**PRSQ – 02 - IMDB Movies**



**Team ID:** PTID-CDA-DEC-24-283

**Project ID:** PRSQ-02

**Project Name:** IMDB Movies

**By**

***Team member 1: -***

**Name: Raasi P**

**Email:** [**rashireddy981212@gmail.com**](mailto:rashireddy981212@gmail.com)

***Team member 2: -***

**Name: Neetu Sharma**

**Email:** [**neets.2012@gmail.com**](mailto:neets.2012@gmail.com)

**Perform the Problem Queries:**

1. Can you get all data about movies?
2. How do you get all data about directors?
3. Check how many movies are present in IMDB.
4. Find these 3 directors: James Cameron; Luc Besson; John Woo.
5. Find all directors with name starting with S.
6. Count female directors.
7. Find the name of the 10th first women directors?
8. What are the 3 most popular movies?
9. What are the 3 most bankable movies?
10. What is the most awarded average vote since the January 1st, 2000?
11. Which movie(s) were directed by Brenda Chapman?
12. Which director made the most movies?
13. Which director is the most bankable?

**Attribute Information:**

**Table Directors:**

Name: Name of the Director

ID: Unique ID of the Directors

Gender: Gender of the Director→ 0/2=Male, 1=Female

Department: Department of the Directors.

**Table Movies:**

ID: Unique Id for Movies

Original title: Movie name

Budget: Budget of the movie

Popularity: Popularity of the Movie

Release Date: Release date for the movies

Revenue: Revenue collected movie

Title: Initial title of the movie

Vote Average: Average IMDB rating

Vote Count: Number of Vote the movie got

Overview: Description of the movie

Tagline: Tagline of the Movie

UID: Unique ID for Movie

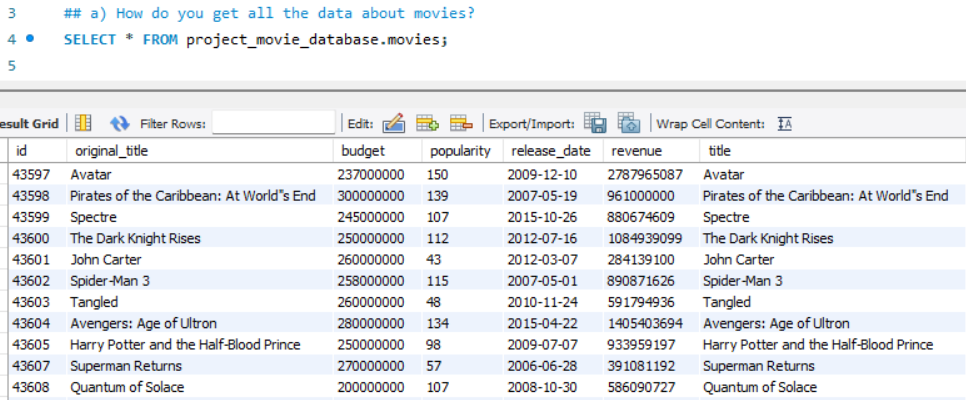
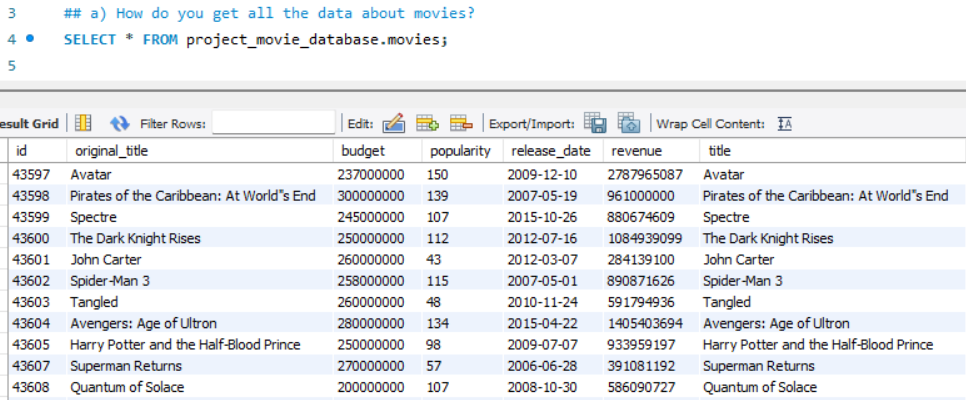
Director ID: Director ID

