July 4, 2022

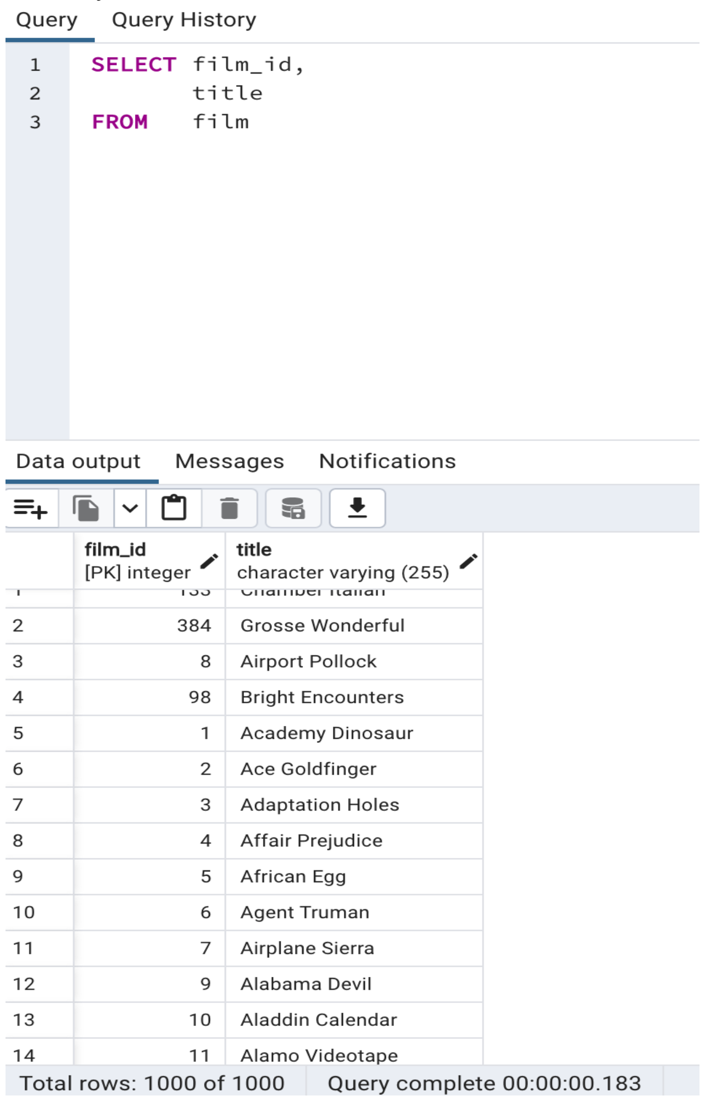
Data Immersion
Database & SQL for Analysis
3.4: Database Querying in SQL
David Guillen Aroche

1. Refining query:

SELECT *
FROM *film*

1a.- Select only the following columns: *film_id* and *title*.

SELECT *film_id,*
          title
FROM *film*



Query   Query History

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

Data output   Messages   Notifications

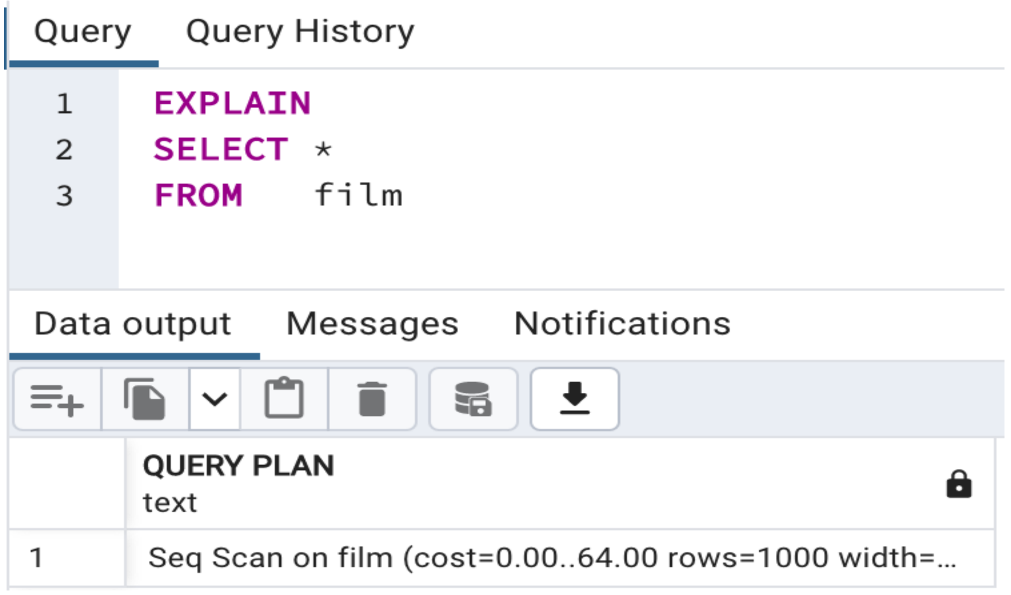
	film_id [PK] integer	title character varying (255)
1	133	Chamber Italian
2	384	Grosse Wonderful
3	8	Airport Pollock
4	98	Bright Encounters
5	1	Academy Dinosaur
6	2	Ace Goldfinger
7	3	Adaptation Holes
8	4	Affair Prejudice
9	5	African Egg
10	6	Agent Truman
11	7	Airplane Sierra
12	9	Alabama Devil
13	10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

