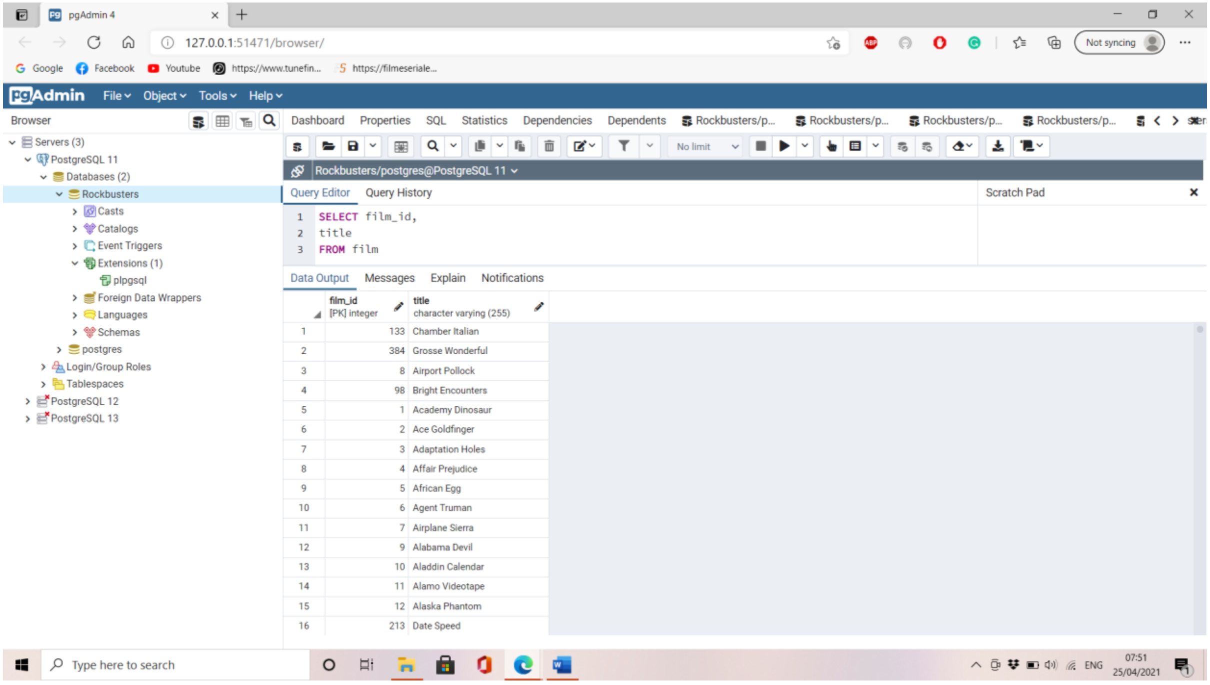
Step 1

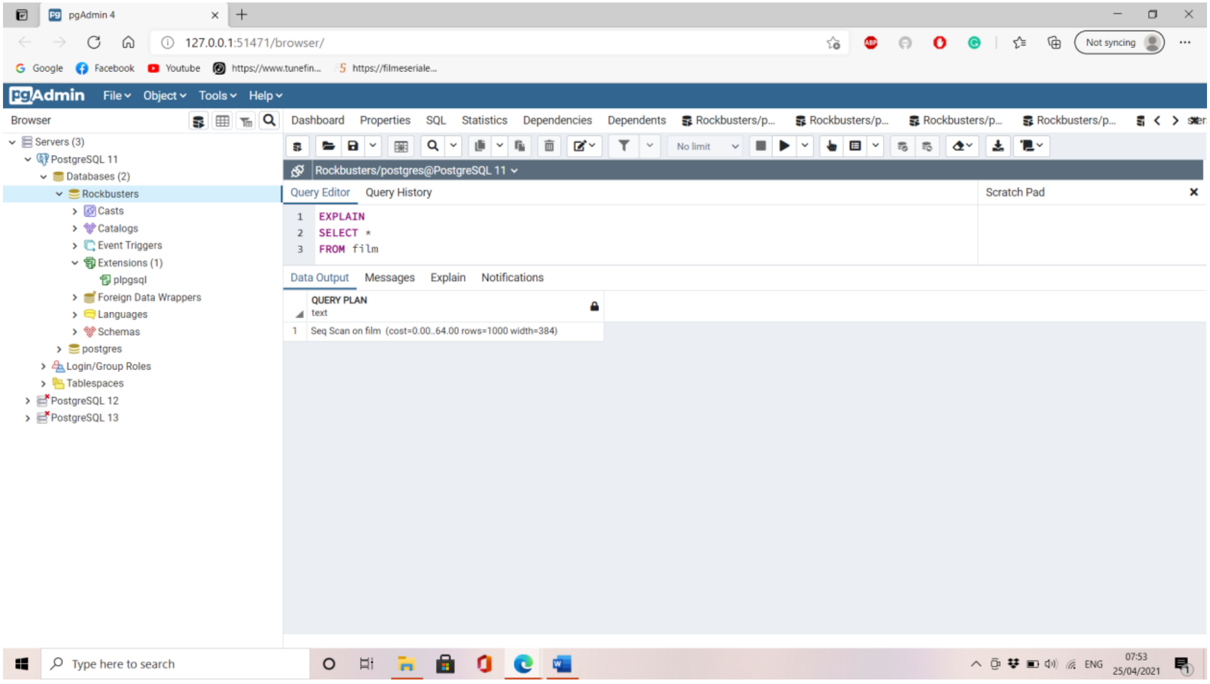
Refining Your Query:

