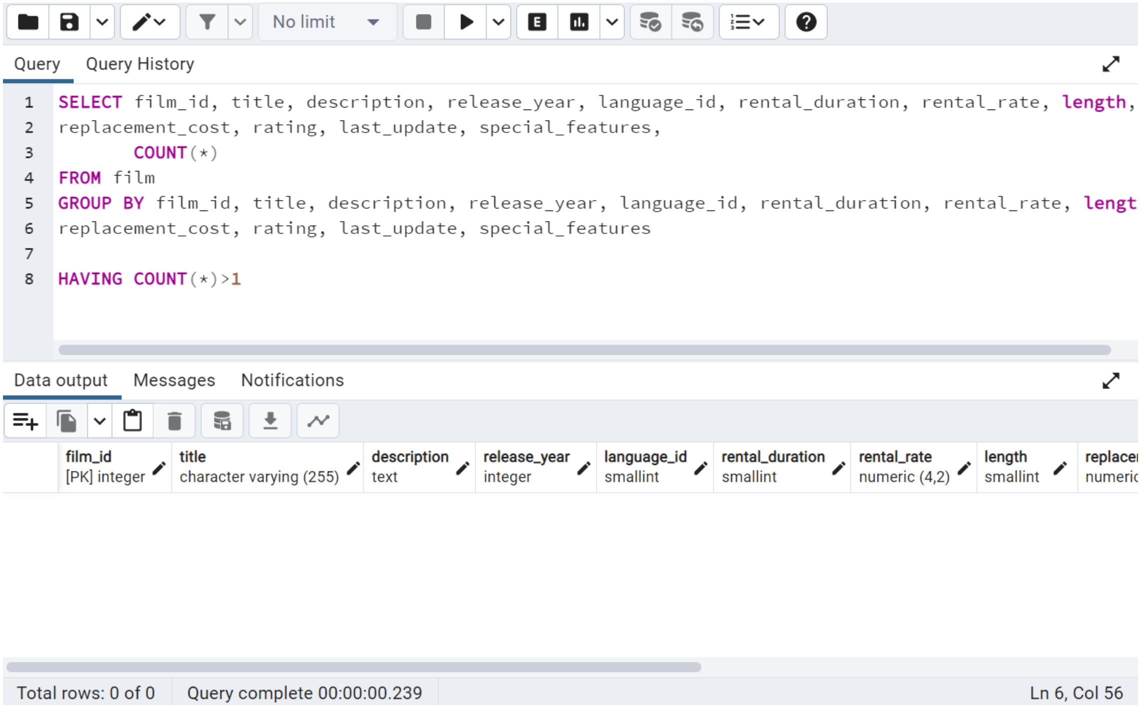
3.6: Summarizing & Cleaning Data in SQL

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. Create a new "Answers 3.6" document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty).

Checking duplicates on film table



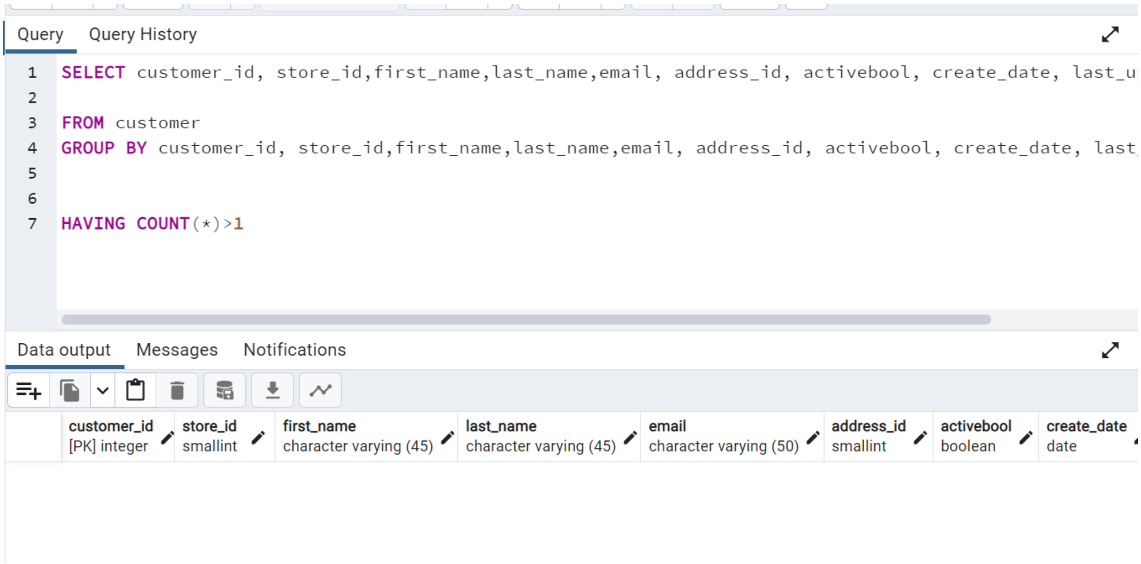
```
SELECT film_id, title, description, release_year, language_id, rental_duration, rental_rate, length,
replacement_cost, rating, last_update, special_features,
       COUNT(*)
FROM film
GROUP BY film_id, title, description, release_year, language_id, rental_duration, rental_rate, lengt
replacement_cost, rating, last_update, special_features

HAVING COUNT(*)>1
```

