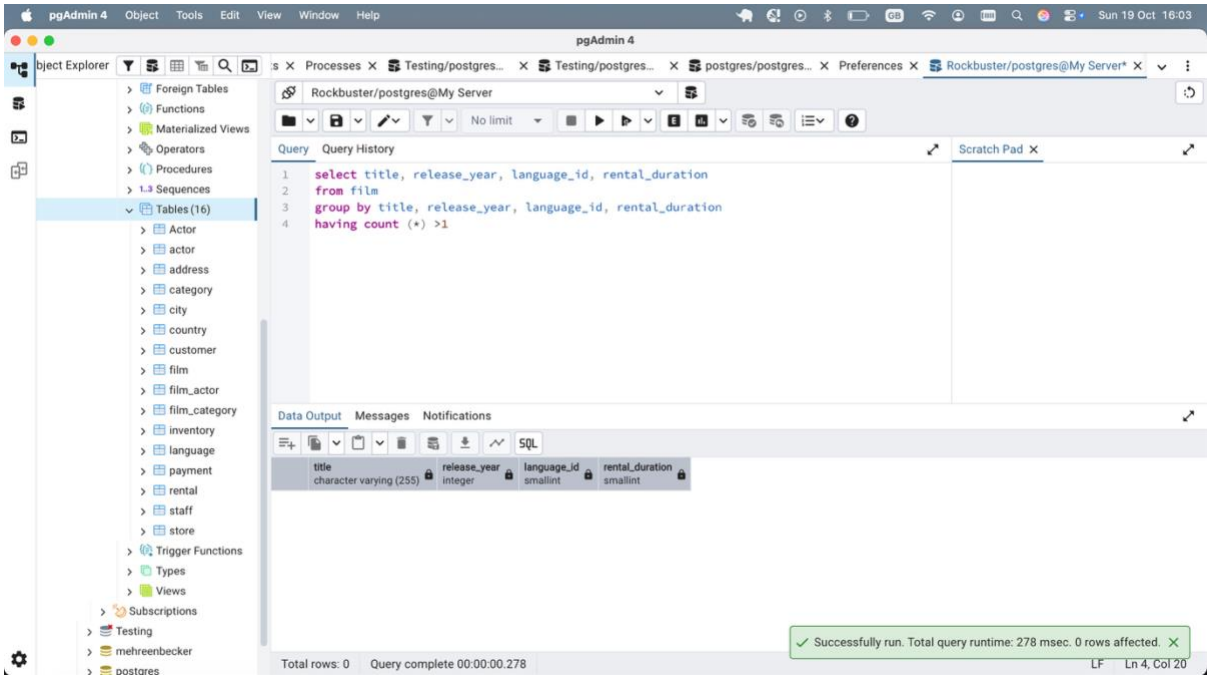
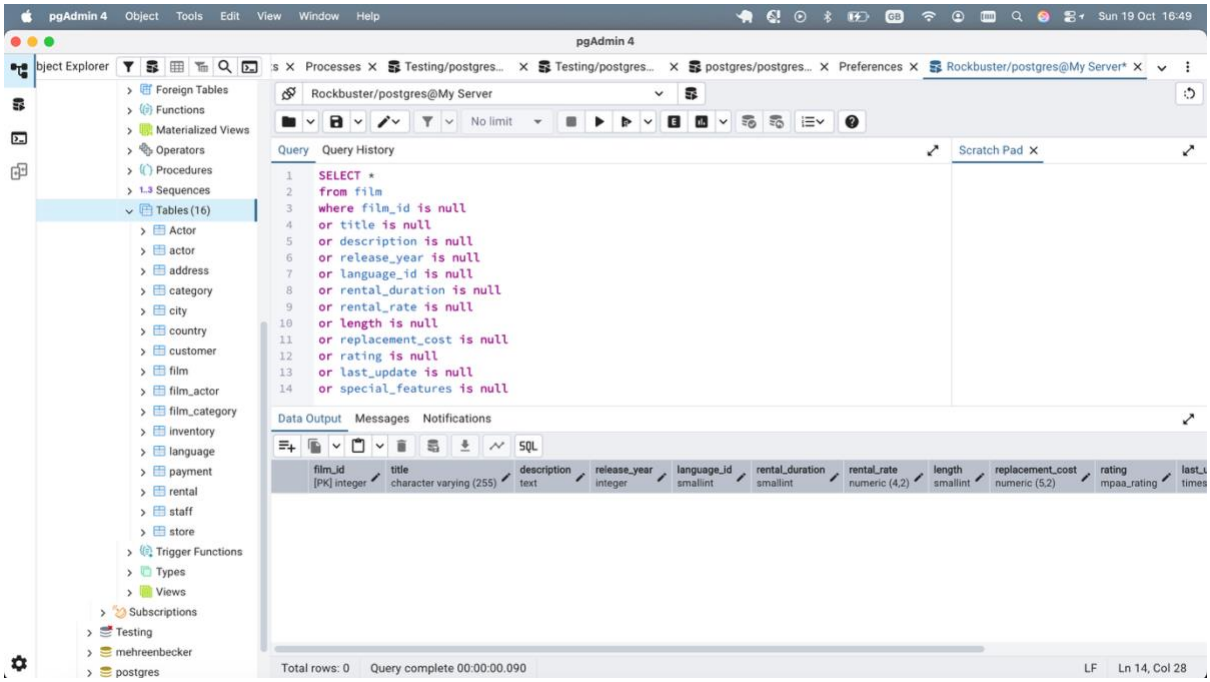
Data Analytics Immersion
3.6: Summarizing & Cleaning Data in SQL
Mehreen Becker