- You need to get some data from the “film” table and decide to use the query SELECT * FROM film.
- You realize that only the “film_id” and “title” columns are needed. Write a new query that selects only those 2 columns.
- Compare the cost of the original query and the revised query, and write a few sentences explaining the comparison. Can you suggest any ways to optimize this query?



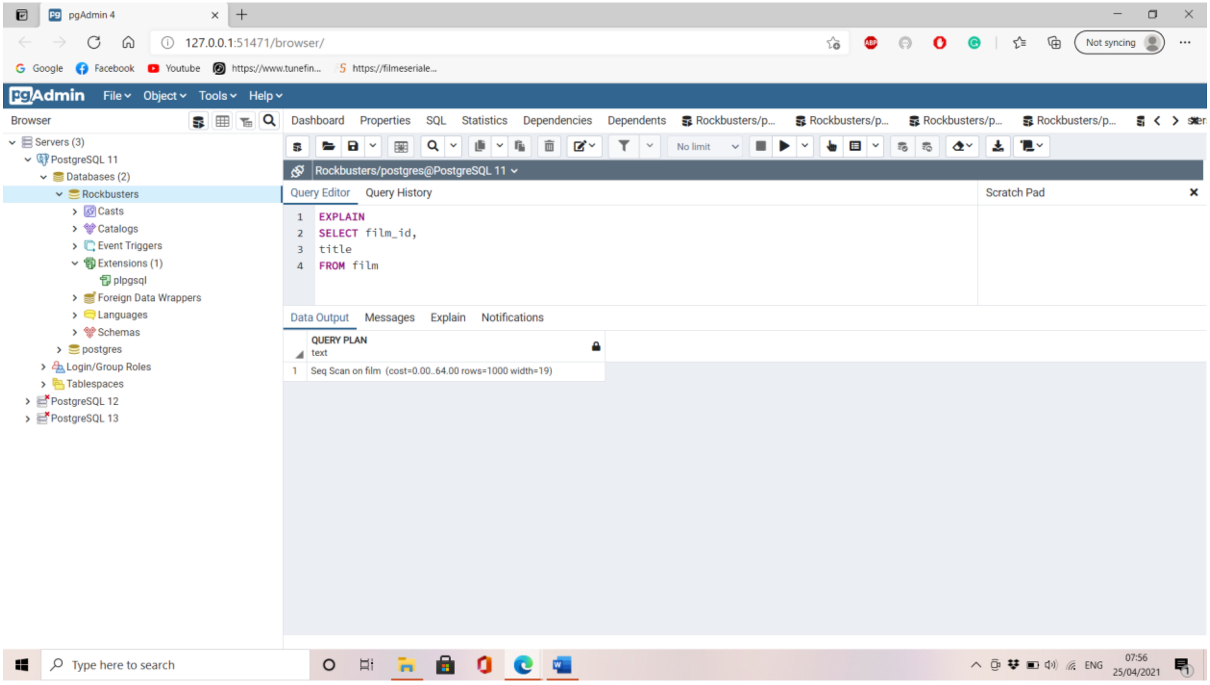
```
SELECT film_id,
title
FROM film
```

	film_id [PK] integer	title character varying (255)
1	133	Chamber Italian
2	384	Grosse Wonderful
3	8	Airport Pollock
4	98	Bright Encounters
5	1	Academy Dinosaur
6	2	Ace Goldfinger
7	3	Adaptation Holes
8	4	Affair Prejudice
9	5	African Egg
10	6	Agent Truman
11	7	Airplane Sierra
12	9	Alabama Devil
13	10	Aladdin Calendar
14	11	Alamo Videotape
15	12	Alaska Phantom
16	213	Date Speed



```
EXPLAIN
SELECT *
FROM film
```