Total rows: 1000 of 1000   Query complete 00:00:00.183

1b.- Compare cost of original query and the optimized query:

- EXPLAIN
  SELECT *
  FROM *film*

This query has a cost of 64, see screenshot below:

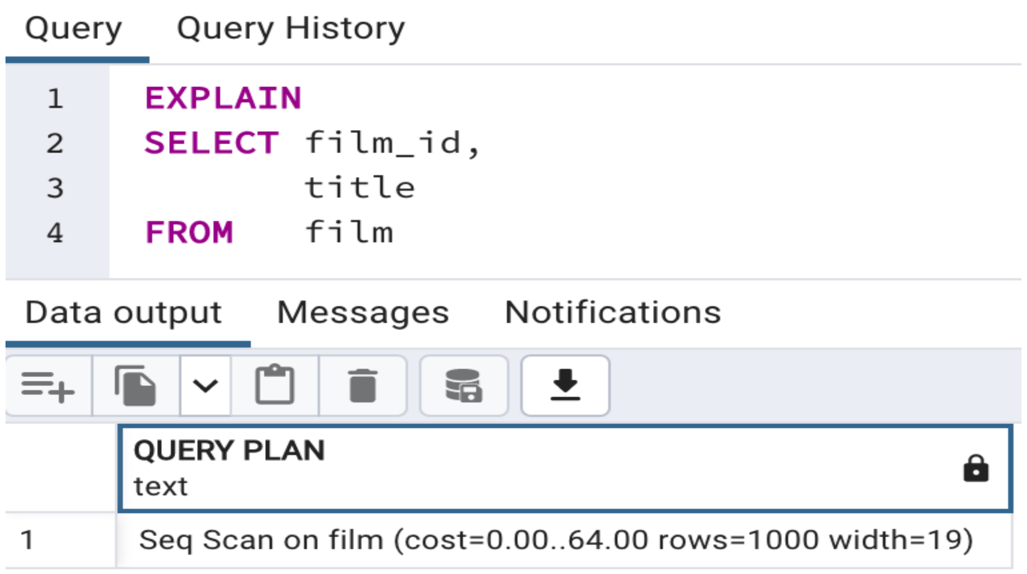


```
EXPLAIN
SELECT *
FROM   film
```

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

- *EXPLAIN*
  SELECT *film_id,*
  title
  FROM *film*

This query also has a cost of 64, see screenshot below:

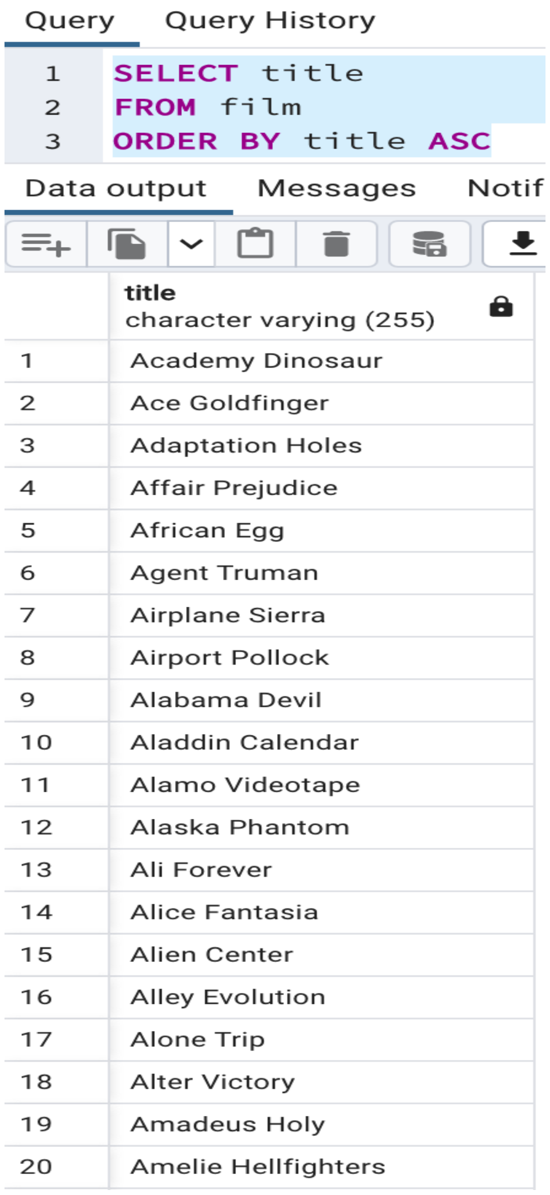


```
EXPLAIN
SELECT film_id,
       title
FROM   film
```