**QUERIES WITH ANSWERS**

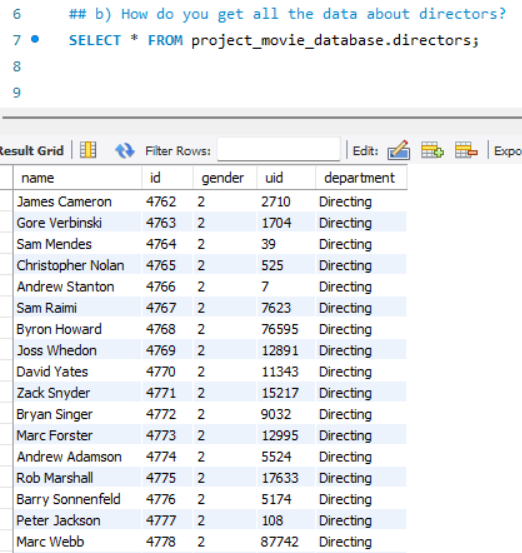
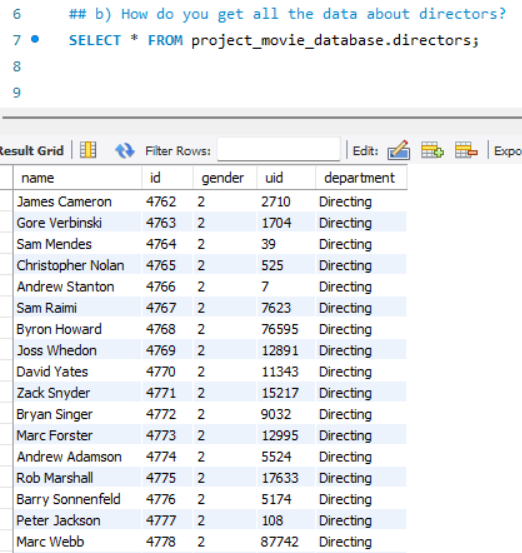
**PERFORM THE PROBLEM QUERIES**

|  |
| --- |
| *a) How do you get all the data about movies?*  SELECT \* FROM project\_movie\_database.movies; |
| *b) How do you get all the data about directors?*  SELECT \* FROM project\_movie\_database.directors; |
| *c) Check how many movies are present in IMDB*  SELECT COUNT(\*) AS movie\_count FROM project\_movie\_database.movies; |
| *d) Find these 3 directors: James Cameron; Luc Besson; John Woo*  SELECT \* FROM project\_movie\_database.directors  WHERE `name` IN ('James Cameron', 'Luc Besson', 'John Woo'); |
| *e) Find all the directors with name starting with S*  SELECT `name` FROM project\_movie\_database.directors  WHERE `name` LIKE 'S%'; |
| *f) Count Female Directors*  SELECT count(\*) AS Count\_of\_female\_directors  FROM project\_movie\_database.directors  WHERE gender=1; |
| *g) Find the name of 10th First women director?*  SELECT `name` AS female\_director  FROM project\_movie\_database.directors  WHERE gender=1 ORDER BY gender  DESC LIMIT 1 OFFSET 9; |
| *h) What are the 3 most popular movies?*  SELECT original\_title, popularity  FROM project\_movie\_database.movies  ORDER BY popularity DESC LIMIT 3; |
| *i) What are the 3 most bankable movies?*  SELECT original\_title, revenue  FROM project\_movie\_database.movies  ORDER BY revenue DESC LIMIT 3; |
| *j) What is the most awarded average vote since the January 1st, 2000?*  SELECT title, vote\_average, release\_date  FROM project\_movie\_database.movies  WHERE release\_date>=2000-01-01  ORDER BY vote\_average DESC LIMIT 1; |
| *k) Which movie(s) were directed by Brenda Chapman?*  SELECT m.original\_title, d.`name`  FROM movies m JOIN directors d  ON m.director\_id=d.id  WHERE d.`name` = "Brenda Chapman"; |
| *l) Which director made the most movies*  SELECT d.`name`, count(m.id) AS most\_movie\_count  FROM movies m JOIN directors d  ON m.director\_id=d.id GROUP BY d.`name`  ORDER BY most\_movie\_count DESC LIMIT 1; |
| *m) Which director is the most bankable?*  SELECT d.`name`, SUM(m.revenue) AS most\_revenue  FROM movies m JOIN directors d  ON m.director\_id=d.id GROUP BY d.`name`  ORDER BY most\_revenue DESC LIMIT 1; |