film_id	title	description	release_year	language_id	rental_duration	rental_rate	length	replace
[PK] integer	character varying (255)	text	integer	smallint	smallint	numeric (4,2)	smallint	numeric

```
Total rows: 0 of 0    Query complete 00:00:00.239                Ln 6, Col 56
```

Checking duplicates on customer table



```
SELECT customer_id, store_id,first_name,last_name,email, address_id, activebool, create_date, last_u

FROM customer
GROUP BY customer_id, store_id,first_name,last_name,email, address_id, activebool, create_date, last


HAVING COUNT(*)>1
```

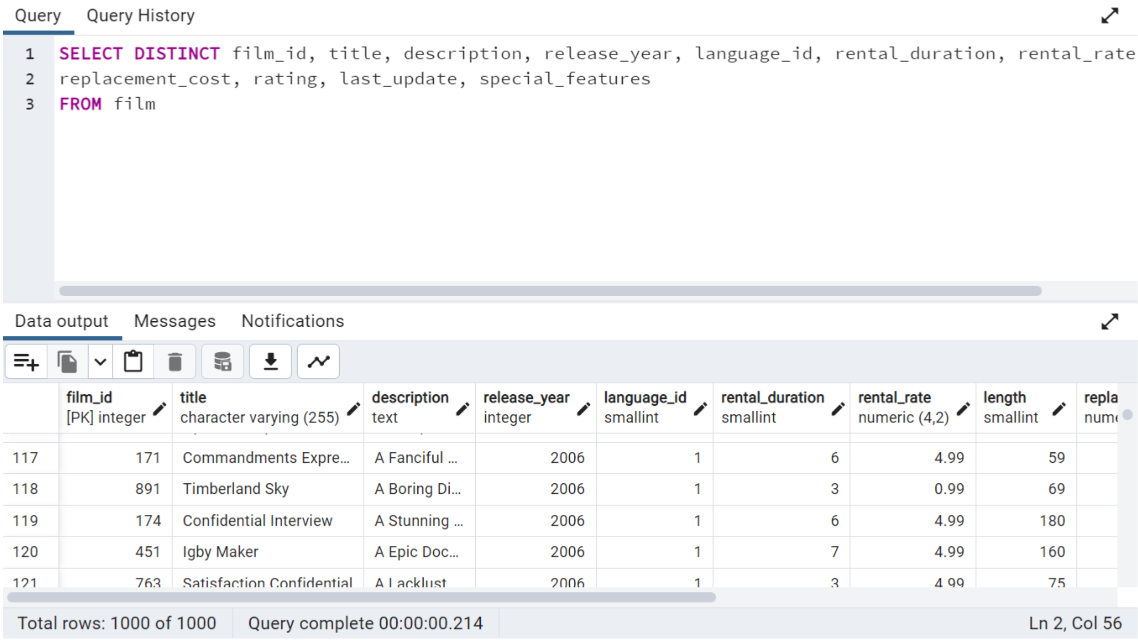
customer_id	store_id	first_name	last_name	email	address_id	activebool	create_date
[PK] integer	smallint	character varying (45)	character varying (45)	character varying (50)	smallint	boolean	date

There is no duplicate values neither on film or customer tables. If there were any, we can:

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

We have to be quite careful when deleting data, so maybe it´s better to create the view

