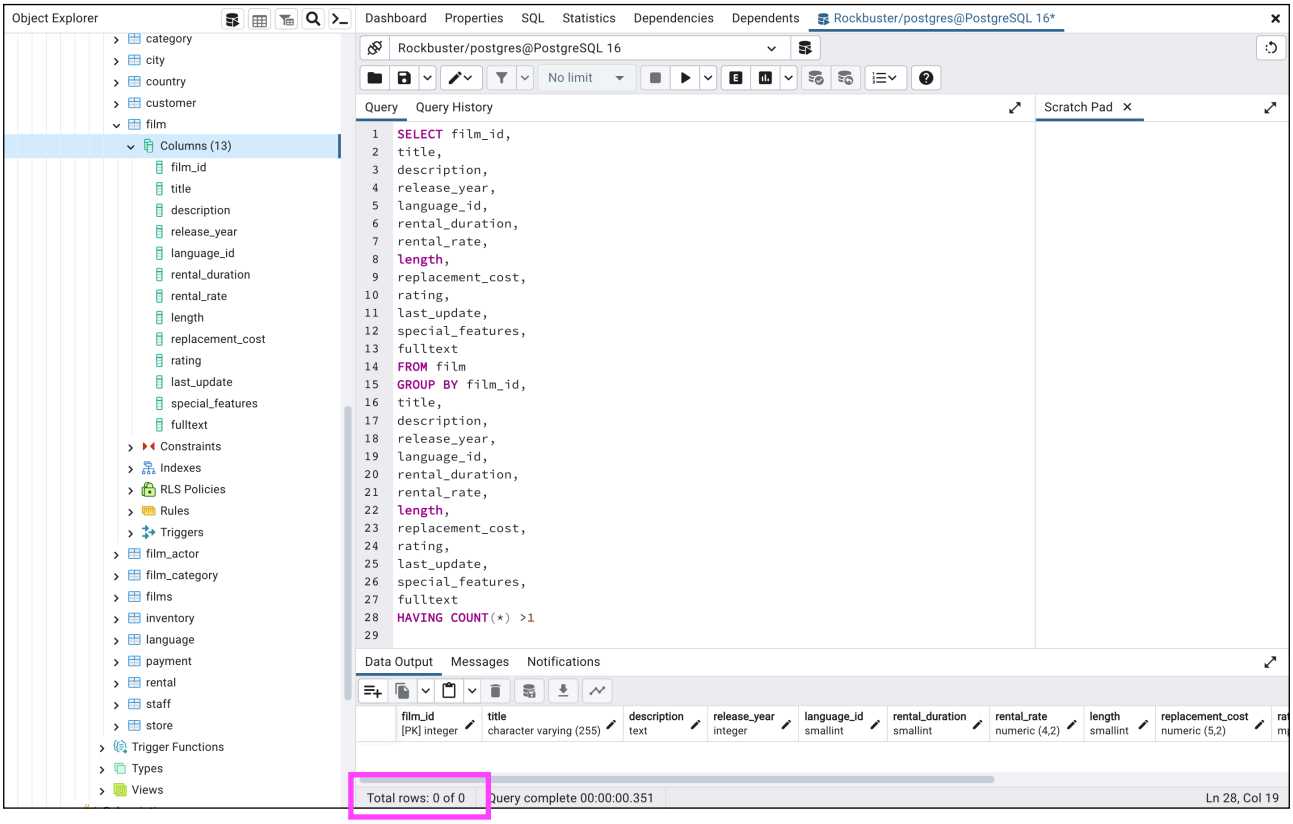
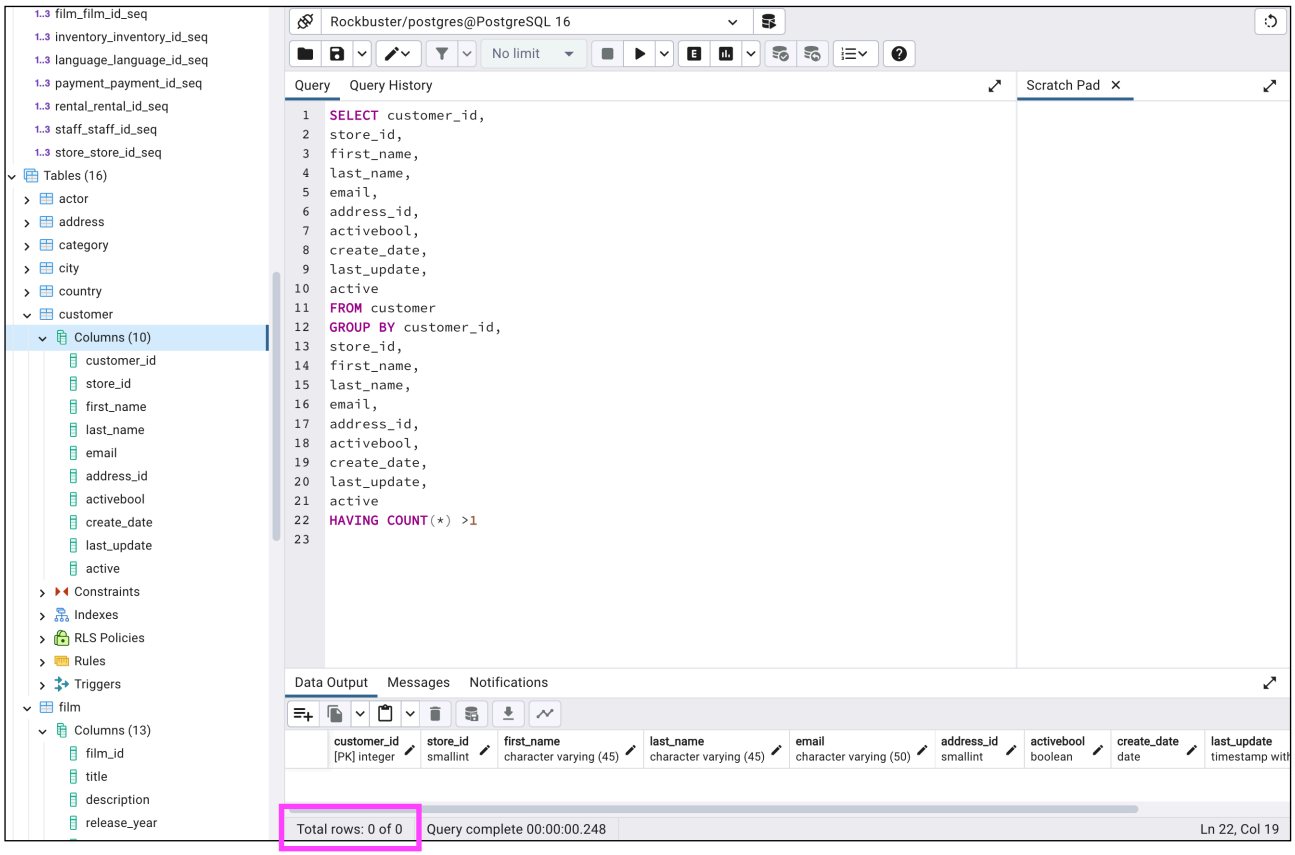
1. **Check for and clean dirty data:** Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values.