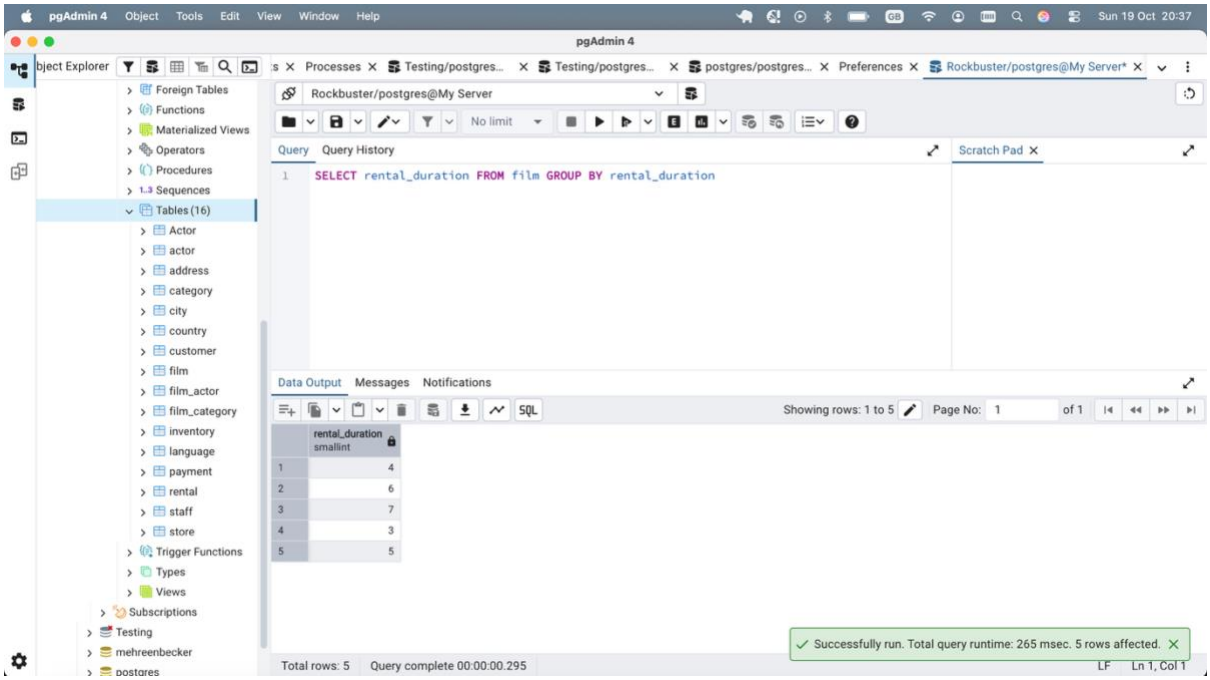
Rockbuster's database engineers have loaded some new data into the database, and your manager has asked you to clean and profile it. Follow the instructions below to complete their request:

