

## 1. Refining Your Query

- a. 

```
SELECT film_id,  
       title  
FROM film
```
- b. Original query:
  - cost = 64
  - rows = 1000
  - width = 384

The screenshot shows the Rockbuster PostgreSQL interface. The left sidebar displays the database structure, including 'Rockbuster' and its various components. The main panel is divided into 'Query Editor' and 'Query History'. The 'Query Editor' contains the following SQL query:

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

Below the query editor, the 'Data Output' tab is selected, showing the 'QUERY PLAN' for the query. The plan indicates a 'Seq Scan on film' with a cost of 0.00..64.00, 1000 rows, and a width of 384 bytes.

### Optimized query:

- cost = 64
- rows = 1000
- width = 19

The screenshot shows the Rockbuster PostgreSQL interface with the optimized query entered in the 'Query Editor'.

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

The 'Data Output' tab shows the 'QUERY PLAN' for the optimized query, which is a 'Seq Scan on film' with a cost of 0.00..64.00, 1000 rows, and a significantly reduced width of 19 bytes.

Since both queries have the same cost and rows, the key difference is the width. The original query width is 384 bytes and the optimized query width is 19 bytes. This means that the estimated average width of rows is much lower in the optimized query, thus the preferred option.

## 2. Ordering the Data

The screenshot shows the Rockbuster PostgreSQL interface. The top navigation bar includes Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The current view is the SQL editor, showing a query to select title, release\_year, and rental\_rate from the film table, ordered by rental\_rate DESC, release\_year DESC, and title. The query is as follows:

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

Below the query editor, the Data Output tab is active, displaying the results of the query. The output table has columns: title (character varying (255)), release\_year (integer), and rental\_rate (numeric (4,2)). The results are ordered by rental\_rate DESC, release\_year DESC, and title.

	title	release_year	rental_rate
1	Ace Goldfinger	2006	4.99

## 3. Grouping Data

The screenshot shows the Rockbuster PostgreSQL interface. The top navigation bar includes Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. The current view is the SQL editor, showing a query to select DISTINCT rating, AVG(rental\_rate), MIN(rental\_duration), and MAX(rental\_duration) from the film table, grouped by rating. The query is as follows:

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

Below the query editor, the Data Output tab is active, displaying the results of the query. The output table has columns: rating (mpaa\_rating), avg (numeric), min (smallint), and max (smallint). The results are grouped by rating.

	rating	avg	min	max
1	G	2.888876404494382	3	7
2	PG	3.0518556701030928	3	7
3	PG-13	3.034843049327354	3	7
4	R	2.9387179487179487	3	7
5	NC-17	2.970952380952381	3	7

## 4. Database Migration

Steps for migrating data from an outside source to the Rockbuster data warehouse:

1. Data architect/engineer – extract and validate data from the outside source and load it into a staging area.
2. Data engineer/analyst – clean, map and transform data in staging area.
3. Data engineer/analyst – load data from staging area to data warehouse and verify new data is added.