DATAMITES - INTERNSHIP

PRSQ-02-IMDB MOVIES

Project ID: **PRSQL-01 IMDB MOVIES**

Project Team ID: **PTID-CDA-SEP-25-1036**

Database Management System (DBMS)**: SQL SERVER**

Language**: SQL**

Project By**: RAKSHITA SHANKRAPPA APPAJI**

This professionally enhanced report now contains detailed explanations below every SQL query screenshot for better understanding and presentation quality.

TABLE OF CONTENTS

Introduction:

This report presents an overview and analysis of the **IMDB Movies database**, which contains structured information about movies and their directors. The objective of this analysis is to use SQL queries to explore the dataset and extract meaningful insights related to movie performance, director activity, popularity, revenue, and audience ratings.

Project Overview:

This report presents an overview and analysis of the IMDB Movies database, which contains structured information about movies and their directors. The objective of this analysis is to use SQL queries to explore the dataset and extract meaningful insights related to movie performance, director activity, popularity, revenue, and audience ratings.

The database consists of two main tables:

- **Directors Table**: Contains director details such as name, gender, department, and a unique director ID.
- **Movies Table**: Contains movie-related information including title, budget, revenue, popularity, release date, ratings, and a director reference ID.

Both tables are related through a **one-to-many relationship**, where one director can direct multiple movies. The tables are joined using the director ID as the key.

The movies table consists of data like:

- id
- original title
- title
- budget
- revenue
- popularity
- release date
- vote average
- vote count
- overview
- tagline
- uid
- director_id

The directors table consists of data like:

- id
- name
- gender (0/2 = Male, 1 = Female)
- department
- uid

Quires need to be performed:

a) Can you get all data about movies?

b) How do you get all data about directors?

c) Check how many movies are present in IMDB.

d) Find these 3 directors: James Cameron; Luc Besson; John Woo

e) Find all directors with name starting with S.

f) Count female directors.

g) Find the name of the 10th first women directors?

h) What are the 3 most popular movies?

i) What are the 3 most bankable movies?

j) What is the most awarded average vote since the January 1st, 2000?

k) Which movie(s) were directed by Brenda Chapman?

l) Which director made the most movies?

m) Which director is the most bankable?

Exploring the database:

Exploring the database means first understanding what data is available and how it is organized. In this project, the database contains two main tables: Movies and Directors. The Movies table stores information about films such as their names, release dates, ratings, popularity, budget, and revenue. The Directors table stores details about the people who directed these movies, including their names and gender.

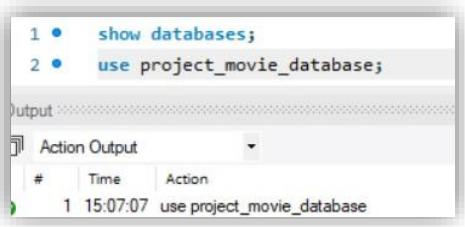
During database exploration, we mainly focus on viewing all records, checking how many movies and directors are present, and understanding how both tables are connected using the director ID. This step helps us know what columns exist, what type of data each column holds, and how we can combine movie and director information.

Overall, exploring the database helps us become familiar with the data before performing deeper analysis or writing complex SQL queries.

SHOW DATABASES:

Show database;

Use project_movie_database;



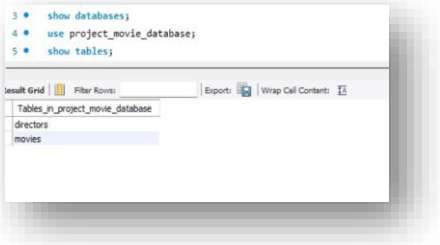
.This query is used to display all the databases available in the SQL Server. From this, we can identify our working database named:

USE PROJECT_MOVIE_DATABASE;

This query is used to select the specific database project_movie_database to work with in the Database Management System (DBMS).
It allows us to perform all further SQL operations inside this database.

SHOW TABLES;

Show table;



This query displays all the tables present in the selected database.

