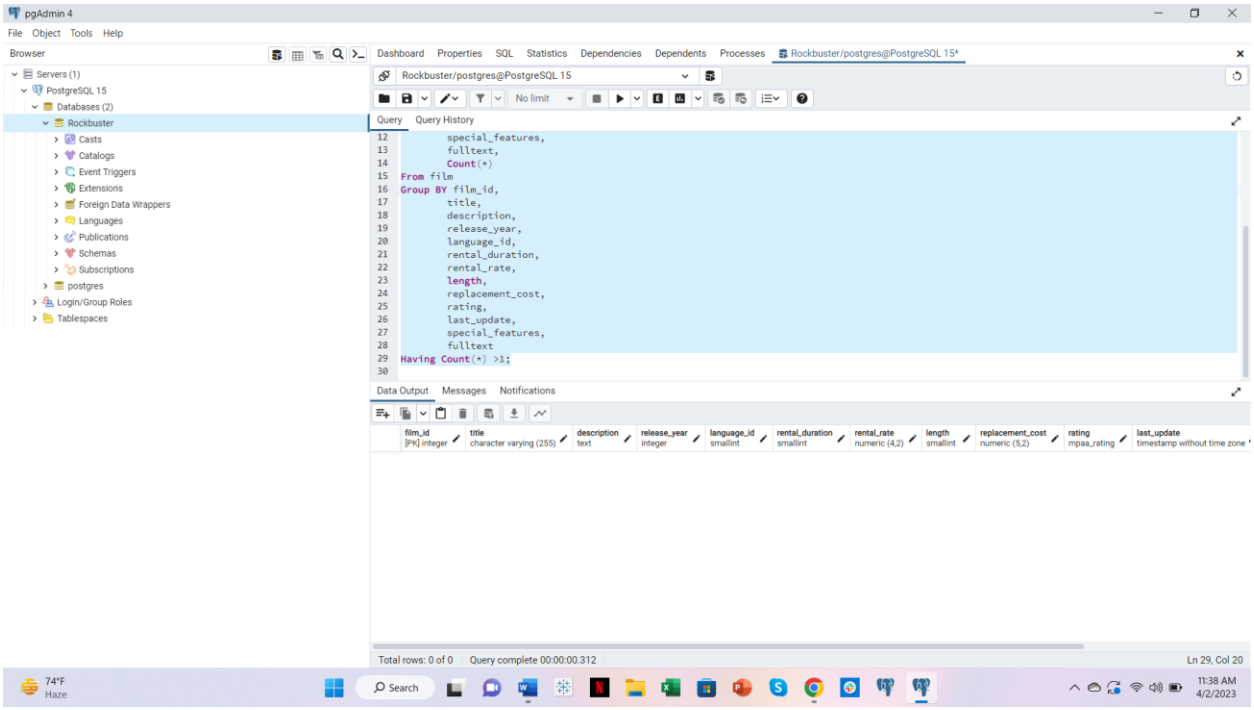
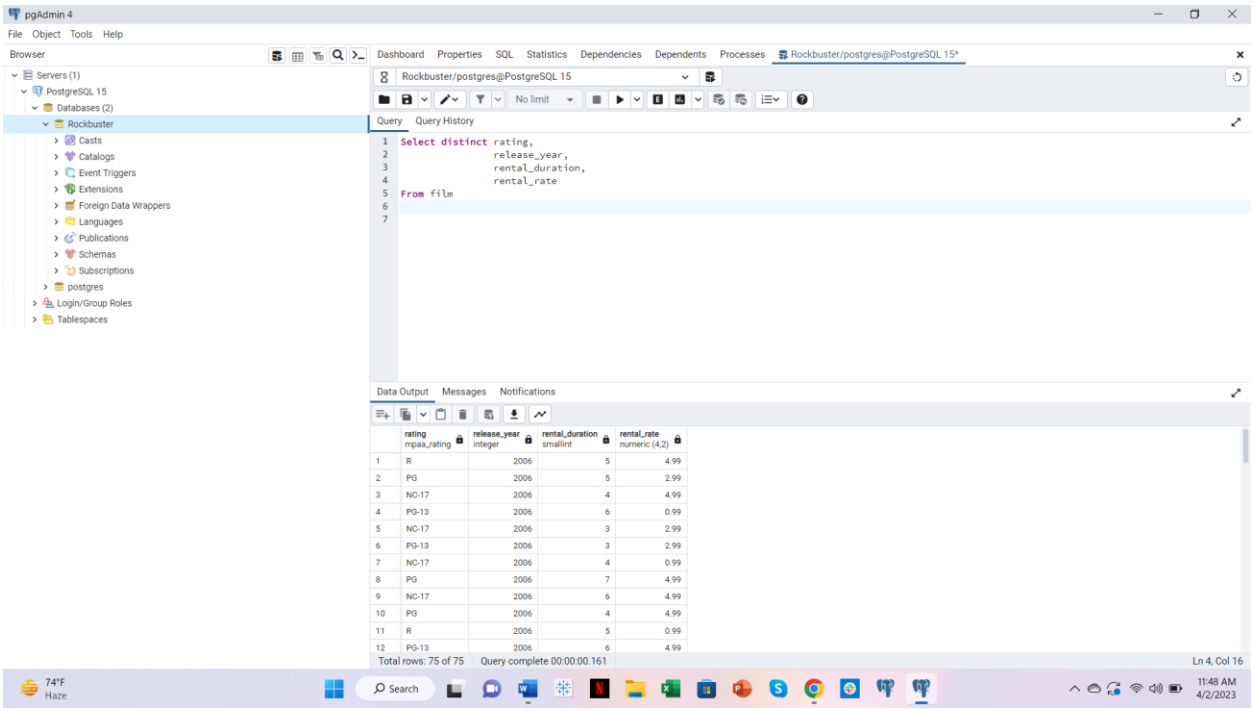
Exercise 3.6

