

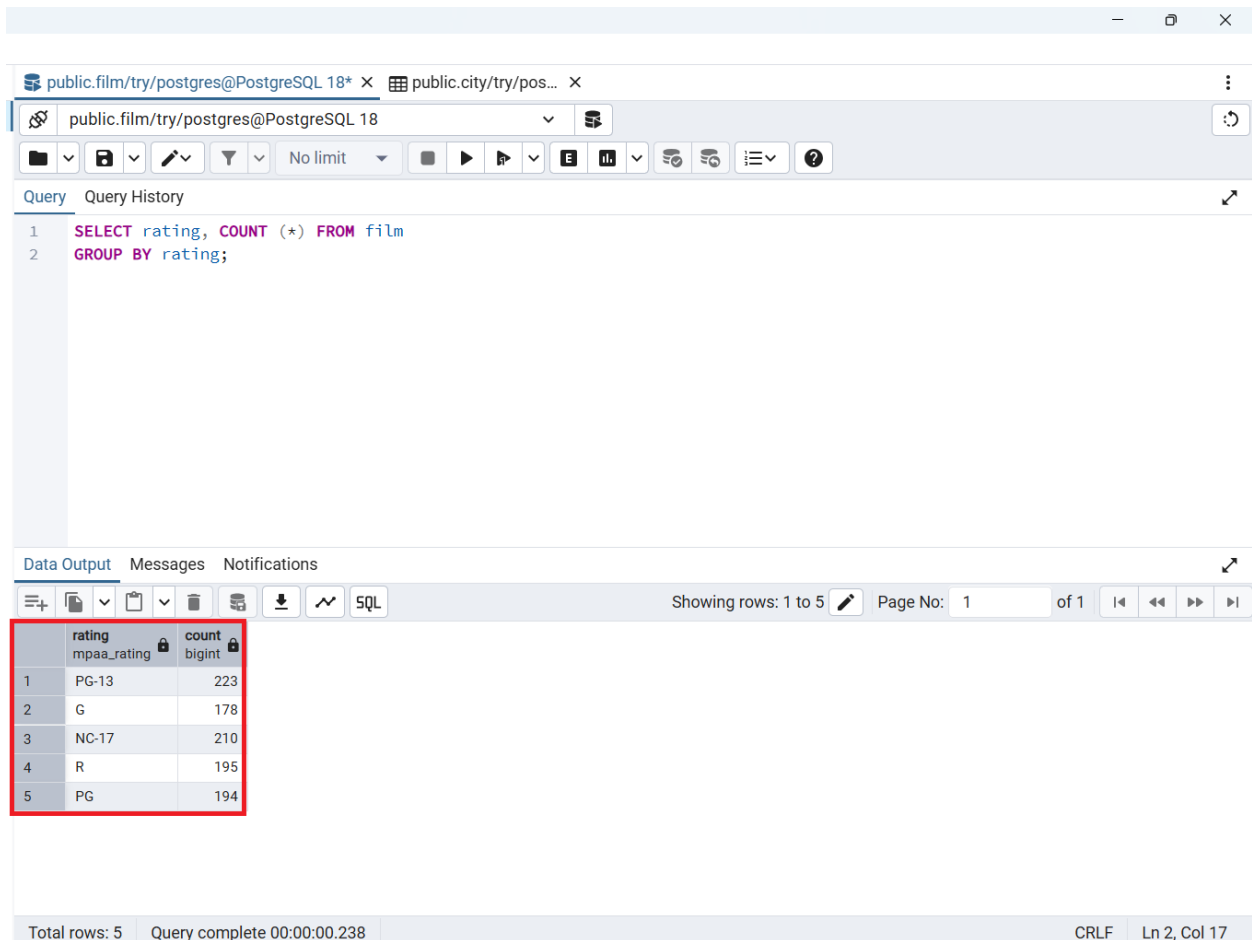
## Assignment 7: DVD Rental Queries

This assignment focuses on practicing **SQL queries** on the DVD Rental database. The main topics include **grouping data, counting entries, filtering grouped data, and sorting results**.

These exercises help improve understanding of aggregate functions, conditional logic, and working with grouped data, which are important for analyzing information in a database.

### Query 1:

**Description:** Count how many films there are for each rating in the film table. This query shows the number of films in every rating group.



