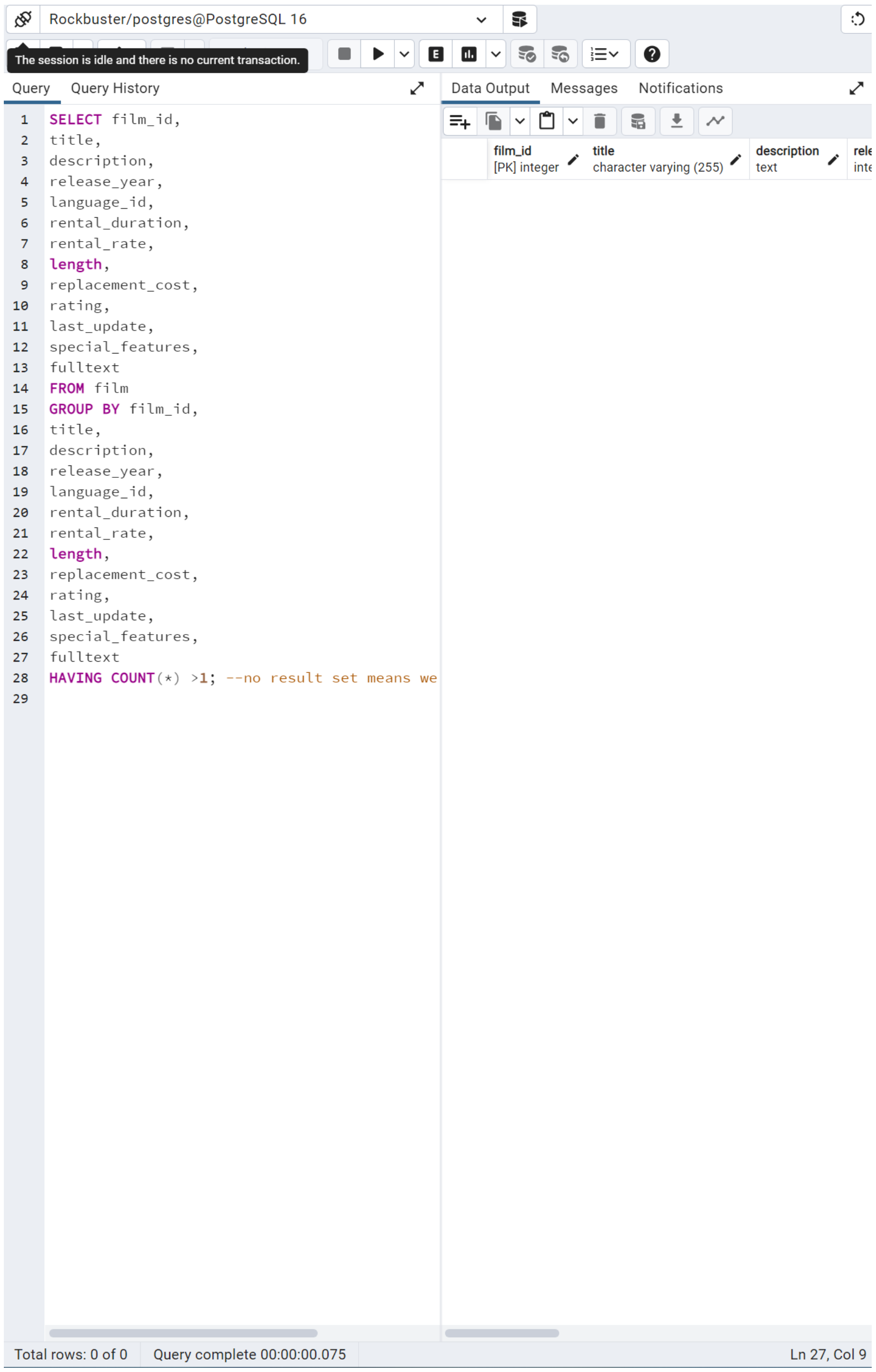
Alexander Coley
11-19-23
Exercise 3.6

Question 1:
Film table-Duplicate Data


Rockbuster/postgres@PostgreSQL 16
The session is idle and there is no current transaction.
Query
Query History
Data Output
Messages
Notifications