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

2. Ordering Data
2a.1

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



Query Query History

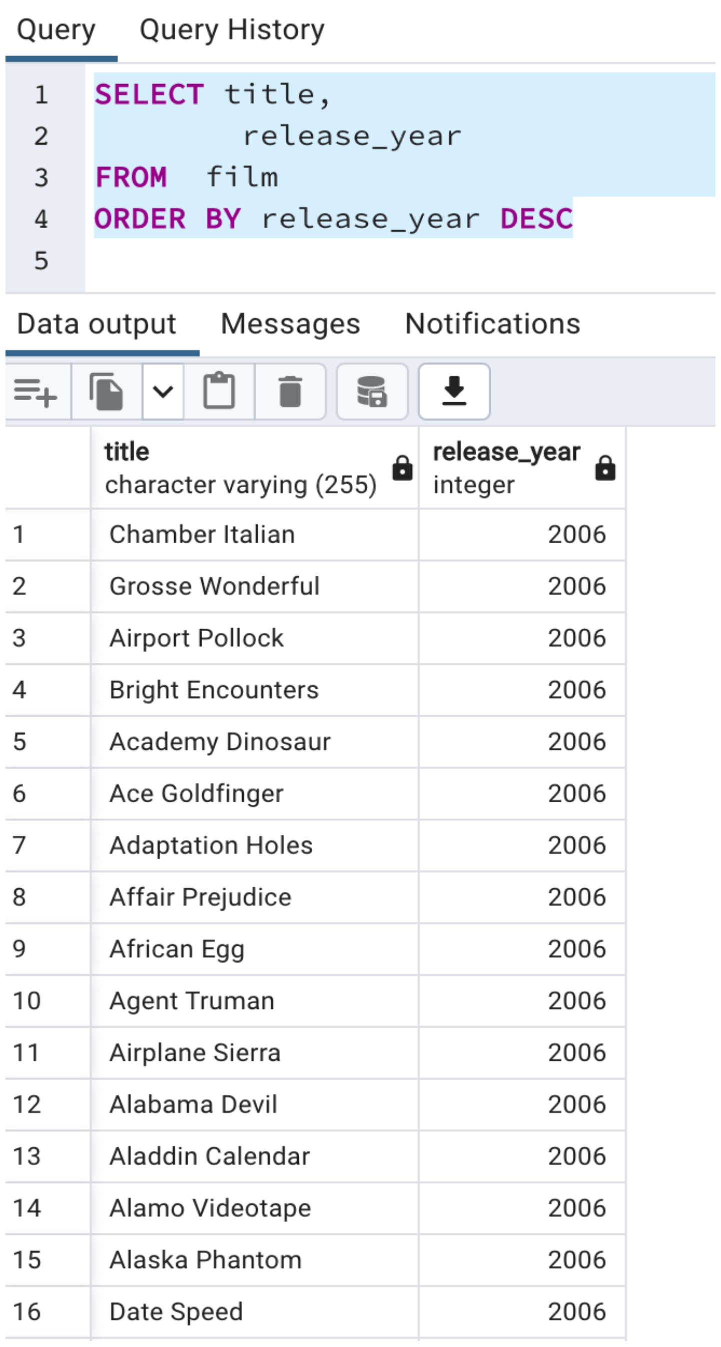
```
SELECT title
FROM film
ORDER BY title ASC
```

Data output Messages Notif

```
    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
15  Alien Center
16  Alley Evolution
17  Alone Trip
18  Alter Victory
19  Amadeus Holy
20  Amelie Hellfighters
```

2a.2

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

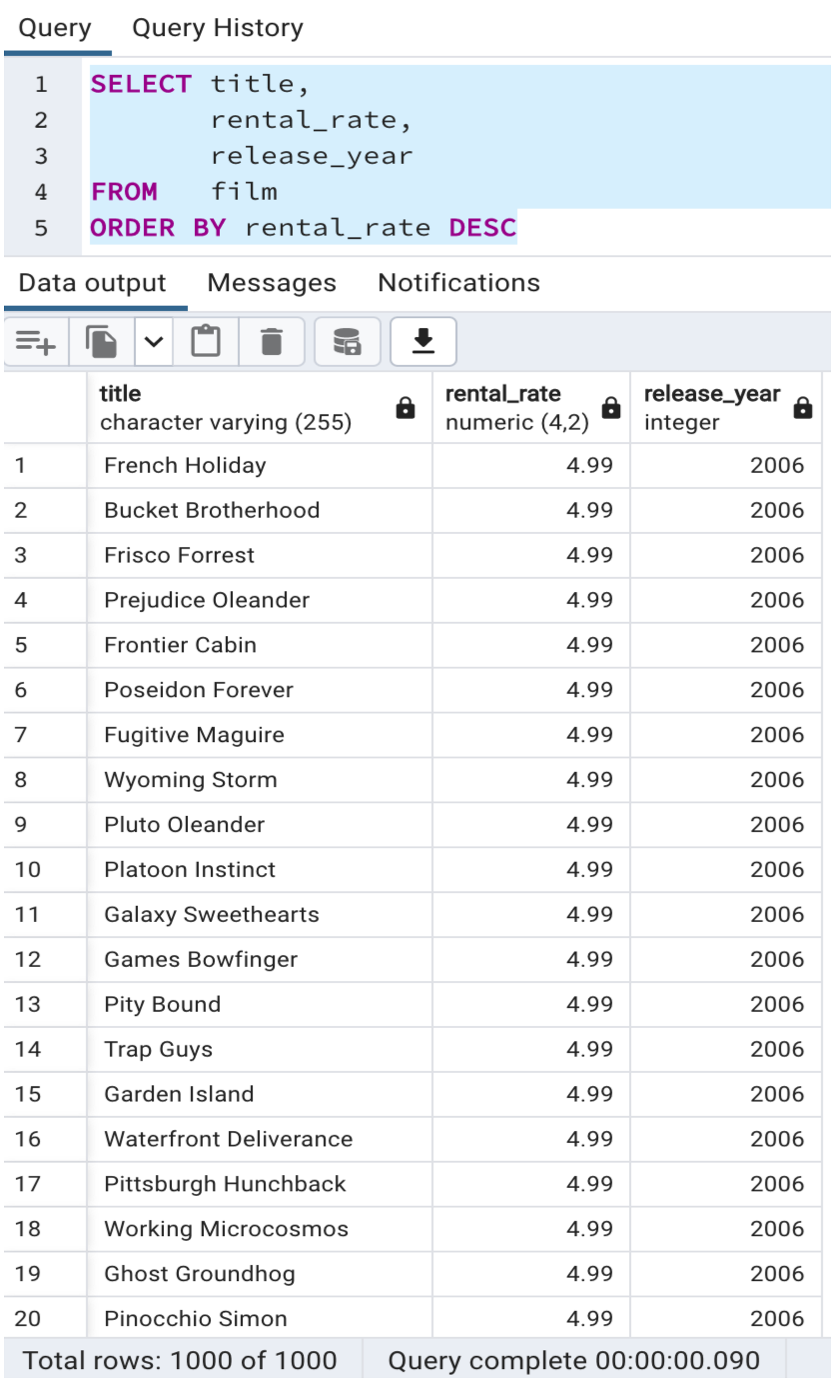


```
SELECT title,
        release_year
FROM  film
ORDER BY release_year DESC
```

