

Project Title: Exploratory Data Analysis on Film Data Using PostgreSQL

TIJANI OLAMIDE @TECHWAY CONSULT



[This Photo](#) by Unknown Author is licensed under [CC BY](#)

TASK 2

What are the top 10 highest-grossing films in the database, and when were they released?

The screenshot shows a PostgreSQL database interface with a query editor and a results table. The query editor contains the following SQL query:

```
--What are the top 10 highest-grossing films in the database, and when were they released?  
SELECT title, gross, release_year  
FROM films  
WHERE gross IS NOT NULL  
LIMIT 10
```

The results table displays the top 10 highest-grossing films. The table has columns for an index, title, gross, and release_year. The data is as follows:

	title	gross	release_year
1	Over the Hill to the Poorhouse	3000000	1920
2	Metropolis	26435	1927
3	Pandora's Box	9950	1929
4	The Broadway Melody	2808000	1929
5	42nd Street	2300000	1933
6	Top Hat	3000000	1935
7	Modern Times	163245	1936
8	Snow White and the Seven Dwarfs	184925485	1937
9	Gone with the Wind	198655278	1939

Total rows: 10 of 10 Query complete 00:00:00.102

TASK 3

How many films in the database were released in each country, and what are the top five countries?

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main window displays a SQL query in the 'Query' tab, which is executed against a database named 'Film/postgres@Local_db'. The query is as follows:

```
1 SELECT country, COUNT(title) AS number_of_films
2 FROM films
3 GROUP BY country
4 ORDER BY number_of_films DESC
5 LIMIT 5;
```

Below the query editor, the 'Data Output' tab shows the results of the query in a table format. The table has two columns: 'country' (character varying) and 'number_of_films' (bigint). The results are as follows:

	country	number_of_films
1	USA	3750
2	UK	443
3	France	153
4	Canada	123
5	Germany	97

The status bar at the bottom indicates 'Total rows: 5 of 5' and 'Query complete 00:00:00.093'.

TASK 4

How many films are available in each language, and what are the top three languages represented?

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query and its results. The query is as follows:

```
1 SELECT language, COUNT(title) AS number_of_films
2 FROM films
3 GROUP BY language
4 ORDER BY number_of_films DESC
5 LIMIT 3;
```