DUPLICATE DATA



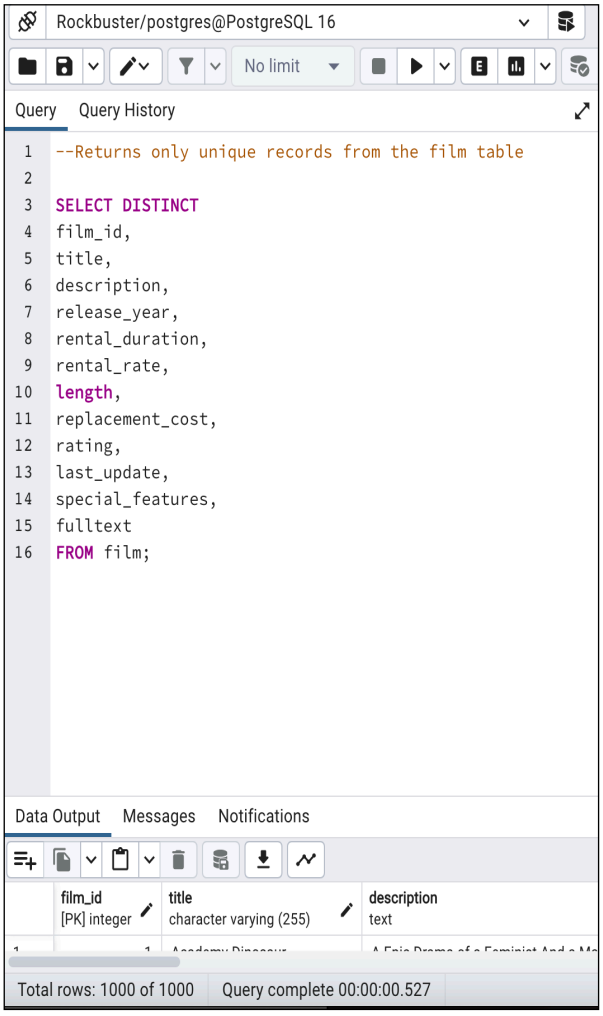


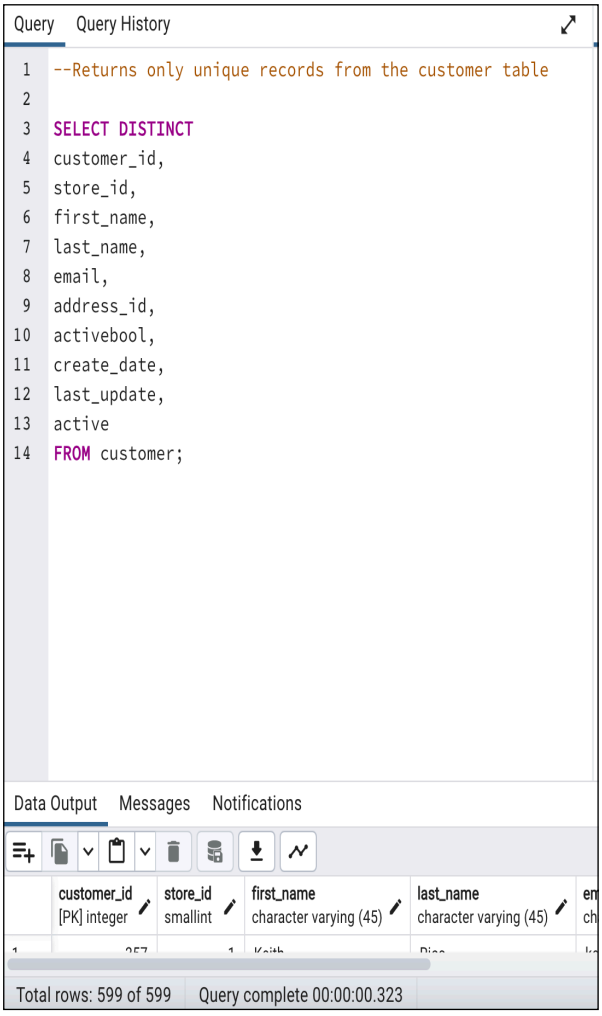
No duplicates were found in either table, however, if there had beed duplicated data, there are two ways to fix them:

1. Create a virtual table, known as a “view,” where you select only unique records.
2. Delete the duplicate record from the table or view.

According to the lesson of this exercise, creating a new view is the standard way for a data analyst to handle duplicate records. On the other hand, deleting data is not recommended and sometimes it is even unauthorised.