	QUERY PLAN text
1	Seq Scan on film (cost=0.00..64.00 rows=1000 width=384)



```
EXPLAIN
SELECT film_id,
title
FROM film
```

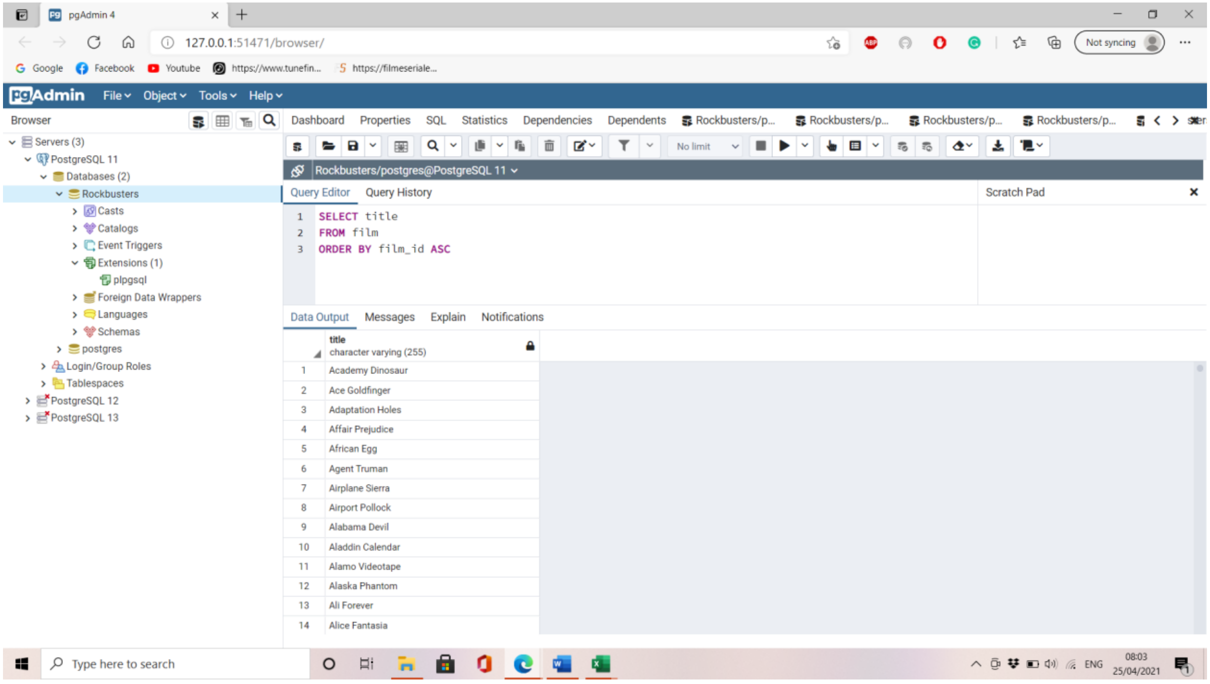
	QUERY PLAN text
1	Seq Scan on film (cost=0.00..64.00 rows=1000 width=19)

- The cost for the original query was 0-64 for 1000 rows and 384 columns and the cost for the revised query is 0-62 with 1000 rows and 19 columns. They are similar in cost but refining the revised query more by ordering it could save on cost a little more.