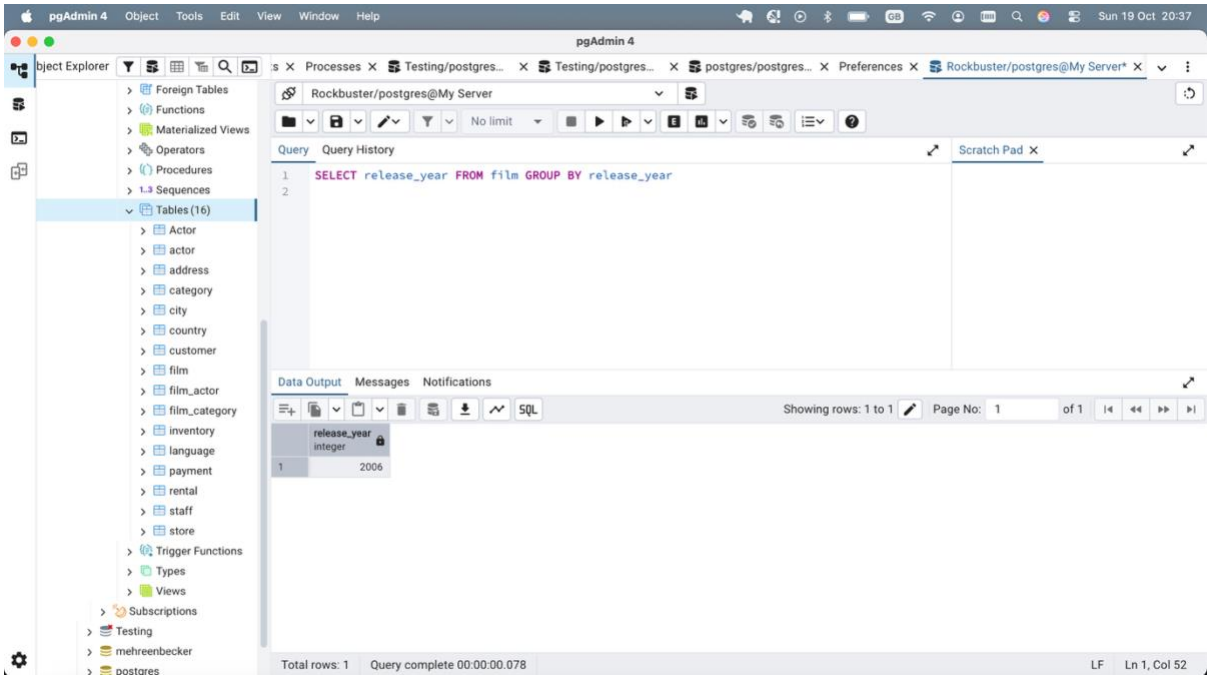
1. **Check for and clean dirty data: Find out if the film table and the customer table contain any dirty data, specifically non-uniform or duplicate data, or missing values. Create a new "Answers 3.6" document and copy-paste your queries into it. Next to each query write 2 to 3 sentences explaining how you would clean the data (even if the data is not dirty)**