	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
15	Alaska Phantom	2006
16	Date Speed	2006

2a.3

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


Query Query History

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

Data output Messages Notifications

	title character varying (255)	rental_rate numeric (4,2)	release_year integer
1	French Holiday	4.99	2006
2	Bucket Brotherhood	4.99	2006
3	Frisco Forrest	4.99	2006
4	Prejudice Oleander	4.99	2006
5	Frontier Cabin	4.99	2006
6	Poseidon Forever	4.99	2006
7	Fugitive Maguire	4.99	2006
8	Wyoming Storm	4.99	2006
9	Pluto Oleander	4.99	2006
10	Platoon Instinct	4.99	2006
11	Galaxy Sweethearts	4.99	2006
12	Games Bowfinger	4.99	2006
13	Pity Bound	4.99	2006
14	Trap Guys	4.99	2006
15	Garden Island	4.99	2006
16	Waterfront Deliverance	4.99	2006
17	Pittsburgh Hunchback	4.99	2006
18	Working Microcosmos	4.99	2006
19	Ghost Groundhog	4.99	2006
20	Pinocchio Simon	4.99	2006

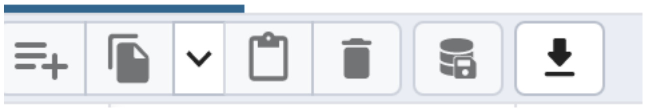
Total rows: 1000 of 1000 Query complete 00:00:00.090

2b.- Extract the data output from the query into a csv file for the film collection department to analyze it in Excel.

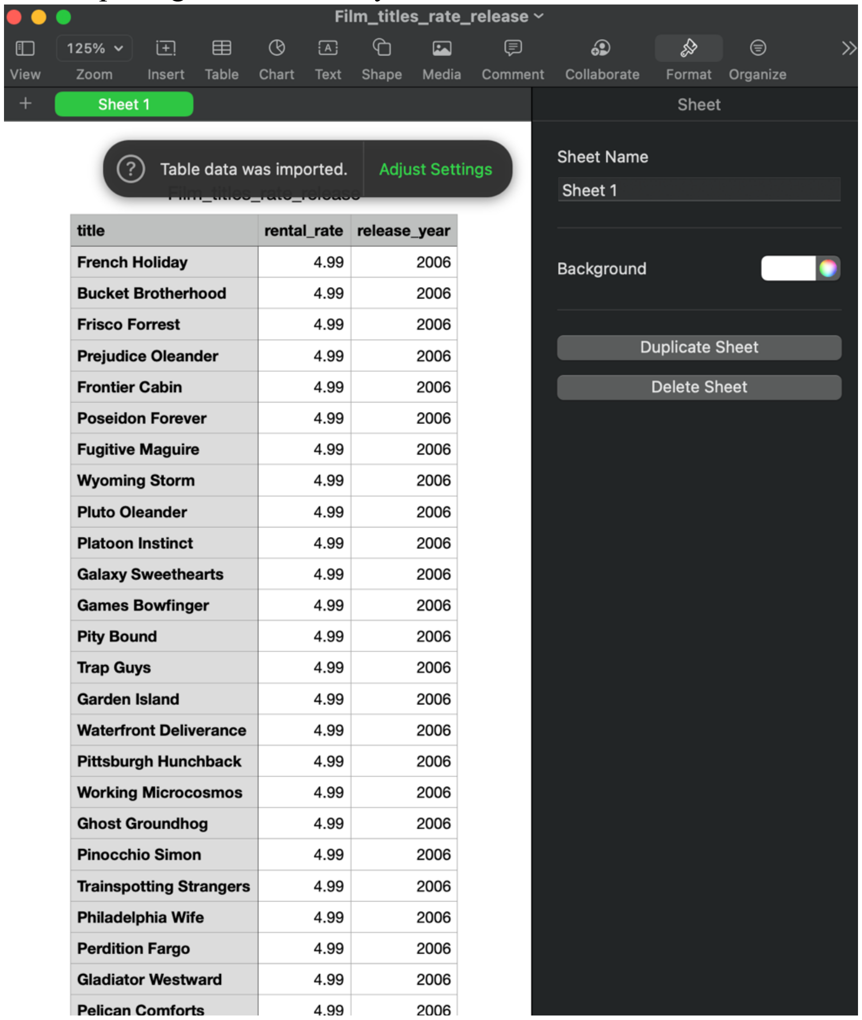
There are several ways to export data into a csv file, but for didactical reasons, only two other ways will be referenced.