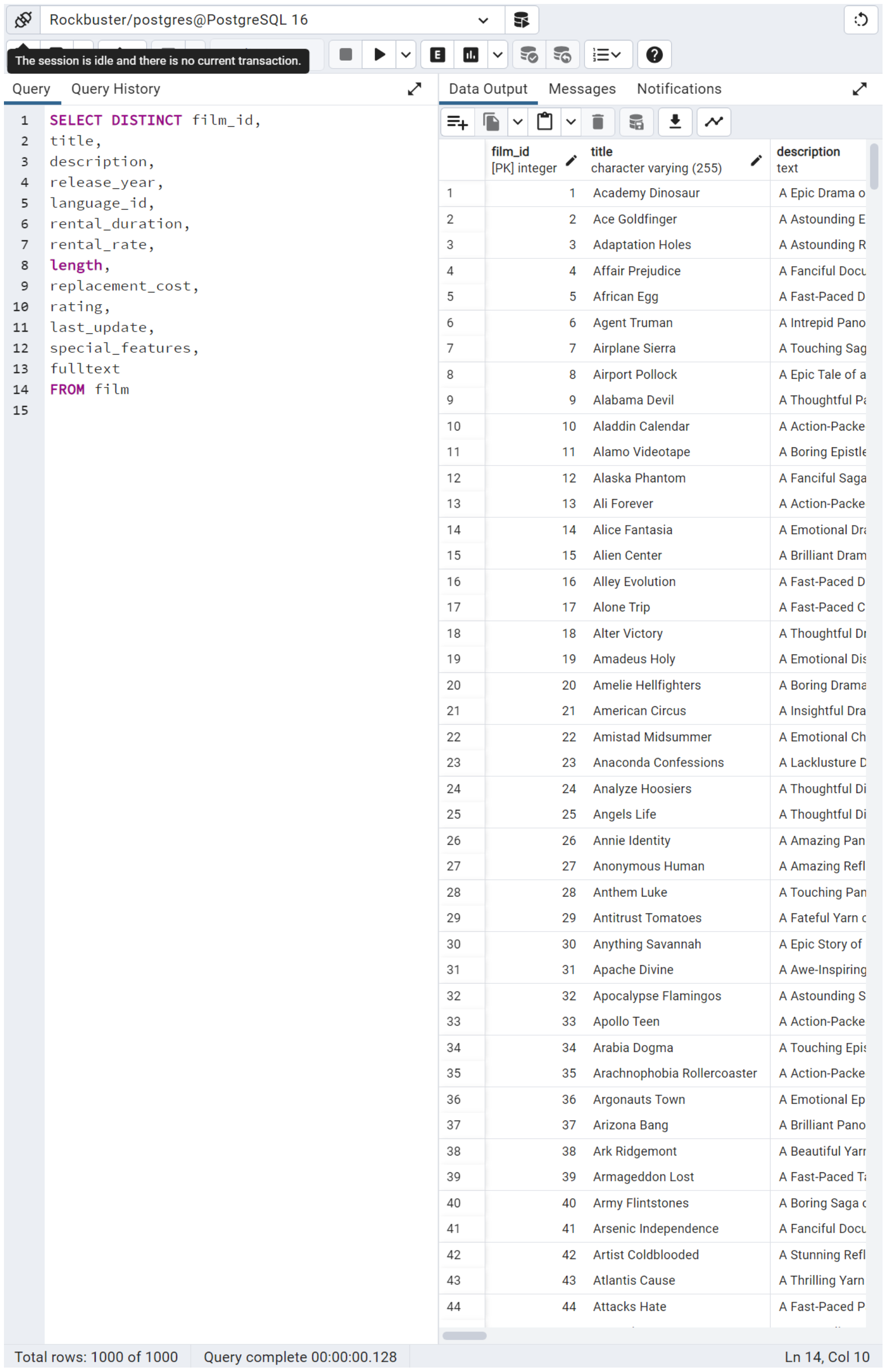
```
SELECT film_id,
title,
description,
release_year,
language_id,
rental_duration,
rental_rate,
length,
replacement_cost,
rating,
last_update,
special_features,
fulltext
FROM film
GROUP BY film_id,
title,
description,
release_year,
language_id,
rental_duration,
rental_rate,
length,
replacement_cost,
rating,
last_update,
special_features,
fulltext
HAVING COUNT(*) >1; --no result set means we
```

film_id [PK] integer	title character varying (255)	description text	rele inte

Total rows: 0 of 0   Query complete 00:00:00.075   Ln 27, Col 9

I didn't find any duplicates in this data set. Although, if there were any duplicates, I would delete the duplicate columns to keep the data set consistent. Deleting these duplicates would also help with visualizations later on.

Non-uniform data


Rockbuster/postgres@PostgreSQL 16

The session is idle and there is no current transaction.