Step 2

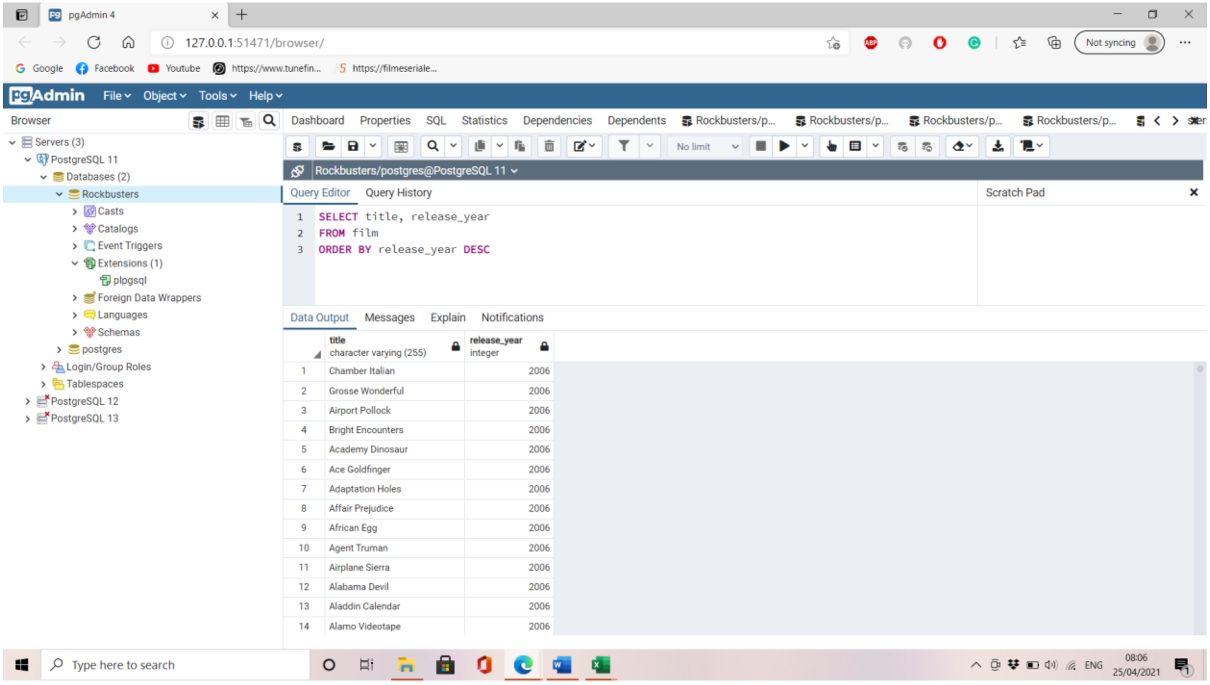
Ordering the Data:

- In the pgAdmin Query Tool, run a query that selects every film from the “film” table, with the movies sorted by title from A to Z, then by most recent release year, and then by highest to lowest rental rate.
- Extract the data output of your query into a csv file for the film collection department to analyse in Excel. (You may need to explore how to save your output as a csv file in the Query Tool.)



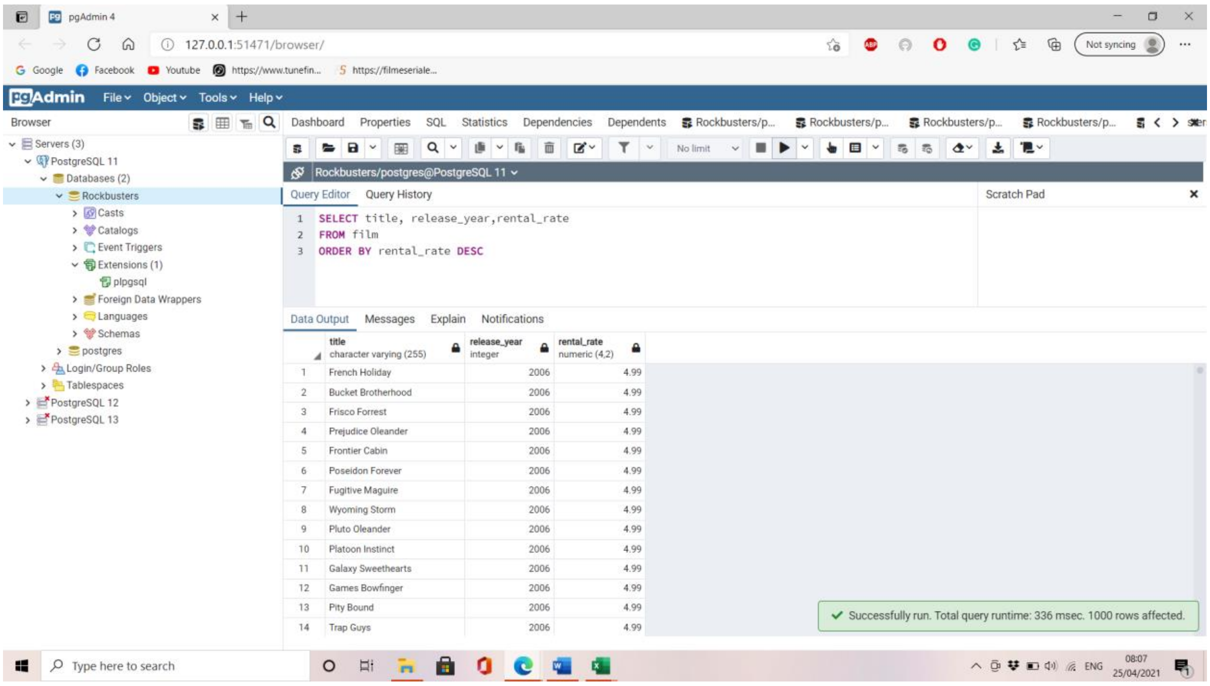
```sql
SELECT title
FROM film
ORDER BY film_id ASC
```

