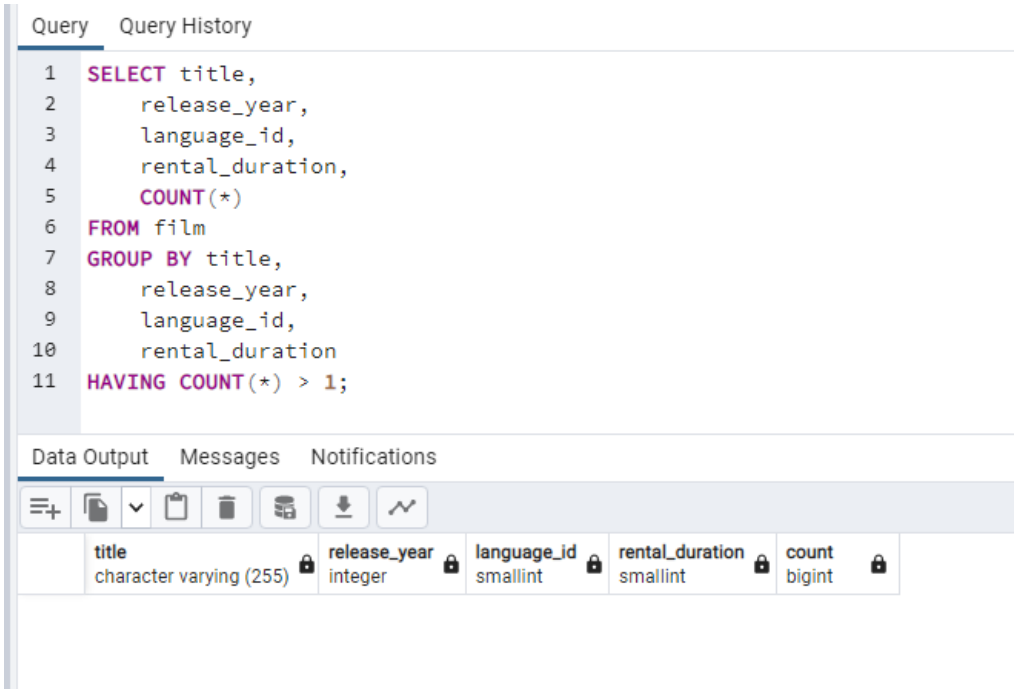
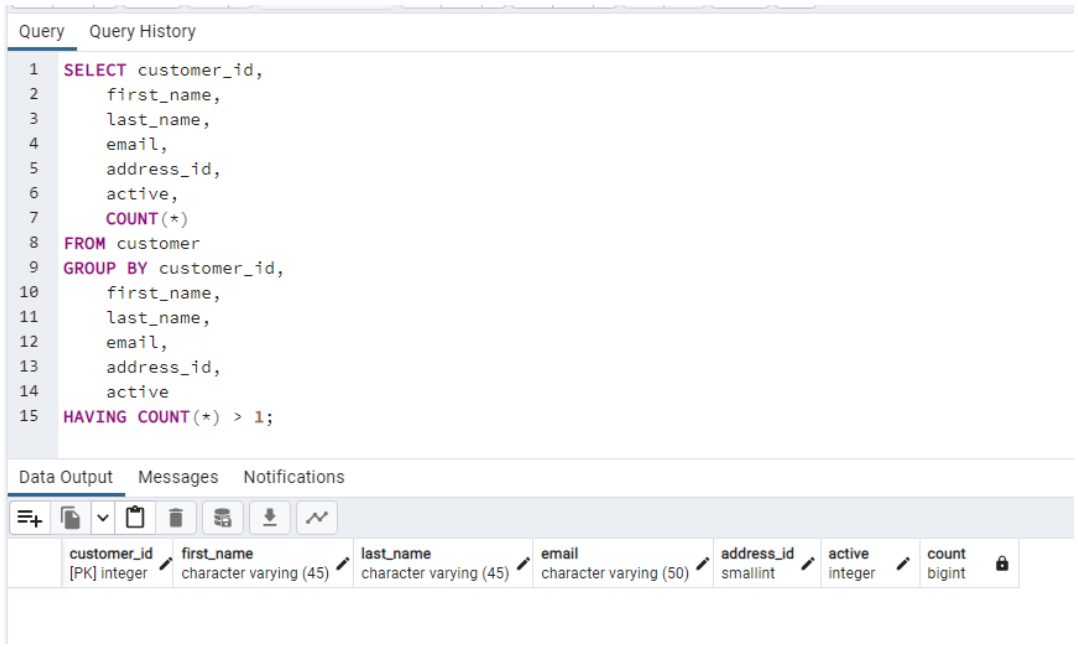
William Su