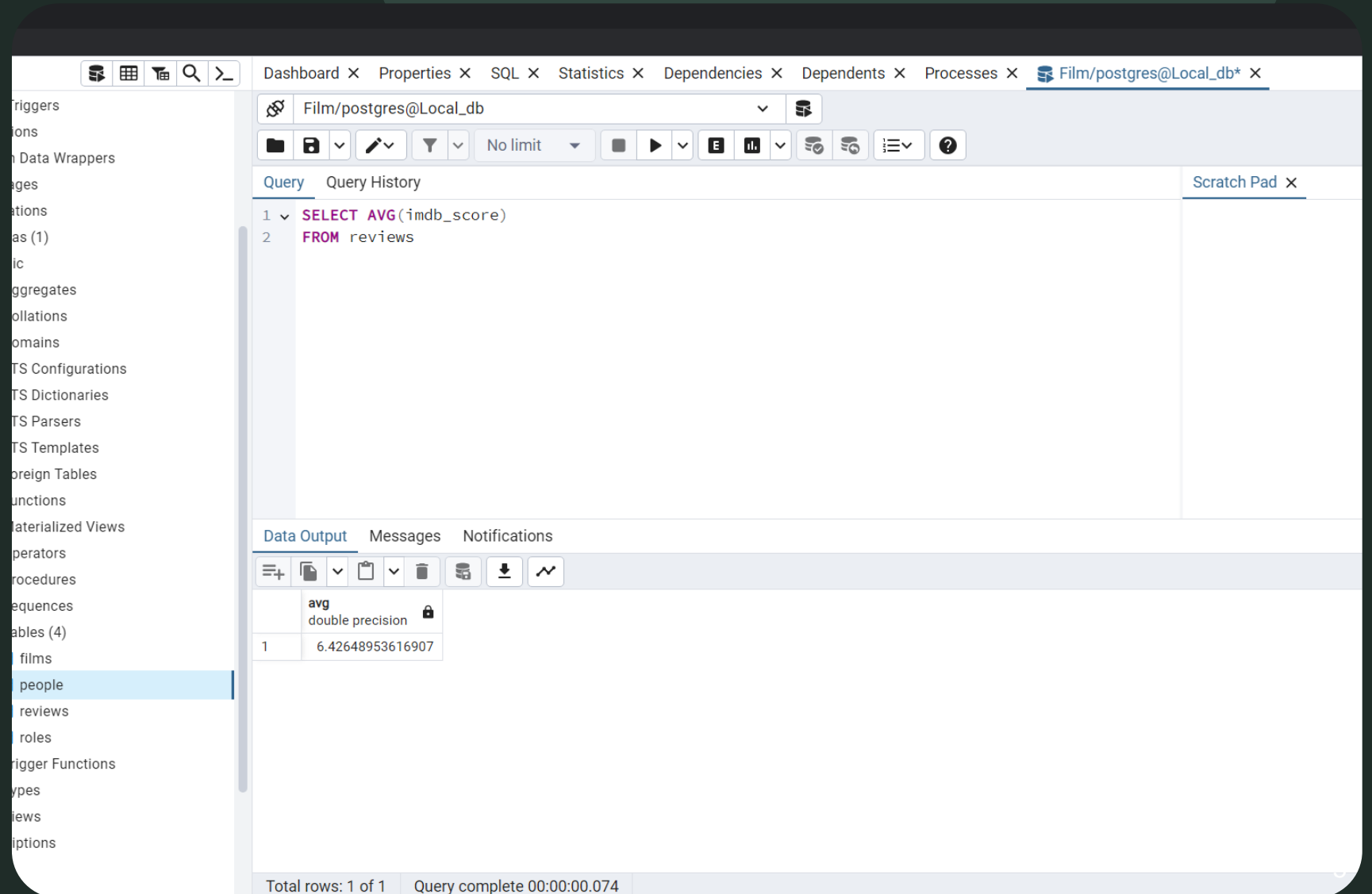
The results are shown in a table with the following data:

	language character varying	number_of_films bigint
1	English	4635
2	French	72
3	Spanish	40

The interface also shows a status bar at the bottom indicating "Total rows: 3 of 3" and "Query complete 00:00:00.106".

TASK 5

What is the average IMDb score for films in the database?



The screenshot shows a PostgreSQL database interface with a query editor and a results pane. The query editor contains the following SQL query:

```
1 SELECT AVG(imdb_score)
2 FROM reviews
```

The results pane displays the output of the query, showing a single row with the average IMDb score for films in the database. The results are as follows:

	avg
1	6.42648953616907

The interface also shows a sidebar with a list of database objects, including tables, views, and functions. The 'reviews' table is highlighted in the sidebar. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.074'.

TASK 6

Which country has made the highest profit from movies? [Tip: To avoid a biased result, use average instead of sum.]

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query in the 'Query' tab, and the 'Data Output' tab shows the results of the query. The query is a SQL statement that selects the country with the highest average profit from movies, using the 'AVG' function to avoid a biased result. The result shows Taiwan as the country with the highest average profit.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

Query Query History Scratch Pad ×

```
1 --Which country has made the highest profit from movies? [Tip: To avoid a biased result, use
2 --average instead of sum.]
3 SELECT country, AVG(gross-budget) AS avg_profit
4 FROM films
5 GROUP BY country
6 HAVING AVG(gross-budget) IS NOT NULL
7 ORDER BY avg_profit DESC
8 LIMIT 1
```

Data Output Messages Explain × Notifications

	country character varying	avg_profit numeric
1	Taiwan	49340682.000000000000

Total rows: 1 of 1 Query complete 00:00:00.128

TASK 7

Which
movie made
the highest
profit in the
21st
century?

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query in the 'Query' tab, and the 'Data Output' tab shows the results of the query. The query filters for movies released after 2000 and orders them by average profit in descending order, limiting the results to one row. The result shows 'Star Wars: Episode VII - The Force Awakens' as the movie with the highest average profit.

