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

Team member 2: -

Name: Neetu Sharma

Email: neets.2012@gmail.com

Perform the Problem Queries:

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?

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:

```
3     ## a) How do you get all the data about movies?
4 •   SELECT * FROM project_movie_database.movies;
```

esult Grid | Filter Rows: | Edit: | Export/Import: | Wrap Cell Content:

id	original_title	budget	popularity	release_date	revenue	title
43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar
43598	Pirates of the Caribbean: At World"s End	300000000	139	2007-05-19	961000000	Pirates of the Caribbean: At World"s End
43599	Spectre	245000000	107	2015-10-26	880674609	Spectre
43600	The Dark Knight Rises	250000000	112	2012-07-16	1084939099	The Dark Knight Rises
43601	John Carter	260000000	43	2012-03-07	284139100	John Carter
43602	Spider-Man 3	258000000	115	2007-05-01	890871626	Spider-Man 3
43603	Tangled	260000000	48	2010-11-24	591794936	Tangled
43604	Avengers: Age of Ultron	280000000	134	2015-04-22	1405403694	Avengers: Age of Ultron
43605	Harry Potter and the Half-Blood Prince	250000000	98	2009-07-07	933959197	Harry Potter and the Half-Blood Prince
43607	Superman Returns	270000000	57	2006-06-28	391081192	Superman Returns
43608	Quantum of Solace	200000000	107	2008-10-30	586090727	Quantum of Solace

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.

```
6     ## b) How do you get all the data about directors?
7 •   SELECT * FROM project_movie_database.directors;
```

esult Grid | Filter Rows: | Edit: | Expo

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
Marc Forster	4773	2	12995	Directing
Andrew Adamson	4774	2	5524	Directing
Rob Marshall	4775	2	17633	Directing
Barry Sonnenfeld	4776	2	5174	Directing
Peter Jackson	4777	2	108	Directing
Marc Webb	4778	2	87742	Directing

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.

```
11      ## c) Check how many movies are present in IMDB
12 •    SELECT COUNT(*) AS movie_count FROM project_movie_database.movies;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

movie_count
47

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.

```
16      ## d) Find these 3 directors: James Cameron; Luc Besson; John Woo
17 •    SELECT * FROM project_movie_database.directors
18      WHERE `name` IN ('James Cameron', 'Luc Besson', 'John Woo');
```

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

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.

```
20      ## e) Find all the directors with name starting with S
21 •    SELECT `name` FROM project_movie_database.directors
22      WHERE `name` LIKE 'S%';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name
Sam Mendes
Sam Raimi
Shane Black
Steven Spielberg
Stephen Sommers
Shawn Levy
Steve Hickner
Simon Wells
Steven Soderbergh
Simon West
Stefen Fangmeier
Spike Jonze
Steve Martino
Sergei Bodrov

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.

```
26      ## f) Count Female Directors
27 •    SELECT count(*) AS Count_of_female_directors
28      FROM project_movie_database.directors
29      WHERE gender=1;
```

Result Grid | Filter Rows: | Export: | Wrap

Count_of_female_directors
150

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.

```
33      ## g) Find the name of 10th First women director?
34 •    SELECT `name` AS female_director
35      FROM project_movie_database.directors
36      WHERE gender=1 ORDER BY gender
37      DESC LIMIT 1 OFFSET 9;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Cont

female_director
Angelina Jolie

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.

```
39      ## h) What are the 3 most popular movies?
40 •    SELECT original_title, popularity
41      FROM project_movie_database.movies
42      ORDER BY popularity DESC LIMIT 3;
```

Result Grid | Filter Rows: | Export:

original_title	popularity
Jurassic World	418
Captain America: Civil War	198
Avatar	150

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

```
44      ## i) What are the 3 most bankable movies?
45 •    SELECT original_title, revenue
46      FROM project_movie_database.movies
47      ORDER BY revenue DESC LIMIT 3;
```

Result Grid | Filter Rows: | Export:

original_title	revenue
Avatar	2787965087
Titanic	1845034188
The Avengers	1519557910

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

```
49      ## j) What is the most awarded average vote since the January 1st, 2000?
50 •    SELECT title, vote_average, release_date
51      FROM project_movie_database.movies
52      WHERE release_date>=2000-01-01
53      ORDER BY vote_average DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

title	vote_average	release_date
The Dark Knight Rises	7.6	2012-07-16

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.

```
55      ## k) Which movie(s) were directed by Brenda Chapman?
56 •    SELECT m.original_title, d.`name`
57      FROM movies m JOIN directors d
58      ON m.director_id=d.id
59      WHERE d.`name` = "Brenda Chapman";
```

Result Grid | Filter Rows: | Export: | Wrap Cell Conten

original_title	name

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.

```
61      ## l) Which director made the most movies
62 •    SELECT d.`name`, count(m.id) AS most_movie_count
63      FROM movies m JOIN directors d
64      ON m.director_id=d.id GROUP BY d.`name`
65      ORDER BY most_movie_count DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap Ce

name	most_movie_count
Gore Verbinski	3

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.

```
67      ## m) Which director is the most bankable?
68 •    SELECT d.`name`, SUM(m.revenue) AS most_revenue
69      FROM movies m JOIN directors d
70      ON m.director_id=d.id GROUP BY d.`name`
71      ORDER BY most_revenue DESC LIMIT 1;
```

Result Grid | Filter Rows: | Export: | Wrap C

name	most_revenue
James Cameron	4632999275

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.