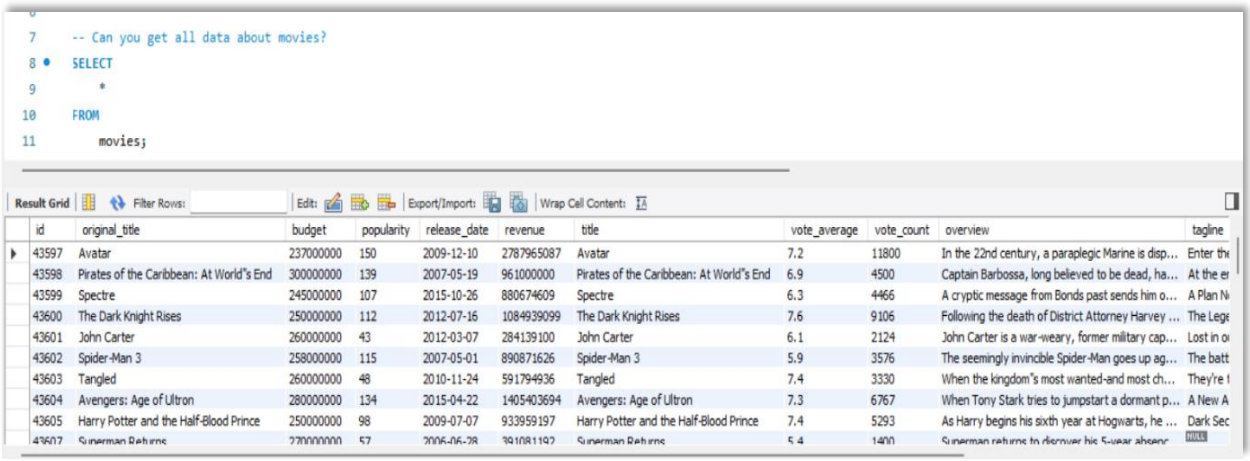
The tables available in this database are:

- movies
- directors

QUIRES AND THEIR OUTPUTS

a. Can you get all data about movies?

SELECT * FROM movies;



id	original_title	budget	popularity	release_date	revenue	title	vote_average	vote_count	overview	tagline
43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar	7.2	11800	In the 22nd century, a paraplegic Marine is disp...	Enter the
43598	Pirates of the Caribbean: At World"s End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World"s End	6.9	4500	Captain Barbossa, long believed to be dead, ha...	At the er
43599	Spectre	245000000	107	2015-10-26	880674609	Spectre	6.3	4466	A cryptic message from Bonds past sends him o...	A Plan N
43600	The Dark Knight Rises	250000000	112	2012-07-16	1084939099	The Dark Knight Rises	7.6	9106	Following the death of District Attorney Harvey ...	The Lege
43601	John Carter	260000000	43	2012-03-07	284139100	John Carter	6.1	2124	John Carter is a war-weary, former military cap...	Lost in o
43602	Spider-Man 3	258000000	115	2007-05-01	890871626	Spider-Man 3	5.9	3576	The seemingly invincible Spider-Man goes up ag...	The batt
43603	Tangled	260000000	48	2010-11-24	591794936	Tangled	7.4	3330	When the kingdom"s most wanted-and most ch...	They're
43604	Avengers: Age of Ultron	280000000	134	2015-04-22	1405403694	Avengers: Age of Ultron	7.3	6767	When Tony Stark tries to jumpstart a dormant p...	A New A
43605	Harry Potter and the Half-Blood Prince	250000000	98	2009-07-07	933959197	Harry Potter and the Half-Blood Prince	7.4	5293	As Harry begins his sixth year at Hogwarts, he ...	Dark Sec
43607	Superman Returns	270000000	57	2006-06-28	391081192	Superman Returns	5.4	1400	Superman returns to discover his 5-year absenc	NULL

Description: This query displays complete information of all movies stored in the database using SELECT * FROM movies. It retrieves columns such as movie ID, title, budget, revenue, popularity, release date, vote average and vote count. This helps to understand the full dataset before performing analysis.

b. How do you get all data about directors?

