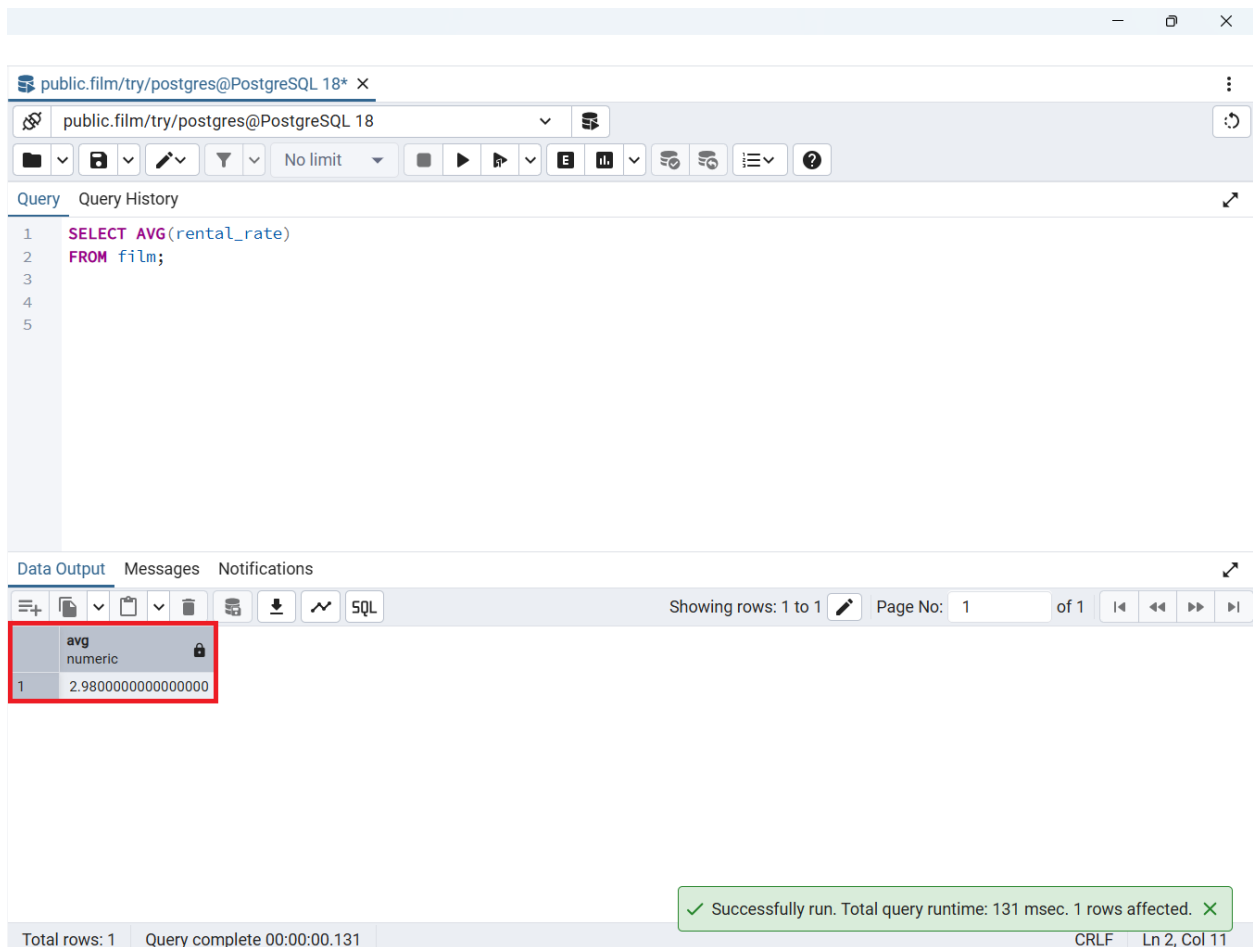


## Assignment 6: DVD Rental Queries

This query focuses on identifying and counting distinct values in the DVD Rental database. Specifically, it retrieves the number of unique replacement\_cost values for films that have a length greater than 150 minutes. The task emphasizes the use of conditional filtering with the WHERE clause and the COUNT(DISTINCT ...) aggregate function to ensure that only unique cost values are counted. This exercise helps strengthen understanding of data uniqueness, filtering conditions, and aggregate functions in SQL.

### Query 1:

**Description:** This query finds the average rental\_rate in the film table, demonstrating how to use the AVG() function to summarize numerical data in SQL.



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

```
1 SELECT AVG(rental_rate)
2 FROM film;
```

The query results are displayed in a table with the following structure:

	avg
1	2.9800000000000000

The result row is highlighted with a red box. The interface also shows a status bar at the bottom indicating: "Total rows: 1", "Query complete 00:00:00.131", and a green message box stating "Successfully run. Total query runtime: 131 msec. 1 rows affected."

## Query 2:

**Description:** This query counts the number of films in the film table whose titles start with the letter 'C', demonstrating the use of COUNT() with the LIKE operator to filter data in SQL.

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

```
1 SELECT COUNT(*)
2 FROM film
3 WHERE title LIKE 'C%';
4
```

The query has been executed, and the results are displayed in the "Data Output" tab. The results show a single row with a count of 92.

	count bigint
1	92

At the bottom of the interface, the status bar indicates: "Total rows: 1", "Query complete 00:00:00.103", "CRLF", and "Ln 4, Col 1".

### Query 3:

**Description:** This query finds the longest film (length) with a rental\_rate of 0.99 in the film table, using the MAX() function combined with a WHERE filter to analyze specific data in SQL.

The screenshot shows a PostgreSQL query editor interface. The query is: `SELECT MAX(length) FROM film WHERE rental_rate = 0.99;`. The result is displayed in a table with one row and one column, showing the value 184. The interface includes a toolbar with various icons for query execution, a query history panel, and a data output panel. The status bar at the bottom indicates "Total rows: 1" and "Query complete 00:00:00.294".

	max smallint
1	184

Total rows: 1    Query complete 00:00:00.294    CRLF    Ln 2, Col 26

## Query 4:

**Description:** In this query counts the number of unique `replacement_cost` values for films longer than 150 minutes, demonstrating the use of `COUNT(DISTINCT ...)` with a `WHERE` filter to analyze distinct data in SQL.

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

```
1 SELECT COUNT(DISTINCT replacement_cost)
2 FROM film
3 WHERE length > 150;
4
```

The query is executed, and the results are displayed in the 'Data Output' tab. The results show a single row with a count of 21.

	count
1	21

The status bar at the bottom indicates 'Total rows: 1', 'Query complete 00:00:00.058', and 'Ln 4, Col 1'.