Chrisie Hunter

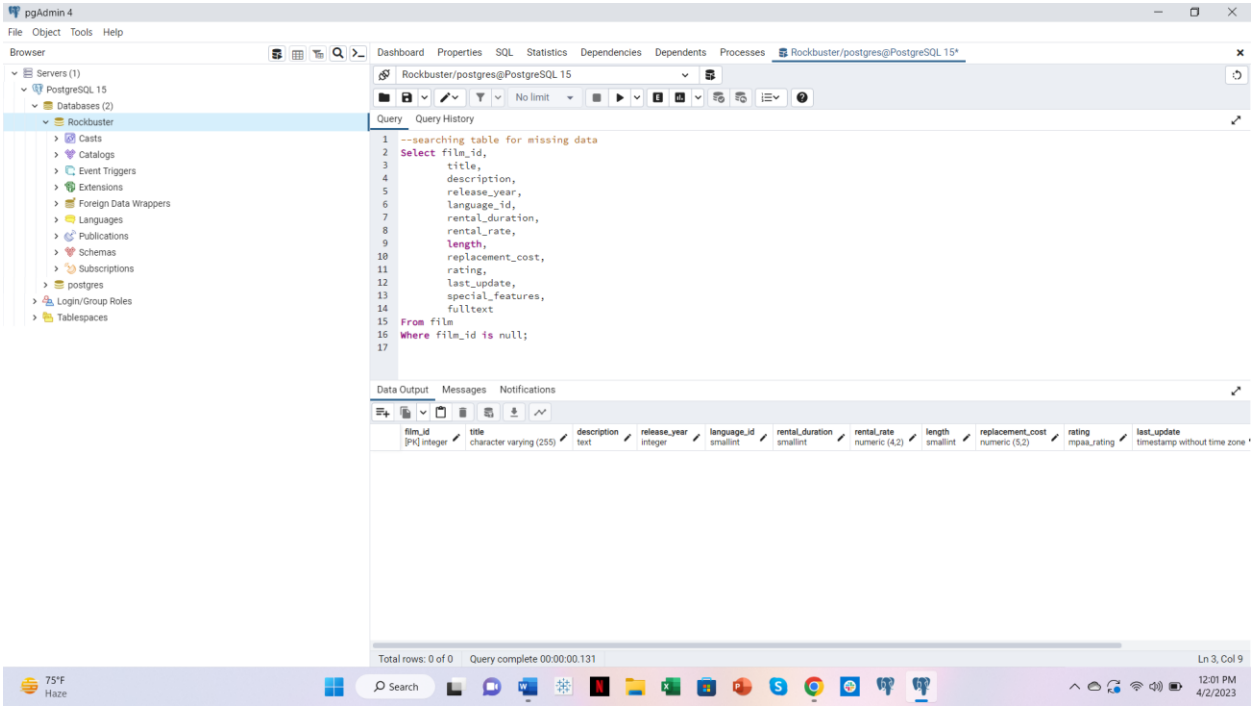
1. Checking for duplicate records in the film table.