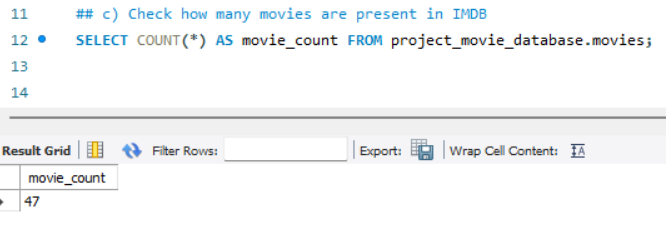
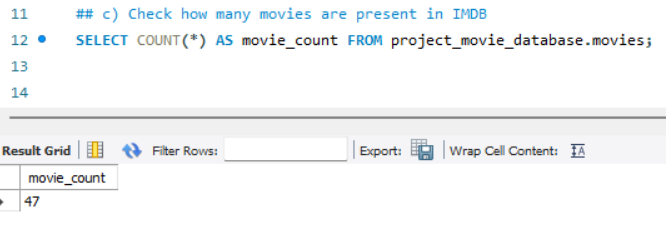
**MySQL Execution with Explanation:**



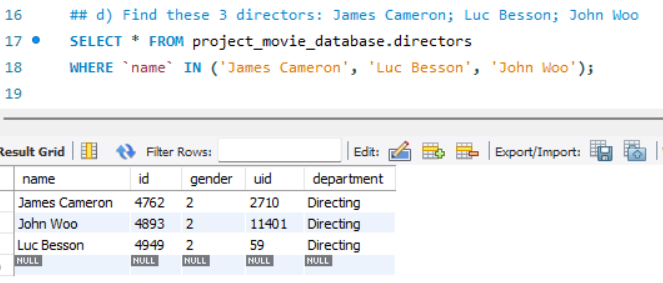
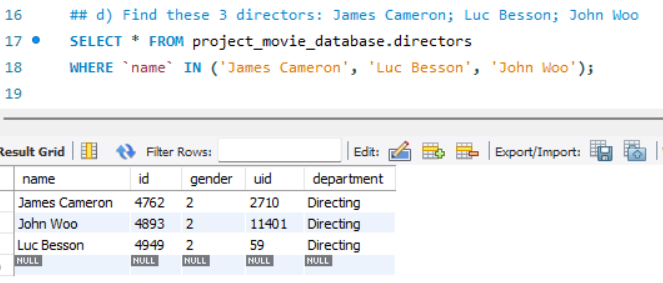
**Statement Explanation:** The above SELECT Statement retrieves all columns and rows from the movies table, providing detailed information about every movie in the dataset as shown in the result.



