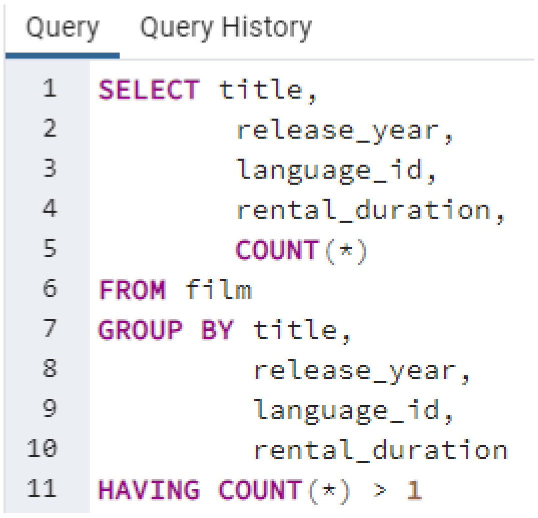
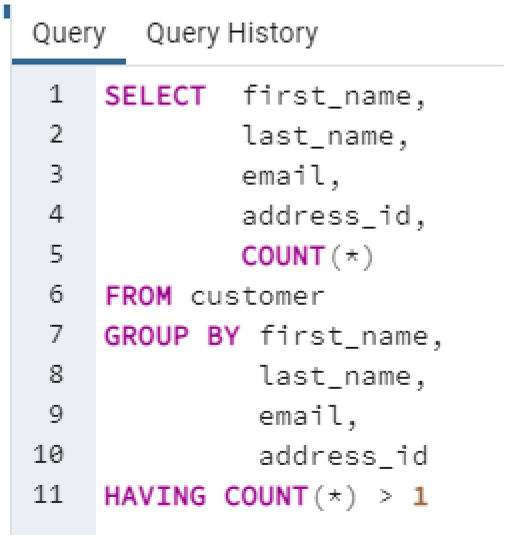
Task 3.6 – Summarizing & Cleaning Data in SQL

1.



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



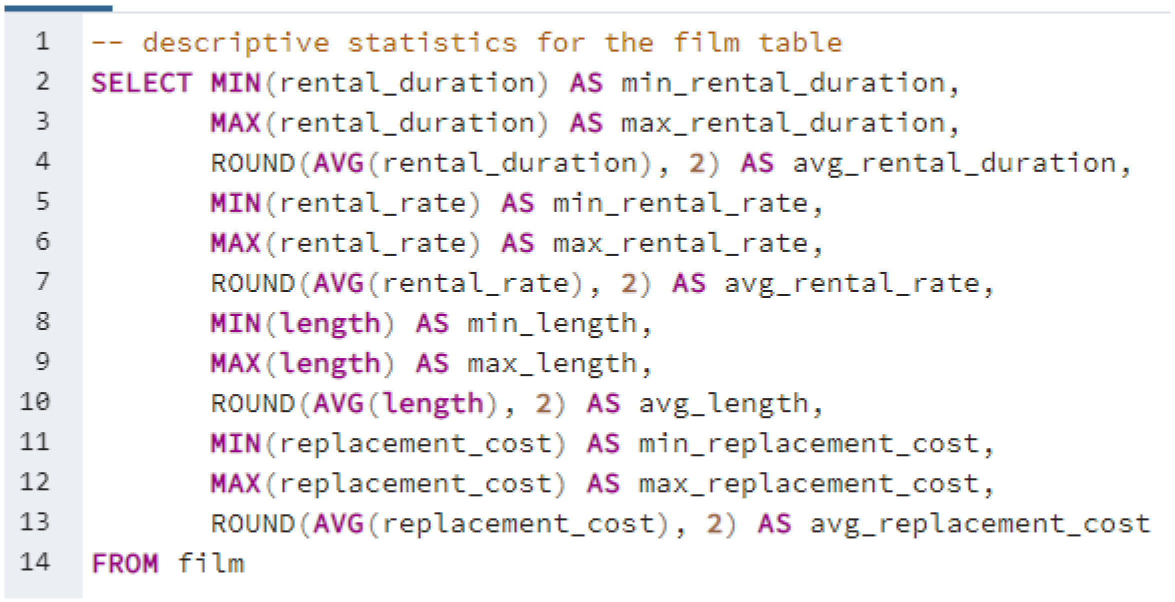
```sql
SELECT  first_name,
        last_name,
        email,
        address_id,
        COUNT(*)
FROM customer
GROUP BY first_name,
         last_name,
         email,
         address_id
HAVING COUNT(*) > 1
```

This query will check for any duplicate data, and in this case, no result set was returned meaning there were no duplicates. If there were duplicate records, there are two ways to fix them.