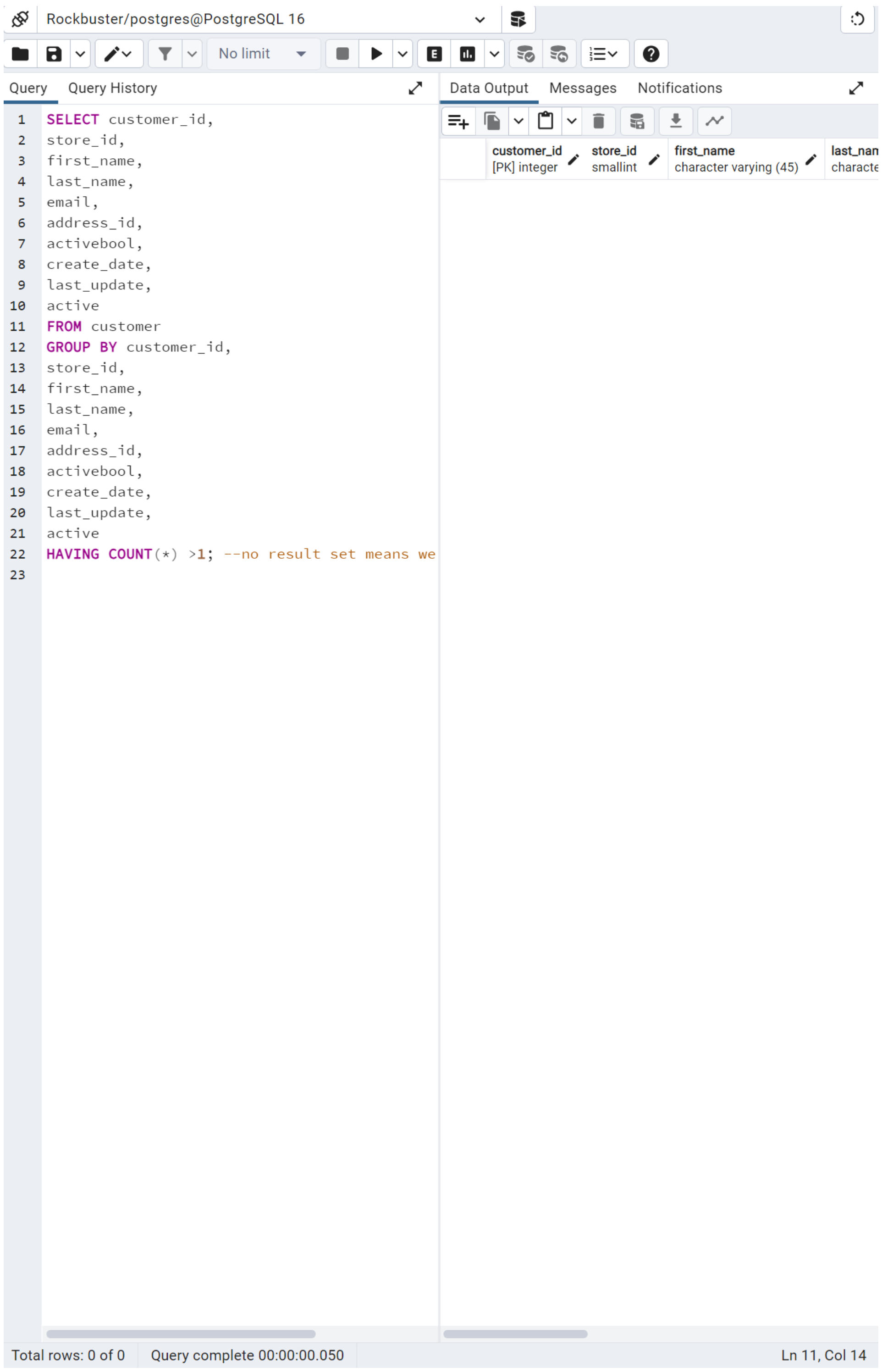
```sql
SELECT DISTINCT film_id,
title,
description,
release_year,
language_id,
rental_duration,
rental_rate,
length,
replacement_cost,
rating,
last_update,
special_features,
fulltext
FROM film
```

	film_id [PK] integer	title character varying (255)	description text
1	1	Academy Dinosaur	A Epic Drama o
2	2	Ace Goldfinger	A Astounding E
3	3	Adaptation Holes	A Astounding R
4	4	Affair Prejudice	A Fanciful Docu
5	5	African Egg	A Fast-Paced D
6	6	Agent Truman	A Intrepid Pano
7	7	Airplane Sierra	A Touching Sag
8	8	Airport Pollock	A Epic Tale of a
9	9	Alabama Devil	A Thoughtful Pa
10	10	Aladdin Calendar	A Action-Packe
11	11	Alamo Videotape	A Boring Epistle
12	12	Alaska Phantom	A Fanciful Saga
13	13	Ali Forever	A Action-Packe
14	14	Alice Fantasia	A Emotional Dr
15	15	Alien Center	A Brilliant Dram
16	16	Alley Evolution	A Fast-Paced D
17	17	Alone Trip	A Fast-Paced C
18	18	Alter Victory	A Thoughtful Dr
19	19	Amadeus Holy	A Emotional Dis
20	20	Amelie Hellfighters	A Boring Drama
21	21	American Circus	A Insightful Dra
22	22	Amistad Midsummer	A Emotional Ch
23	23	Anaconda Confessions	A Lacklusture D
24	24	Analyze Hoosiers	A Thoughtful Di
25	25	Angels Life	A Thoughtful Di
26	26	Annie Identity	A Amazing Pan
27	27	Anonymous Human	A Amazing Refl
28	28	Anthem Luke	A Touching Pan
29	29	Antitrust Tomatoes	A Fateful Yarn c
30	30	Anything Savannah	A Epic Story of
31	31	Apache Divine	A Awe-Inspiring
32	32	Apocalypse Flamingos	A Astounding S
33	33	Apollo Teen	A Action-Packe
34	34	Arabia Dogma	A Touching Epis
35	35	Arachnophobia Rollercoaster	A Action-Packe
36	36	Argonauts Town	A Emotional Ep
37	37	Arizona Bang	A Brilliant Pano
38	38	Ark Ridgemont	A Beautiful Yarn
39	39	Armageddon Lost	A Fast-Paced Ta
40	40	Army Flintstones	A Boring Saga c
41	41	Arsenic Independence	A Fanciful Docu
42	42	Artist Coldblooded	A Stunning Refl
43	43	Atlantis Cause	A Thrilling Yarn
44	44	Attacks Hate	A Fast-Paced P

Total rows: 1000 of 1000 Query complete 00:00:00.128 Ln 14, Col 10

There isn't any non-uniform data. The data sent back 1000 out of 1000 rows so everything is consistent.

