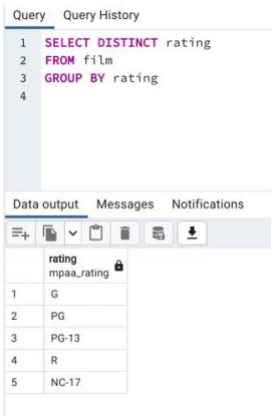
3.6 **Answers-** Summarizing & Cleaning

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.

   Finding non-uniform data

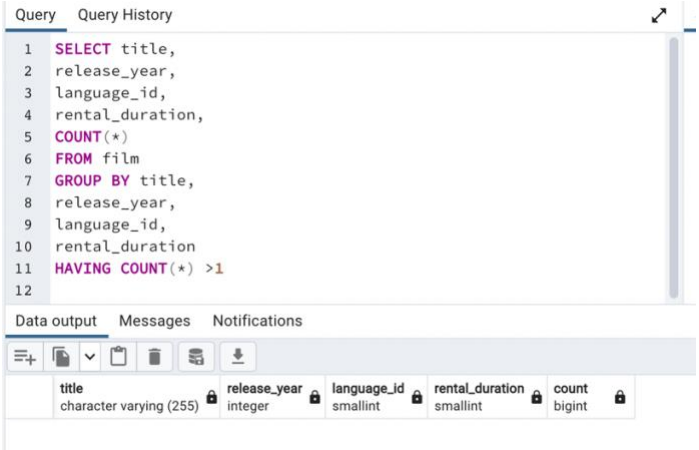
   ```
   SELECT DISTINCT rating
   FROM film
   GROUP BY rating
   ```



   - Review a few random values to check for inconsistencies using the `GROUP BY` and `DISTINCT` keywords.

Finding Duplicates in film table

```
SELECT title,
release_year,
language_id,
rental_duration,
COUNT(*)
FROM film
GROUP BY title,
release_year,
language_id,
rental_duration
HAVING COUNT(*) >1
```



Finding Duplicates in Customer table

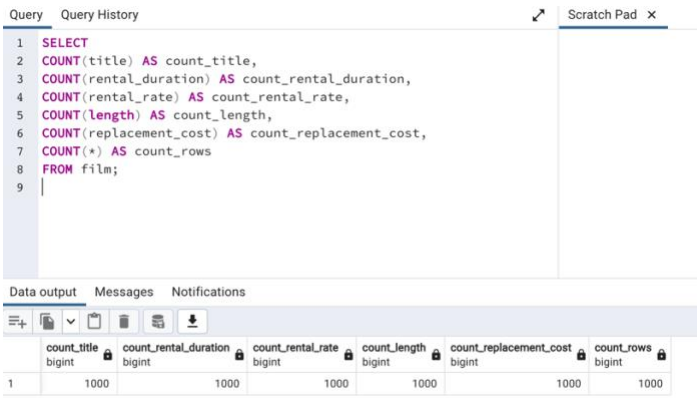
```
SELECT first_name,
last_name,
address_id,
email,
active,
COUNT(*)
FROM customer
GROUP BY first_name,
last_name,
address_id,
email,
active
HAVING COUNT(*) >1;
```

- **Duplicate – There are 2 options.** Create a virtual table, known as a "view," where you select only unique records or delete the duplicate record from the table or view (not advised).

Finding Missing Values in film table

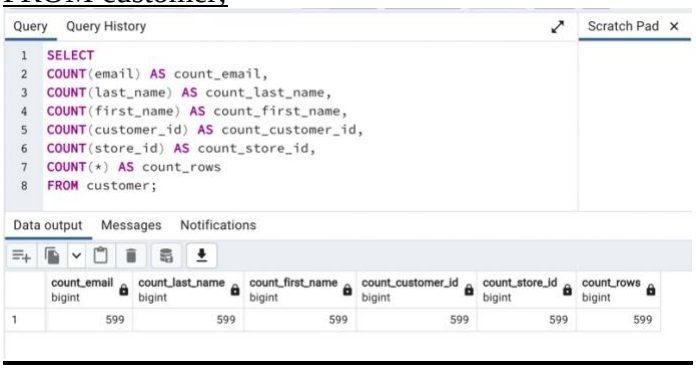
SELECT
COUNT(title) AS count_title,
COUNT(rental_duration) AS count_rental_duration,
COUNT(rental_rate) AS count_rental_rate,

COUNT(length) AS count_length,
COUNT(replacement_cost) AS count_replacement_cost,
COUNT(*) AS count_rows
FROM film;