1. Stating the query and then "F8" or "Save Results to File":

With the last button in the following ribbon, you can download the result of the query:

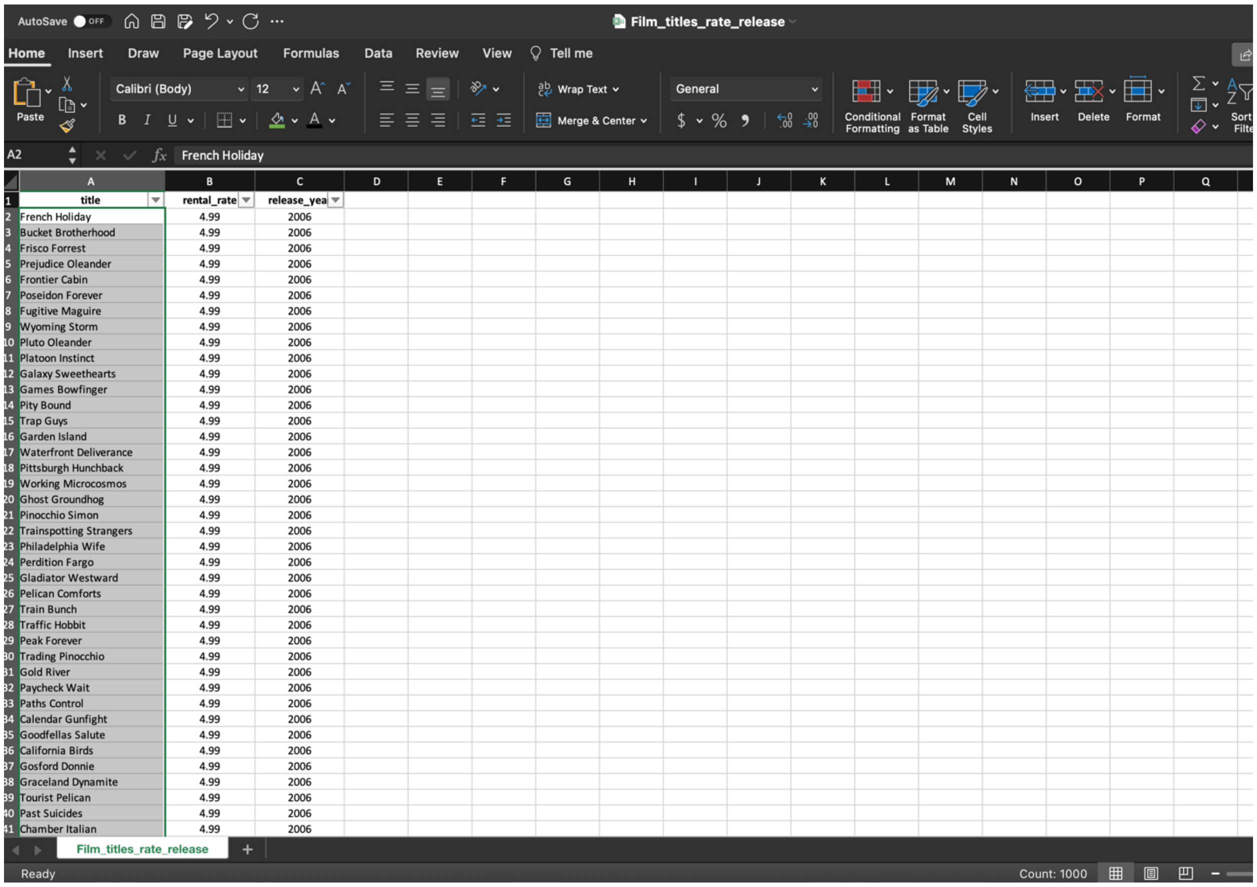


Then, opening the csv directly from Excel:



