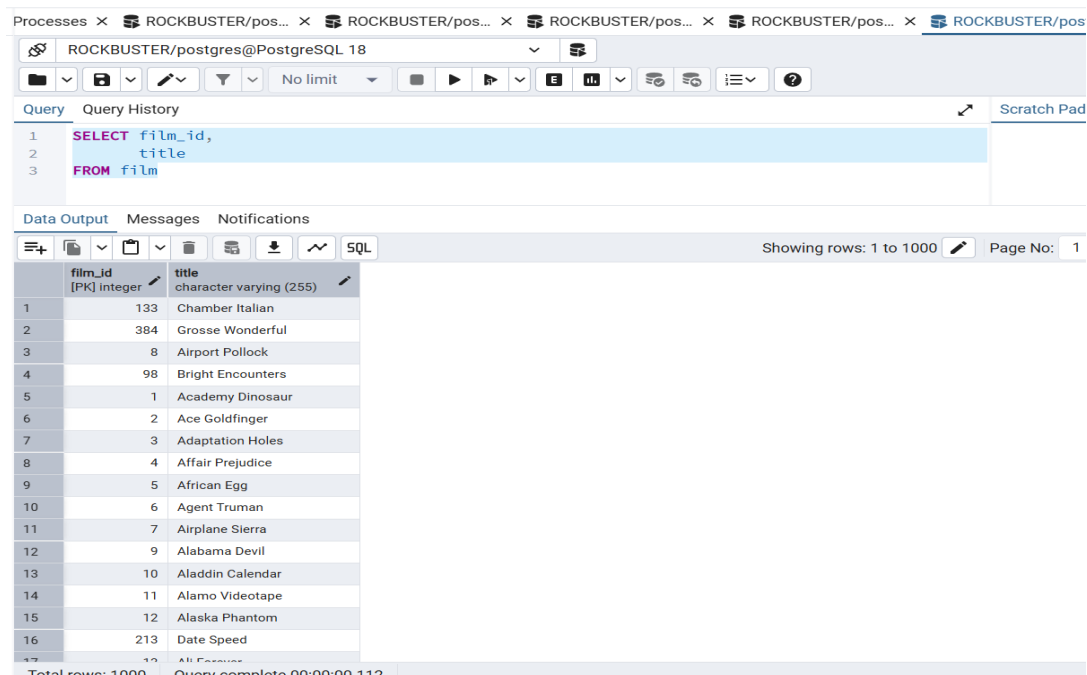


Databases & SQL for Analysts

3.4: Database Querying in SQL

Refining Your Query:

1/



The screenshot shows a PostgreSQL query editor interface. The top bar displays the connection name 'ROCKBUSTER/postgres@PostgreSQL 18'. The query editor contains the following SQL query:

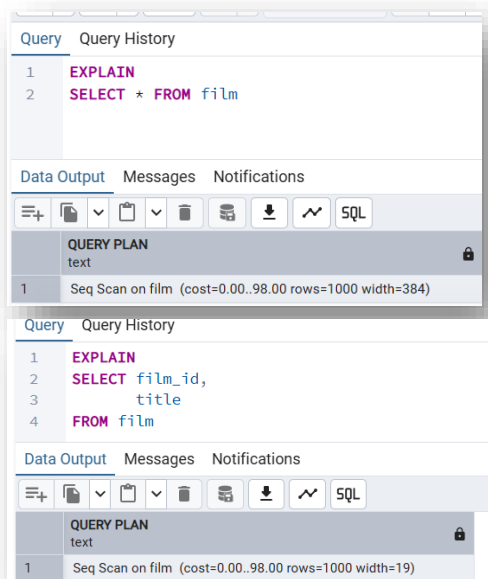
```
1 SELECT film_id,  
2      title  
3 FROM film
```

Below the query editor, the 'Data Output' tab is active, showing the results of the query. The results are displayed in a table with two columns: 'film_id' (integer, primary key) and 'title' (character varying (255)). The table shows 17 rows of data, with the first 16 rows visible in the screenshot. The status bar at the bottom indicates 'Showing rows: 1 to 1000' and 'Page No: 1'.