Checking incoherences on film table



Query Query History

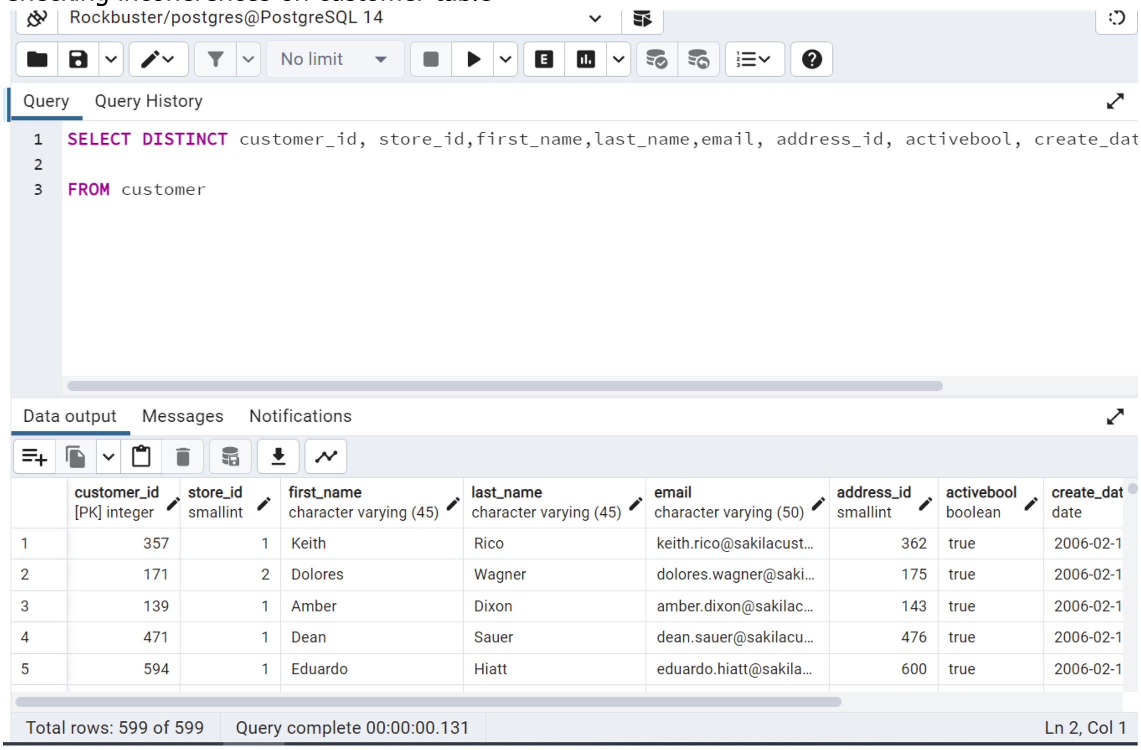
```
SELECT DISTINCT film_id, title, description, release_year, language_id, rental_duration, rental_rate
replacement_cost, rating, last_update, special_features
FROM film
```

Data output Messages Notifications

	film_id [PK] integer	title character varying (255)	description text	release_year integer	language_id smallint	rental_duration smallint	rental_rate numeric (4,2)	length smallint	repla num
117	171	Commandments Expre...	A Fanciful ...	2006	1	6	4.99	59	
118	891	Timberland Sky	A Boring Di...	2006	1	3	0.99	69	
119	174	Confidential Interview	A Stunning ...	2006	1	6	4.99	180	
120	451	Igby Maker	A Epic Doc...	2006	1	7	4.99	160	
121	763	Satisfaction Confidential	A Lacklust	2006	1	3	4.99	75	

Total rows: 1000 of 1000 Query complete 00:00:00.214 Ln 2, Col 56

Checking incoherences on customer table



Rockbuster/postgres@PostgreSQL 14

No limit

Query Query History

```
SELECT DISTINCT customer_id, store_id,first_name,last_name,email, address_id, activebool, create_dat

FROM customer
```

Data output Messages Notifications

	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_name character varying (45)	email character varying (50)	address_id smallint	activebool boolean	create_dat date
1	357	1	Keith	Rico	keith.rico@sakilacust...	362	true	2006-02-1
2	171	2	Dolores	Wagner	dolores.wagner@saki...	175	true	2006-02-1
3	139	1	Amber	Dixon	amber.dixon@sakilac...	143	true	2006-02-1
4	471	1	Dean	Sauer	dean.sauer@sakilacu...	476	true	2006-02-1
5	594	1	Eduardo	Hiatt	eduardo.hiatt@sakila...	600	true	2006-02-1

Total rows: 599 of 599 Query complete 00:00:00.131 Ln 2, Col 1

It seems also not to be any incoherence on the information of both tables. If we´ve found any, we can fix it with an UPDATE function when nulls

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. Copy-paste your SQL queries and their outputs into your answers document.