Customer Table-Duplicate Data



Rockbuster/postgres@PostgreSQL 16

No limit

Query | Query History

```
SELECT customer_id,
store_id,
first_name,
last_name,
email,
address_id,
activebool,
create_date,
last_update,
active
FROM customer
GROUP BY customer_id,
store_id,
first_name,
last_name,
email,
address_id,
activebool,
create_date,
last_update,
active
HAVING COUNT(*) >1; --no result set means we
```

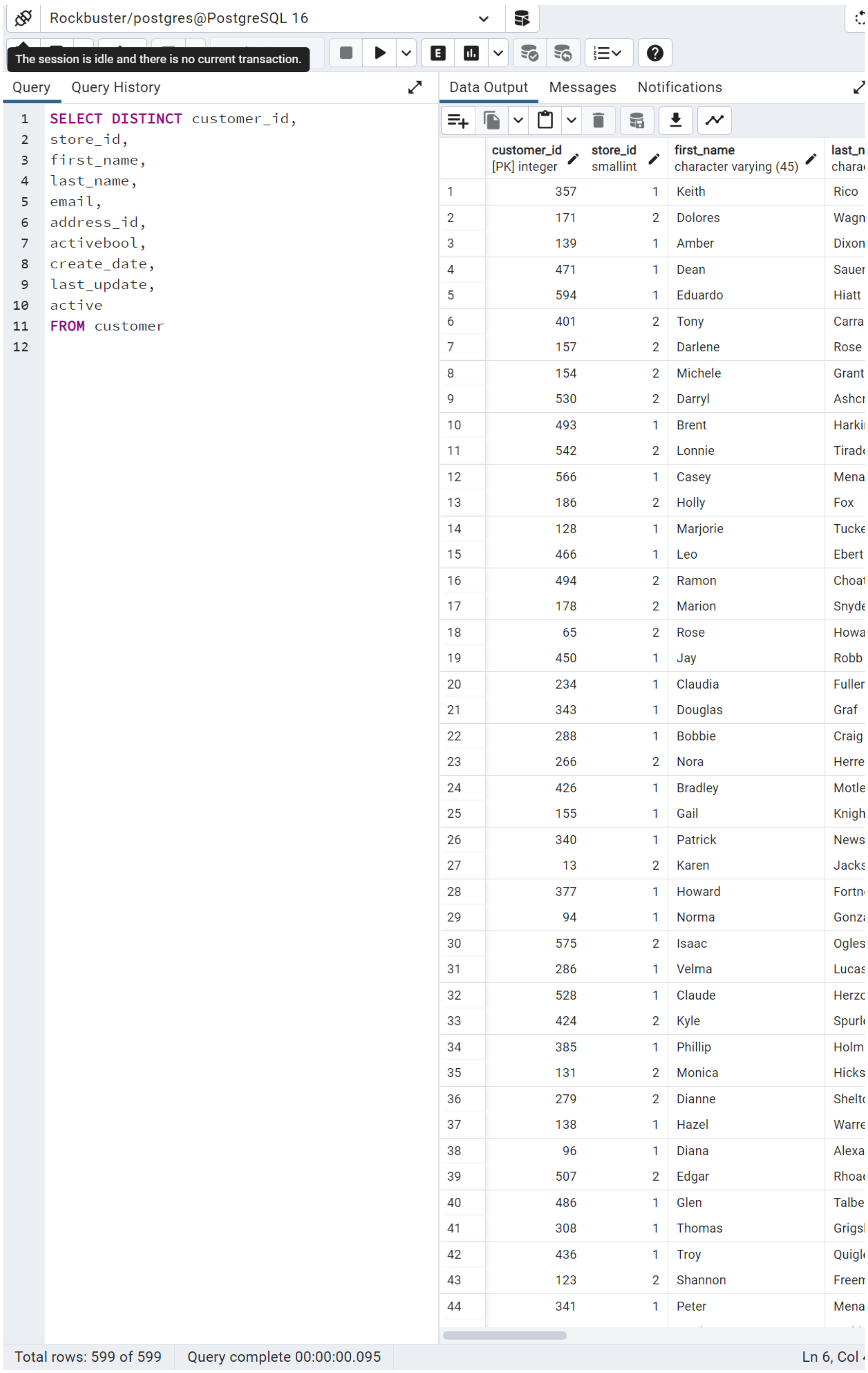
Data Output | Messages | Notifications

customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_nan characte

Total rows: 0 of 0 Query complete 00:00:00.050 Ln 11, Col 14

There weren't any duplicates for the customer table. If there were any duplicates I would do the same as before and delete the duplicates as to not hurt my progress in making the visualizations.

