



# SQL PROJECT ON IMDB MOVIES

START SLIDE





# HELLO!

My name is Mallikarjun Kulali. In this project, I used SQL to perform detailed analysis on IMDB movies data and answer a range of analytical questions.



# INTRODUCTION

- Analyzed IMDB movies dataset
- Used SQL to answer business and analytical questions
- Focused on movies, directors, revenue, popularity, and ratings
- Demonstrates SQL querying, analysis, and interpretation skills





# DATASET DESCRIPTION

## 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



# A) CAN YOU GET ALL DATA ABOUT MOVIES?

```
SELECT *  
FROM movies;
```

Result Grid   Filter Rows   Edit   Export/Import   Wrap Cell Content									
id	original_title	budget	popularity	release_date	revenue	title	vote_average	vote_count	overview
43597	Avatar	237000000	150	2009-12-10	2787965087	Avatar	7.2	11800	In the 22nd century, a paraplegic Marine is disp..
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..
43599	Spectre	245000000	107	2015-10-26	880674609	Spectre	6.3	4466	A cryptic message from Bonds past sends him o..
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 .
43601	John Carter	260000000	43	2012-03-07	284139100	John Carter	6.1	2124	John Carter is a war-weary, former military cap..
43602	Spider-Man 3	258000000	115	2007-05-01	890871626	Spider-Man 3	5.9	3576	The seemingly invincible Spider-Man goes up ag..
43603	Tangled	260000000	48	2010-11-24	591794936	Tangled	7.4	3330	When the kingdom's most wanted-and most ch..
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..
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 ..
43607	Superman Returns	270000000	57	2006-06-28	391081192	Superman Returns	5.4	1400	Superman returns to discover his 5-year absenc..
43608	Quantum of Solace	200000000	107	2008-10-30	586090727	Quantum of Solace	6.1	2965	Quantum of Solace continues the adventures of.



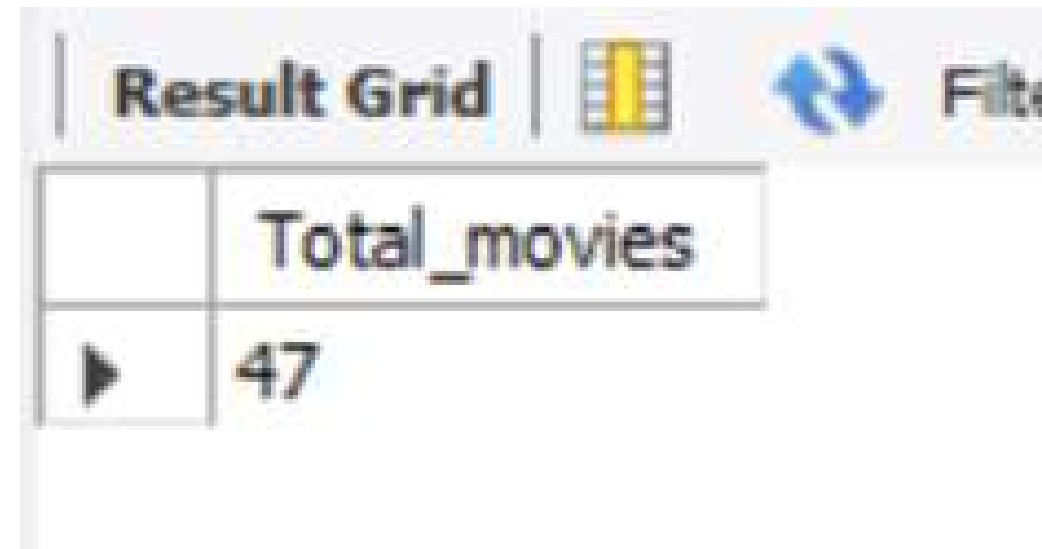
## B) HOW DO YOU GET ALL DATA ABOUT DIRECTORS?

```
SELECT *  
FROM directors;
```

Result Grid					
Filter Rows:					
	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

## C) CHECK HOW MANY MOVIES ARE PRESENT IN IMDB.

```
SELECT COUNT(*) AS Total_movies  
FROM movies;
```



The screenshot shows a database interface with a 'Result Grid' tab. The grid contains one column named 'Total\_movies' and one row with the value '47'. There are icons for refreshing the data and filtering the results.

	Total_movies
▶	47



D) FIND THESE 3 DIRECTORS: JAMES CAMERON ; LUC BESSON ; JOHN WOO

```
SELECT *  
FROM directors  
WHERE Name IN ('James Cameron', 'Luc Besson', 'John Woo');
```

Result Grid					
Filter Rows:					
	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



## E) FIND ALL DIRECTORS WITH NAME STARTING WITH S.

```
SELECT *  
FROM directors  
WHERE Name LIKE 'S%';
```

Result Grid					
Filter Rows:					
	name	id	gender	uid	department
▶	Sam Mendes	4764	2	39	Directing
	Sam Raimi	4767	2	7623	Directing
	Shane Black	4784	2	1108	Directing
	Steven Spielberg	4799	2	488	Directing
	Stephen Sommers	4815	2	7775	Directing
	Shawn Levy	4842	2	17825	Directing
	Steve Hickner	4852	2	44113	Directing
	Simon Wells	4855	2	21879	Directing
	Steven Soderbergh	4909	2	1884	Directing
	Simon West	4930	2	12786	Directing
	Stefen Fangmeier	4931	0	25453	Directing
	Spike Jonze	4932	2	5953	Directing

## F) COUNT FEMALE DIRECTORS.

```
SELECT COUNT(*) AS Female_directors_count  
FROM directors  
WHERE gender = 1;
```

Result Grid		Filter Rows:
	Female_directors_count	
▶	150	



## G) FIND THE NAME OF THE 10TH FIRST WOMEN DIRECTORS?

```
SELECT *  
FROM directors  
WHERE gender = 1  
order by id  
LIMIT 1 OFFSET 9;
```