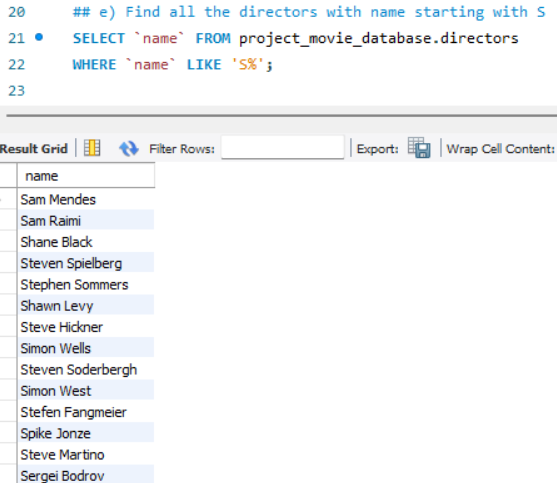
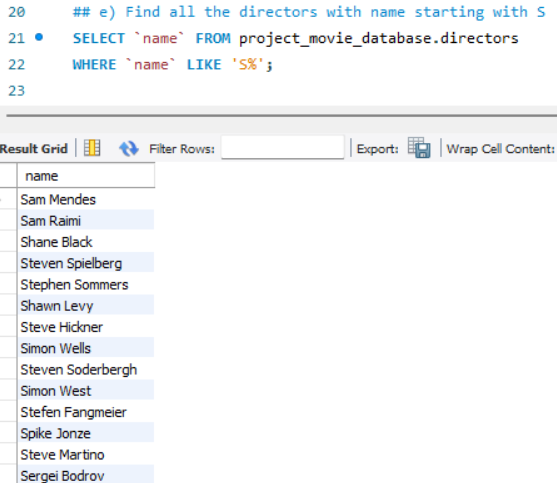
**Statement Explanation:** The above SELECT Statement retrieves all columns and rows from the directors table, providing detailed information about every director in the dataset as shown in the result.



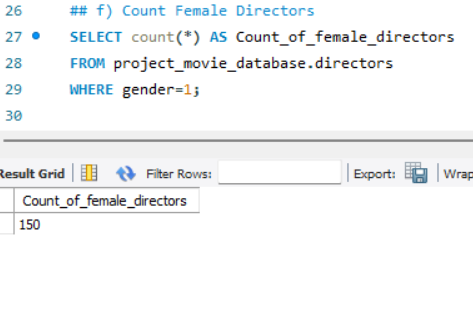
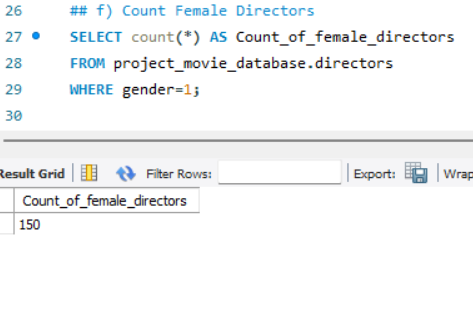
**Statement Explanation:** The above SELECT Statement uses the count function to count all rows in the movies table, returning the total number of movies listed as shown in the result.



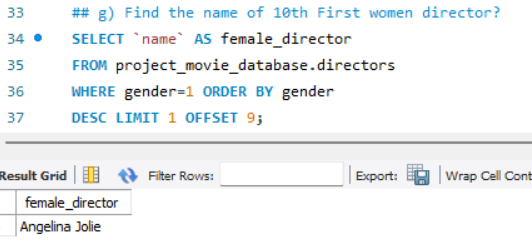
**Statement Explanation:** The above SELECT Statement filters the directors table using the WHERE clause with the IN operator to match any of the three specified names as shown in the result.



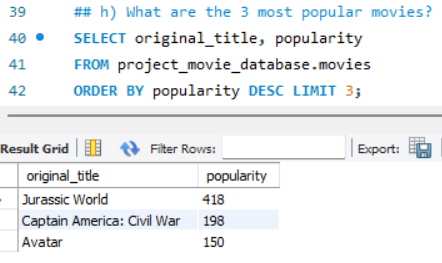
**Statement Explanation:** The above SELECT Statement uses the LIKE operator with the wildcard % to find directors whose names start with the letter "S" as shown in the result.