NON-UNIFORM

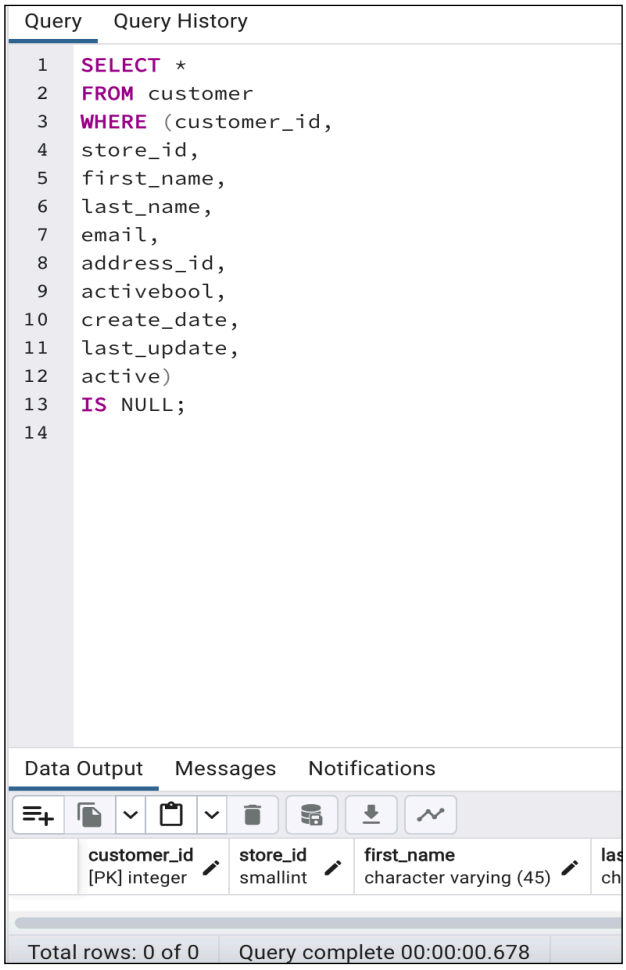


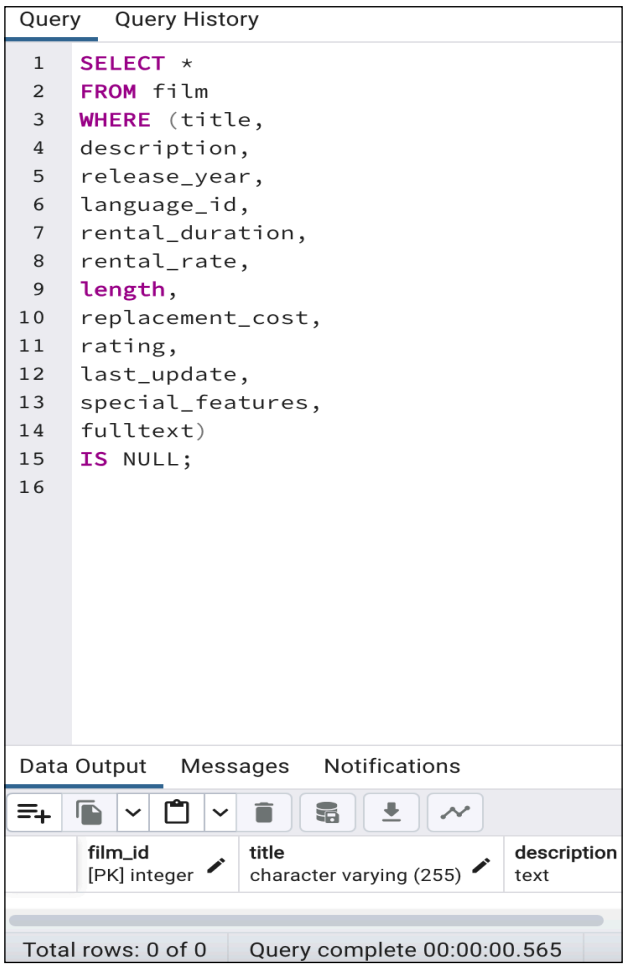


There were no non-uniform data found in either film/customer tables; there is no standard approach to deal with non -uniform data, but in case it had been found, what I would have done to fix that is try to update the values using the following command:

```
UPDATE x
SET x = ‘x’
WHERE x IN (‘x’)…
```

MISSING DATA





There were no missing values found in both tables; but in any case the suggested technique to avoid that is to ignore columns with a high percent of missing values by omitting whichever column you want to ignore, with the following SELECT statement: *ignored in select because it has a lot of missing values.*
The second suggested approach is to impute the missing values using statistical methods such as AVG, MIN, MAX. Using the UPDATE command.