SELECT * FROM directors;

```
12
13      -- How do you get all data about directors?
14 •    SELECT
15          *
16      FROM
17          directors;
```

Result Grid | Filter Rows: | Edit:

name	id	gender	uid	department
James Cameron	4762	2	2710	Directing
Gore Verbinski	4763	2	1704	Directing
Sam Mendes	4764	2	39	Directing
Christopher Nolan	4765	2	525	Directing
Andrew Stanton	4766	2	7	Directing
Sam Raimi	4767	2	7623	Directing
Byron Howard	4768	2	76595	Directing
Joss Whedon	4769	2	12891	Directing
David Yates	4770	2	11343	Directing
Zack Snyder	4771	2	15217	Directing
Bryan Singer	4772	2	9032	Directing

directors 4 ×

Description: This query retrieves all records from the director's table. It lists director names, gender, department and their unique identifiers. This helps to understand the number of directors and their attributes in the dataset.

c. Check how many movies are present in IMDB.

SELECT COUNT (*) AS total_movies FROM movies;

```
19      -- Check how many movies are present in IMDB
20 •    SELECT
21          COUNT(*)
22      FROM
23          movies;
24
```

Result Grid | Filter Rows: | Export: | Wrap

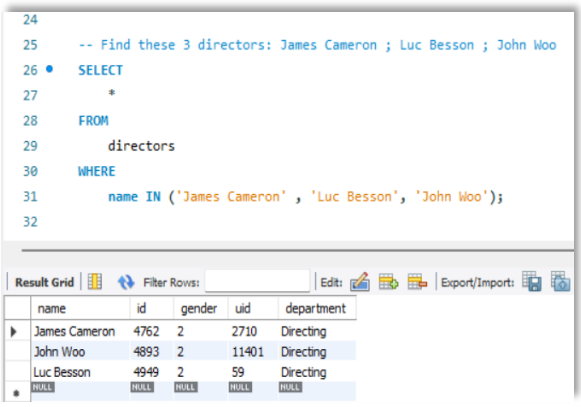
COUNT(*)
47

Description: This query counts how many movies are stored in the database using COUNT (*) function. It returns a single numeric value representing the total number of movies available for analysis.

d. Find these 3 directors: James Cameron; Luc Besson; John Woo

SELECT * FROM directors

WHERE name IN ('James Cameron','Luc Besson','John Woo');



```
24
25    -- Find these 3 directors: James Cameron ; Luc Besson ; John Woo
26    SELECT
27        *
28    FROM
29        directors
30    WHERE
31        name IN ('James Cameron' , 'Luc Besson', 'John Woo');
32
```

Result Grid | Filter Rows: | Edit: | Export/Import:

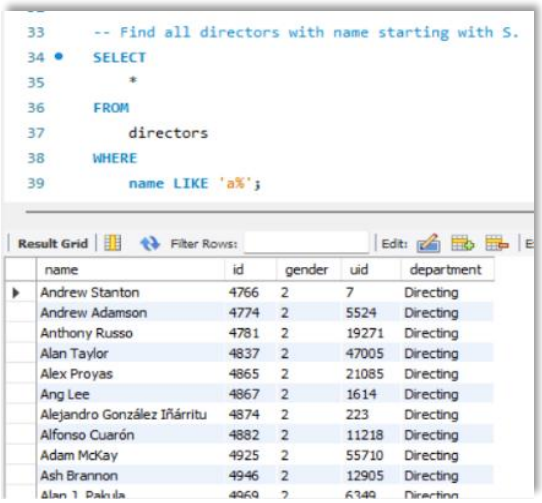
name	id	gender	uid	department
James Cameron	4762	2	2710	Directing
John Woo	4893	2	11401	Directing
Luc Besson	4949	2	59	Directing
NULL	NULL	NULL	NULL	NULL

Description: This query filters and retrieves information about three famous directors: James Cameron, Luc Besson and John Woo. The WHERE clause with IN condition is used to match multiple names in a single query.

e. Find all directors with name starting with S.

