

Answer 3.4: Database Querying in SQL

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?

❖ ORIGINAL QUERY

The screenshot shows a database query interface with two panels. The left panel displays the query `EXPLAIN SELECT * FROM film` under the 'Query' tab. The right panel shows the 'Query History' tab with the same query. Below the query history, the 'Data output' tab is active, showing a 'QUERY PLAN' section with the text 'Seq Scan on film (cost=0.00..64.00 rows=1000 width=388)'. The 'Messages' tab is also visible, showing a success message: 'Successfully run. Total query runtime: 227 msec. 1 rows affected.'

❖ REVISED QUERY

The screenshot shows a database query interface with two panels. The left panel displays the query `EXPLAIN SELECT film_id, title FROM film` under the 'Query' tab. The right panel shows the 'Query History' tab with the same query. Below the query history, the 'Data output' tab is active, showing a 'QUERY PLAN' section with the text 'Seq Scan on film (cost=0.00..64.00 rows=1000 width=19)'. The 'Messages' tab is also visible, showing a success message: 'Successfully run. Total query runtime: 111 msec. 1 rows affected.'

Both queries have the same cost which is 0.00-64.00, however, the revised total query runtime is faster than the original as it's only focusing on the two specific columns which are needed. To further optimize this query one can focus the writing of the query to give only the information needed. For example:

The screenshot shows a database query interface with two panels. The left panel displays the query `EXPLAIN SELECT title, COUNT(film_id) FROM film GROUP BY title` under the 'Query' tab. The right panel shows the 'Query History' tab with the same query. Below the query history, the 'Data output' tab is active, showing a 'QUERY PLAN' section with the text 'Seq Scan on film (cost=0.00..64.00 rows=1000 width=19)'. The 'Messages' tab is also visible, showing a success message: 'Successfully run. Total query runtime: 102 msec. 3 rows affected.'

Although this has the same cost as the original and Revised query, the total query runtime is a little faster.

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 analyze in Excel. To do this, click the button “Save results to file”:

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

Data output		Messages		Notifications	
+	+	+	+	+	+
	title	release_year	rental_rate		
	character varying (255)	integer	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 Trin	2006	0.99		
Total rows: 1000 of 1000		Query complete 00:00:00.286			

Film data CSV

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

Data output		Messages		Notifications	
+	+	+	+	+	+
	rating	avg_rental_rate			
	mpaa_rating	numeric			
1	R	2.9387179487179			
2	NC-17	2.9709523809523			
3	G	2.8888764044943			
4	PG	3.0518556701030			
5	PG-13	3.0348430493273			

[Avg_rental_rate CSV file](#)

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

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

Data output		Messages		Notifications	
	rating		min_rental_rate		
	mpaa_rating		numeric		
1	R		0.99		
2	NC-17		0.99		
3	G		0.99		
4	PG		0.99		
5	PG-13		0.99		

[Minimum rental rate CSV file](#)

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

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

Data output		Messages		Notifications	
	rating		max_rental_rate		
	mpaa_rating		numeric		
1	R		4.99		
2	NC-17		4.99		
3	G		4.99		
4	PG		4.99		
5	PG-13		4.99		

[Max rental rate CSV file](#)

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 analyze it.
- Can you outline the procedure for migrating the data and who will be responsible for it?
 - ❖ The data engineer is responsible for this, and the procedure to load data on user behaviour will be to use the ETL process. Firstly, the data would be extracted from its source and then transform by converting to meet the format required of the current data in the data warehouse before loading into the data warehouse.
 - What problems do you foresee if you start analyzing the data before it's been loaded into the data warehouse?
 - ❖ It will be difficult to analyze and time-consuming as not loading the user behaviour data into the data warehouse will require manual analysis.

BONUS TASK

5. 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?

Query







Query History

```
1 SELECT rating, MAX (replacement_cost) AS
2 Max_replacement_cost,
3 MIN (replacement_cost) AS Min_replacement_cost
4 FROM film
5 GROUP BY rating
6 ORDER BY rating
```

Data output

Messages

Notifications



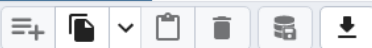
	rating mpaa_rating	max_replacement_cost numeric	min_replacement_cost numeric
1	G	29.99	9.99
2	PG	29.99	9.99
3	PG-13	29.99	9.99
4	R	29.99	9.99
5	NC-17	29.99	9.99

Another way is Using ORDER BY CASE

Query Query History

```
1 SELECT rating,  
2 MAX (replacement_cost) AS Max_replacement_cost,  
3 MIN (replacement_cost) AS Min_replacement_cost  
4 FROM film  
5 GROUP BY rating  
6 ORDER BY CASE WHEN rating = 'G' THEN 1  
7                WHEN rating = 'PG' THEN 2  
8                WHEN rating = 'PG-13' THEN 3  
9                WHEN rating = 'R' THEN 4  
10               ELSE 5 END
```

Data output Messages Notifications



	rating mpaa_rating 🔒	max_replacement_cost numeric 🔒	min_replacement_cost numeric 🔒
1	G	29.99	9.99
2	PG	29.99	9.99
3	PG-13	29.99	9.99
4	R	29.99	9.99
5	NC-17	29.99	9.99