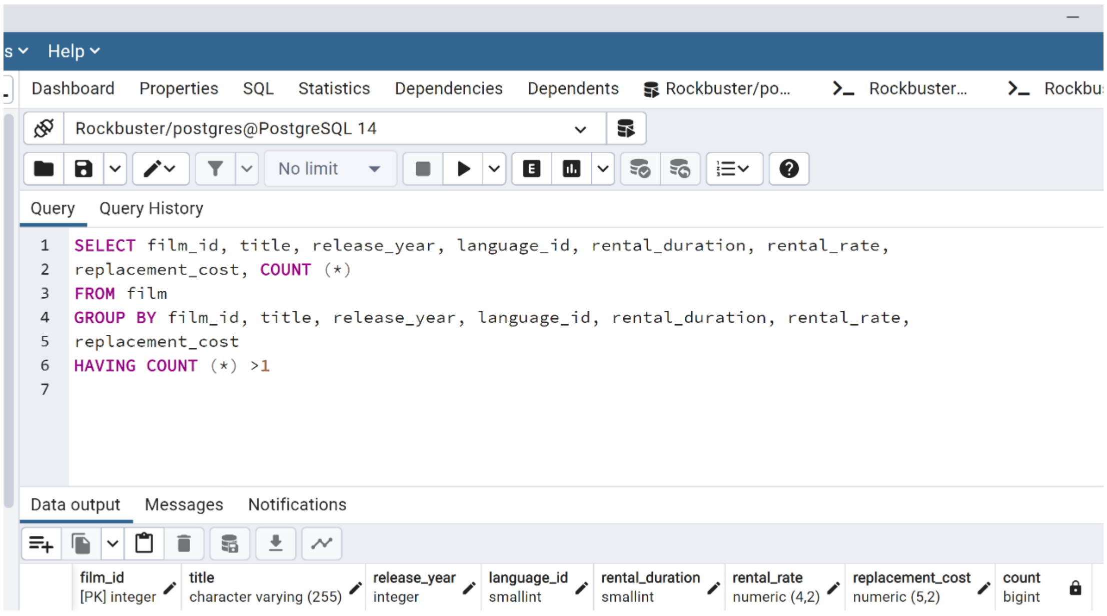
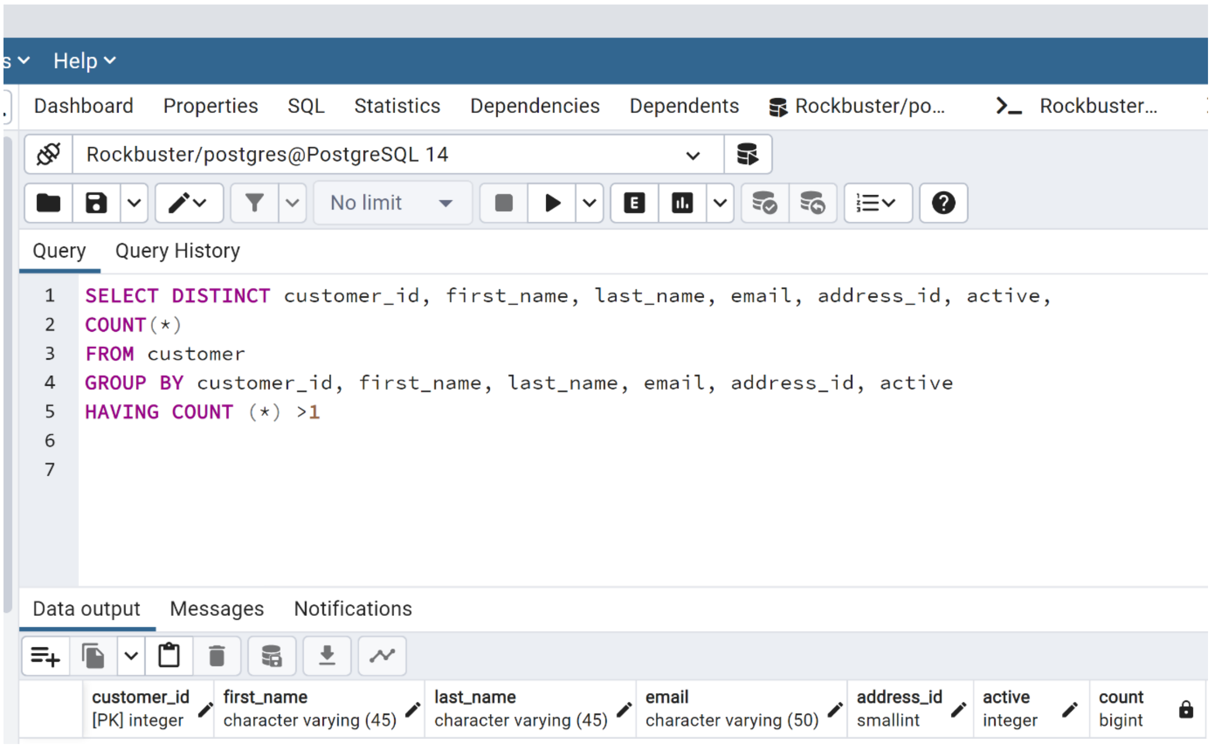
Summarizing & Cleaning Data in SQL