Film_titles_rate_release

title	rental_rate	release_year
French Holiday	4.99	2006
Bucket Brotherhood	4.99	2006
Frisco Forrest	4.99	2006
Prejudice Oleander	4.99	2006
Frontier Cabin	4.99	2006
Poseidon Forever	4.99	2006
Fugitive Maguire	4.99	2006
Wyoming Storm	4.99	2006
Pluto Oleander	4.99	2006
Platoon Instinct	4.99	2006
Galaxy Sweethearts	4.99	2006
Games Bowfinger	4.99	2006
Pity Bound	4.99	2006
Trap Guys	4.99	2006
Garden Island	4.99	2006
Waterfront Deliverance	4.99	2006
Pittsburgh Hunchback	4.99	2006
Working Microcosmos	4.99	2006
Ghost Groundhog	4.99	2006
Pinocchio Simon	4.99	2006
Trainspotting Strangers	4.99	2006
Philadelphia Wife	4.99	2006
Perdition Fargo	4.99	2006
Gladiator Westward	4.99	2006
Pelican Comforts	4.99	2006



title	rental_rate	release_yea
French Holiday	4.99	2006
Bucket Brotherhood	4.99	2006
Frisco Forrest	4.99	2006
Prejudice Oleander	4.99	2006
Frontier Cabin	4.99	2006
Poseidon Forever	4.99	2006
Fugitive Maguire	4.99	2006
Wyoming Storm	4.99	2006
Pluto Oleander	4.99	2006
Platoon Instinct	4.99	2006
Galaxy Sweethearts	4.99	2006
Games Bowfinger	4.99	2006
Pity Bound	4.99	2006
Trap Guys	4.99	2006
Garden Island	4.99	2006
Waterfront Deliverance	4.99	2006
Pittsburgh Hunchback	4.99	2006
Working Microcosmos	4.99	2006
Ghost Groundhog	4.99	2006
Pinocchio Simon	4.99	2006
Trainspotting Strangers	4.99	2006
Philadelphia Wife	4.99	2006
Perdition Fargo	4.99	2006
Gladiator Westward	4.99	2006
Pelican Comforts	4.99	2006
Train Bunch	4.99	2006
Traffic Hobbit	4.99	2006
Peak Forever	4.99	2006
Trading Pinocchio	4.99	2006
Gold River	4.99	2006
Paycheck Wait	4.99	2006
Paths Control	4.99	2006
Calendar Gunfight	4.99	2006
Goodfellas Salute	4.99	2006
California Birds	4.99	2006
Gosford Donnie	4.99	2006
Graceland Dynamite	4.99	2006
Tourist Pelican	4.99	2006
Past Suicides	4.99	2006
Chamber Italian	4.99	2006

There are also other ways to create a csv., two of many are by using the "\COPY" and "COPY" commands.

- "\COPY" command used for a copy in the local systems, meaning that the user does not necessarily have to have superuser access to the database.

```
\COPY
        (SELECT*
        FROM film)
TO
        ('relative_path/film_table_-_partial.csv' CSV HEADER)
```

The relative path is a relative sequence of file where the csv file will be saved.

- "COPY" command used for a copy in the server's side.

```
COPY
        (SELECT*
        FROM film)
TO
        ('absolute/path/to/save/film_table_-_partial.csv' CSV HEADER)
```

Both ways must be done through *SQL shell (psql).*

Note: both ways were tried both in PgAdmin4 and SQL shell (psql), but in both cases the error is as follows,

ERROR: relative path not allowed for COPY to file SQL state: 42602

This error comes to attention considering that the database has been saved in the same location as where the SQL query is being made.

3. Grouping Data

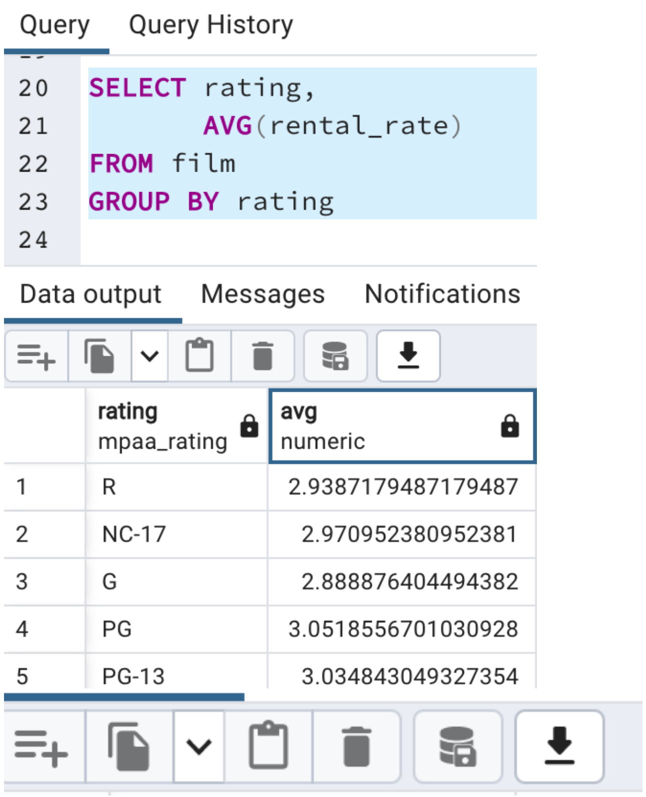
Write a query to retrieve the correct answers to the following questions, then extract results as a CSV file.