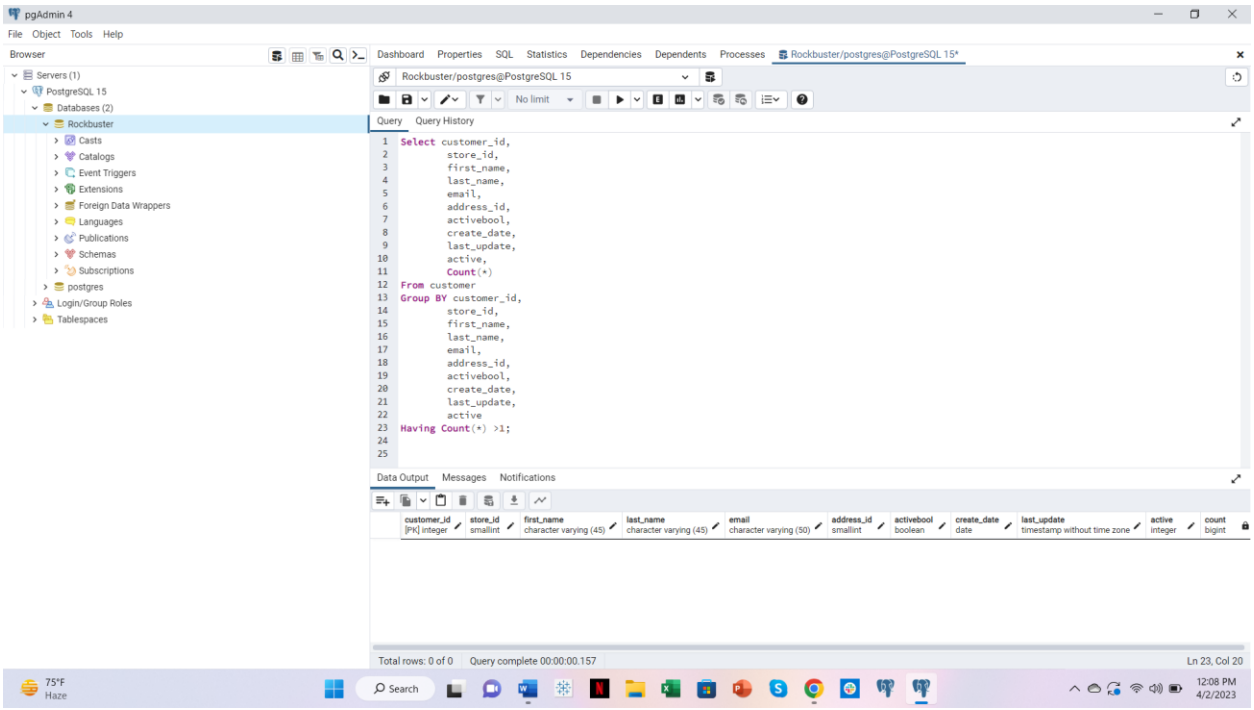
Checking several columns for non-uniform data. Doesn't appear to be any issues.