1. **Check for and clean dirty data:**



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

film_id [PK] integer	title character varying (255)	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	replacement_cost numeric (5,2)	count bigint

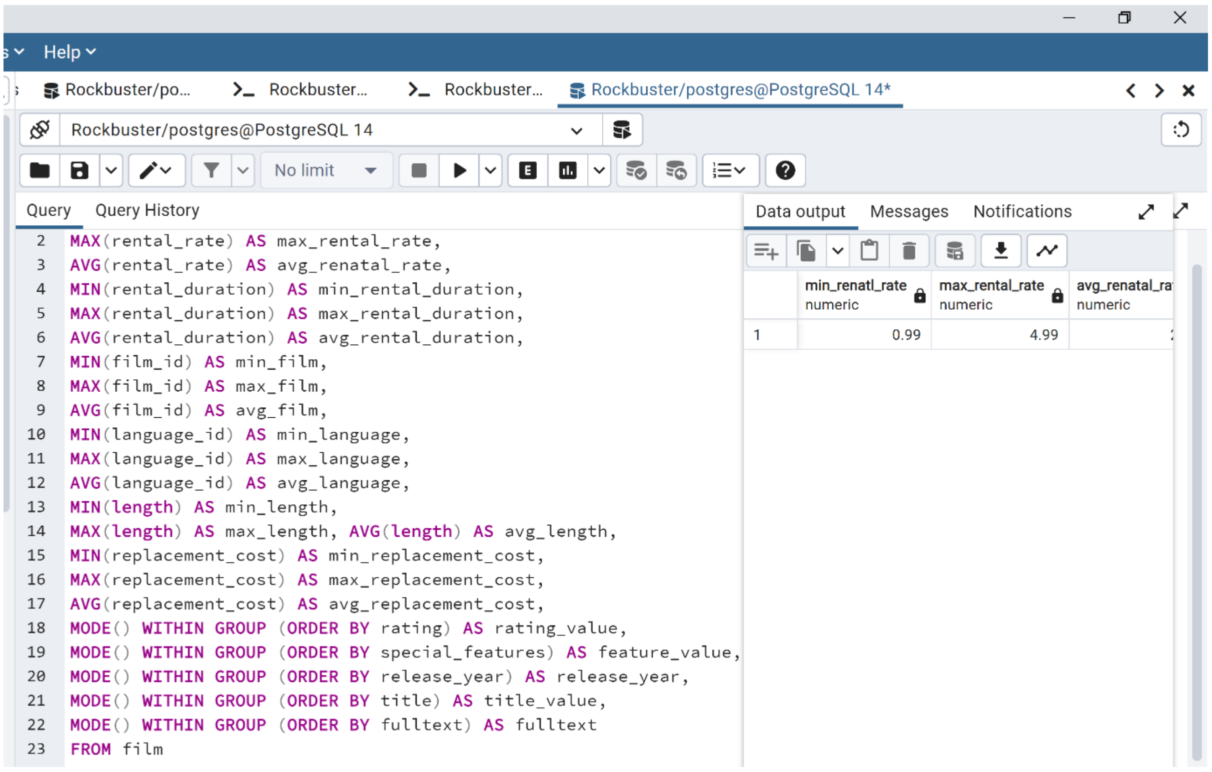


```
SELECT DISTINCT customer_id, first_name, last_name, email, address_id, active,
COUNT(*)
FROM customer
GROUP BY customer_id, first_name, last_name, email, address_id, active
HAVING COUNT (*) >1
```

customer_id [PK] integer	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	active integer	count bigint

There is no returned duplicate value. Duplicate data can be solved by deleting the records if permissible or using a GROUP BY or DISTINCT to select required data.