Triggers
Functions
Data Wrappers
Tables
Views
Aggregates
Collations
Domains
TS Configurations
TS Dictionaries
TS Parsers
TS Templates
Foreign Tables
Functions
Materialized Views
Operators
Procedures
Sequences
Tables (4)
films
people
reviews
roles
Trigger Functions
Types
Views
Options

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

Query Query History Scratch Pad ×

```
1 SELECT title, release_year, AVG(gross-budget) AS avg_profit
2 FROM films
3 WHERE release_year > 2000
4 GROUP BY title, release_year
5 HAVING AVG(gross-budget) IS NOT NULL
6 ORDER BY avg_profit DESC
7 LIMIT 1
8
```

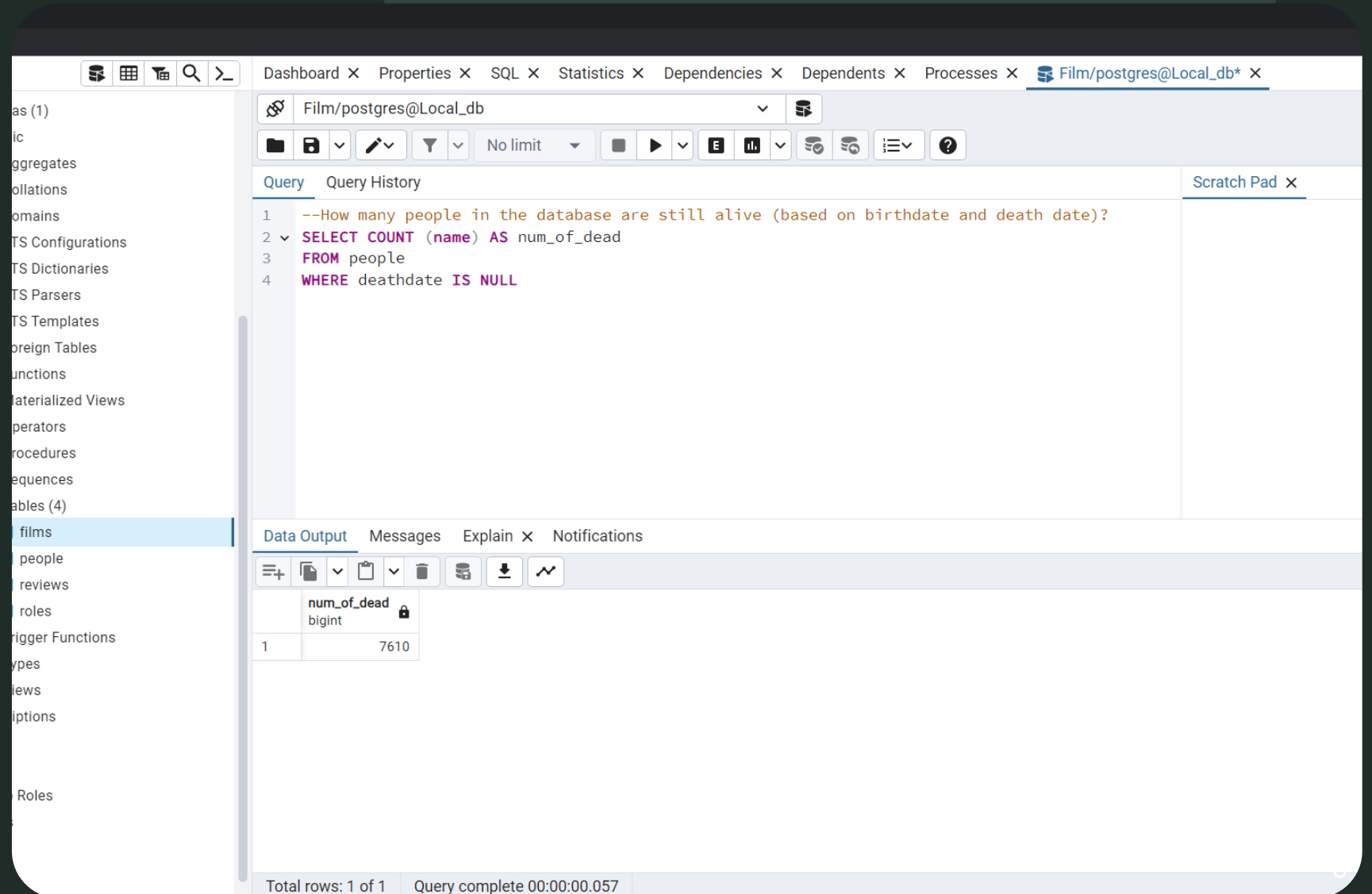
Data Output Messages Notifications

	title	release_year	avg_profit
1	Star Wars: Episode VII - The Force Awakens	2015	691627416.00000000

Total rows: 1 of 1 Query complete 00:00:00.117

TASK 8

How many people in the database are still alive (based on birthdate and death date)?



The screenshot shows a PostgreSQL database management tool interface. The left sidebar lists database objects, with 'films' selected. The main window displays a SQL query in the 'Query' tab:

```
--How many people in the database are still alive (based on birthdate and death date)?  
1 SELECT COUNT (name) AS num_of_dead  
2 FROM people  
3 WHERE deathdate IS NULL  
4
```

The 'Data Output' tab shows the results of the query:

	num_of_dead bigint
1	7610

The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.057'.

TASK 9

Which year
has the
highest
number of
movie
releases?

The screenshot shows a PostgreSQL database management tool interface. The left sidebar contains a list of database objects, with 'people' highlighted. The main window displays a SQL query in the 'Query' tab, which asks for the year with the highest number of movie releases. The 'Data Output' tab shows the results of the query, which is a single row for the year 2009 with 260 releases. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.062'.