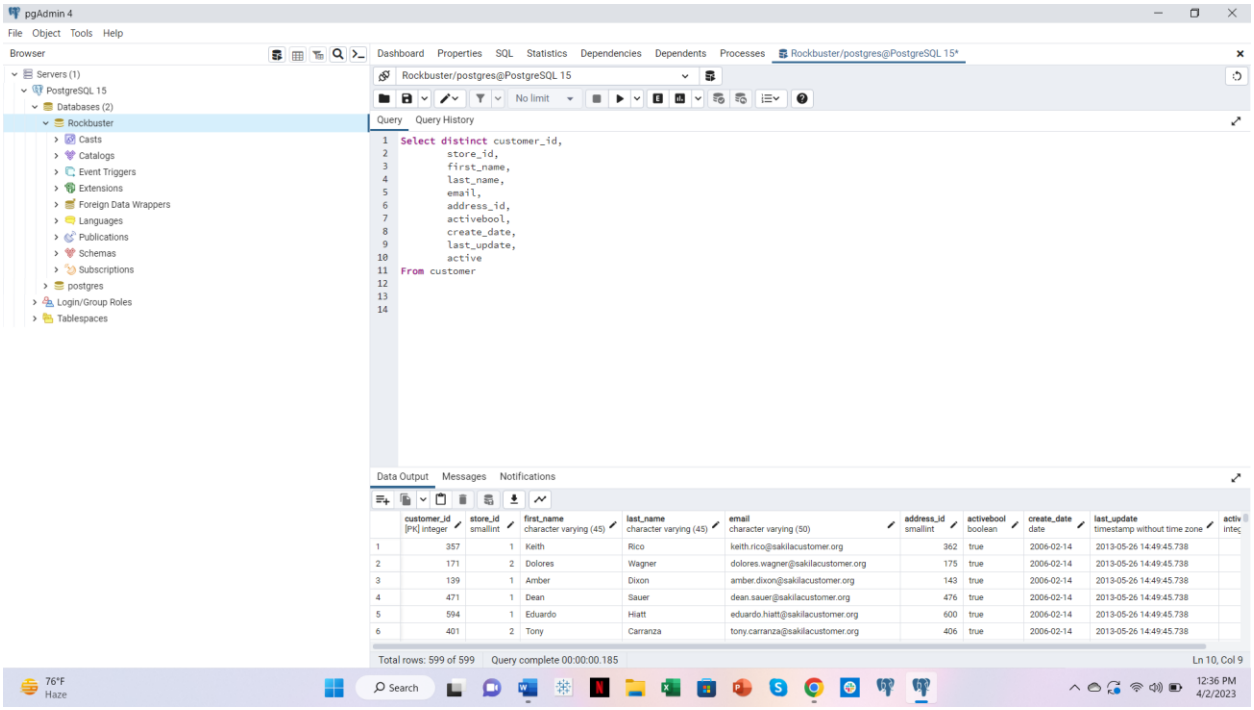
Checking for null values. This query was used to check film_id, rating, rental_duration, and release year. No missing data found.