SELECT * FROM directors

WHERE name LIKE 'S%';



```
33    -- Find all directors with name starting with S.
34    SELECT
35        *
36    FROM
37        directors
38    WHERE
39        name LIKE 'a%';
```

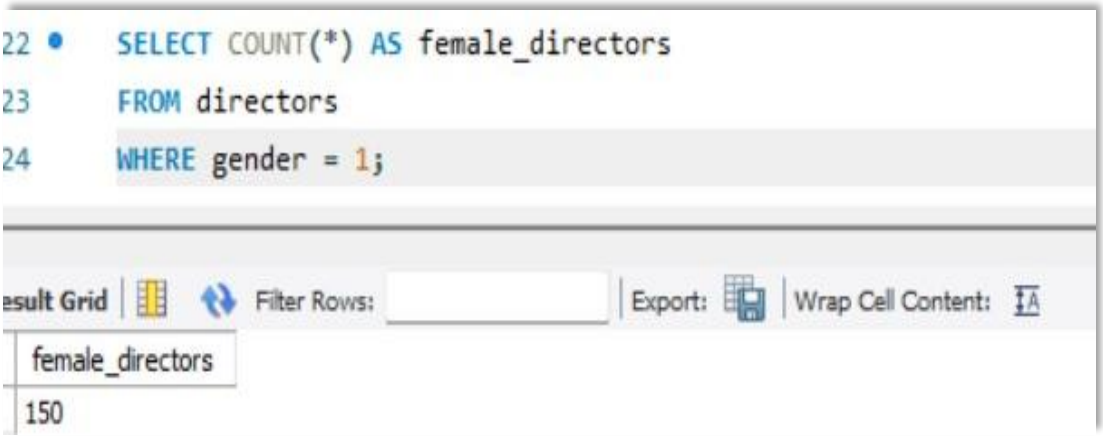
Result Grid | Filter Rows: | Edit:

name	id	gender	uid	department
Andrew Stanton	4766	2	7	Directing
Andrew Adamson	4774	2	5524	Directing
Anthony Russo	4781	2	19271	Directing
Alan Taylor	4837	2	47005	Directing
Alex Proyas	4865	2	21085	Directing
Ang Lee	4867	2	1614	Directing
Alejandro González Iñárritu	4874	2	223	Directing
Alfonso Cuarón	4882	2	11218	Directing
Adam McKay	4925	2	55710	Directing
Ash Brannon	4946	2	12905	Directing
Alan J. Pakula	4969	2	6349	Directing

Description: This query displays all directors whose names start with the letter 'S' using LIKE 'S%'. It is useful for pattern matching and filtering based on text values.

f. Count female directors.

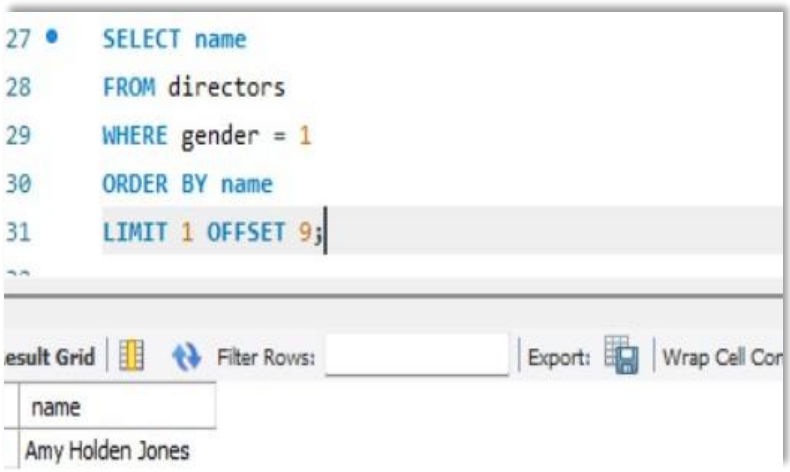
SELECT COUNT (*) AS female_directors FROM
director WHERE gender = 1;



Description: This query counts total female directors in the database by checking gender = 1. It helps in gender-based analysis and representation study.

g. Find the name of the 10th first women directors?

