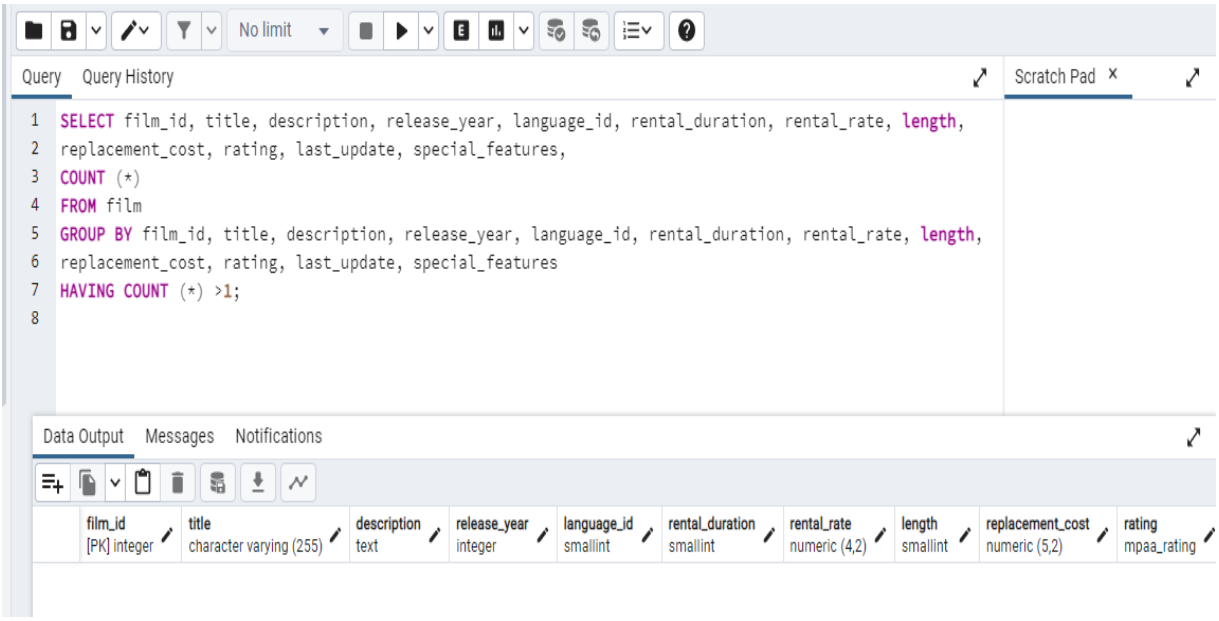
3.6 Summarizing & Cleaning Data in SQL

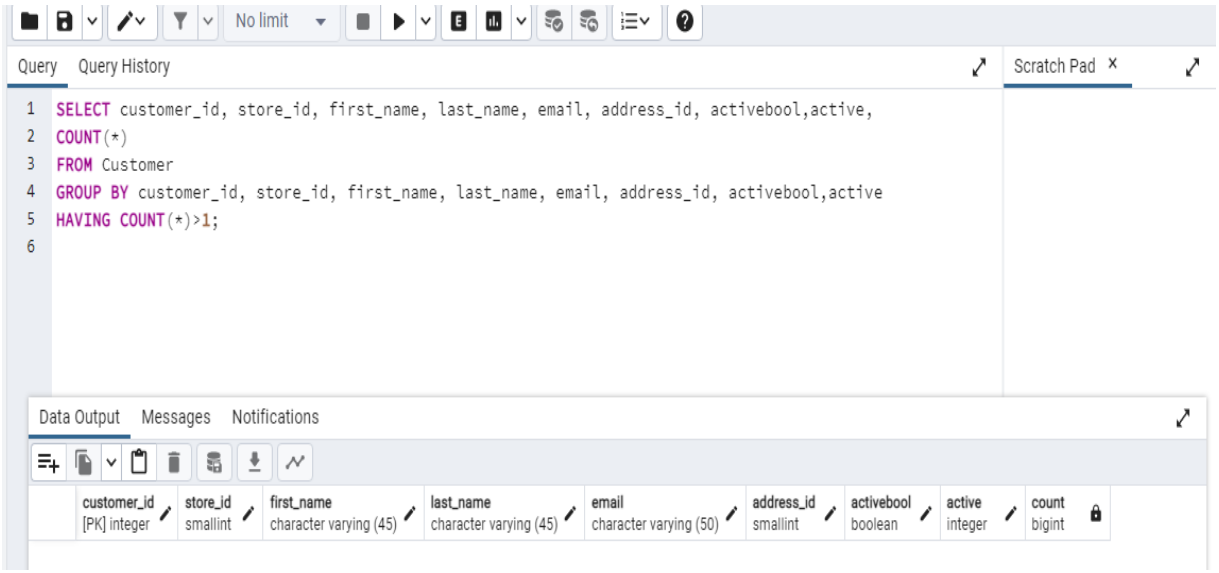
Q1. Check for and clean dirty data:

