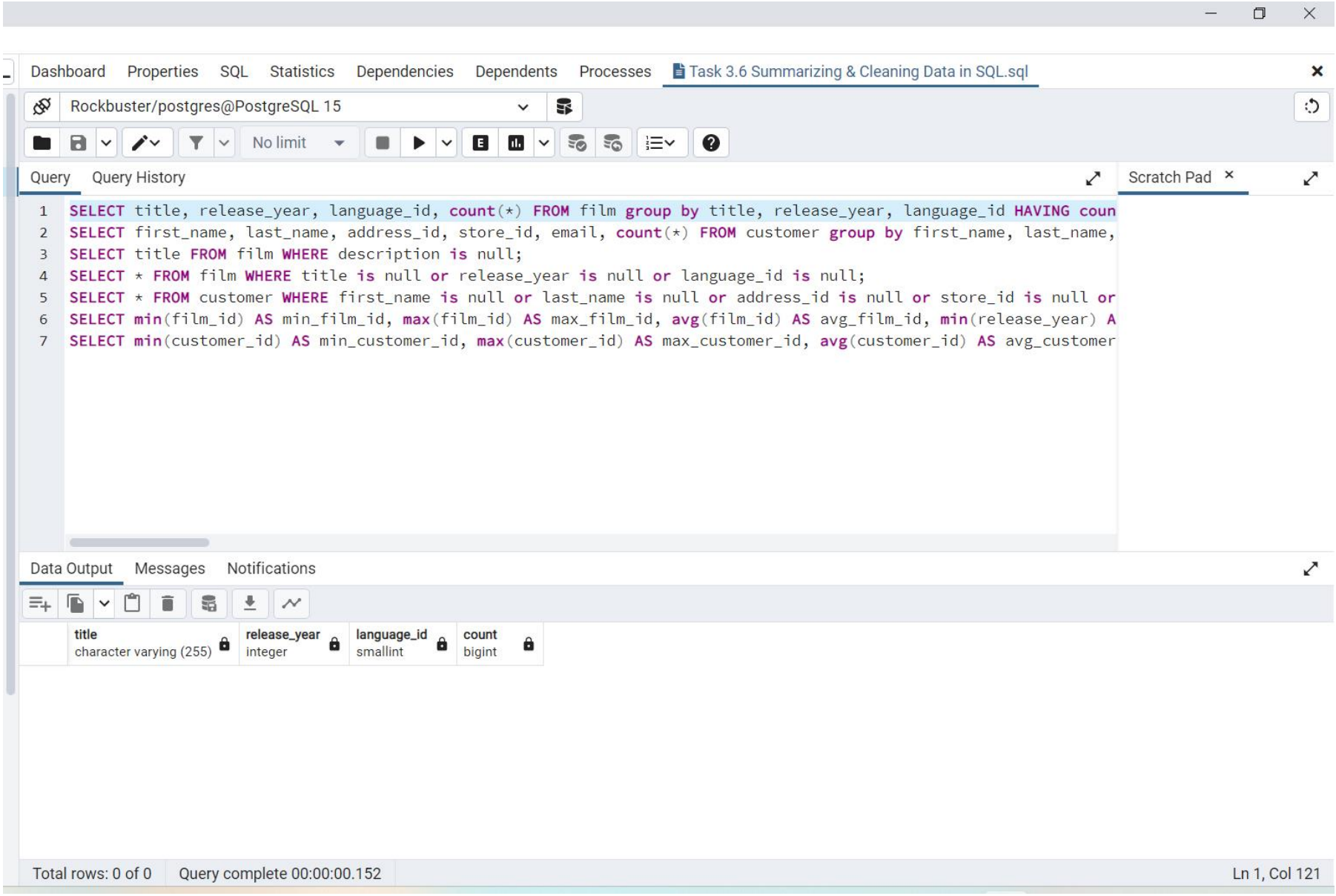
