

STEP 1:

File Object Tools Help

Object Explorer

- > AIS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > 1.3 Sequences
- > Tables (16)
 - > actor
 - > address
 - > category
 - > city
 - > country
 - > customer
 - > film
 - > film_2
 - > film_actor
 - > film_category
 - > inventory
 - > language
 - > payment
 - > rental
 - > staff

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/ < > P

Rockbuster/postgres@PostgreSQL 15

Query Query History Scratch Pad x

```
1 EXPLAIN SELECT * FROM film
```

Data Output Messages Notifications

QUERY PLAN
text

1	Seq Scan on film (cost=0.00..64.00 rows=1000 width=38...
---	--

Total rows: 1 of 1 Query complete 00:00:00.179 Ln 1, Col 27

File Object Tools Help

Object Explorer

- > AIS Parsers
- > FTS Templates
- > Foreign Tables
- > Functions
- > Materialized Views
- > Operators
- > Procedures
- > 1.3 Sequences
- > Tables (16)
 - > actor
 - > address
 - > category
 - > city
 - > country
 - > customer
 - > film
 - > film_2
 - > film_actor
 - > film_category
 - > inventory
 - > language
 - > payment
 - > rental
 - > staff

Dashboard Properties SQL Statistics Dependencies Dependents Processes Rockbuster/ < > P

Rockbuster/postgres@PostgreSQL 15

Query Query History Scratch Pad x

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

Data Output Messages Notifications

QUERY PLAN
text

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

Total rows: 1 of 1 Query complete 00:00:00.089 Ln 1, Col 29

The cost of the original Query and revised query both are same, however, the columns returned are different in both queries. I think, the revised query is already optimized as compared to original query. However, we can use LIMIT to limit the number of rows returned. We can also use WHERE clause to improve the query performance by reducing the number of rows.

Step2:

The screenshot shows a PostgreSQL query editor interface. At the top, the connection is set to 'Rockbuster/postgres@PostgreSQL 15'. The query editor has a toolbar with icons for file operations, filters, and execution. The query is as follows:

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

Below the query editor, the 'Data Output' tab is active, displaying the results of the query in a table. The table has three columns: 'title' (character varying), 'release_year' (integer), and 'rental_rate' (numeric (4,2)). The results are sorted by title in ascending order, then by release_year in descending order, and finally by rental_rate in ascending order. The first six rows are visible:

	title	release_year	rental_rate
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

At the bottom of the interface, the status bar shows 'Total rows: 1000 of 1000', 'Query complete 00:00:00.143', and the cursor position 'Ln 2, Col 53'.

Step3:

Rockbuster/postgres@PostgreSQL 15

Query Query History Scratch Pad x

```

1 SELECT rating,AVG(rental_rate) FROM film
2 GROUP BY rating

```

	rating mpaa_rating	avg numeric
1	PG	3.0518556701030928
2	R	2.9387179487179487
3	NC-17	2.9709523809523810
4	PG-13	3.0348430493273543
5	G	2.8888764044943820

Total rows: 5 of 5 Query complete 00:00:00.139 Ln 2, Col 16

Query

Query History

Scratch Pad

1

SELECT rating,MIN(rental_duration),MAX(rental_duration) FROM film

2

GROUP BY rating

Data Output

Messages

Notifications

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

Total rows: 5 of 5

Query complete 00:00:00.117

Ln 1, Col 55

STEP4:

Procedure:

While the process of migrating the data is generally the responsibility of the data engineer, they would likely use the ETL process. First, the data is Extracted from the relevant data source, which in this case would be the new tool that is collecting data on user behavior in the Android App. This data would then be Transformed which converts the data into the appropriate format. Finally, the data is Loaded into the existing data warehouse for us to analyze.

Potential Issues:

If we analyze the data from this one source before loading it into the data warehouse, we're likely going to be missing some relevant information. For one, it is just one source of information, which may create skewed analysis. If it were in combination with our other available data it would be a more complete.

Additionally, we may be less familiar with the format the data is in, or there could be errors in the data