Finding Missing Values in customer table

SELECT
COUNT(email) AS count_email,
COUNT(last_name) AS count_last_name,
COUNT(first_name) AS count_first_name,
COUNT(customer_id) AS count_customer_id,
COUNT(store_id) AS count_store_id,
Answers 3.6 2
COUNT(*) AS count_rows
FROM customer;



- Missing Values - We can either ignore columns with a high percentage of missing values or *impute* missing values with a column average: Ex.
  imputing missing values with the AVG value
  UPDATE tablename
  SET = AVG(col1)
  WHERE col1 IS NULL

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,

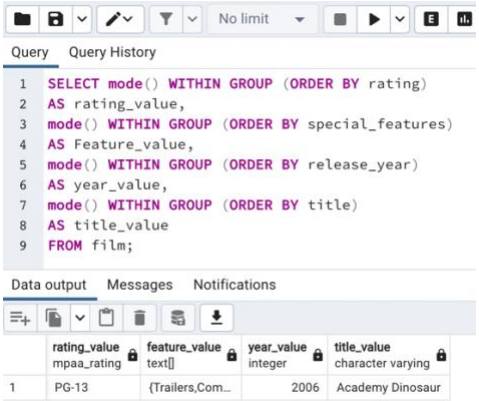
maximum, and average values. For non-numerical columns, calculate the mode value.

Descriptive Stats for film table (numerical columns)

```
SELECT MIN(rental_rate) AS min_renatl_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
```

Mode value for film table (non-numerical)

```
SELECT mode() WITHIN GROUP (ORDER BY rating)
AS rating_value,
mode() WITHIN GROUP (ORDER BY special_features)
AS Feature_value,
mode() WITHIN GROUP (ORDER BY release_year)
AS year_value,
mode() WITHIN GROUP (ORDER BY title)
AS title_value
FROM film;
```



```
SELECT mode() WITHIN GROUP (ORDER BY rating)
AS rating_value,
mode() WITHIN GROUP (ORDER BY special_features)
AS Feature_value,
mode() WITHIN GROUP (ORDER BY release_year)
AS year_value,
mode() WITHIN GROUP (ORDER BY title)
AS title_value
FROM film;
```

	rating_value mpaa_rating	feature_value text[]	year_value integer	title_value character varying
1	PG-13	{Trailers,Com...	2006	Academy Dinosaur

Descriptive Stats for customer table(numerical columns)

```
SELECT MIN(active) AS min active,
MAX(active) AS max active,
AVG(active) AS avg active,
MIN(address id) AS min address,
MAX(address id) AS max address,
AVG(address id) AS avg address,
MIN(customer id) AS min customer,
MAX(customer id) AS max customer,
AVG(customer id) AS avg customer,
MIN(store id) AS min store,
MAX(store id) AS max store,
AVG(store id) AS avg store
FROM customer;
```

```
MIN(address_id) AS min_address,
MAX(address_id) AS max_address,
AVG(address_id) AS avg_address,
MIN(customer_id) AS min_customer,
MAX(customer_id) AS max_customer,
AVG(customer_id) AS avg_customer,
MIN(store_id) AS min_store,
MAX(store_id) AS max_store,
AVG(store_id) AS avg_store
FROM customer;
```

	min_active integer	max_active integer	avg_active numeric	min_address smallint	max_address smallint	avg_address numeric	min_customer integer	max_customer integer	avg_customer numeric	min_store smallint	max_store smallint	avg_store numeric
1	0	1	0.974958263	5	605	304.724540901	1	599	300	1	2	1.455759599

Mode value for customer table(numerical column)

```
SELECT mode() WITHIN GROUP (ORDER BY first_name)
AS first_name_value,
mode() WITHIN GROUP (ORDER BY last_name)
AS last_name_value,
mode() WITHIN GROUP (ORDER BY email)
AS email_value
FROM customer;
```

Query Query History

```sql
SELECT mode() WITHIN GROUP (ORDER BY first_name)
AS first_name_value,
mode() WITHIN GROUP (ORDER BY last_name)
AS last_name_value,
mode() WITHIN GROUP (ORDER BY email)
AS email_value
FROM customer;
```

Data output Messages Notifications

	first_name_value character varying	last_name_value character varying	email_value character varying
1	Jamie	Abney	aaron.selby@sak...

3. Reflect on your work

Based on your previous experience, which tool (Excel or SQL) do you think is more effective for data profiling, and why?

SQL is more efficient at computing data profiles. Once the code is written, it can be applied to other data sets, it just takes time to get use to and lot’s of trial and error.