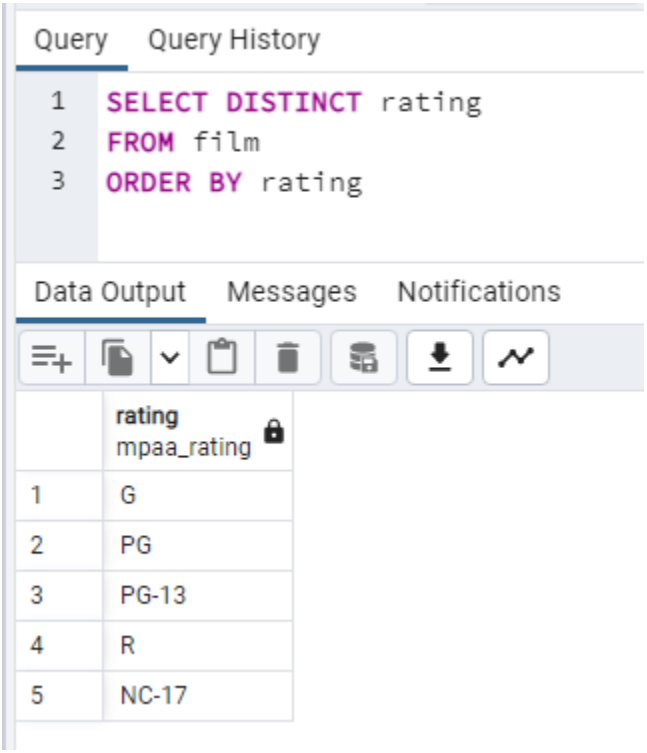
Task 3.6 – Summarizing and Cleaning Data in SQL

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).





There are no duplicates in either data table. In the case that there was “dirty” data, I would clean the data by either creating a secondary virtual table, or a view, that selects only the unique data, or I would delete the duplicate records from the table.



There was no non-uniform data in the film table. If there was, this could be corrected by using the UPDATE, SET, WHERE commands.

In the case of missing data, it would depend on how much of the data is missing, and whether or not the data can be substituted. For example, if a large amount of data values is missing, the column could be ignored when performing queries. If the data is numerical, then an average can be imputed instead.

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.

```
SELECT MIN(film_id) AS minimum_film_id,
        MAX(film_id) AS maximum_film_id,
        AVG(film_id) AS average_film_id,
        MIN(release_year) AS minimum_release_year,
        MAX(release_year) AS maximum_release_year,
        AVG(release_year) AS average_release_year,
        MIN(language_id) AS minimum_language_id,
        MAX(language_id) AS maximum_language_id,
        AVG(language_id) AS average_language_id,
        MIN(rental_duration) AS min_rental_duration,
        MAX(rental_duration) AS max_rental_duration,
        AVG(rental_duration) AS avg_rental_duration,
        MIN(rental_rate) AS min_rental_rate,
        MAX(rental_rate) AS max_rental_rate,
        AVG(rental_rate) AS avg_rental_rate,
        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,
        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,
        MODE() WITHIN GROUP (ORDER BY special_features) AS modal_specials,
        MODE() WITHIN GROUP (ORDER BY fulltext) AS modal_fulltext
FROM film
```

Query | Query History | Scratch Pad

```sql
SELECT MIN(film_id) AS minimum_film_id,
    MAX(film_id) AS maximum_film_id,
    AVG(film_id) AS average_film_id,
    MIN(release_year) AS minimum_release_year,
    MAX(release_year) AS maximum_release_year,
    AVG(release_year) AS average_release_year,
    MIN(language_id) AS minimum_language_id,
    MAX(language_id) AS maximum_language_id,
    AVG(language_id) AS average_language_id,
    MIN(rental_duration) AS min_rental_duration,
    MAX(rental_duration) AS max_rental_duration,
    AVG(rental_duration) AS avg_rental_duration,
    MIN(rental_rate) AS min_rental_rate,
    MAX(rental_rate) AS max_rental_rate,
    AVG(rental_rate) AS avg_rental_rate,
    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,
    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,
    MODE() WITHIN GROUP (ORDER BY special_features) AS modal_specials,
    MODE() WITHIN GROUP (ORDER BY fulltext) AS modal_fulltext
FROM film
```

Data Output | Messages | Notifications

	minimum_film_id integer	maximum_film_id integer	average_film_id numeric	minimum_release_year integer	maximum_release_year integer	average_release_year numeric	minimum_language_id smallint	maximum_language_id smallint	average_language_id numeric	min_rental_duration smallint	max_rent smallint
1	1	1000	500.500000000000	2006	2006	2006.000000000000000	1	1	1.000000000000000000	3	

```
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_adddress_id,
        MIN(active) AS min_active,
        MAX(active) AS max_active,
        AVG(active) AS avg_active,
        MIN(create_date) AS min_create_date,
        MAX(create_date) AS max_create_date,
        MODE() WITHIN GROUP (ORDER BY create_date) AS modal_create_date,
        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,
        MODE() WITHIN GROUP (ORDER BY activebool) AS modal_activebool
FROM customer
```

```
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_adddress_id,
    MIN(active) AS min_active,
    MAX(active) AS max_active,
    AVG(active) AS avg_active,
    MIN(create_date) AS min_create_date,
    MAX(create_date) AS max_create_date,
    MODE() WITHIN GROUP (ORDER BY create_date) AS modal_create_date,
    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,
    MODE() WITHIN GROUP (ORDER BY activebool) AS modal_activebool
FROM customer
```

Query | Query History | Scratch Pad

Data Output | Messages | Notifications

	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_adddress_id numeric	min_active integer	max_active integer	avg_active numeric	min_create_date date	max_create_date date	modal_create_d date
1	1	599	300.000000000000	1	2	1.45575959933	5	605	304.724540901502	0	1	0.9749582637	2006-02-14	2006-02-14	2006-02-14

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 seems to have felt faster while trying to come up with data summarizing. While working on problem 2, it felt very time-consuming typing up the same query for a different column over and over again, as well as including an alias for all of them. I do understand, however, that it does take some time and practice to be able to use SQL quickly and efficiently, and it is much faster than Excel when working with large amounts of data.