3a.- What is the average rate of each rating category?

Query:

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

Please, see the following screenshots for answers:

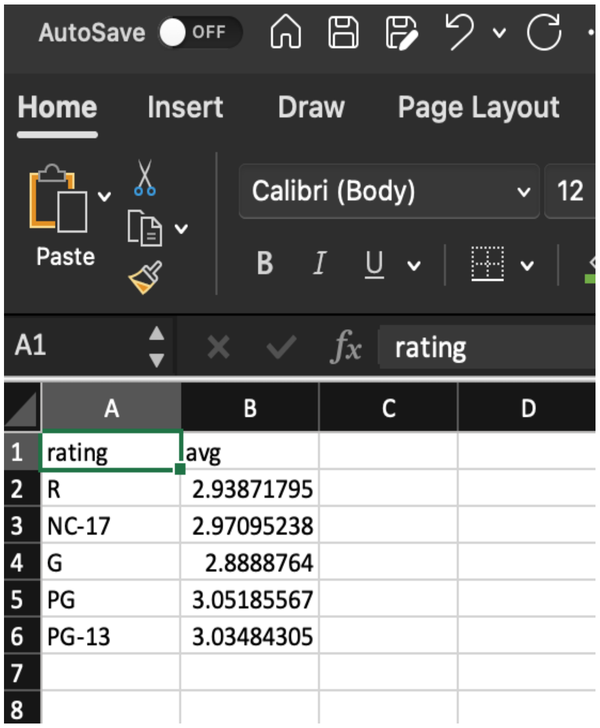


Query Query History

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

Data output Messages Notifications

	rating mpaa_rating	avg numeric
1	R	2.9387179487179487
2	NC-17	2.970952380952381
3	G	2.888876404494382
4	PG	3.0518556701030928
5	PG-13	3.034843049327354

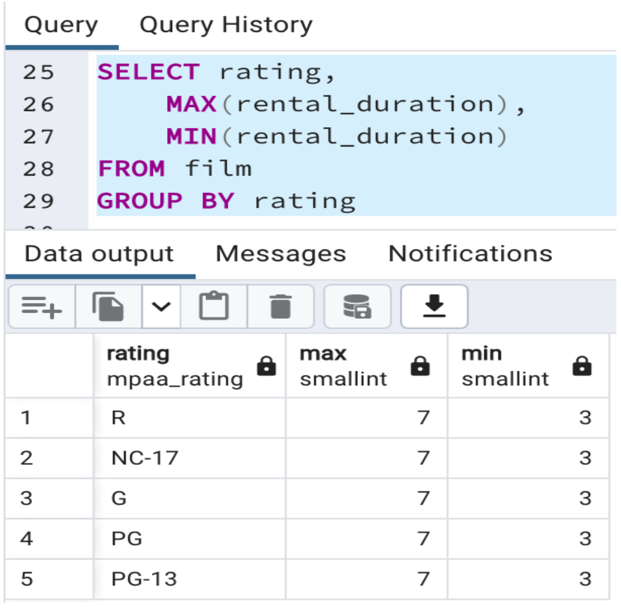


	A	B	C	D
1	rating	avg		
2	R	2.93871795		
3	NC-17	2.97095238		
4	G	2.8888764		
5	PG	3.05185567		
6	PG-13	3.03484305		
7				
8				

3b.- What are the minimum and maximum rental durations for each rental category?

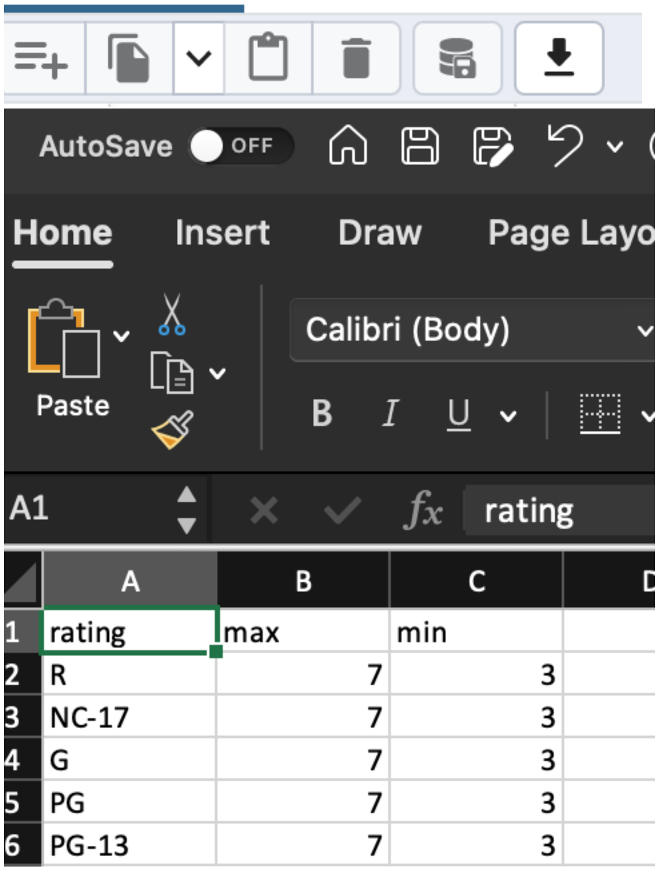
Query:
```
SELECT rating,
   MAX(rental_duration),
   MIN  (rental_duration)
FROM   film
GROUP BY rating
```

Please, see screenshots below for answers:



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

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



rating	max	min
R	7	3
NC-17	7	3
G	7	3
PG	7	3
PG-13	7	3

4. Database Migration
The 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 it can be analyzed.

4a.- Outline the procedure for migrating the data and who will be responsible for it
A Data Migration Process called ETL (Extract, Transform, Load) must be done before starting the analyzing process.