```
--Which year has the highest number of movie releases?  
SELECT release_year, COUNT(title) AS num_of_releases  
FROM films  
GROUP BY release_year  
ORDER BY COUNT(title) DESC  
LIMIT 1
```

	release_year integer	num_of_releases bigint
1	2009	260

Total rows: 1 of 1 Query complete 00:00:00.062

TASK 10

Determine
the top 10
people with
the most
roles in the
database

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area is divided into a query editor and a results pane. The query editor contains a SQL query to find the top 10 people with the most roles. The results pane shows a table with 10 rows of data.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

Query Query History Scratch Pad ×

```
1 --Determine the top 10 people with the most roles in the database.
2 SELECT name, COUNT(roles) AS num_roles
3 FROM roles
4 JOIN people ON roles.id = people.id
5 GROUP BY name
6 ORDER BY num_roles DESC
7 LIMIT 10
```

Data Output Messages Explain × Notifications

	name character varying	num_roles bigint
1	David Cross	1
2	Masayuki Ochiai	1
3	Scott Glenn	1
4	Jacob Vargas	1
5	Phil Fish	1
6	Jack Carson	1
7	Brian Van Holt	1
8	Steve Gonsalves	1
9	Roger Jackson	1

Total rows: 10 of 10 Query complete 00:00:00.093

TASK 11

Who are the
top 10
actors or
directors
with the
most roles
in the
database?

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query in the 'Query' tab, and the 'Data Output' tab shows the results of the query. The query is designed to find the top 10 people with the most roles in the database. The results table has two columns: 'name' and 'num_roles'. The data is as follows:

	name	num_roles
1	David Cross	1
2	Masayuki Ochiai	1
3	Scott Glenn	1
4	Jacob Vargas	1
5	Phil Fish	1
6	Jack Carson	1
7	Brian Van Holt	1
8	Steve Gonsalves	1
9	Roger Jackson	1

Total rows: 10 of 10 Query complete 00:00:00.093

TASK 12

Identify how many people in the database are still alive.