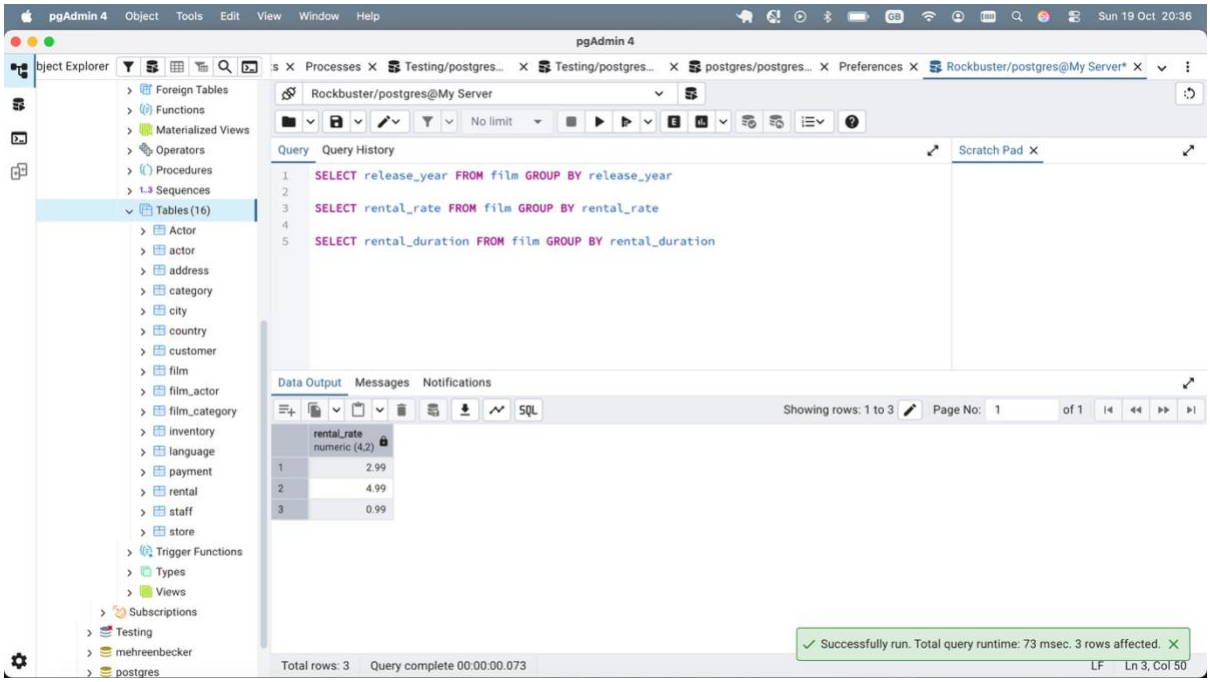


I would first identigy duplicate records by analyzing key fields (SELECT column, COUNT (*) FROM table GROUP BY column HAVING COUNT (*) >1)to see where the same values appear more than once and then I would decide which records to keep and remove the extra duplicates to ensue only unique and accurate data remains.