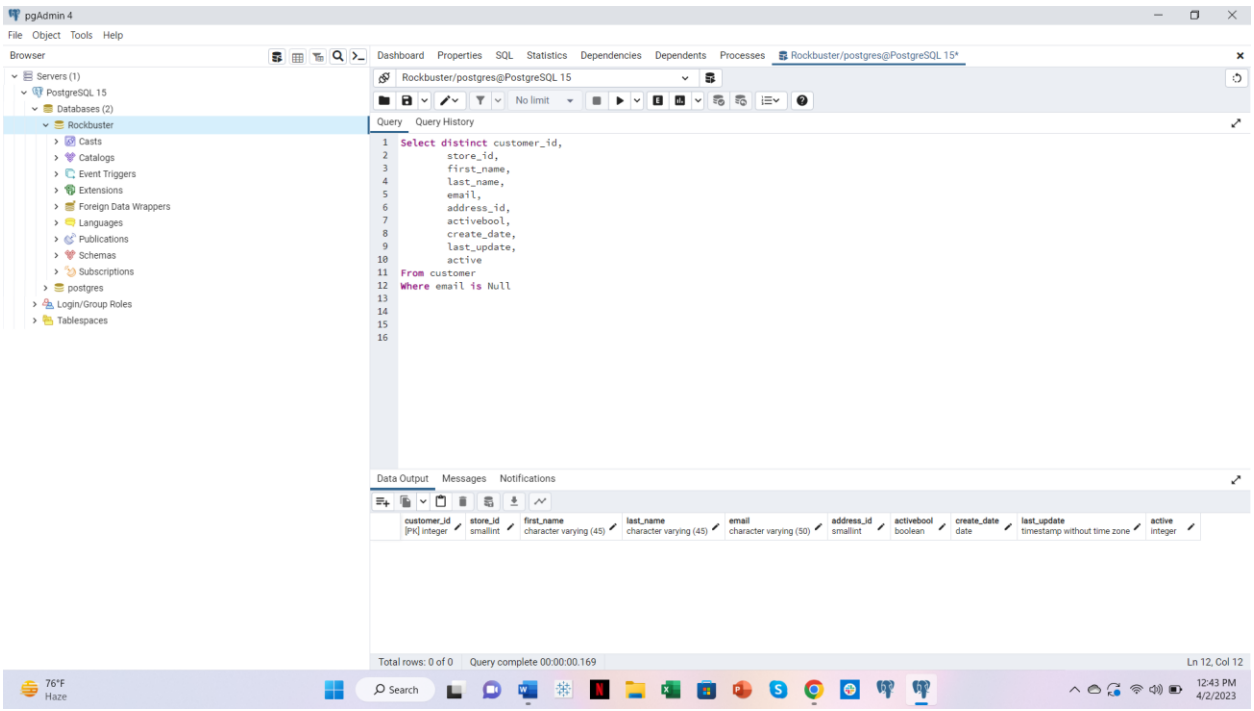
Checking for duplicates in Customer table.



Checking for non-uniform data. Nothing appears to be non-uniform.



Checking for missing data. A check was run on store_id, email, first_name, last_name