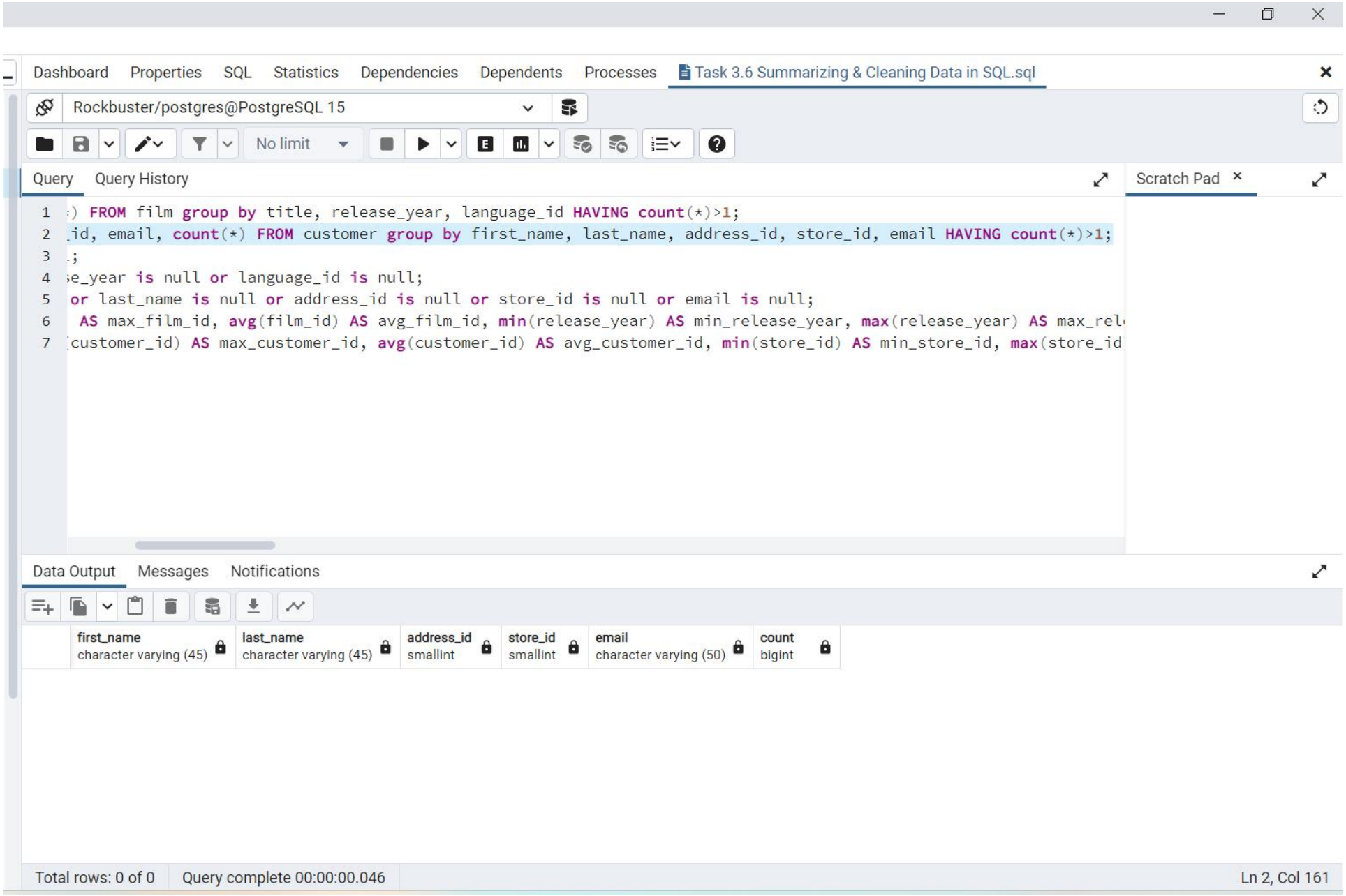
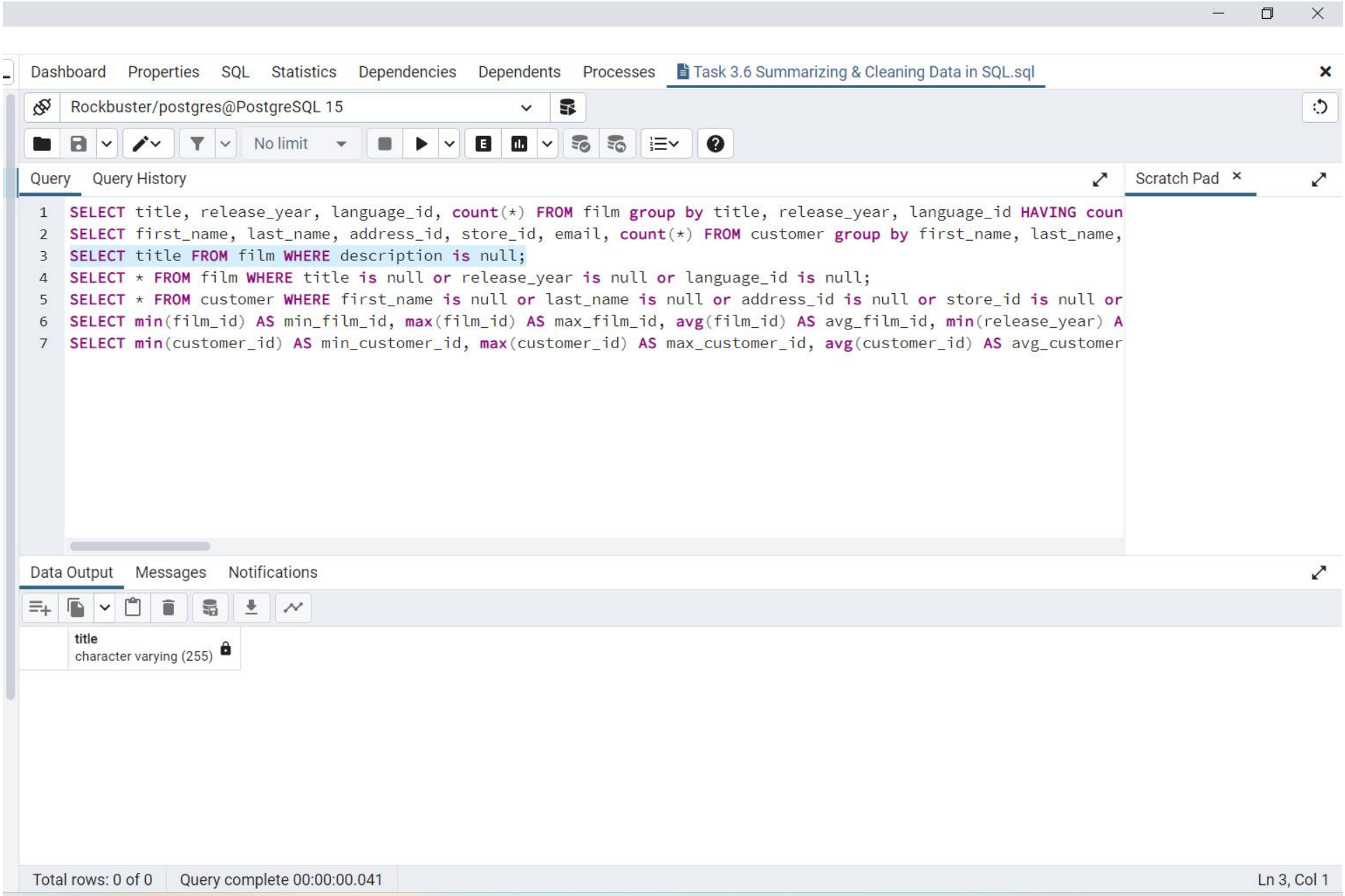
Summarizing & Cleaning Data in SQL