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query in a text editor, and below it, the results of the query are shown in a table. The query is: `-- Identify how many people in the database are still alive.`, `SELECT COUNT (name) AS num_of_dead_people`, `FROM people`, `WHERE deathdate IS NOT NULL`. The results table has one column, `num_of_dead_people`, and one row with the value `787`. The status bar at the bottom indicates 'Total rows: 1 of 1' and 'Query complete 00:00:00.098'.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

No limit

Query Query History Scratch Pad ×

```
1 -- Identify how many people in the database are still alive.
2 SELECT COUNT (name) AS num_of_dead_people
3 FROM people
4 WHERE deathdate IS NOT NULL
```

Data Output Messages Explain × Notifications

	num_of_dead_people
1	787

Total rows: 1 of 1 Query complete 00:00:00.098

TASK 13

Calculate
the average
number of
user and
critic
reviews for
films

The screenshot shows a database management interface with a sidebar on the left containing a tree view of database objects. The main area displays a SQL query in the 'Query' tab, and the 'Data Output' tab shows the results of the query. The query is a SELECT statement that calculates the average number of user reviews for each film, grouped by the number of critic reviews. The results are displayed in a table with two columns: 'num_critic' (integer) and 'avg_num_user' (numeric). The table contains 9 rows of data. The status bar at the bottom indicates that the query is complete and has returned 530 rows.

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

Query Query History Scratch Pad ×

```
1 --Calculate the average number of user and critic reviews for films
2 SELECT num_critic, AVG (num_user) AS avg_num_user
3 FROM reviews
4 GROUP BY num_critic
5
```

Data Output Messages Explain × Notifications

	num_critic integer	avg_num_user numeric
1	828	4080.0000000000000000
2	[null]	7.2592592592592593
3	184	287.7000000000000000
4	87	112.9166666666666667
5	273	578.3333333333333333
6	394	397.0000000000000000
7	51	93.8500000000000000
8	272	294.6666666666666667
9	70	135.5000000000000000

Total rows: 530 of 530 Query complete 00:00:00.115

Dashboard × Properties × SQL × Statistics × Dependencies × Dependents × Processes × Film/postgres@Local_db* ×

Film/postgres@Local_db

Query Query History Scratch Pad ×

```
1 --Identify films with the highest number of user and critic review
2 ✓ SELECT title, num_critic, AVG(num_user) AS avg_num_user
3 FROM films
4 JOIN reviews ON films.id = reviews.film_id
5 WHERE num_critic IS NOT NULL
6 GROUP BY title, num_critic
7 ORDER BY AVG(num_user) DESC, num_critic DESC
8 LIMIT 1
9
10
11
12
```

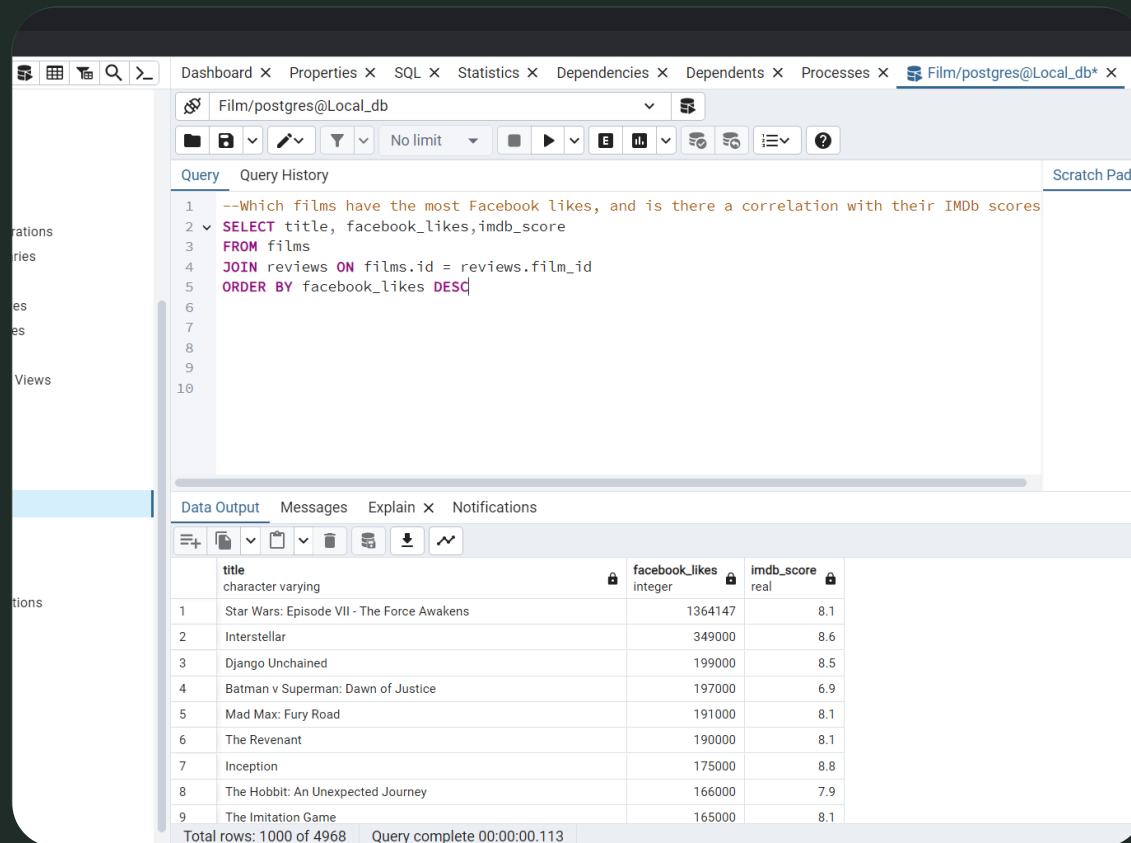
Data Output Messages Explain × Notifications