The screenshot shows a PostgreSQL query editor interface. The query being executed is:

```
1 SELECT rating, COUNT (*) FROM film
2 GROUP BY rating;
```

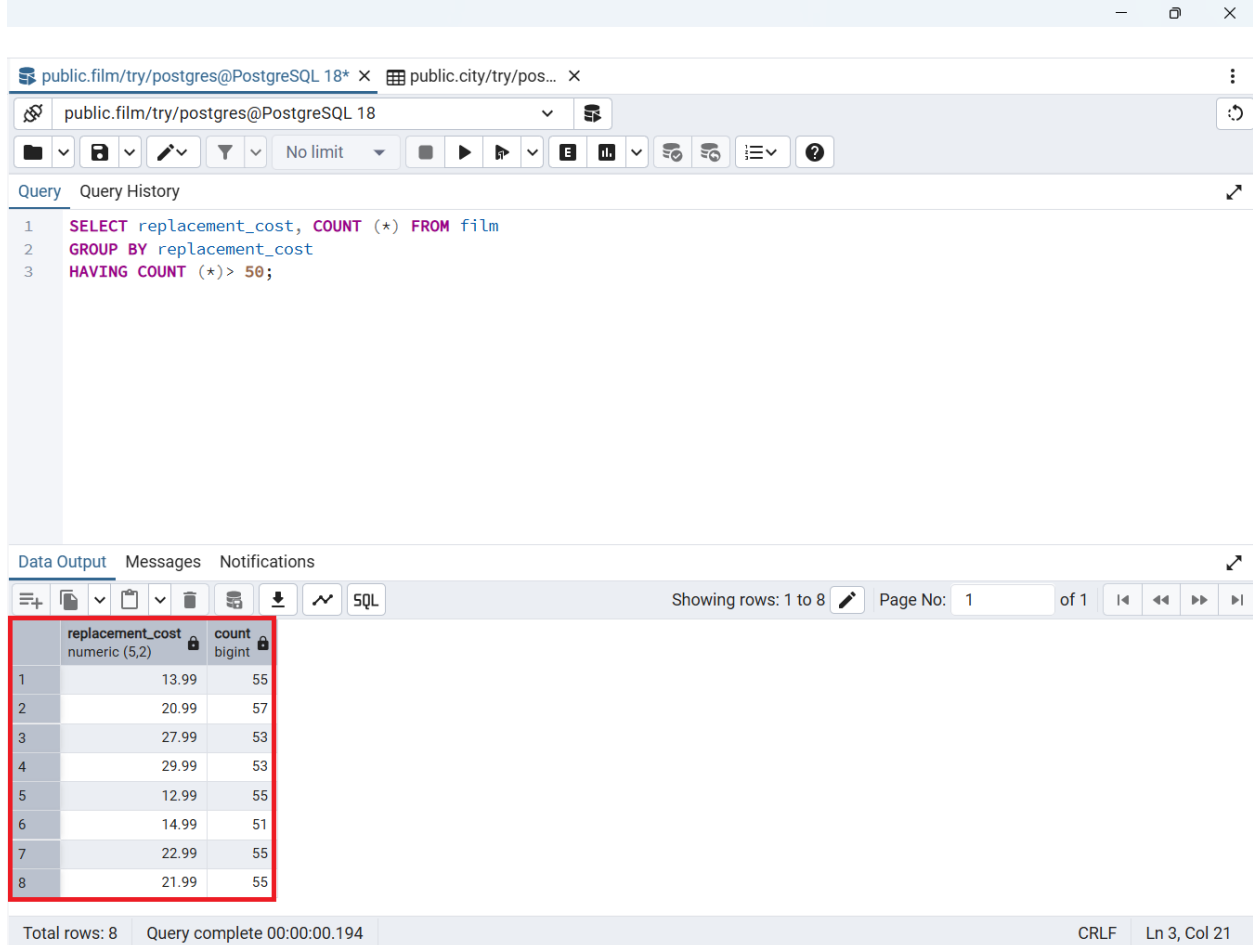
The results are displayed in a table with the following data:

	rating	count
1	PG-13	223
2	G	178
3	NC-17	210
4	R	195
5	PG	194

The table is highlighted with a red border. The interface also shows the query history, data output, and status bar indicating the query is complete.