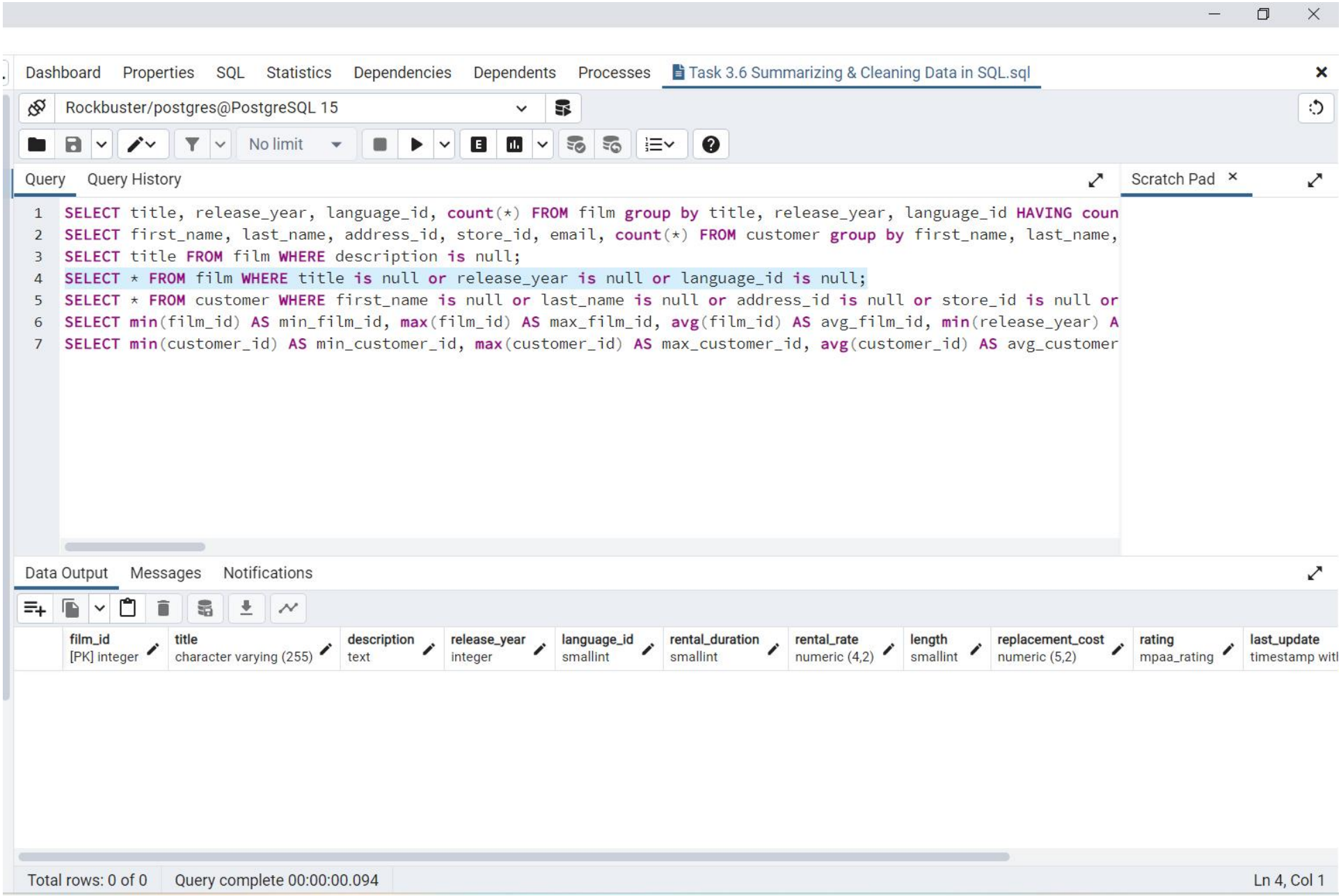
Step 1

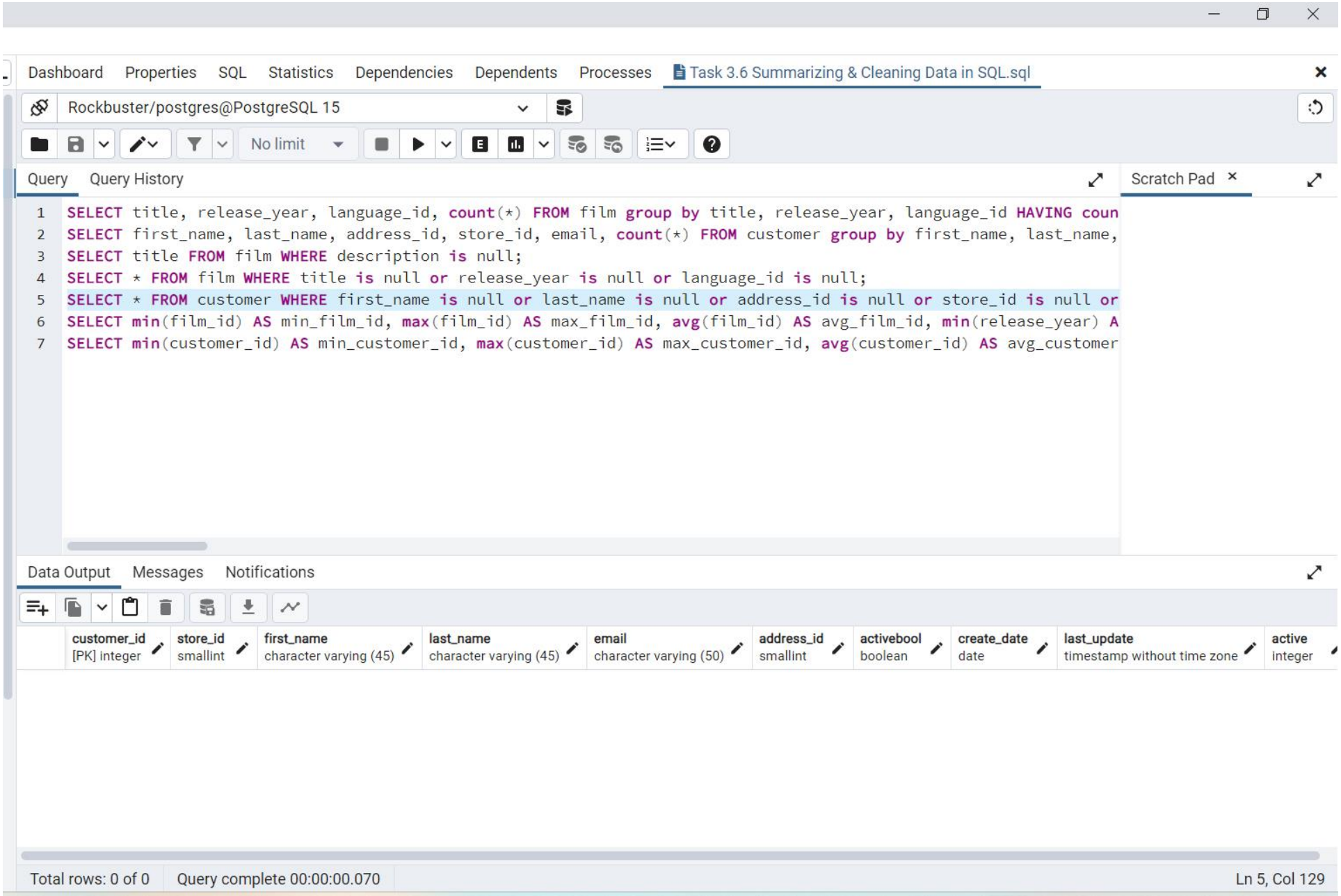
No duplicates, no missing data as well as non-uniform




Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.6 Summarizing & Cleaning Data in SQL.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1) FROM film group by title, release_year, language_id HAVING count(*)>1;
2 id, email, count(*) FROM customer group by first_name, last_name, address_id, store_id, email HAVING count(*)>1;
3 ;
4 e_year is null or language_id is null;
5 or last_name is null or address_id is null or store_id is null or email is null;
6 AS max_film_id, avg(film_id) AS avg_film_id, min(release_year) AS min_release_year, max(release_year) AS max_rel
7 customer_id) AS max_customer_id, avg(customer_id) AS avg_customer_id, min(store_id) AS min_store_id, max(store_id
Data Output
Messages
Notifications
first_name character varying (45)
last_name character varying (45)
address_id smallint
store_id smallint
email character varying (50)
count bigint
Total rows: 0 of 0
Query complete 00:00:00.046
Ln 2, Col 161


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.6 Summarizing & Cleaning Data in SQL.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT title, release_year, language_id, count(*) FROM film group by title, release_year, language_id HAVING coun
2 SELECT first_name, last_name, address_id, store_id, email, count(*) FROM customer group by first_name, last_name,
3 SELECT title FROM film WHERE description is null;
4 SELECT * FROM film WHERE title is null or release_year is null or language_id is null;
5 SELECT * FROM customer WHERE first_name is null or last_name is null or address_id is null or store_id is null or
6 SELECT min(film_id) AS min_film_id, max(film_id) AS max_film_id, avg(film_id) AS avg_film_id, min(release_year) A
7 SELECT min(customer_id) AS min_customer_id, max(customer_id) AS max_customer_id, avg(customer_id) AS avg_customer
Data Output
Messages
Notifications
title
character varying (255)
Total rows: 0 of 0
Query complete 00:00:00.041
Ln 3, Col 1