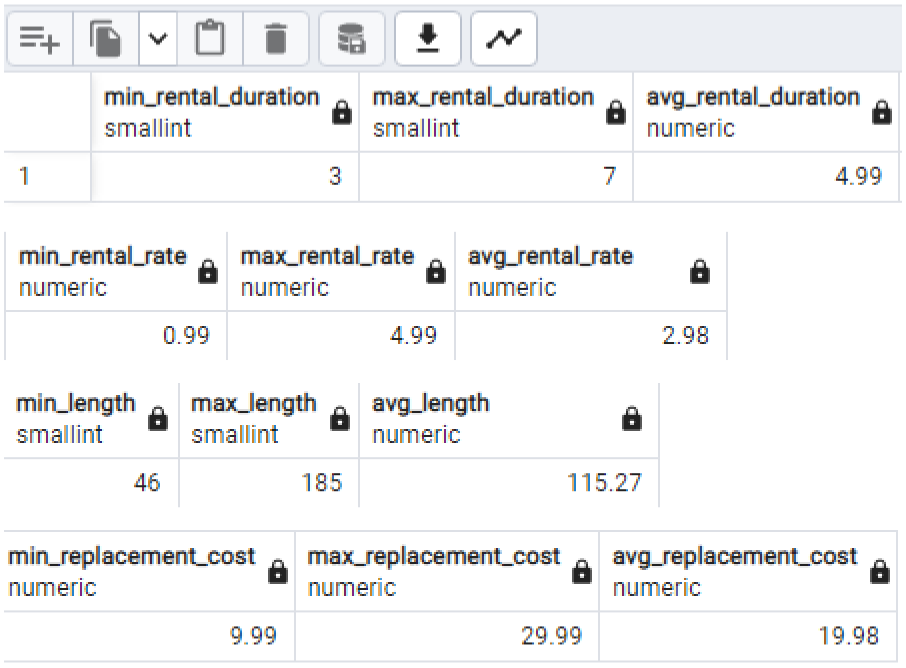
a. I could create a virtual table (VIEW) and select only the unique records.
b. I could delete the duplicate record from the table or view.

We could also use GROUP BY or DISTINCT to find any non-uniform data.

2.



```sql
-- descriptive statistics for the film table
SELECT MIN(rental_duration) AS min_rental_duration,
       MAX(rental_duration) AS max_rental_duration,
       ROUND(AVG(rental_duration), 2) AS avg_rental_duration,
       MIN(rental_rate) AS min_rental_rate,
       MAX(rental_rate) AS max_rental_rate,
       ROUND(AVG(rental_rate), 2) AS avg_rental_rate,
       MIN(length) AS min_length,
       MAX(length) AS max_length,
       ROUND(AVG(length), 2) AS avg_length,
       MIN(replacement_cost) AS min_replacement_cost,
       MAX(replacement_cost) AS max_replacement_cost,
       ROUND(AVG(replacement_cost), 2) AS avg_replacement_cost
FROM film
```