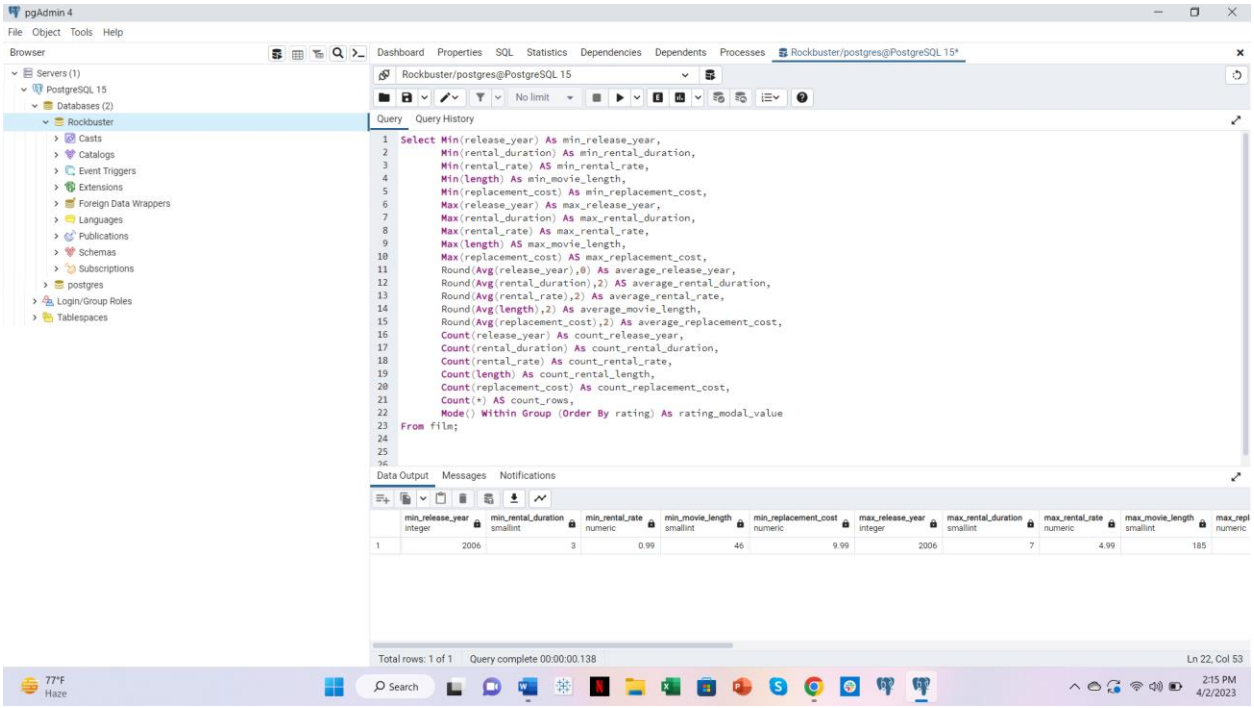
If duplicates were located, creating a view would allow you to use the Group By function, which would return unique cases only. Another option would be to delete the records using the Delete command instead of Select. Then using a Where command combined with Not In to delete the duplicate record.

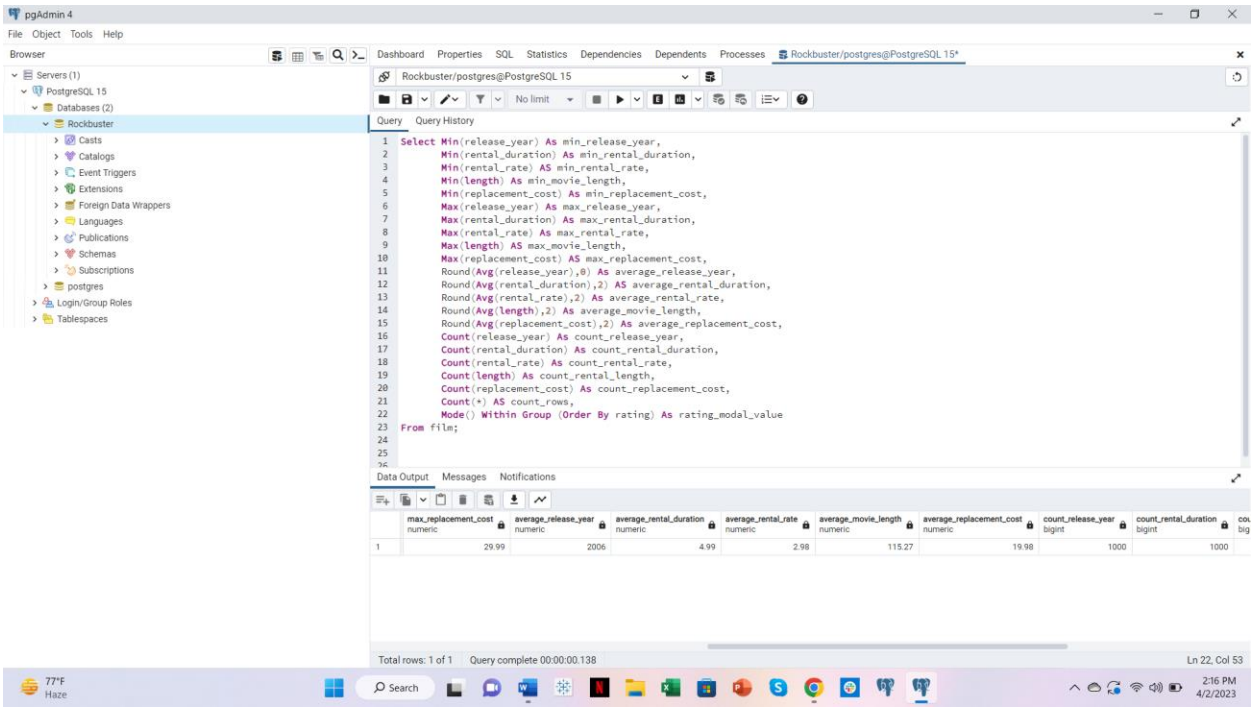
For non-uniform data, the Update command would need to be used, with Set and Where / In syntax. Set is where you tell it what you want it to be and Where / In is where you tell it which values to update.