Non-Uniform Data


Rockbuster/postgres@PostgreSQL 16

The session is idle and there is no current transaction.

Query   Query History

```sql
SELECT DISTINCT customer_id,
store_id,
first_name,
last_name,
email,
address_id,
activebool,
create_date,
last_update,
active
FROM customer

```

Data Output   Messages   Notifications

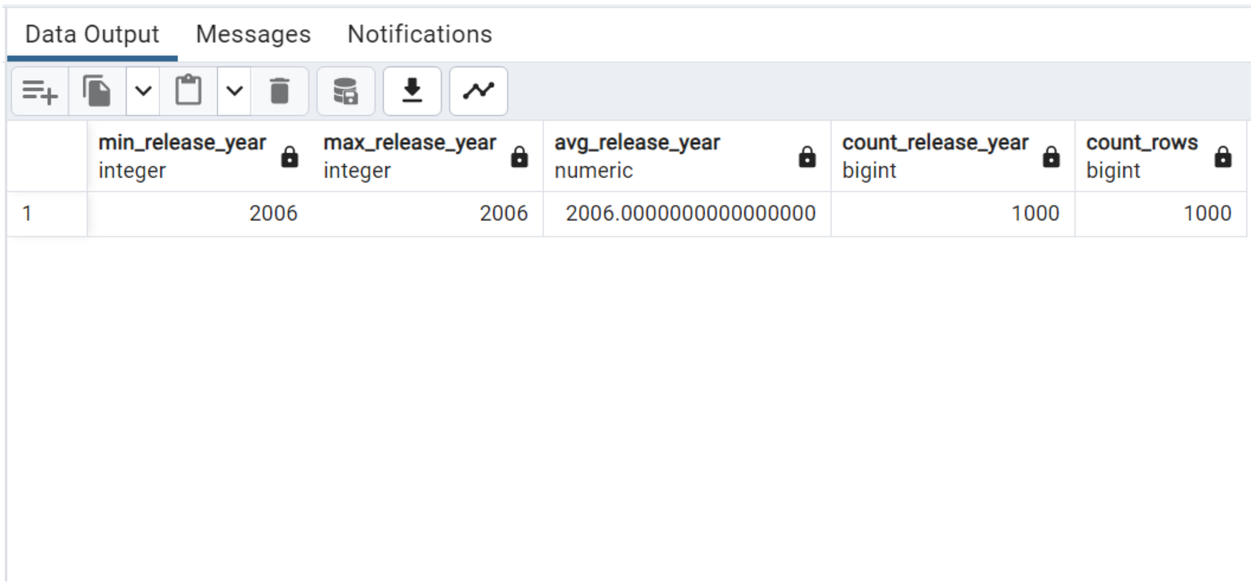
	customer_id [PK] integer	store_id smallint	first_name character varying (45)	last_n chara
1	357	1	Keith	Rico
2	171	2	Dolores	Wagn
3	139	1	Amber	Dixon
4	471	1	Dean	Sauer
5	594	1	Eduardo	Hiatt
6	401	2	Tony	Carra
7	157	2	Darlene	Rose
8	154	2	Michele	Grant
9	530	2	Darryl	Ashcı
10	493	1	Brent	Harki
11	542	2	Lonnie	Tirad
12	566	1	Casey	Mena
13	186	2	Holly	Fox
14	128	1	Marjorie	Tucke
15	466	1	Leo	Ebert
16	494	2	Ramon	Choat
17	178	2	Marion	Snyde
18	65	2	Rose	Howa
19	450	1	Jay	Robb
20	234	1	Claudia	Fuller
21	343	1	Douglas	Graf
22	288	1	Bobbie	Craig
23	266	2	Nora	Herre
24	426	1	Bradley	Motle
25	155	1	Gail	Knigh
26	340	1	Patrick	News
27	13	2	Karen	Jacks
28	377	1	Howard	Fortn
29	94	1	Norma	Gonz
30	575	2	Isaac	Ogles
31	286	1	Velma	Lucas
32	528	1	Claude	Herzc
33	424	2	Kyle	Spurl
34	385	1	Phillip	Holm
35	131	2	Monica	Hicks
36	279	2	Dianne	Shelt
37	138	1	Hazel	Warre
38	96	1	Diana	Alexa
39	507	2	Edgar	Rhoa
40	486	1	Glen	Talbe
41	308	1	Thomas	Grigs
42	436	1	Troy	Quigl
43	123	2	Shannon	Freen
44	341	1	Peter	Mena

Total rows: 599 of 599   Query complete 00:00:00.095   Ln 6, Col

The data was uniform here as well as it returned the maximum amount of rows. If there were any missing values, if I didn’t have access to what was missing, I would make the values zero.