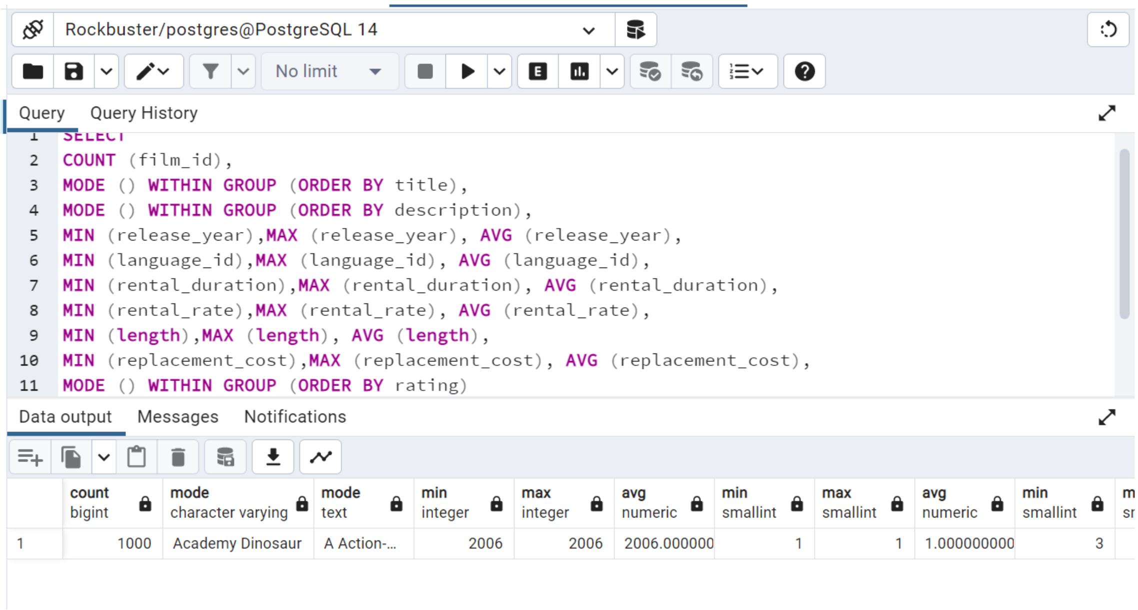
   *I think it does not make sense to calculate MIN/MAX/AVG when we are talking about ids, so I´ve chosen COUNT function here

   FILM TABLE

Columns	Data Type	Descriptive statistic
film_id	SERIAL	COUNT
title	CHARACTER VARYING(25)	MODE
description	TEXT	MODE
release_year	YEAR	MIN/MAX/AVG
language_id	SMALLINT	MODE
rental_duration	SMALLINT	MIN/MAX/AVG
rental_rate	NUMERIC(4.2)	MIN/MAX/AVG
length	SMALLINT	MIN/MAX/AVG
replacement_cost	NUMERIC(5.2)	MIN/MAX/AVG
rating	mpaa_rating	MODE
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	N/A
special_features	TEXT[]	N/A
fulltext	TSVECTOR	N/A

```
SELECT
COUNT (film_id),
MODE () WITHIN GROUP (ORDER BY title),
MODE () WITHIN GROUP (ORDER BY description),
MIN (release_year),MAX (release_year), AVG (release_year),
MIN (language_id),MAX (language_id), AVG (language_id),
MIN (rental_duration),MAX (rental_duration), AVG (rental_duration),
MIN (rental_rate),MAX (rental_rate), AVG (rental_rate),
MIN (length),MAX (length), AVG (length),
MIN (replacement_cost),MAX (replacement_cost), AVG (replacement_cost),
MODE () WITHIN GROUP (ORDER BY rating)

FROM film
```