	title character varying (255)
1	Academy Dinosaur
2	Ace Goldfinger
3	Adaptation Holes
4	Affair Prejudice
5	African Egg
6	Agent Truman
7	Airplane Sierra
8	Airport Pollock
9	Alabama Devil
10	Aladdin Calendar
11	Alamo Videotape
12	Alaska Phantom
13	Ali Forever
14	Alice Fantasia



```sql
SELECT title, release_year
FROM film
ORDER BY release_year DESC
```

	title character varying (255)	release_year integer
1	Chamber Italian	2006
2	Grosse Wonderful	2006
3	Airport Pollock	2006
4	Bright Encounters	2006
5	Academy Dinosaur	2006
6	Ace Goldfinger	2006
7	Adaptation Holes	2006
8	Affair Prejudice	2006
9	African Egg	2006
10	Agent Truman	2006
11	Airplane Sierra	2006
12	Alabama Devil	2006
13	Aladdin Calendar	2006
14	Alamo Videotape	2006



```
SELECT title, release_year,rental_rate
FROM film
ORDER BY rental_rate DESC
```

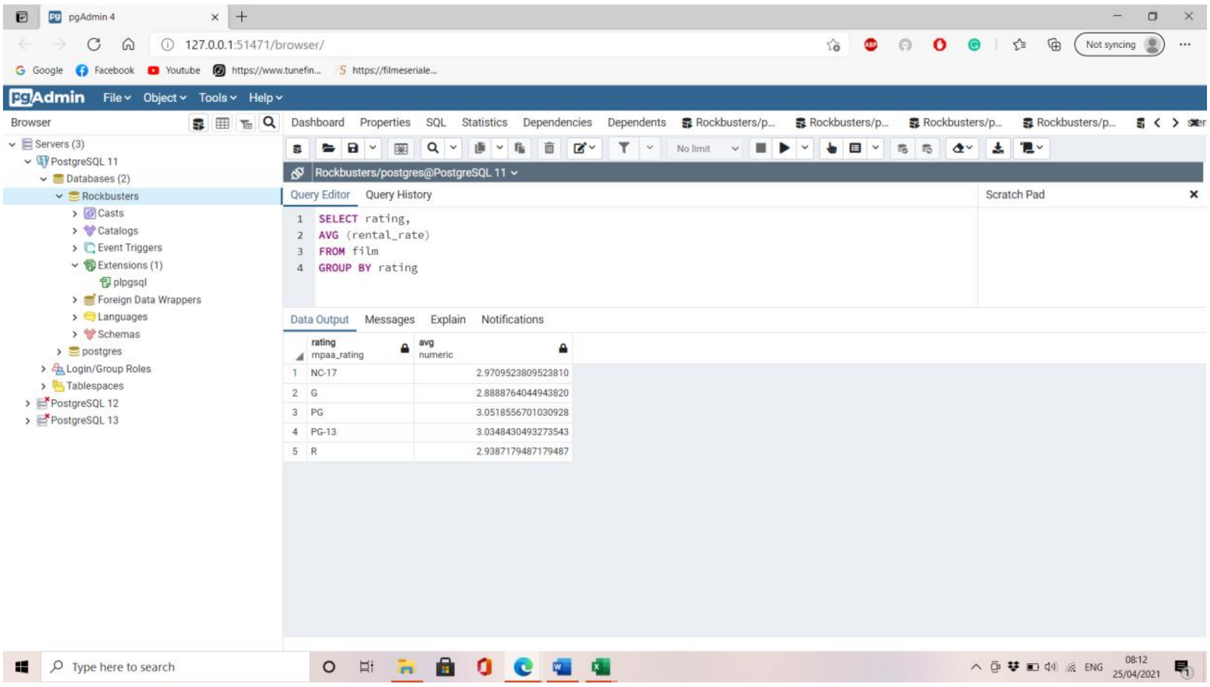
	title character varying (255)	release_year integer	rental_rate numeric (4,2)
1	French Holiday	2006	4.99
2	Bucket Brotherhood	2006	4.99
3	Frisco Forrest	2006	4.99
4	Prejudice Oleander	2006	4.99
5	Frontier Cabin	2006	4.99
6	Poseidon Forever	2006	4.99
7	Fugitive Maguire	2006	4.99
8	Wyoming Storm	2006	4.99
9	Pluto Oleander	2006	4.99
10	Platoon Instinct	2006	4.99
11	Galaxy Sweethearts	2006	4.99
12	Games Bowfinger	2006	4.99
13	Pity Bound	2006	4.99
14	Trap Guys	2006	4.99

```
Successfully run. Total query runtime: 336 msec. 1000 rows affected.
```