	title character varying	num_critic integer	avg_num_user numeric
1	The Lord of the Rings: The Fellowship of the Ring	297	5060.00000000000000000000

Total rows: 1 of 1 Query complete 00:00:00.103

TASK 14

Identify films with the highest number of user and critic reviews



Dashboard x Properties x SQL x Statistics x Dependencies x Dependents x Processes x Film/postgres@Local_db* x

Film/postgres@Local_db

Query Query History Scratch Pad x

```

1 --Which films have the most Facebook likes, and is there a correlation with their IMDb scores
2 SELECT title, facebook_likes, imdb_score
3 FROM films
4 JOIN reviews ON films.id = reviews.film_id
5 ORDER BY facebook_likes DESC
6
7
8
9
10

```

Data Output Messages Explain x Notifications

	title character varying	facebook_likes integer	imdb_score real
1	Star Wars: Episode VII - The Force Awakens	1364147	8.1
2	Interstellar	349000	8.6
3	Django Unchained	199000	8.5
4	Batman v Superman: Dawn of Justice	197000	6.9
5	Mad Max: Fury Road	191000	8.1
6	The Revenant	190000	8.1
7	Inception	175000	8.8
8	The Hobbit: An Unexpected Journey	166000	7.9
9	The Imitation Game	165000	8.1

Total rows: 1000 of 4968 Query complete 00:00:00.113

TASK 15

Which films have the most Facebook likes, and is there a correlation with their IMDb scores?

Project Report

LIST OF THE FIRST TEN GROSSING COUNTRY

"Over the Hill to the Poorhouse"

"Metropolis"

"Pandora's Box"

"The Broadway Melody"

"42nd Street"

"Top Hat"

"Modern Times"

"Snow White and the Seven Dwarfs"

"Gone with the Wind"

"The Wizard of Oz"

Project Report

FIRST FIVE COUNTRIES WITH HIGHEST NUMBER OF RELEASED MOVIE

USA,
UK,
France,
Canada,
Germany

COUNTRY WITH THE HIGHEST PROFIT

TAIWAN

AVERAGE IMDB SCORES

6.42

MOVIE WITH THE HIGHEST PROFIT

STAR WARS:EPISODE VII- THE FORCE AWAKENS

YEAR WITH HIGHEST NUMBER OF RELEASED MOVIE

2009

TOP THREE LANGUAGES WITH THE MOST RELEASE FILMS

ENGLISH

FRENCH

SPANISH