	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	All Features

Total rows: 1000 Query complete: 00:00:00.112

2/



The only difference is in the width: reducing the width from 384 bytes to 19 bytes.

We can say that there is no big difference but the revised query is more efficient in practice because it transfers less data from disk to memory.

Can you suggest any ways to optimize this query?

The query shown is: `SELECT film_id, title FROM film LIMIT 10;`

	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

Ordering the Data:

run a query that selects every film from the “film” table, with the movies sorted by title from A to Z ; by most recent release year, by highest to lowest rental rate.

Query

Query History

1

2

3

4

5

6

7

8

SELECT

|

title,

release_year,

rental_rate

FROM film

ORDER BY title ASC,

release_year DESC,

rental_rate DESC;

Data Output

Messages

Notifications

≡+

📄

▼

📋

▼

🗑️

📦

⬇️

📈

SQL

	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

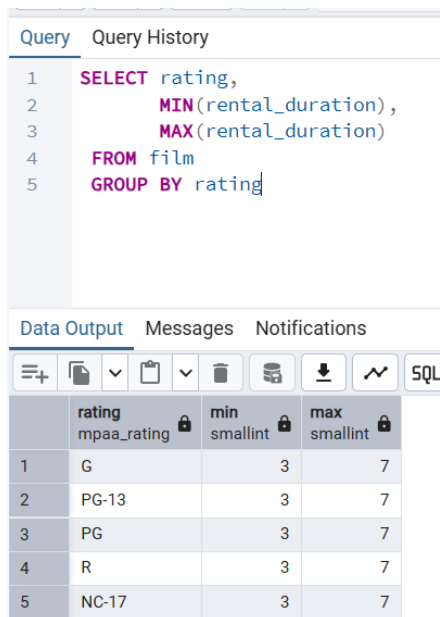
Grouping Data:

1/

Query	Query History
1	SELECT rating,
2	AVG (rental_rate)
3	FROM film
4	GROUP BY rating

Data Output	Messages	Notifications																		
<div> <div>≡</div> <div>📄</div> <div>▼</div> <div>📋</div> <div>▼</div> <div>🗑️</div> <div>📦</div> <div>⬇️</div> <div>📈</div> <div>SQL</div> </div> <table> <tr> <th></th><th>rating mpaa_rating</th><th>avg numeric</th></tr> <tr><td>1</td><td>G</td><td>2.8888764044943820</td></tr> <tr><td>2</td><td>PG-13</td><td>3.0348430493273543</td></tr> <tr><td>3</td><td>PG</td><td>3.0518556701030928</td></tr> <tr><td>4</td><td>R</td><td>2.9387179487179487</td></tr> <tr><td>5</td><td>NC-17</td><td>2.9709523809523810</td></tr> </table>		rating mpaa_rating	avg numeric	1	G	2.8888764044943820	2	PG-13	3.0348430493273543	3	PG	3.0518556701030928	4	R	2.9387179487179487	5	NC-17	2.9709523809523810		
	rating mpaa_rating	avg numeric																		
1	G	2.8888764044943820																		
2	PG-13	3.0348430493273543																		
3	PG	3.0518556701030928																		
4	R	2.9387179487179487																		
5	NC-17	2.9709523809523810																		

2/



The screenshot shows a SQL query editor with a query window and a results window. The query window contains the following SQL code:

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

The results window shows the output of the query, which is a table with 5 rows and 4 columns. The columns are: rating, mpaa_rating, min, and max. The data is as follows:

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

Database Migration:

Can you outline the procedure for migrating the data?

- **Extract:** The first step involves collecting the data from multiple data sources.
- **Transform:** During this step, the extracted data is converted into another format. This could mean calculating ages from dates of birth or combining multiple data points like area codes and telephone numbers to get a contact number, for example.
- **Load:** At this point the transformed data is inserted or loaded into the new database.

who will be responsible for it?

ETL is a data engineer's job. However, an awareness of the basic concepts is important for data analysis

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

Analyzing data before it is fully loaded into the data warehouse leads to biased results, as some extractions or transformations may be incomplete and provide a partial view. Furthermore, raw data often contains duplicates, null values, or inconsistent units, which compromises the reliability of the insights.