Step 3

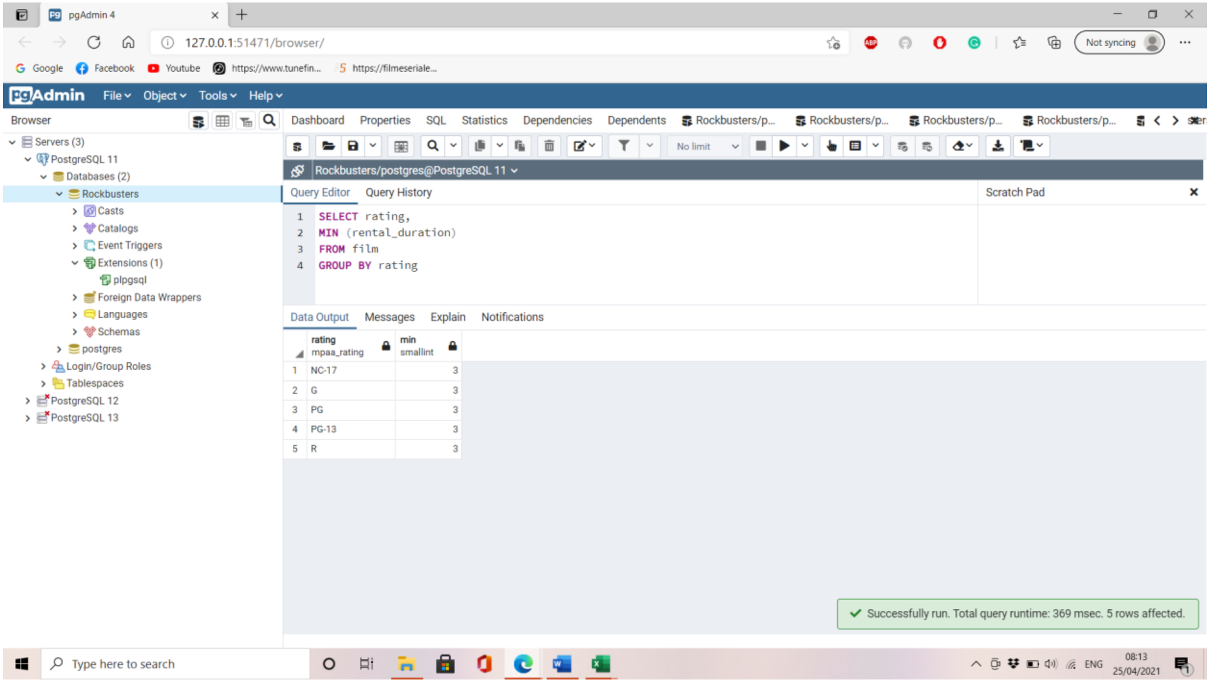
Grouping Data:

- The strategy department has asked you the questions below. Write a SQL query to retrieve the correct answers, then extract your results as a csv file.
- What is the average rental rate for each rating category?
- What are the minimum and maximum rental durations for each rating category?



```
SELECT rating,
AVG (rental_rate)
FROM film
GROUP BY rating
```