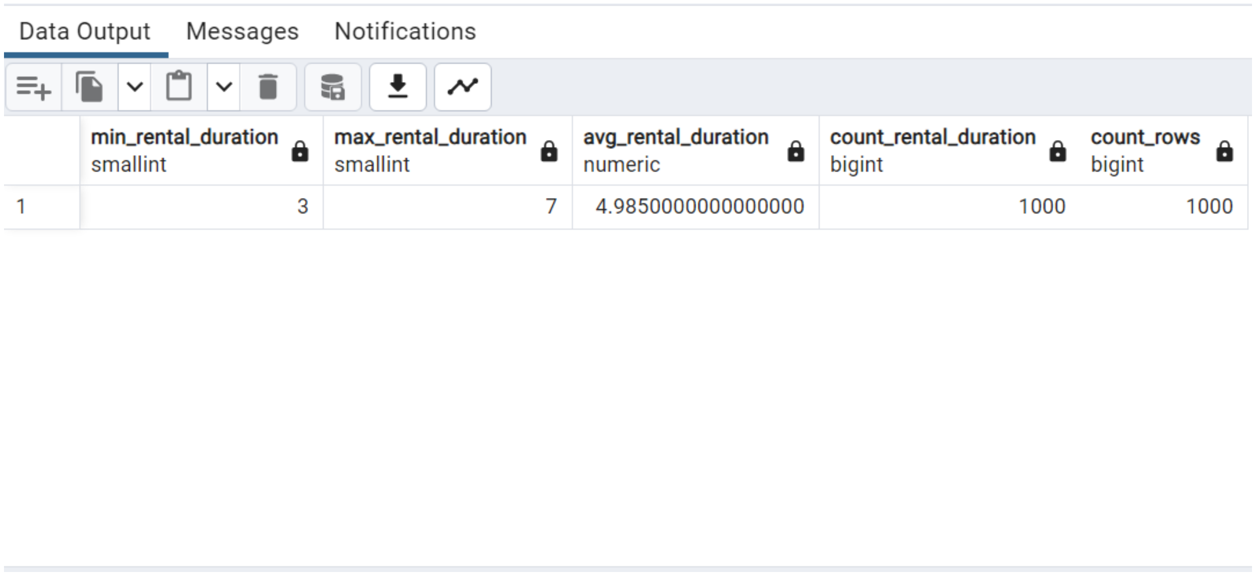
Question 2:
Numerical Values for Film Table

Release year
SELECT MIN (release_year) AS min_release_year ,
MAX (release_year) AS max_release_year,
AVG (release_year) AS avg_release_year,
COUNT (release_year) AS count_release_year,
COUNT (*) AS count_rows
FROM film



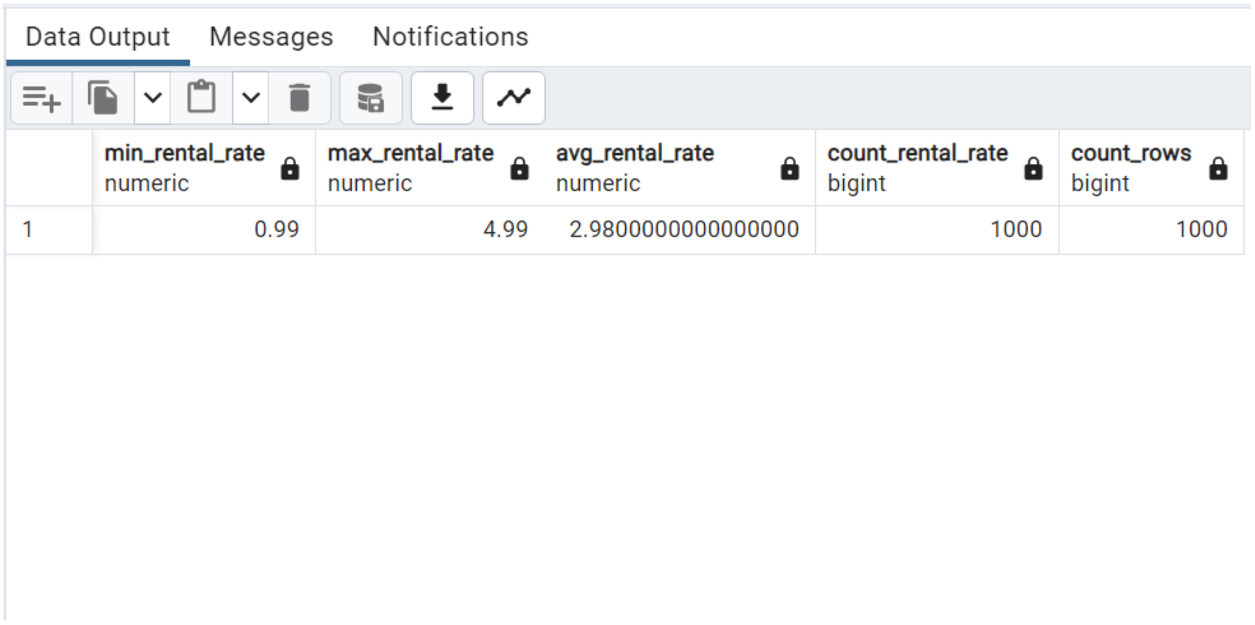
	min_release_year integer	max_release_year integer	avg_release_year numeric	count_release_year bigint	count_rows bigint
1	2006	2006	2006.0000000000000000	1000	1000