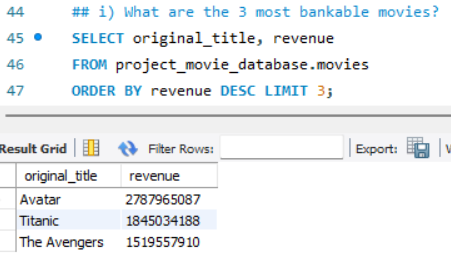
**Statement Explanation:** The above SELECT Statement filters the directors table where the gender column equals 1 (assuming 1 represents females) and counts the matching rows as shown in the result.



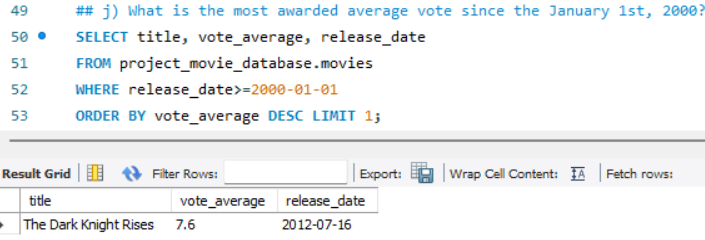
**Statement Explanation:** The above SELECT Statement filters for female directors (gender = 1), orders them by id in ascending order, skips the first 9 rows (OFFSET 9), and retrieves the 10th row (LIMIT 1) as shown in the result.



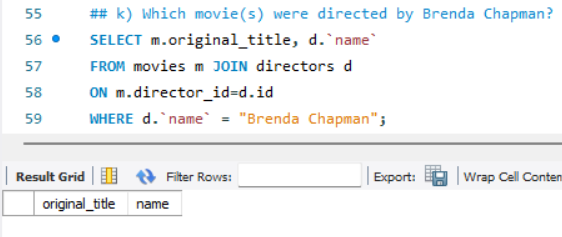
**Statement Explanation:** The above SELECT Statement orders movies by their popularity in descending order and retrieves the top 3 as shown in the result.



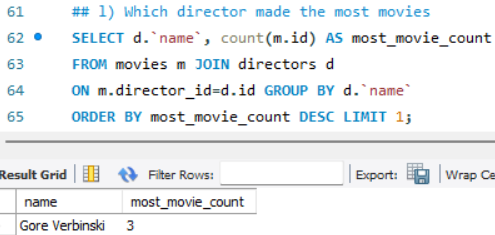
**Statement Explanation:** The above SELECT Statement orders movies by their revenue in descending order and retrieves the top 3 as shown in the result.



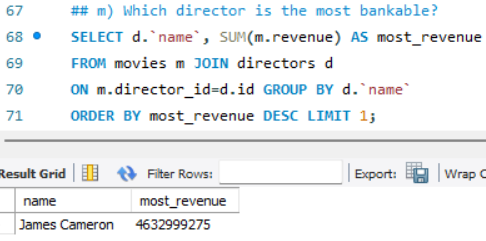
**Statement Explanation:** The above SELECT Statement filters for movies released on or after 2000-01-01, orders them by vote\_average in descending order, and retrieves the top movie as shown in the result.



**Statement Explanation:** The above SELECT Statement performs an INNER JOIN between movies and directors tables on director\_id and id. Filters the result for Brenda Chapman as shown in the result.



**Statement Explanation:** The above SELECT Statement joins movies and directors, groups result by director name, counts movies for each director, and retrieves the one with the highest count as shown in the result.



**Statement Explanation:** The above SELECT Statement joins movies and directors, groups result by director name, sums up the revenues for each director, and retrieves the one with the highest revenue as shown in the result.