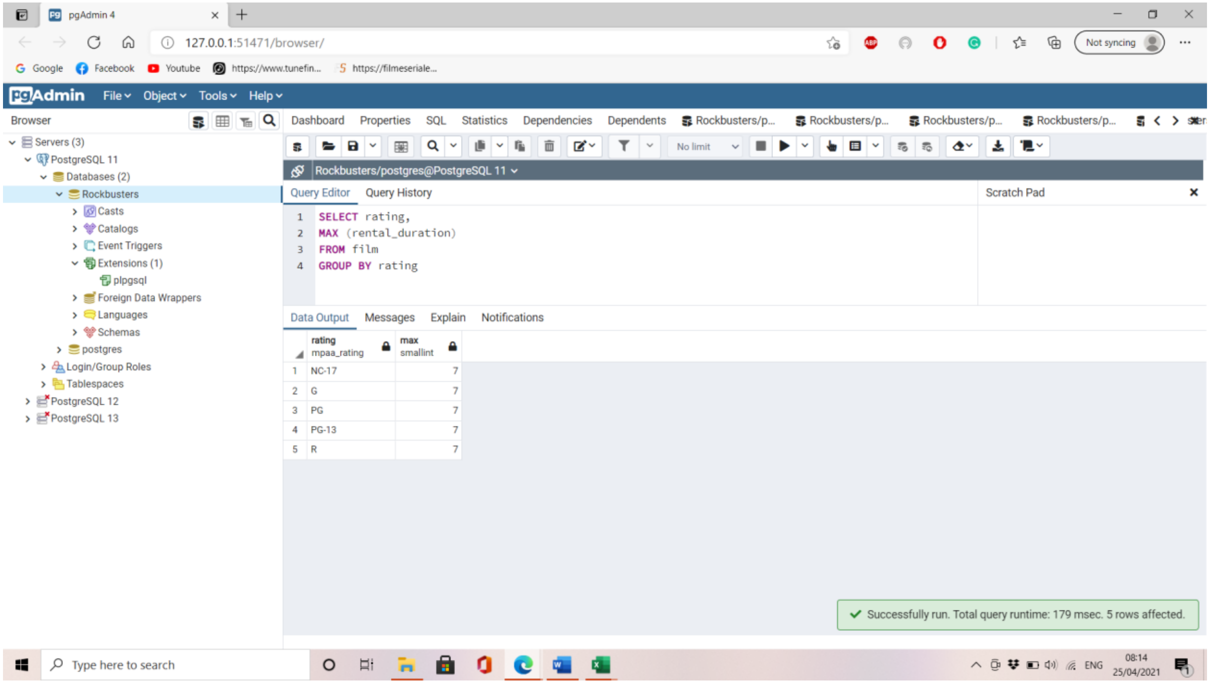
	rating mpaa_rating	avg numeric
1	NC-17	2.9709523809523810
2	G	2.8888764044943820
3	PG	3.0518556701030928
4	PG-13	3.0348430493273543
5	R	2.9387179487179487



```
SELECT rating,
MIN (rental_duration)
FROM film
GROUP BY rating
```

	rating mpaa_rating	min smallint
1	NC-17	3
2	G	3
3	PG	3
4	PG-13	3
5	R	3

✓ Successfully run. Total query runtime: 369 msec. 5 rows affected.



```
SELECT rating,
MAX (rental_duration)
FROM film
GROUP BY rating
```

	rating mpaa_rating	max smallint
1	NC-17	7
2	G	7
3	PG	7
4	PG-13	7
5	R	7

✓ Successfully run. Total query runtime: 179 msec. 5 rows affected.

Step 4

Database Migration:

- Your team has decided to use an external tool to collect data on user behaviour in the new Rockbuster Android app. Data collected from this new source will need to be loaded into the data warehouse before you can analyse it.
- Can you outline the procedure for migrating the data and who will be responsible for it?

➢ The ETL procedure should be a data engineer´s task but knowing the process as a data analyst would help in coordinating the process. As first step data should be extracted from different data sources (for checking user behaviour would be helpful to see, what customer search mostly, payment tracking of a specific period, most rented movies). Then it´s transformed in an appropriate format and then loaded in the data warehouse.

- What problems do you foresee if you start analysing the data before it has been loaded into the data warehouse?

➢ Data might be incomplete or duplicate as it comes from different sources, could be involuntary manipulated.