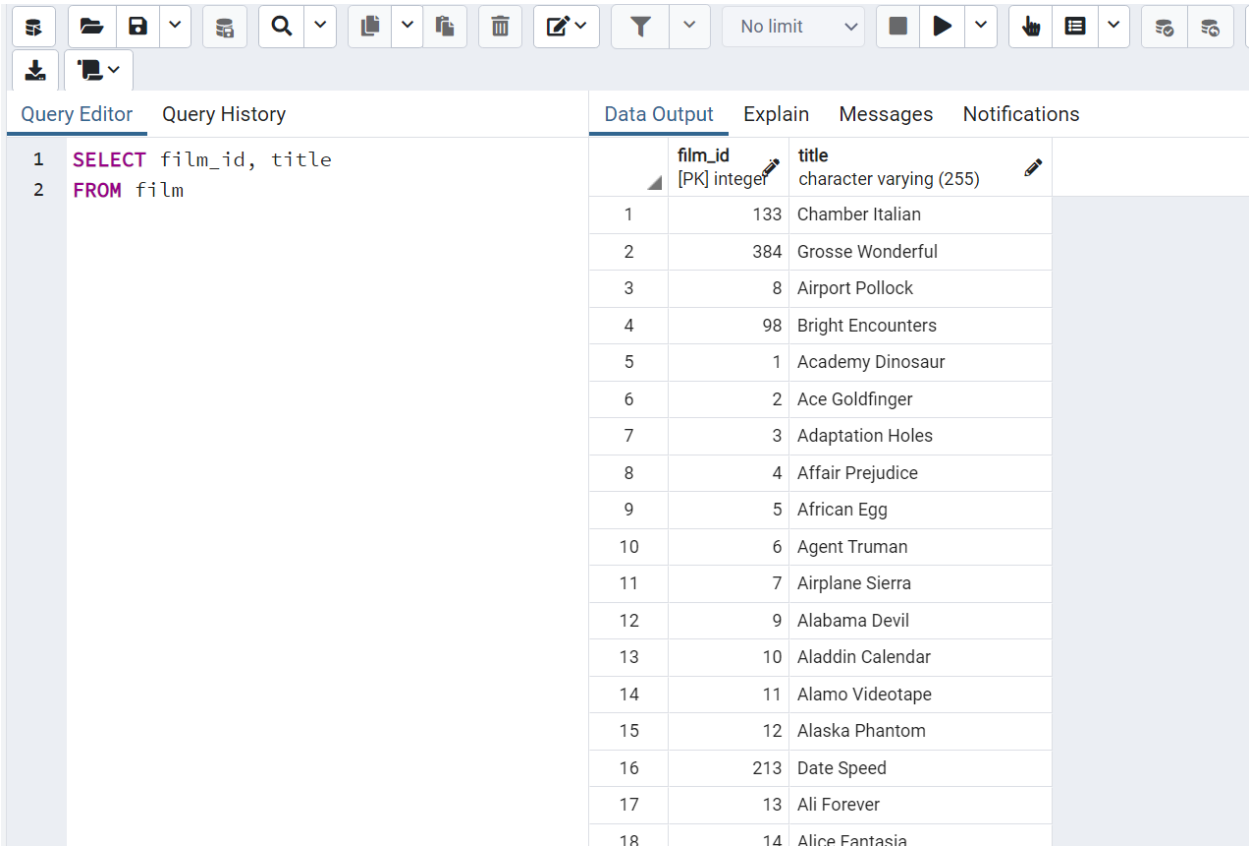
1) **Refining Your Query:** You need to get some data from the "film" table and decide to use the query `SELECT * FROM film`.

- You realize that only the "film_id" and "title" columns are needed. Write a new query that selects only those 2 columns.



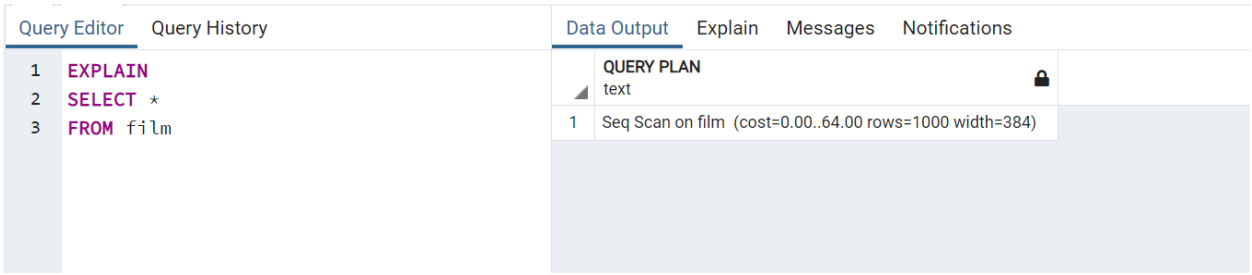
Query Editor | Query History

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

Data Output | Explain | 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
15	12	Alaska Phantom
16	213	Date Speed
17	13	Ali Forever
18	14	Alice Fantasia

- 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?