Result Grid					
	name	id	gender	uid	department
▶	Angelina Jolie	5086	1	11701	Directing
•	NULL	NULL	NULL	NULL	NULL

## H) WHAT ARE THE 3 MOST POPULAR MOVIES?

```
SELECT ID, original_title, title, popularity, release_date
FROM movies
ORDER BY popularity DESC
LIMIT 3;
```

Result Grid	Filter Rows:	Export:	Wrap Cell Content:	Fetch rows
ID	original_title	title	popularity	release_date
43625	Jurassic World	Jurassic World	418	2015-06-09
43623	Captain America: Civil War	Captain America: Civil War	198	2016-04-27
43597	Avatar	Avatar	150	2009-12-10



# I) WHAT ARE THE 3 MOST BANKABLE MOVIES?

```
SELECT ID, original_title, title, revenue, release_date
FROM movies
ORDER BY revenue DESC
LIMIT 3;
```

Result Grid					
	ID	original_title	title	revenue	release_date
▶	43597	Avatar	Avatar	2787965087	2009-12-10
	43622	Titanic	Titanic	1845034188	1997-11-18
	43613	The Avengers	The Avengers	1519557910	2012-04-25

## J) WHAT IS THE MOST AWARDED AVERAGE VOTE SINCE THE JANUARY 1ST, 2000?

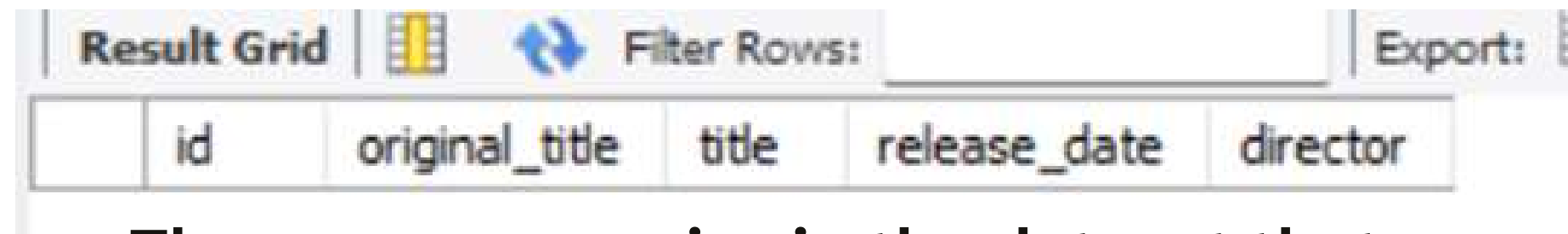
```
SELECT id, original_title, title, vote_average, vote_count, release_date
FROM movies
WHERE release_date >= '2000-01-01'
AND vote_average = (
    SELECT MAX(vote_average)
    FROM movies
    WHERE release_date >= '2000-01-01'
);
```

Result Grid						
Filter Rows: <input type="text"/>						
Edit:   						
Export/Import:  						
Wrap Cell Content: <input type="checkbox"/>						
	id	original_title	title	vote_average	vote_count	release_date
▶	43600	The Dark Knight Rises	The Dark Knight Rises	7.6	9106	2012-07-16
	43619	The Hobbit: The Desolation of Smaug	The Hobbit: The Desolation of Smaug	7.6	4524	2013-12-11
	43639	Toy Story 3	Toy Story 3	7.6	4597	2010-06-16
•	NULL	NULL	NULL	NULL	NULL	NULL



## K) WHICH MOVIE(S) WERE DIRECTED BY BRENDA CHAPMAN?

```
SELECT m.id, m.original_title, m.title, m.release_date, d.name AS director
FROM movies m
JOIN directors d ON m.director_id = d.id
WHERE d.name = 'Brenda Chapman';
```



The screenshot shows a database interface with a 'Result Grid' tab. Above the grid are icons for a table, a refresh button, and a 'Filter Rows:' input field. To the right is an 'Export:' button. The grid has five columns: 'id', 'original\_title', 'title', 'release\_date', and 'director'. The grid is currently empty.

	id	original_title	title	release_date	director
--	----	----------------	-------	--------------	----------

There are no movies in the dataset that were directed by Brenda Chapman.

## L) WHICH DIRECTOR MADE THE MOST MOVIES?

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

Result Grid				Filter Rows:	
	id	name	movie_count		
▶	4763	Gore Verbinski	3		



## M) WHICH DIRECTOR IS THE MOST BANKABLE?

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

Result Grid				Filter Rows:	
	id	name	total_revenue		
▶	4762	James Cameron	4632999275		





# INSIGHTS

- Clear trends in popularity and revenue
- Directors significantly influence movie performance
- Vote averages alone can be misleading without vote counts
- SQL efficiently answers complex analytical questions
- Using LIMIT and ORDER BY allowed ranking top-performing movies efficiently.

# TOOLS

- MySQL / SQL Workbench
- Relational Databases
- Data Analysis Techniques



# CONCLUSION

- SQL helped uncover meaningful insights from movie data
- Demonstrated ability to work with structured datasets
- Project strengthened understanding of SQL, joins, and analytical reasoning





The background of the image is a photograph of a movie theater audience. The audience is seated in rows of red seats, and the scene is dimly lit, typical of a cinema. A large, semi-transparent red rectangle is overlaid on the center of the image, serving as a background for the text. In the top-left and bottom-right corners, there are decorative elements consisting of black and white film strips. The text "THANK YOU" is written in a large, bold, white, sans-serif font, centered within the red overlay.

**THANK  
YOU**