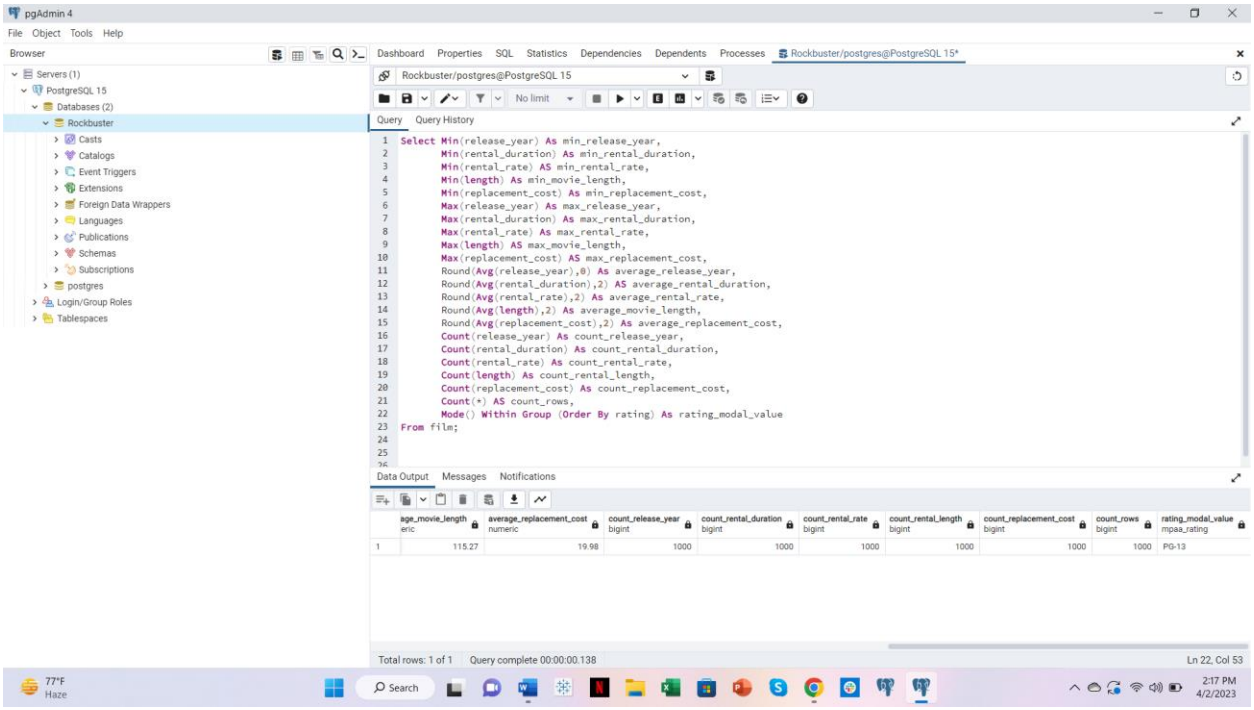
Incorrect data is far more difficult to address. This generally requires going back to the source owner. However, sometime, depending on the possible answers or range, it is easier to determine what the value should be.