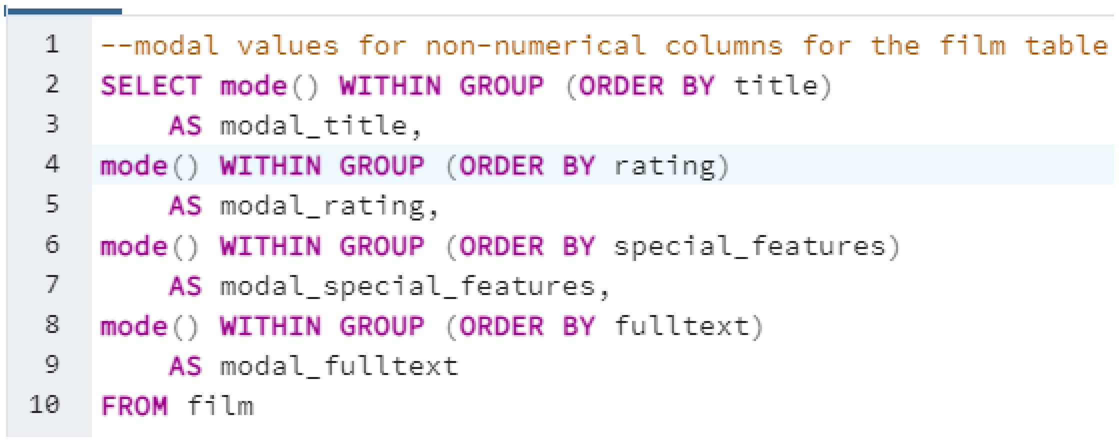


	min_rental_duration smallint	max_rental_duration smallint	avg_rental_duration numeric
1	3	7	4.99

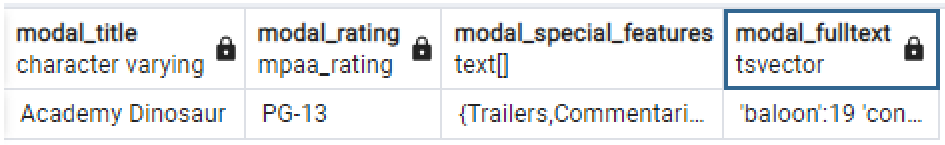
min_rental_rate numeric	max_rental_rate numeric	avg_rental_rate numeric
0.99	4.99	2.98

min_length smallint	max_length smallint	avg_length numeric
46	185	115.27

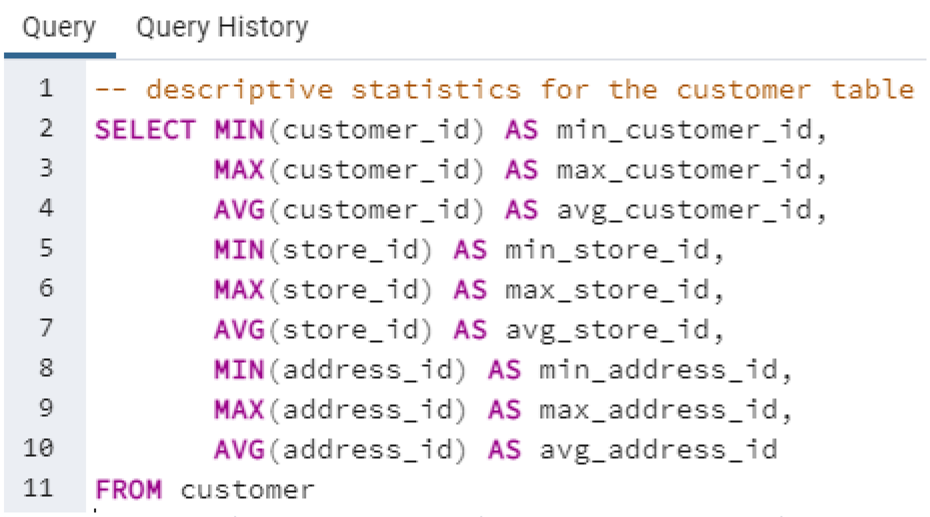
min_replacement_cost numeric	max_replacement_cost numeric	avg_replacement_cost numeric
9.99	29.99	19.98



```sql
--modal values for non-numerical columns for the film table
SELECT mode() WITHIN GROUP (ORDER BY title)
    AS modal_title,
mode() WITHIN GROUP (ORDER BY rating)
    AS modal_rating,
mode() WITHIN GROUP (ORDER BY special_features)
    AS modal_special_features,
mode() WITHIN GROUP (ORDER BY fulltext)
    AS modal_fulltext
FROM film
```

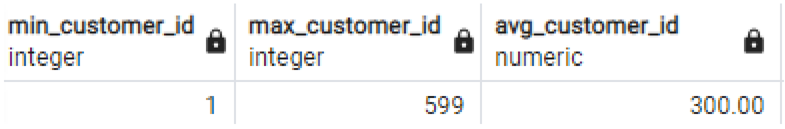


modal_title character varying	modal_rating mpaa_rating	modal_special_features text[]	modal_fulltext tsvector
Academy Dinosaur	PG-13	{Trailers,Commentari...	'baloon':19 'con...