1.1 Duplicate Data

Duplicate Data Check from Film Table



Duplicate Data Check from Customer table

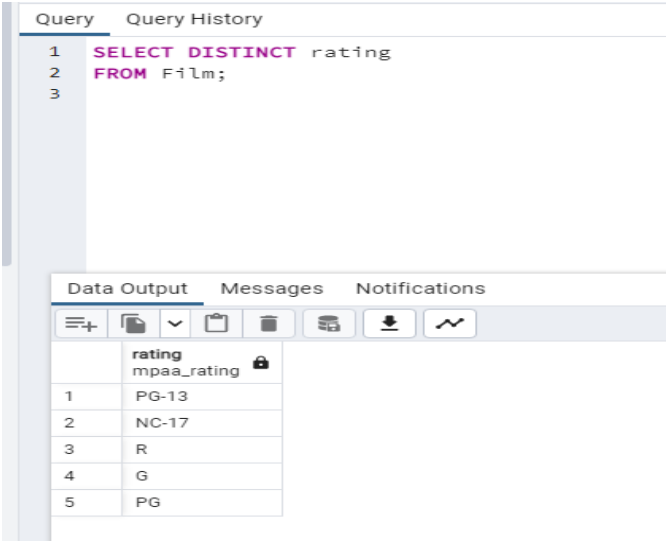


After running the Query, the result is not showing any Duplicate Value. There are two ways of removing duplicate value

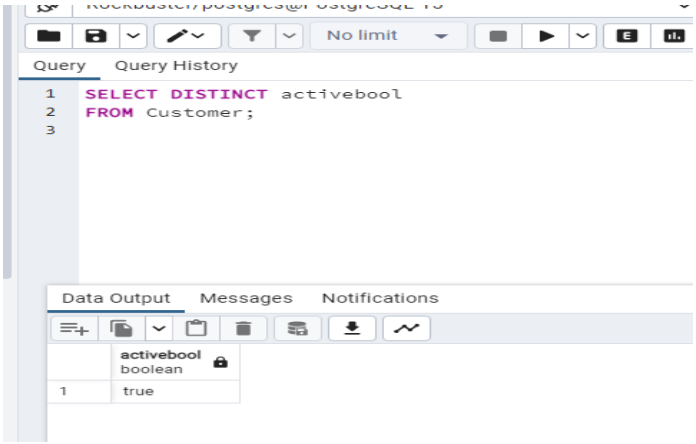
a) Creating a View Table where we can select the Uniques record.
b) Deleting the Duplicate records from the Table.

1.2 Non Uniform Data

Non Uniform Data check of Film table



Non Uniform Data check of Customer table



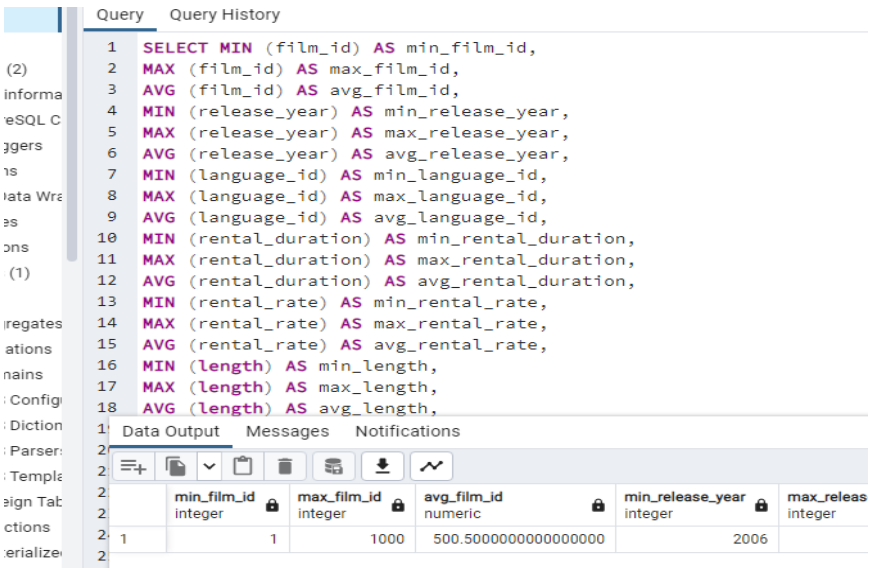
In Non Uniform Data, we can Update the Value. Once we figured out uniformity in data, we can update the value with the help of UPDATE query together with SET and WHERE query.