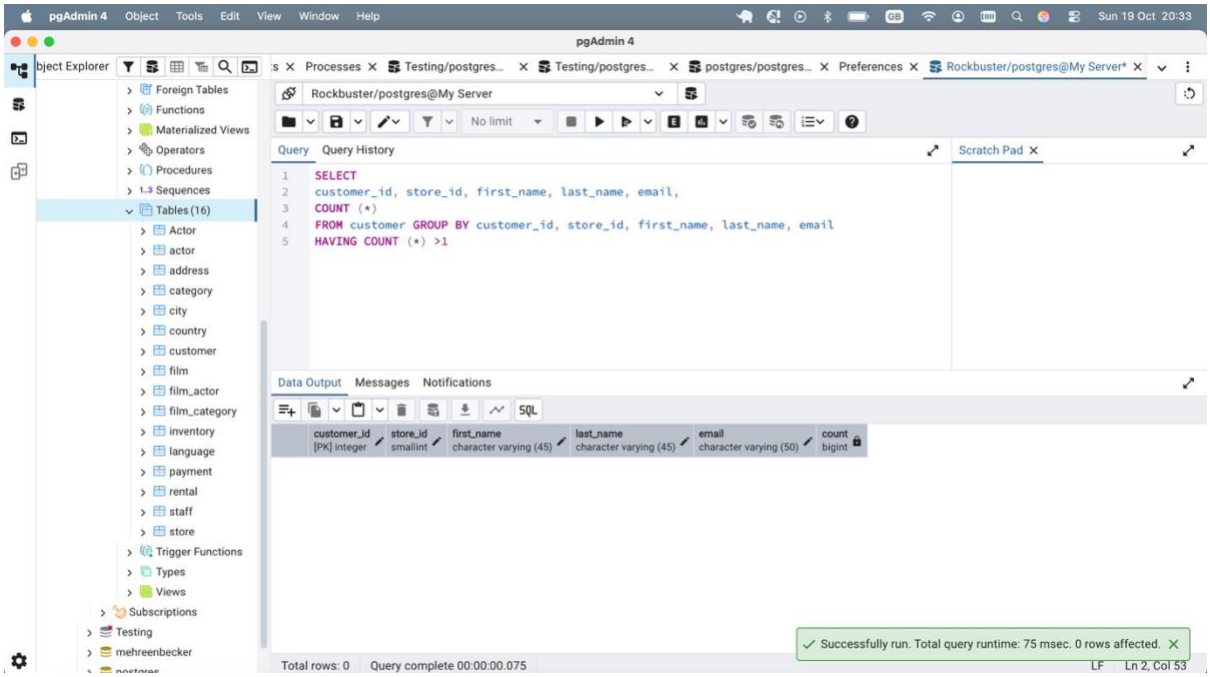

pgAdmin 4
Object Explorer
Rockbuster/postgres@My Server
Query
Query History
Scratch Pad
SELECT *
from film
where film_id is null
or title is null
or description is null
or release_year is null
or language_id is null
or rental_duration is null
or rental_rate is null
or length is null
or replacement_cost is null
or rating is null
or last_update is null
or special_features is null
Data Output
Messages
Notifications
Total rows: 0
Query complete 00:00:00.090
Ln 14, Col 28