Missing data is common and can be corrected through imputing values or ignore columns with a high percentage of missing data.

2. Screen shots for film_table. Screen shot 2 and 3 are just showing my columns at the bottom.




pgAdmin 4
File Object Tools Help
Browser
Servers (1)
PostgreSQL 15
Databases (2)
Rockbuster
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*
Rockbuster/postgres@PostgreSQL 15
No limit
Query Query History
Select Min(release_year) As min_release_year,
Min(rental_duration) As min_rental_duration,
Min(rental_rate) AS min_rental_rate,
Min(length) As min_movie_length,
Min(replacement_cost) As min_replacement_cost,
Max(release_year) As max_release_year,
Max(rental_duration) As max_rental_duration,
Max(rental_rate) As max_rental_rate,
Max(length) AS max_movie_length,
Max(replacement_cost) AS max_replacement_cost,
Round(Avg(release_year),0) As average_release_year,
Round(Avg(rental_duration),2) AS average_rental_duration,
Round(Avg(rental_rate),2) As average_rental_rate,
Round(Avg(length),2) As average_movie_length,
Round(Avg(replacement_cost),2) As average_replacement_cost,
Count(release_year) As count_release_year,
Count(rental_duration) As count_rental_duration,
Count(rental_rate) As count_rental_rate,
Count(length) As count_rental_length,
Count(replacement_cost) As count_replacement_cost,
Count(*) AS count_rows,
Mode() Within Group (Order By rating) As rating_modal_value
From film;
Data Output Messages Notifications
max_replacement_cost numeric
average_release_year numeric
average_rental_duration numeric
average_rental_rate numeric
average_movie_length numeric
average_replacement_cost numeric
count_release_year bigint
count_rental_duration bigint
1 29.99 2006 4.99 2.98 115.27 19.98 1000 1000
Total rows: 1 of 1 Query complete 00:00:00.138
Ln 22, Col 53
77°F Haze
Search
2:16 PM 4/2/2023


pgAdmin 4
File Object Tools Help
Browser
Servers (1)
PostgreSQL 15
Databases (2)
Rockbuster
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
postgres
Login/Group Roles
Tablespaces
Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/postgres@PostgreSQL 15*
Rockbuster/postgres@PostgreSQL 15
No limit
Query Query History
Select Min(release_year) As min_release_year,
Min(rental_duration) As min_rental_duration,
Min(rental_rate) AS min_rental_rate,
Min(length) As min_movie_length,
Min(replacement_cost) As min_replacement_cost,
Max(release_year) As max_release_year,
Max(rental_duration) As max_rental_duration,
Max(rental_rate) As max_rental_rate,
Max(length) AS max_movie_length,
Max(replacement_cost) AS max_replacement_cost,
Round(Avg(release_year),0) As average_release_year,
Round(Avg(rental_duration),2) AS average_rental_duration,
Round(Avg(rental_rate),2) As average_rental_rate,
Round(Avg(length),2) As average_movie_length,
Round(Avg(replacement_cost),2) As average_replacement_cost,
Count(release_year) As count_release_year,
Count(rental_duration) As count_rental_duration,
Count(rental_rate) As count_rental_rate,
Count(length) As count_rental_length,
Count(replacement_cost) As count_replacement_cost,
Count(*) AS count_rows,
Mode() Within Group (Order By rating) As rating_modal_value
From film;
Data Output Messages Notifications
age_movie_length eric
average_replacement_cost numeric
count_release_year bigint
count_rental_duration bigint
count_rental_rate bigint
count_rental_length bigint
count_replacement_cost bigint
count_rows bigint
rating_modal_value mpaa_rating
1 115.27 19.98 1000 1000 1000 1000 1000 1000 PG-13
Total rows: 1 of 1 Query complete 00:00:00.138
Ln 22, Col 53
77°F Haze
Search
2:17 PM 4/2/2023

Screen shots for customer table.