1.3 Missing Data

Another problem is the missing Data in Database. In this case ,either we ignore or don't use the column that having maximum missing value or we can replace the missing value with Impute Value with average or Mode.

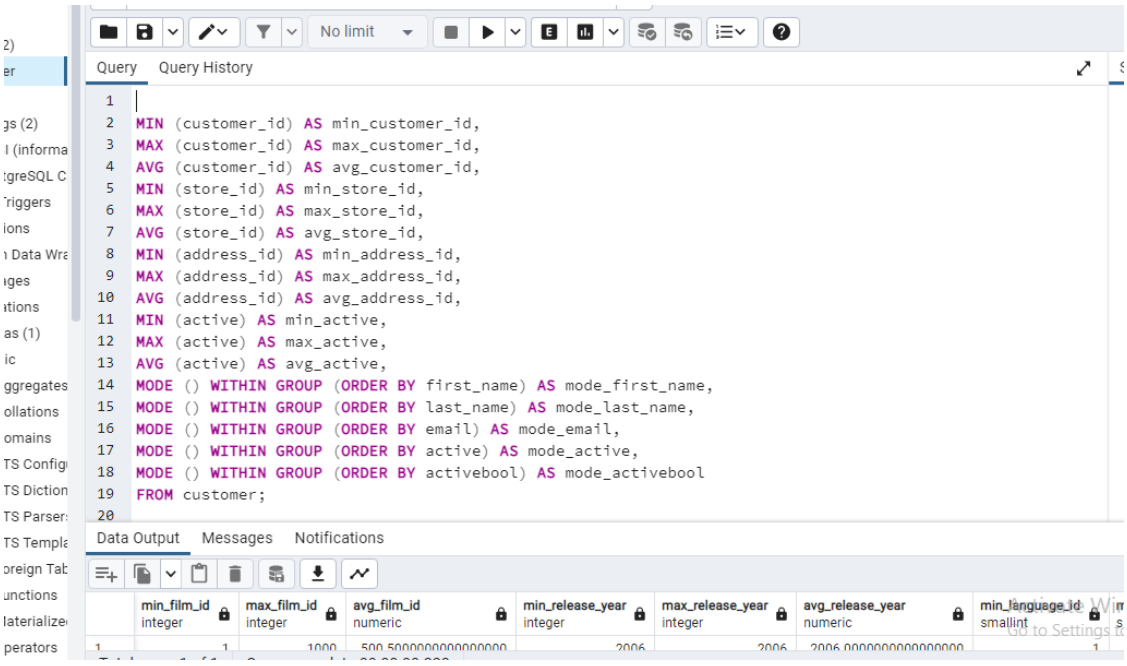
Q2: Summarize your data:
Description Summary of Film Table:

```
SELECT
MIN (film_id) AS min_film_id,
MAX (film_id) AS max_film_id,
AVG (film_id) AS avg_film_id,
MIN (release_year) AS min_release_year,
MAX (release_year) AS max_release_year,
AVG (release_year) AS avg_release_year,
MIN (language_id) AS min_language_id,
MAX (language_id) AS max_language_id,
AVG (language_id) AS avg_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 mode_title,
MODE () WITHIN GROUP (ORDER BY description) AS mode_description,
MODE () WITHIN GROUP (ORDER BY rating) AS mode_rating,
MODE () WITHIN GROUP (ORDER BY special_features) AS mode_special_features,
MODE () WITHIN GROUP (ORDER BY fulltext) AS mode_fulltext
FROM film;
```



Description Summary of Customer Table:

```
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 (active) AS min_active,
MAX (active) AS max_active,
AVG (active) AS avg_active,
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,
MODE () WITHIN GROUP (ORDER BY active) AS mode_active,
MODE () WITHIN GROUP (ORDER BY activebool) AS mode_activebool
FROM customer;
```



Q3 Reflect on the Work

I have worked on both the tool Excel and SQL. I will say, both worked perfectly for data profiling and their effectiveness depend upon the Amount of Data. Like ,For small Data we will consider Excel is best Option and For big data, I recommend SQL. But I also found, that SQL is faster than Excel and give us quick Answers.