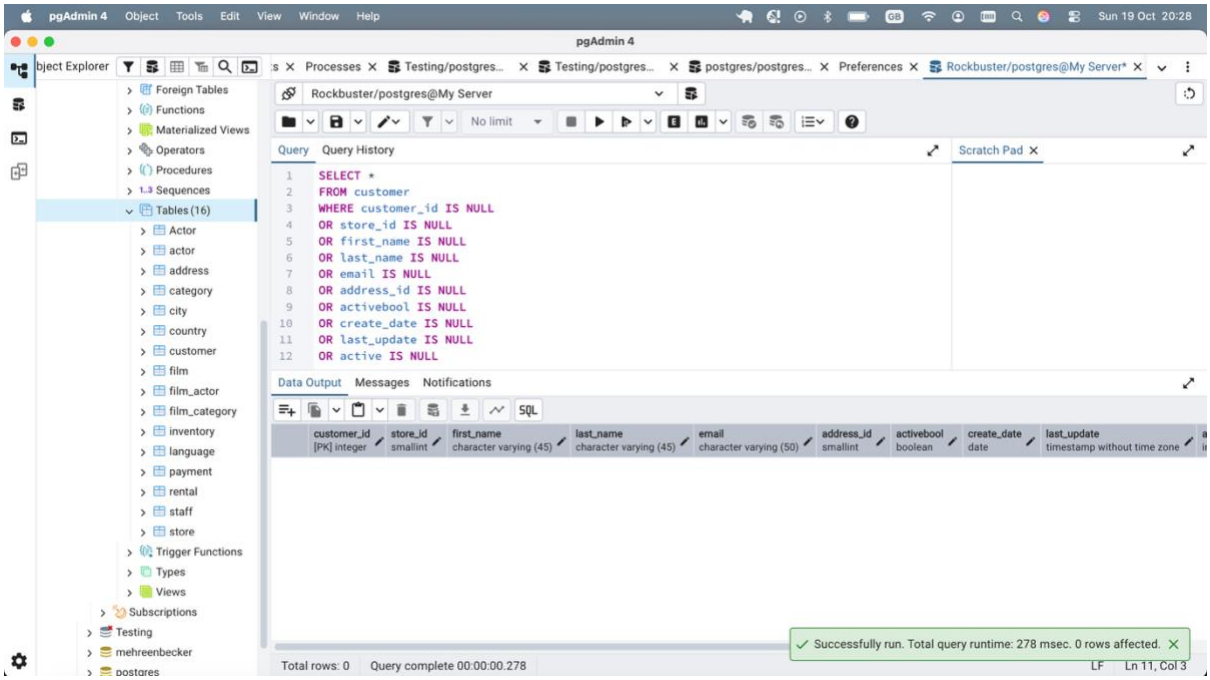

pgAdmin 4
Object Explorer
Rockbuster/postgres@My Server
Query
Query History
Scratch Pad
SELECT rental_duration FROM film GROUP BY rental_duration
Data Output
Messages
Notifications
rental_duration
smallint
1 4
2 6
3 7
4 3
5 5
Total rows: 5
Query complete 00:00:00.295
Successfully run. Total query runtime: 265 msec. 5 rows affected.
Ln 1, Col 1