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.

NUMERICAL VALUES FILM COLUMN

- rental_duration
- rental_rate
- Length
- replacement_cost

Query Query History

```sql
SELECT MAX (rental_duration) AS max_rental_duration,
       MIN (rental_duration) AS min_rental_duration,
       AVG (rental_duration) AS avg_rental_duration,
       MAX (rental_rate) AS max_rental_rate,
       MIN (rental_rate) AS min_rental_rate,
       AVG (rental_rate) AS avg_rental_rate,
       MAX (length) AS max_length,
       MIN (length) AS min_length,
       AVG (length) AS avg_length,
       MAX (replacement_cost) AS max_replacement_cost,
       MIN (replacement_cost) AS min_replacement_cost,
       AVG (replacement_cost) AS avg_replacement_cost
FROM film;
```

	max_rental_duration smallint	min_rental_duration smallint	avg_rental_duration numeric
1	7	3	4.9850000000000000

max_rental_rate numeric	min_rental_rate numeric	avg_rental_rate numeric
4.99	0.99	2.9800000000000000

max_length smallint	min_length smallint	avg_length numeric
185	46	115.2720000000000000

max_replacement_cost numeric	min_replacement_cost numeric	avg_replacement_cost numeric
29.99	9.99	19.9840000000000000

NON-NUMERICAL VALUES FILM TABLE

- film_id
- Title
- language_id
- Rating
- last_update
- special_features
- Fulltext

```sql
SELECT MODE () WITHIN GROUP (ORDER BY film_id) AS film_id_modal_value,
MODE () WITHIN GROUP (ORDER BY title) AS title_modal_value,
MODE () WITHIN GROUP (ORDER BY language_id) AS language_id_modal_value,
MODE () WITHIN GROUP (ORDER BY rating) AS rating_modal_value,
MODE () WITHIN GROUP (ORDER BY last_update) AS last_update_modal_value,
MODE () WITHIN GROUP (ORDER BY special_features) AS special_features_modal_value,
MODE () WITHIN GROUP (ORDER BY fulltext) AS fulltext_modal_value
FROM film;
```

	film_id_modal_value integer	title_modal_value character varying	language_id_modal_value smallint	rating_modal_value mpaa_rating	last_update_modal_value timestamp without time zone
1	1	Academy Dinosaur	1	PG-13	2013-05-26 14:50:58.951

special_features_modal_value text[]	fulltext_modal_value tsvector
{Trailers,Commentaries,"Behind the Scenes"}	'baloon':19 'confront':14 'documentari':5 'feminist':8,11,16 'mile':2 'must':13 'spi':1 'thri…

NON-NUMERICAL VALUES CUSTOMER TABLE

*(no numerical values for the customer table)

- store_id
- Activebool
- create_date
- last_update

```
SELECT MODE () WITHIN GROUP (ORDER BY store_id) AS store_id_modal_value,
MODE () WITHIN GROUP (ORDER BY activebool) AS activebool_modal_value,
MODE () WITHIN GROUP (ORDER BY create_date) AS create_date_modal_value,
MODE () WITHIN GROUP (ORDER BY last_update) AS last_update_modal_value
FROM customer;
```

Data Output | Messages | Notifications

	store_id_modal_value smallint	activebool_modal_value boolean	create_date_modal_value date	last_update_modal_value timestamp without time zone
1	1	true	2006-02-14	2013-05-26 14:49:45.738

3. 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.

Neither is absolutely better than the other because each is designed and intended to be used in different records keeping and manipulation situations, but in this case, in my opinion, once you learn how to structure queries on SQL, it might be faster than the long step-by-step guide for doing this in Excel: Import data, sort data, filter certain values, delete unwanted rows, and so on.