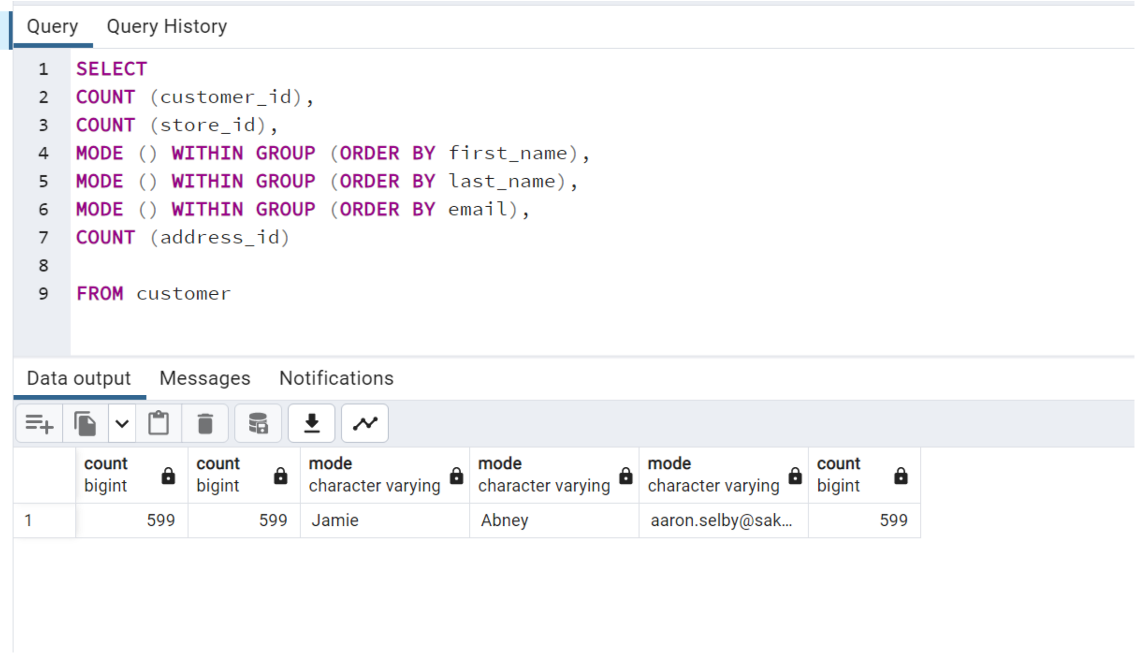
```
SELECT
COUNT (film_id),
MODE () WITHIN GROUP (ORDER BY title),
MODE () WITHIN GROUP (ORDER BY description),
MIN (release_year),MAX (release_year), AVG (release_year),
MIN (language_id),MAX (language_id), AVG (language_id),
MIN (rental_duration),MAX (rental_duration), AVG (rental_duration),
MIN (rental_rate),MAX (rental_rate), AVG (rental_rate),
MIN (length),MAX (length), AVG (length),
MIN (replacement_cost),MAX (replacement_cost), AVG (replacement_cost),
MODE () WITHIN GROUP (ORDER BY rating)
```

	count bigint	mode character varying	mode text	min integer	max integer	avg numeric	min smallint	max smallint	avg numeric	min smallint	m sr
1	1000	Academy Dinosaur	A Action-...	2006	2006	2006.000000	1	1	1.000000000	3	

CUSTOMER TABLE

We don´t have here any numerical character so there is no MIN/MAX/AVG calculation

Columns	Data Type	Descriptive statistic
customer_id	SERIAL	COUNT
store_id	SMALLINT	COUNT
first_name	CHARACTER VARYING(45)	MODE
last_name	CHARACTER VARYING(45)	MODE
email	CHARACTER VARYING(50)	MODE
adress_id	SMALLINT	COUNT
activebool	BOOLEAN	N/A
create_date	DATE	N/A
last_update	TIMESTAMP(6) WITHOUT TIME ZONE	N/A
active	INTEGER	N/A



```
SELECT
COUNT (customer_id),
COUNT (store_id),
MODE () WITHIN GROUP (ORDER BY first_name),
MODE () WITHIN GROUP (ORDER BY last_name),
MODE () WITHIN GROUP (ORDER BY email),
COUNT (address_id)

FROM customer
```

	count bigint	count bigint	mode character varying	mode character varying	mode character varying	count bigint
1	599	599	Jamie	Abney	aaron.selby@sak...	599

3. **Reflect on your work:** Back in Achievement 1 you learned about data profiling in Excel. Based on your previous experience, 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. Write a short paragraph in the running document that you have started.

Same as the last time I answer about this, I don´t think we have the skills on SQL to choose between both tools, because at least if I´m talking about me:

- I´ve been working for more than 15 years with excel so the usability and the velocity with I can do things like filtering and cleaning, summarizing etc is quite high
- Of course I´m not use to work (I have started 1-2 years ago and then was when I realize I need another tool to work with) with big data bases.
- I´ve started just for one week so yet my skills on SQL are soo low, and I don´t even know about best practices, commands (copy/paste/replace), to go as fast as I work with excel

But even with this, I´m convinced on the tool, on the language, and I have all my expectations to be able to work with it

4. Save your "Answers 3.6" document as a PDF and upload it here for your tutor to review.