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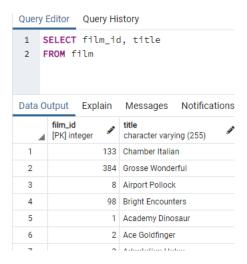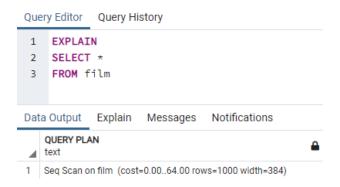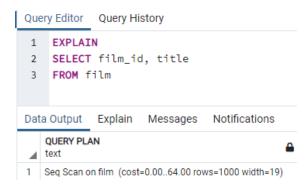Refining Your Query:** You need to get some data from the "film" table and decide to use the query `SELECT * FROM film`.

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

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

   Query Editor   Query History

   1   `SELECT film_id, title`
   2   `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

   ➢ *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

   1   `EXPLAIN`
   2   `SELECT *`
   3   `FROM film`

   Data Output   Explain   Messages   Notifications

   QUERY PLAN
   text

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

   Query Editor   Query History

   1   `EXPLAIN`
   2   `SELECT film_id, title`
   3   `FROM film`

   Data Output   Explain   Messages   Notifications

   QUERY PLAN
   text

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

   Both queries have the same cost results regardless of the data dimensions. Running a script to optimize its full potential at a minimum timeframe seeing that it will be frequently executed.
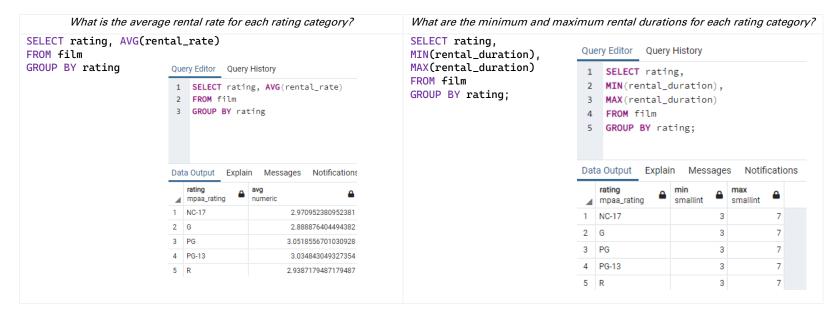
1

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.*

MOVIES (A-Z)	MOST RECENT RELEASE YEAR	HIGHEST TO LOWEST RENTAL RATES
SELECT title, release_year, rental_rate FROM film ORDER BY title ASC	SELECT title, release_year, rental_rate FROM film ORDER BY release_year DESC	SELECT title, release_year, rental_rate FROM film ORDER BY rental_rate DESC

➢ *Extract the data output of your query into a CSV file for the film collection department to analyze in Excel. To do this, click the button "Save results to file":*
DONE

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

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?*

STEP 1: EXTRACT	STEP 2: TRANSFORM	STEP 2: LOAD
Gathering all datasets from several resources.	Normalize data points like ages from birthdates, combine multiple area codes and telephone numbers to extract contact information, etc.	Transformed data is uploaded to the new database.

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

   Performing ETL is essential to exploit the database's full potential in resolving issues, forecasting and many other reasons to help management make an educated decision. For instance, there can be inaccurate query searches for product popularity based on ratings, genres, etc. leading to a potential business loss in both processing time as well as profit.

Bonus Task

You've not yet covered custom sorting; however, let's imagine you've found the two resources below that explain it. Read each one, then try to write a query to answer the following question: What are the minimum and the maximum replacement costs for each rating category ordered by rating as follows: G, PG, PG-13, R, NC-17?

- [SQL Server - Custom Sorting in ORDER BY Clause](#)
- [Custom Order By in SQL Server](#)

```
SELECT rating,
      MIN(replacement_cost),
      MAX(replacement_cost)
FROM film
GROUP BY rating
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'
                 ELSE 6
                 END ASC
```