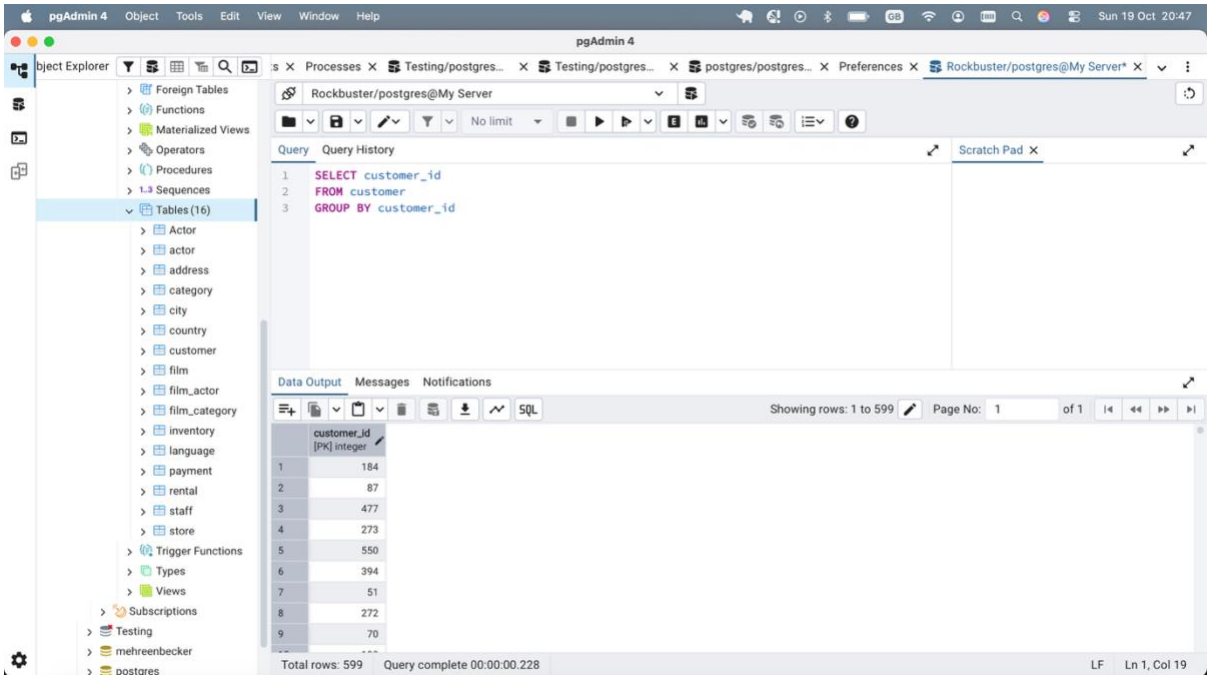


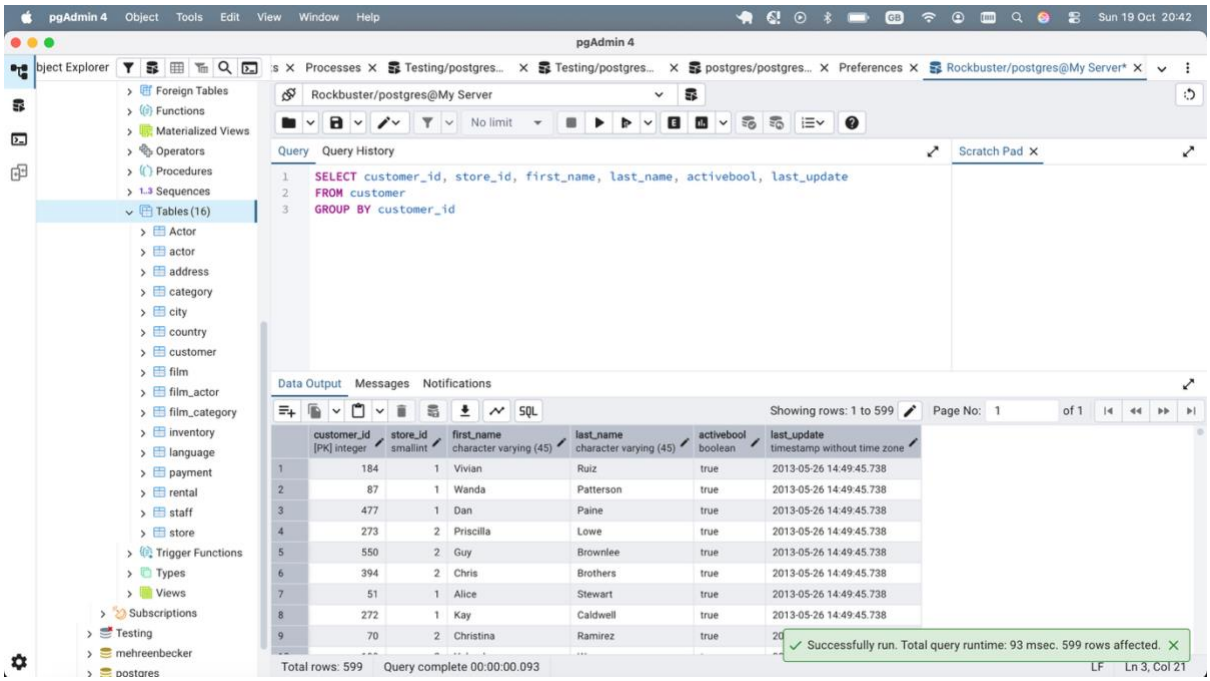


Customer Table

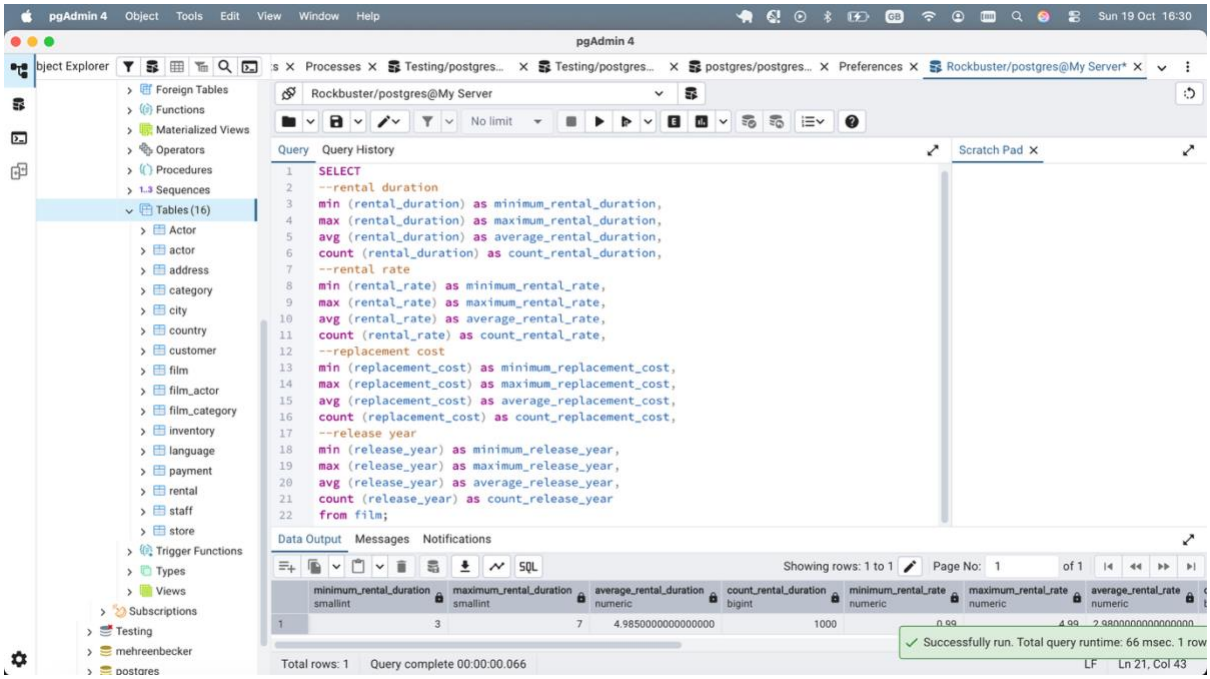

pgAdmin 4 Object Tools Edit View Window Help
Sun 19 Oct 20:33
pgAdmin 4
Object Explorer
Processes
Testing/postgres...
Testing/postgres...
postgres/postgres...
Preferences
Rockbuster/postgres@My Server*
Rockbuster/postgres@My Server
No limit
Query
Query History
Scratch Pad
SELECT
customer_id, store_id, first_name, last_name, email,
COUNT (*)
FROM customer GROUP BY customer_id, store_id, first_name, last_name, email
HAVING COUNT (*) >1
Data Output
Messages
Notifications
SQL
customer_id [PK] integer
store_id smallint
first_name character varying (45)
last_name character varying (45)
email character varying (50)
count bigint
Successfully run. Total query runtime: 75 msec. 0 rows affected.
Total rows: 0
Query complete 00:00:00.075
LF
Ln 2, Col 53
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (16)
Actor
actor
address
category
city
country
customer
film
film_actor
film_category
inventory
language
payment
rental
staff
store
Trigger Functions
Types
Views
Subscriptions
Testing
mehreenbecker


pgAdmin 4 Object Tools Edit View Window Help
Sun 19 Oct 20:28
pgAdmin 4
Object Explorer
Processes
Testing/postgres...
Testing/postgres...
postgres/postgres...
Preferences
Rockbuster/postgres@My Server*
Rockbuster/postgres@My Server
No limit
Query
Query History
Scratch Pad
SELECT *
FROM customer
WHERE customer_id IS NULL
OR store_id IS NULL
OR first_name IS NULL
OR last_name IS NULL
OR email IS NULL
OR address_id IS NULL
OR activebool IS NULL
OR create_date IS NULL
OR last_update IS NULL
OR active IS NULL
Data Output