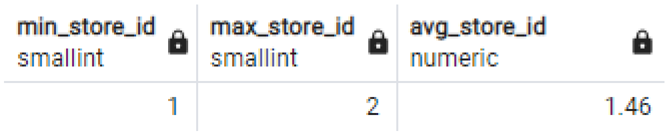


Query Query History

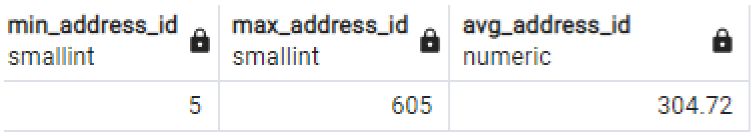
```sql
-- descriptive statistics for the customer table
SELECT MIN(customer_id) AS min_customer_id,
       MAX(customer_id) AS max_customer_id,
       AVG(customer_id) AS avg_customer_id,
       MIN(store_id) AS min_store_id,
       MAX(store_id) AS max_store_id,
       AVG(store_id) AS avg_store_id,
       MIN(address_id) AS min_address_id,
       MAX(address_id) AS max_address_id,
       AVG(address_id) AS avg_address_id
FROM customer
```



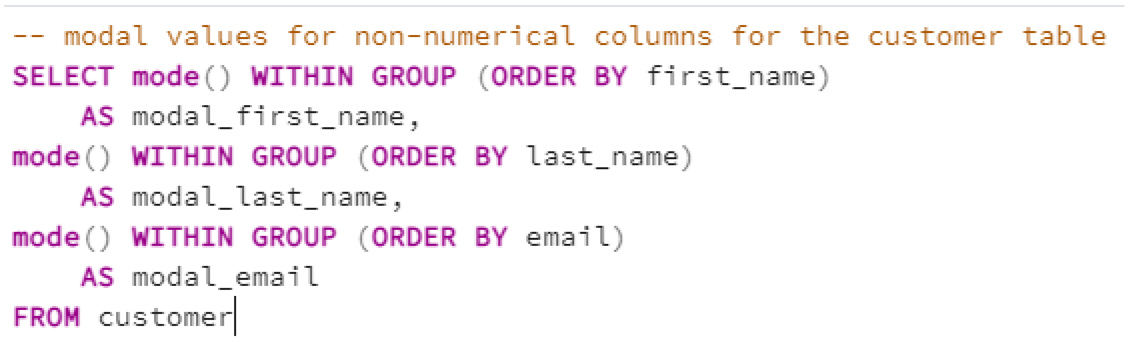
min_customer_id integer	max_customer_id integer	avg_customer_id numeric
1	599	300.00



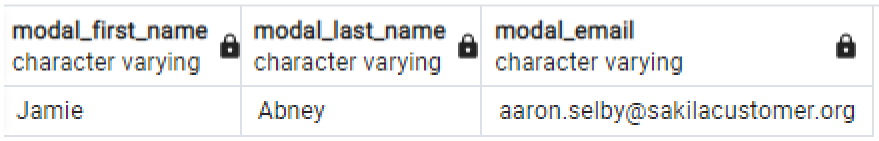
min_store_id smallint	max_store_id smallint	avg_store_id numeric
1	2	1.46



min_address_id smallint	max_address_id smallint	avg_address_id numeric
5	605	304.72



```sql
-- modal values for non-numerical columns for the customer table
SELECT 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
FROM customer
```



modal_first_name character varying	modal_last_name character varying	modal_email character varying
Jamie	Abney	aaron.selby@sakilacustomer.org

3. I lean more towards SQL just because I find it quicker to query or to filter data. There are a ton of commands that can be used, or a combination of commands that can be used to extract exactly what you need. In addition, you can easily save it somewhere and type a new query. Although Excel is more user-friendly at first, I think SQL is more efficient in the long run.