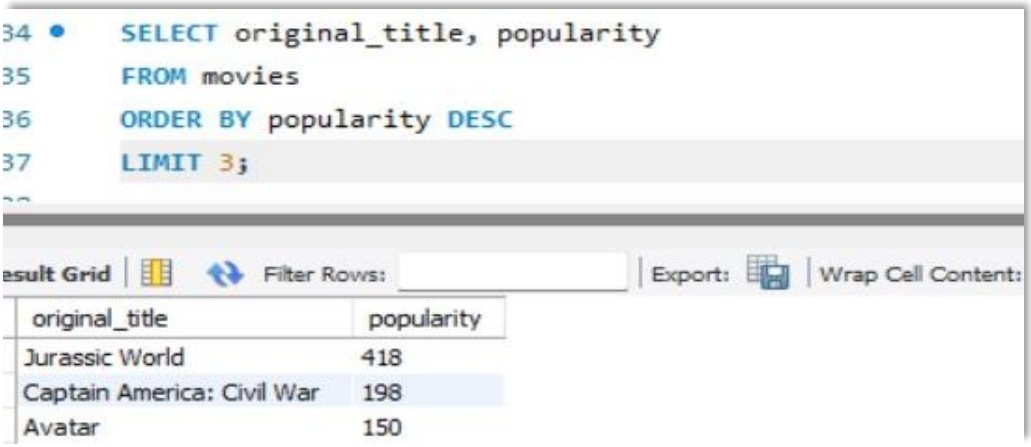
SELECT name FROM directors WHERE gender = 1
ORDER BY name LIMIT 1 OFFSET 9;



Description: This query sorts all female directors alphabetically and retrieves the 10th director using LIMIT and OFFSET. It demonstrates ordering and positional record extraction techniques.

h. What are the 3 most popular movies?

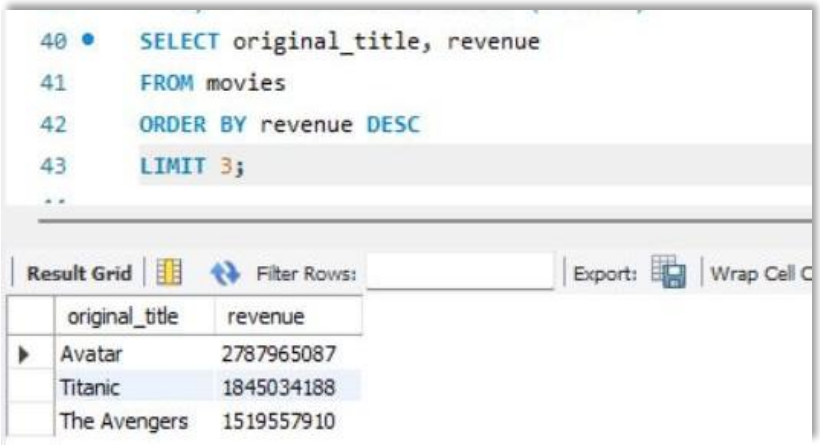
SELECT original_title, popularity FROM movies
ORDER BY popularity DESC LIMIT 3;



Description: This query retrieves the three most popular movies by sorting popularity in descending order. It helps identify movies with highest audience interest levels.

i. What are the 3 most bankable movies?