Messages
Notifications
SQL
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_date date
last_update timestamp without time zone
Successfully run. Total query runtime: 278 msec. 0 rows affected.
Total rows: 0
Query complete 00:00:00.278
LF
Ln 11, Col 3
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (16)
Actor
actor
address
category
city
country
customer
film
film_actor
film_category
inventory
language
payment
rental
staff
store
Trigger Functions
Types
Views
Subscriptions
Testing
mehreenbecker





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.**



Rental Duration

minimum_rental_duration	maximum_rental_duration	average_rental_duration	count_rental_duration
3	7	4.9850000000000000	1000

Rental Rate

minimum_rental_rate	maximum_rental_rate	average_rental_rate	count_rental_rate
0.99	4.99	2.9800000000000000	1000

Replacement Cost

minimum_replacement_cost	maximum_replacement_cost	average_replacement_cost	count_replacement_cost
9.99	29.99	19.9840000000000000	1000

Relear Year

minimum_release_year	maximum_release_year	average_release_year	count_release_year
2006	2006	2006.0000000000000000	1000

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.**

   From my experience, as someone who has been working with SQL for about a week, I think Excel feels more straightforward for basic data profiling since it's easy to filter, sort and use quick functions on (like average etc) without having to write out the queries. However, SQL feels more powerful and is generally faster when it comes to handling

larger datasets. I'm sure that when I learn more and am more familiar with the commands, I will be able to profile data more efficiently.