Rental Duration
SELECT MIN (rental_duration) AS min_rental_duration ,
MAX (rental_duration) AS max_rental_duration,
AVG (rental_duration) AS avg_rental_duration,
COUNT (rental_duration) AS count_rental_duration,
COUNT (*) AS count_rows
FROM film



	min_rental_duration smallint	max_rental_duration smallint	avg_rental_duration numeric	count_rental_duration bigint	count_rows bigint
1	3	7	4.9850000000000000	1000	1000

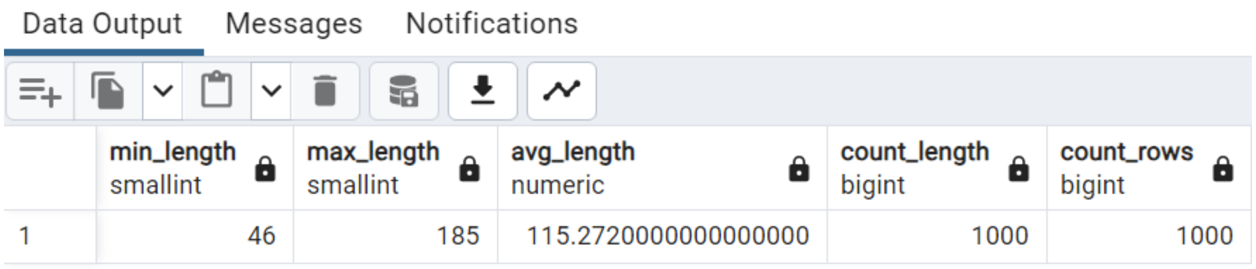
Rental Rate
SELECT MIN (rental_rate) AS min_rental_rate ,
MAX (rental_rate) AS max_rental_rate,
AVG (rental_rate) AS avg_rental_rate,
COUNT (rental_rate) AS count_rental_rate,
COUNT (*) AS count_rows
FROM film



	min_rental_rate numeric	max_rental_rate numeric	avg_rental_rate numeric	count_rental_rate bigint	count_rows bigint
1	0.99	4.99	2.9800000000000000	1000	1000

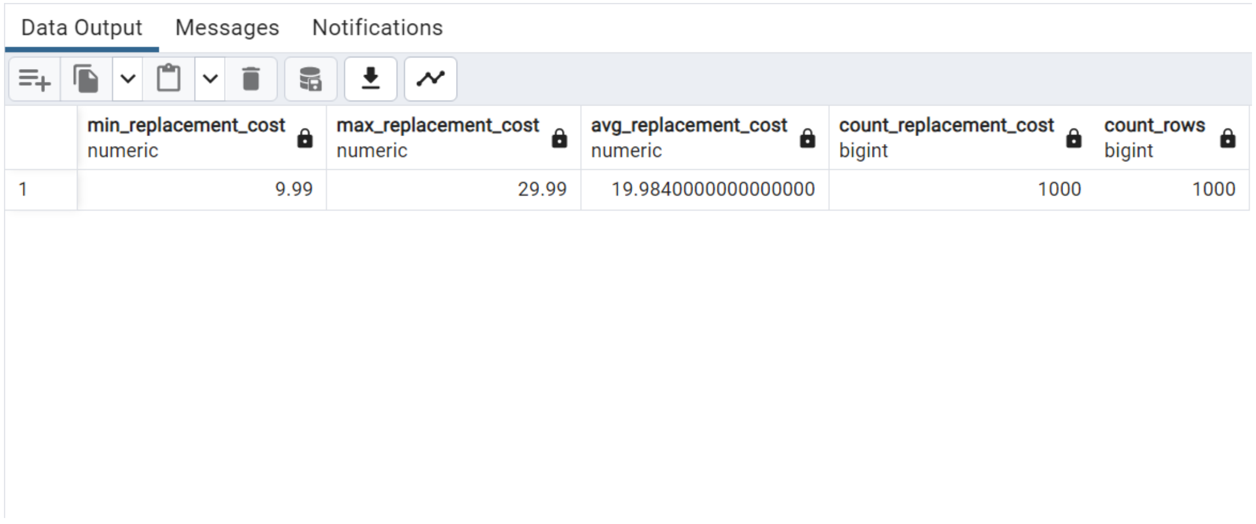
Length
SELECT MIN (length) AS min_length ,
MAX (length) AS max_length,
AVG (length) AS avg_length,

COUNT (length) AS count_length,
COUNT (*) AS count_rows
FROM film



	min_length smallint	max_length smallint	avg_length numeric	count_length bigint	count_rows bigint
1	46	185	115.2720000000000000	1000	1000