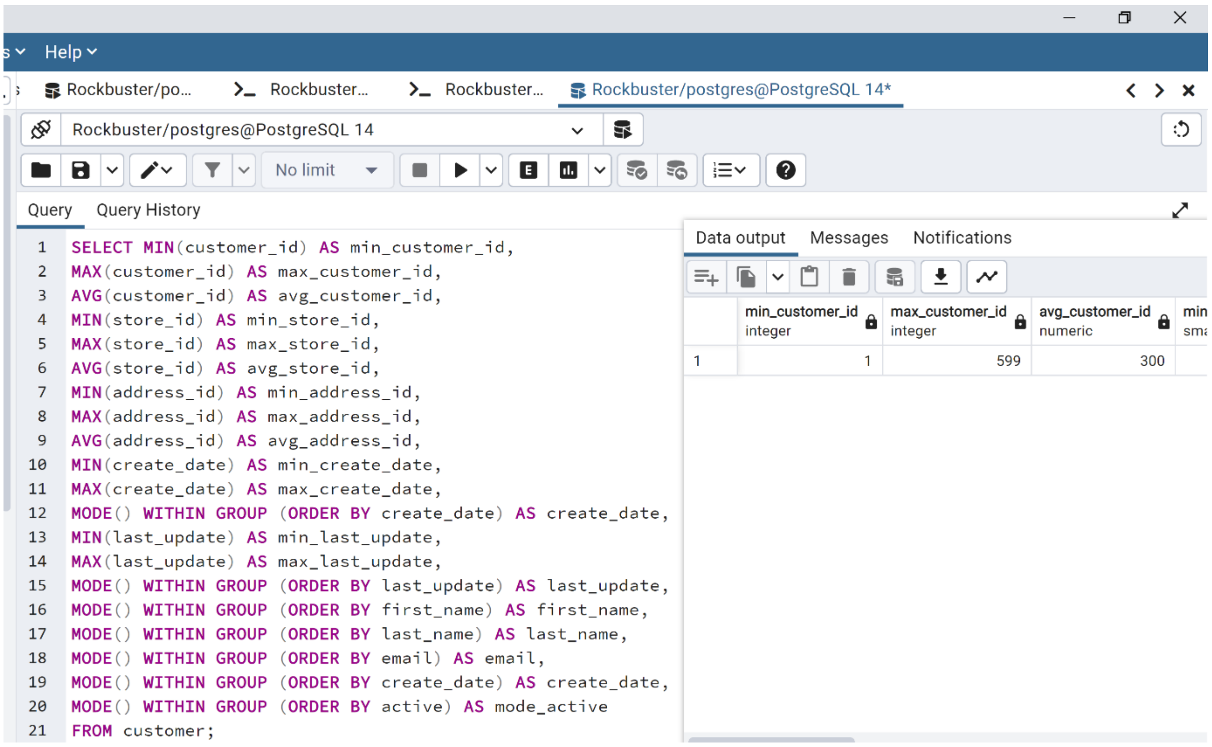
2. **Summarize your data:**



```
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,
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 release_year,
MODE() WITHIN GROUP (ORDER BY title) AS title_value,
MODE() WITHIN GROUP (ORDER BY fulltext) AS fulltext
FROM film
```

```
     min_renatl_rate   max_rental_rate   avg_renatal_ra
     numeric           numeric           numeric
1             0.99              4.99
```

min_renatl_rate	0.99
max_rental_rate	4.99
avg_renatal_rate	2.98
min_rental_duration	3
max_rental_duration	7
avg_rental_duration	4.985
min_film	1
max_film	1000
avg_film	500.5
min_language	1
max_language	1
avg_language	1
min_length	46
max_length	185
avg_length	115.272
min_replacement_cost	9.99
max_replacement_cost	29.99
avg_replacement_cost	19.984
rating_value	PG-13
feature_value	{Trailers,Commentaries,"Behind the Scenes"}
release_year	2006
title_value	Academy Dinosaur



Help

Rockbuster/po... Rockbuster... Rockbuster... Rockbuster/postgres@PostgreSQL 14*

Rockbuster/postgres@PostgreSQL 14

No limit

Query Query History

```sql
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(create_date) AS min_create_date,
MAX(create_date) AS max_create_date,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MIN(last_update) AS min_last_update,
MAX(last_update) AS max_last_update,
MODE() WITHIN GROUP (ORDER BY last_update) AS last_update,
MODE() WITHIN GROUP (ORDER BY first_name) AS first_name,
MODE() WITHIN GROUP (ORDER BY last_name) AS last_name,
MODE() WITHIN GROUP (ORDER BY email) AS email,
MODE() WITHIN GROUP (ORDER BY create_date) AS create_date,
MODE() WITHIN GROUP (ORDER BY active) AS mode_active
FROM customer;
```

Data output Messages Notifications

```
   min_customer_id  max_customer_id  avg_customer_id  min
   integer          integer          numeric          sma
1                1              599              300
```

min_customer_id	1
max_customer_id	599
avg_customer_id	300
min_store_id	1
max_store_id	2
avg_store_id	1.455759599
min_address_id	5
max_address_id	605
avg_address_id	304.7245409
min_create_date	14/02/2006
max_create_date	14/02/2006
create_date	14/02/2006
min_last_update	49:45.7
max_last_update	49:45.7
last_update	49:45.7
first_name	Jamie
last_name	Abney
email	aaron.selby@sakilacustomer.org
create_date-2	14/02/2006
mode_active	1

3. **Reflect on your work:** Using Excel would be more effective when dealing with small amounts of data, or few columns within a data table to categorize and data profile. However, large amounts of data need SQL to be more effective.