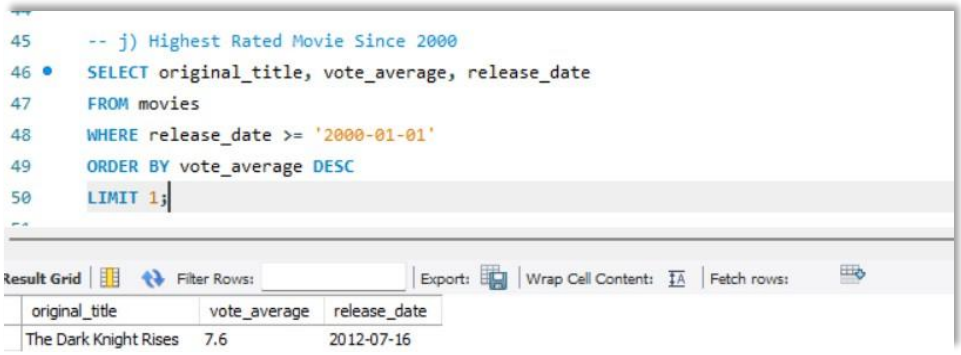
SELECT original_title, revenue
FROM movies
ORDER BY revenue DESC
LIMIT 3;



Description: This query lists the top three highest revenue generating movies. It is useful for financial performance analysis and profitability evaluation.

j. What is the most awarded average vote since the January 1st, 2000?

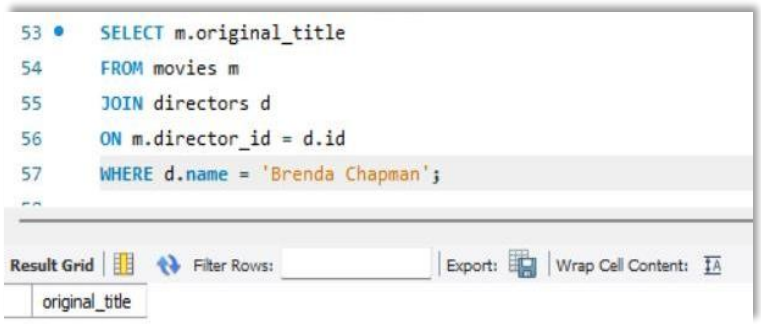
```
SELECT original_title, vote_average, release_date
FROM movies
WHERE release_date >= '2000-01-01'
ORDER BY vote_average DESC
LIMIT 1;
```



Description: This query finds the highest voted/rated movie released after January 1st 2000. It filters movies by date, sorts by ratings and picks the best performing one.

k. Which movie(s) were directed by Brenda Chapman?

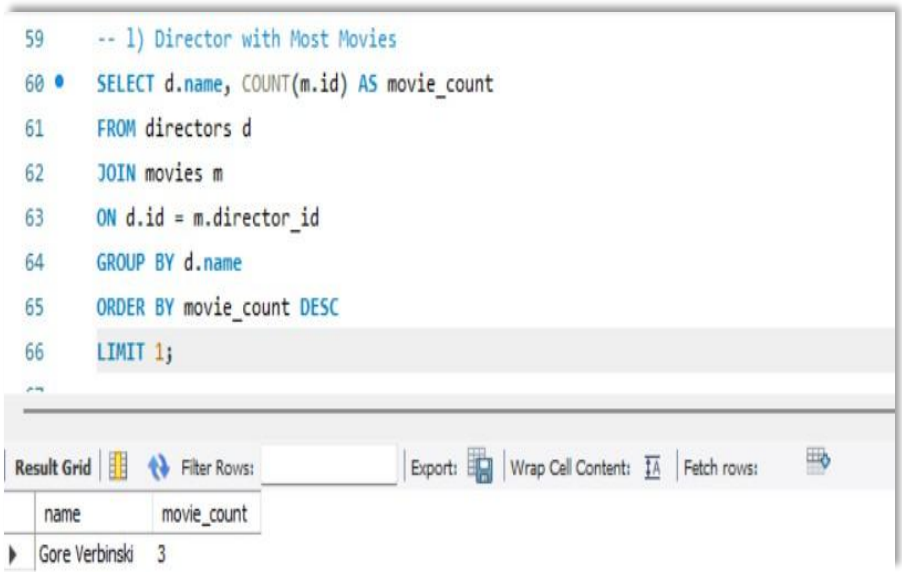
```
SELECT m.original title
FROM movies m
JOIN directors d
ON m.director id = d.id
WHERE d.name = 'Brenda Chapman';
```



Description: This query identifies movies directed by Brenda Chapman using JOIN between movies and directors' tables. It demonstrates relational database linking and lookup.

l. Which director made the most movies?