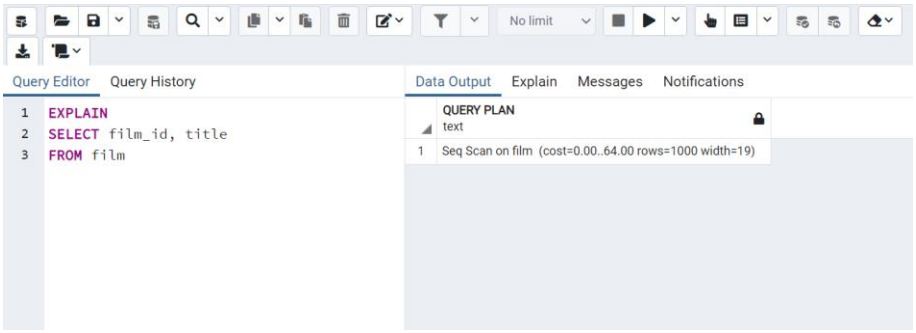


Query Editor | Query History

```
EXPLAIN
SELECT *
FROM film
```

Data Output | Explain | Messages | Notifications

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



The cost is showing exactly the same for either query. The first query took 28 msec while the second took 27 msec. So the second would be slightly more efficient. It would really depend on the processing power of your computer. If you only need part of the data it is always best to only pull up what you need through because then you don't have to do any sifting to get exactly what you want displayed.

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.

Query Editor | Query History

```
1 SELECT title, release_year, rental_rate
2 FROM film
3 ORDER BY title, release_year, rental_rate
4 DESC
```

Data Output | Explain | Messages | Notifications

	title character varying (255)	release_year integer	rental_rate numeric (4,2)
1	Academy Dinosaur	2006	0.99
2	Ace Goldfinger	2006	4.99
3	Adaptation Holes	2006	2.99
4	Affair Prejudice	2006	2.99
5	African Egg	2006	2.99
6	Agent Truman	2006	2.99
7	Airplane Sierra	2006	4.99
8	Airport Pollock	2006	4.99
9	Alabama Devil	2006	2.99
10	Aladdin Calendar	2006	4.99
11	Alamo Videotape	2006	0.99
12	Alaska Phantom	2006	0.99
13	Ali Forever	2006	4.99
14	Alice Fantasia	2006	0.99
15	Alien Center	2006	2.99
16	Alley Evolution	2006	2.99
17	Alone Trip	2006	0.99
18	Alter Victory	2006	0.99

- Extract the data output of your query into a csv file for the film collection department to analyze in Excel. (You may need to explore how to save your output as a csv file in the Query Tool.) ***Attached in submit for this task***

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?

Query Editor | Query History

```
1  SELECT rating, AVG (rental_rate)
2  AS avg_rental_rate
3  FROM film GROUP BY rating
```

Data Output | Explain | Messages | Notifications

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

Attached CSV

- What are the minimum and maximum rental durations for each rating category?

Query Editor | Query History

```
1  SELECT rating,
2  MIN (rental_duration)AS min_rental_duration,
3  MAX (rental_duration)AS max_rental_duration
4  FROM film
5  GROUP BY rating;
```

Data Output | Explain | Messages | Notifications

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

Attached CSV

4) **Database Migration:** Your team has decided to use an external tool to collect data on user behavior in the new Rockbuster Android app. Data collected from this new source will need to be loaded into the data warehouse before you can analyze it.

- Can you outline the procedure for migrating the data and who will be responsible for it?

We would follow the extract, transform and load process. The data would need to be extracted from its source, then re shaped to fit our current formatting. Then we would be using calculations or transforming functions making sure that we can link with the proper keys in the data, then the data would need to be properly loaded in to the database. It also goes without saying that it would then need to be tested to make sure that the process was done properly, and that queries pull what they are supposed to pull for useability.

- What problems do you foresee if you start analyzing the data before it's been loaded into the data warehouse?

There would in general be a lot more work because the data has not been linked to the other data. It could be done, however in a world where efficiency is king, this would just not be the optimal way to carry out actions and take much longer than necessary.