First, ***extract*** the generated data by users from the Rockbuster Android app.

Then, ***transform*** data by, for example, separating information related to the user's characteristics and identification, calculating users KPIs and combining these data points along with the user's own characteristics, etc.

Finally, ***loading*** the formatted data into the data warehouse.

4b.- What problems or issues may arouse if analysis starts before the data can be loaded into the data warehouse

Data analysis insights and/or results will probably be affected as the data manipulation will be done in two separate spaces, and for this would not be virtually possible to extract meaningful information from the whole data in conjunction.

Also, the fact that the data might not be properly formatted to match with that in the data warehouse, could create an issue due to incomplete information or information that cannot be properly analyzed due to its abstract nature.

Bonus.
What are the minimum and maximum replacement cost for each rating category ordered by rating as follows: G, PG, PG-13, R, NC-17.

After some research, two ways were determined but none of them work; the query entry varied to find alternatives with no success

Query:

1.-
```
SELECT rating,
        MIN(replacement_cost),
        MAX(replacement_cost)
FROM film
ORDER BY CASE WHEN rating = 'G' THEN 1
              WHEN rating = 'PG' THEN 2
              WHEN rating = 'PG-13' THEN 3
              WHEN rating = 'R' THEN 4
              WHEN rating = 'NC-17' THEN 5
END
```
```
ERROR: column "film.rating" must appear in the GROUP BY clause or be used in an
aggregate function LINE 1: SELECT rating, ^ SQL state: 42803 Character: 8
```

2.-
```
SELECT rating,
        MIN(replacement_cost),
        MAX(replacement_cost)
FROM film
ORDER BY CASE WHEN rating = 'G' THEN 1,
              WHEN rating = 'PG' THEN 2,
              WHEN rating = 'PG-13' THEN 3,
              WHEN rating = 'R' THEN 4,
              WHEN rating = 'NC-17' THEN 5
END ASC
```
```
ERROR: syntax error at or near "," LINE 5: ORDER BY CASE WHEN rating = 'G' THEN 1, ^
SQL state: 42601 Character: 111
```

3.-
```
SELECT rating,
        MIN(replacement_cost),
        MAX(replacement_cost)
FROM film
ORDER BY rating WHEN rating = ‘G’ THEN 1
                WHEN rating = ‘PG’ THEN 2
                WHEN rating = ‘PG-13’ THEN 3
                WHEN rating = ‘R’ THEN 4
                WHEN rating = ‘NC-17’ THEN 5
```

```
ERROR: syntax error at or near "WHEN" LINE 5: ORDER BY rating WHEN rating = ‘G’ THEN
1 ^ SQL state: 42601 Character: 89
```

4-
```
SELECT MIN(replacement_cost),
         MAX(replacement_cost)
FROM film
ORDER BY rating WHEN rating = ‘G’ THEN 1
                WHEN rating = ‘PG’ THEN 2
                WHEN rating = ‘PG-13’ THEN 3
                WHEN rating = ‘R’ THEN 4
                WHEN rating = ‘NC-17’ THEN 5
```

```
ERROR: syntax error at or near "WHEN" LINE 4: ORDER BY rating WHEN rating = ‘G’ THEN
1 ^ SQL state: 42601 Character: 83
```

5.-
```
SELECT rating,
        MIN(replacement_cost)::int,
        MAX(replacement_cost)::int
FROM film
ORDER BY rating WHEN rating = ‘G’ THEN 1,
                WHEN rating = ‘PG’ THEN 2,
                WHEN rating = ‘PG-13’ THEN 3,
                WHEN rating = ‘R’ THEN 4,,
                WHEN rating = ‘NC-17’ THEN 5
ELSE END
```
```
ERROR: syntax error at or near "WHEN" LINE 5: ORDER BY rating WHEN rating = ‘G’ THEN
1 ^ SQL state: 42601 Character: 99
```

6.-
```
SELECT rating,
        MIN(replacement_cost)::int,
        MAX(replacement_cost)::int
```

```
FROM film
ORDER BY rating WHERE rating = ‘G’ THEN 1
                WHERE rating = ‘PG’ THEN 2
                WHERE rating = ‘PG-13’ THEN 3
                WHERE rating = ‘R’ THEN 4
                WHERE rating = ‘NC-17’ THEN 5
ELSE END
```

```
ERROR: syntax error at or near "WHERE" LINE 5: ORDER BY rating WHERE rating = ‘G’
THEN 1 ^ SQL state: 42601 Character: 99
```