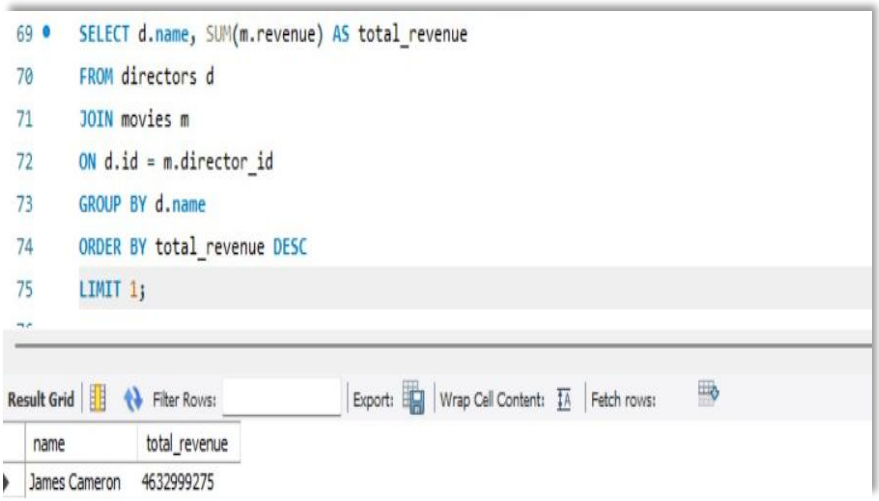
```
SELECT d.name, COUNT(m.id) AS movie_count
FROM directors d
JOIN movies m
ON d.id = m.director_id
GROUP BY d.name
ORDER BY movie_count DESC
LIMIT 1;
```



Description: This query calculates which director has directed the highest number of movies using GROUP BY and COUNT. It helps determine the most productive director.

m. Which director is the most bankable?

```
SELECT d.name, SUM(m.revenue) AS total_revenue
FROM directors d
JOIN movies m
ON d.id = m.director_id
GROUP BY d.name
ORDER BY total_revenue DESC
LIMIT 1;
```



Description: This query identifies the most bankable director based on highest total movie revenue generated. SUM () aggregation with GROUP BY is used to compute revenue contribution of each director.

CONCLUSION:

In conclusion, the IMDB Movies SQL analysis project provides a comprehensive and practical learning experience that bridges the gap between theoretical SQL knowledge and real-world data analysis. By working with a structured relational database containing detailed information about movies and directors, this project enables learners to understand how data is stored, related, and analyzed in professional environments. The use of an IMDB-style dataset makes the learning process engaging and relevant, as it reflects scenarios commonly encountered in entertainment analytics and business intelligence.

Throughout the project, various SQL concepts are applied to solve meaningful analytical problems. These include basic operations such as retrieving records and counting data, as well as advanced techniques like joining multiple tables, applying aggregate functions, filtering based on conditions, and sorting results to identify top-performing movies and directors. Such tasks help in developing a strong foundation in SQL querying and logical thinking, which are essential skills for any data analyst or database professional.

The project also highlights the importance of understanding relationships within a database. By correctly linking the Movies and Directors tables using primary and foreign keys, learners gain practical insight into relational database design and data integrity. This reinforces best practices in database management and demonstrates how accurate joins are crucial for producing reliable analytical results.

Additionally, the analysis goes beyond simple data retrieval by encouraging evaluation of real-world metrics such as movie popularity, revenue, audience ratings, and director performance. The inclusion of gender-based analysis further adds value by promoting awareness of diversity and representation within the dataset. These aspects collectively enhance analytical thinking and demonstrate how data can be used to answer both business-driven and socially relevant questions.

Overall, this project serves as an effective end-to-end SQL practice exercise. It strengthens technical proficiency, improves problem-solving abilities, and builds confidence in handling complex datasets. By completing this analysis, learners are better prepared to apply SQL skills in real-world scenarios, supporting data-driven decision-making and advancing their capabilities in data analytics and database management.