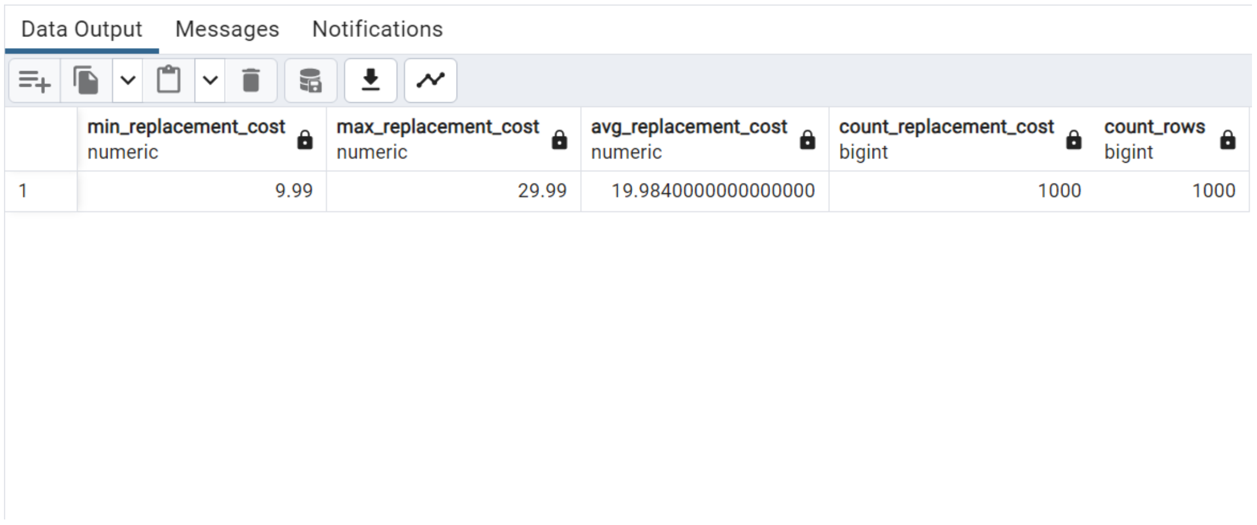
Replacement Cost
SELECT MIN (replacement_cost) AS min_replacement_cost ,
MAX (replacement_cost) AS max_replacement_cost,
AVG (replacement_cost) AS avg_replacement_cost,
COUNT (replacement_cost) AS count_replacement_cost,
COUNT (*) AS count_rows
FROM film



	min_replacement_cost numeric	max_replacement_cost numeric	avg_replacement_cost numeric	count_replacement_cost bigint	count_rows bigint
1	9.99	29.99	19.9840000000000000	1000	1000

Non numerical values

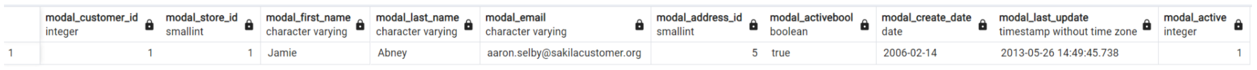
```
SELECT MODE() WITHIN GROUP (ORDER BY film_id) AS modal_film_id,
MODE() WITHIN GROUP (ORDER BY title) AS modal_title,
MODE() WITHIN GROUP (ORDER BY description) AS modal_description,
MODE() WITHIN GROUP (ORDER BY language_id) AS modal_language_id,
MODE() WITHIN GROUP (ORDER BY rating) AS modal_rating,
MODE() WITHIN GROUP (ORDER BY last_update) AS modal_last_update,
MODE() WITHIN GROUP (ORDER BY special_features) AS modal_special_features,
MODE() WITHIN GROUP (ORDER BY fulltext) AS modal_fulltext
FROM film
```



	min_replacement_cost numeric	max_replacement_cost numeric	avg_replacement_cost numeric	count_replacement_cost bigint	count_rows bigint
1	9.99	29.99	19.9840000000000000	1000	1000

Customer Table *only contains non numerical values

```
SELECT MODE() WITHIN GROUP (ORDER BY customer_id) AS modal_customer_id,
MODE() WITHIN GROUP (ORDER BY store_id) AS modal_store_id,
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,
MODE() WITHIN GROUP (ORDER BY address_id) AS modal_address_id,
MODE() WITHIN GROUP (ORDER BY activebool) AS modal_activebool,
MODE() WITHIN GROUP (ORDER BY create_date) AS modal_create_date,
MODE() WITHIN GROUP (ORDER BY last_update) AS modal_last_update,
MODE() WITHIN GROUP (ORDER BY active) AS modal_active
FROM customer
```



	modal_customer_id integer	modal_store_id smallint	modal_first_name character varying	modal_last_name character varying	modal_email character varying	modal_address_id smallint	modal_activebool boolean	modal_create_date date	modal_last_update timestamp without time zone	modal_active integer
1	1	1	Jamie	Abney	aaron.selby@sakilacustomer.org	5	true	2006-02-14	2013-05-26 14:49:45.738	1

Question 3:

As much as I've learned from SQL so far I definitely prefer it over excel. Excel is still very useful but I feel as though the functions in SQL are easier to use and understand. It's also more visually appealing to me personally. I hope to learn more about SQL functions in the future as I grow as an analyst.