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.6 Summarizing & Cleaning Data in SQL.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT title, release_year, language_id, count(*) FROM film group by title, release_year, language_id HAVING coun
2 SELECT first_name, last_name, address_id, store_id, email, count(*) FROM customer group by first_name, last_name,
3 SELECT title FROM film WHERE description is null;
4 SELECT * FROM film WHERE title is null or release_year is null or language_id is null;
5 SELECT * FROM customer WHERE first_name is null or last_name is null or address_id is null or store_id is null or
6 SELECT min(film_id) AS min_film_id, max(film_id) AS max_film_id, avg(film_id) AS avg_film_id, min(release_year) A
7 SELECT min(customer_id) AS min_customer_id, max(customer_id) AS max_customer_id, avg(customer_id) AS avg_customer
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
replacement_cost numeric (5,2)
rating mpaa_rating
last_update timestamp witl
Total rows: 0 of 0
Query complete 00:00:00.094
Ln 4, Col 1


Dashboard
Properties
SQL
Statistics
Dependencies
Dependents
Processes
Task 3.6 Summarizing & Cleaning Data in SQL.sql
Rockbuster/postgres@PostgreSQL 15
No limit
Query
Query History
Scratch Pad
1 SELECT title, release_year, language_id, count(*) FROM film group by title, release_year, language_id HAVING coun
2 SELECT first_name, last_name, address_id, store_id, email, count(*) FROM customer group by first_name, last_name,
3 SELECT title FROM film WHERE description is null;
4 SELECT * FROM film WHERE title is null or release_year is null or language_id is null;
5 SELECT * FROM customer WHERE first_name is null or last_name is null or address_id is null or store_id is null or
6 SELECT min(film_id) AS min_film_id, max(film_id) AS max_film_id, avg(film_id) AS avg_film_id, min(release_year) A
7 SELECT min(customer_id) AS min_customer_id, max(customer_id) AS max_customer_id, avg(customer_id) AS avg_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_date date
last_update timestamp without time zone
active integer
Total rows: 0 of 0
Query complete 00:00:00.070
Ln 5, Col 129

Step 2

Descriptive statistics for film table. For numerical columns, the finding of minimum, maximum, and average values

```
SELECT min(film_id) AS min_film_id, max(film_id) AS max_film_id, avg(film_id) AS avg_film_id, min(release_year) AS min_release_year, max(release_year) AS max_release_year, avg(release_year) AS avg_release_year, MODE() WITHIN GROUP (ORDER BY language_id) as mode_language_id,min(rental_duration) AS min_rental_duration, max(rental_duration) AS max_rental_duration, avg(rental_duration) AS avg_rental_duration,min(rental_rate) AS min_rental_rate, max(rental_rate) AS max_rental_rate, avg(rental_rate) AS avg_rental_rate,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 mode_rating FROM film;
```

Descriptive statistics for Customer table. For numerical columns, the finding of minimum, maximum, and average values

```
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, MODE() within group (order by activebool) AS mode_activebool FROM customer;
```

Film	
min_film_id	1
max_film_id	1001
avg_film_id	501
min_release_year	2006
max_release_year	2019
avg_release_year	2006.012987
mode_language_id	1
min_rental_duration	3
max_rental_duration	7
avg_rental_duration	4.983016983
min_rental_rate	0.99
max_rental_rate	4.99
avg_rental_rate	2.982007992
min_length	46
max_length	185
avg_length	115.272
min_replacement_cost	9.99
max_replacement_cost	29.99
avg_replacement_cost	19.98400599
mode_rating	PG-13

Customer	
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
mode_activebool	TRUE

Step 3

The only way to make SQL at this point is better that excel is practice, no doubt with mastering commands or syntax SQL will be faster and more accurate specially when it comes to huge database