## Query 2:

**Description:** The query shows the number of films for each replacement cost and only includes the costs that have more than 50 films. It helps find which replacement costs are most common.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT replacement_cost, COUNT (*) FROM film
2 GROUP BY replacement_cost
3 HAVING COUNT (*) > 50;
```

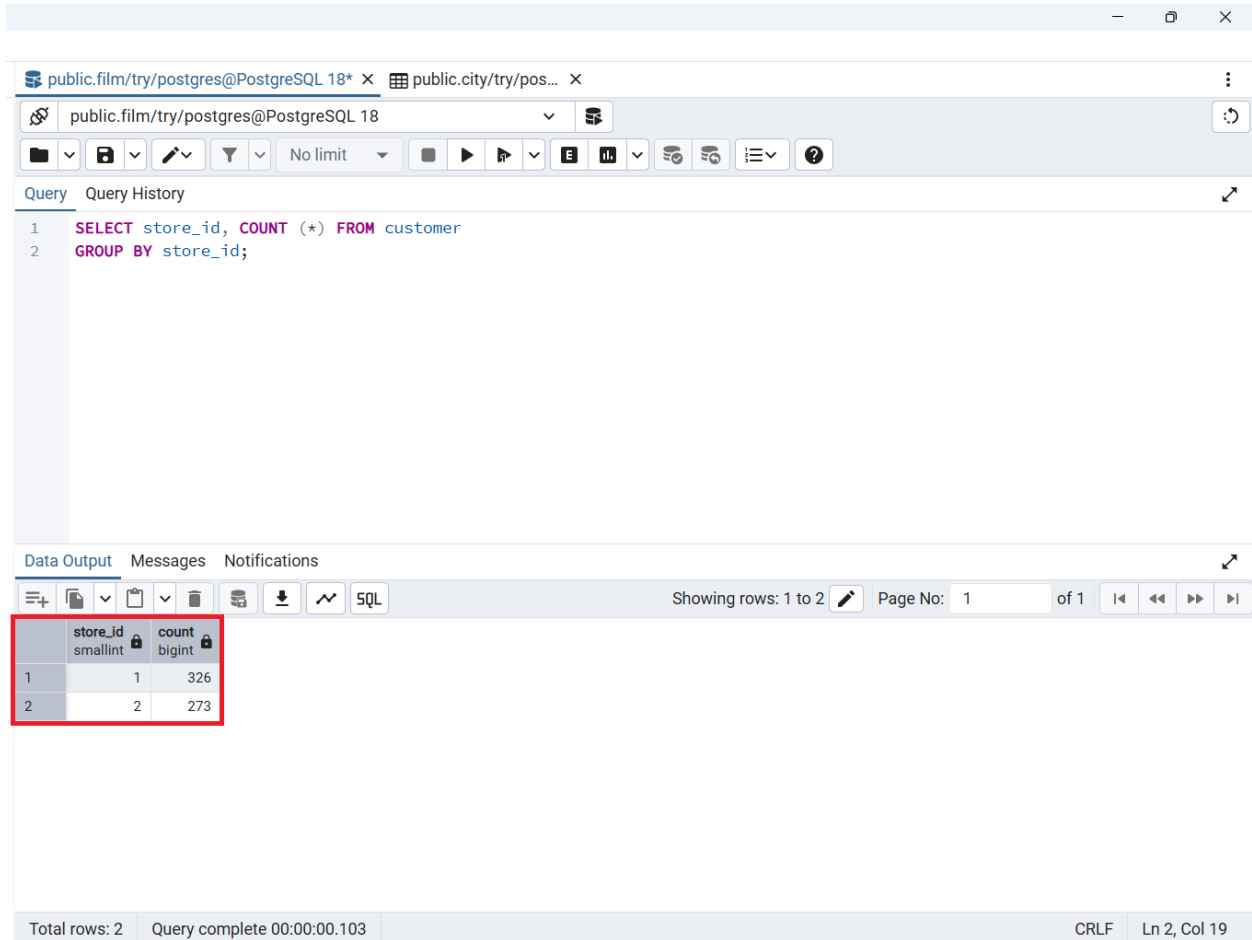
The results are displayed in a table with 8 rows. The first two columns are 'replacement\_cost' (numeric(5,2)) and 'count' (bigint). The data is as follows:

	replacement_cost numeric(5,2)	count bigint
1	13.99	55
2	20.99	57
3	27.99	53
4	29.99	53
5	12.99	55
6	14.99	51
7	22.99	55
8	21.99	55

The table is highlighted with a red border. The interface also shows a status bar at the bottom indicating 'Total rows: 8', 'Query complete 00:00:00.194', and 'CRLF Ln 3, Col 21'.

### Query 3:

**Description:** The query shows how many customers belong to each store. It helps understand the number of customers for every store.



The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT store_id, COUNT (*) FROM customer
2 GROUP BY store_id;
```

The results are displayed in a table with the following data:

	store_id smallint	count bigint
1	1	326
2	2	273

The table is highlighted with a red border. The interface also shows a status bar at the bottom indicating "Total rows: 2" and "Query complete 00:00:00.103".

## Query 4:

**Description:** The query shows the number of cities in each country and finds the country with the most cities. It helps see which country has the highest number of cities.

The screenshot shows a PostgreSQL query editor interface. The query is as follows:

```
1 SELECT (country_id), COUNT (*) FROM city
2 GROUP BY country_id
3 ORDER BY COUNT (*) DESC
4 LIMIT 1;
```

The query has been executed, and the results are displayed in the 'Data Output' tab. The results show one row with the following data:

	country_id smallint	count bigint
1	44	60

The interface also shows the status bar at the bottom: 'Total rows: 1', 'Query complete 00:00:00.093', 'CRLF', and 'Ln 4, Col 9'.