SQL PROJECT

DESCRIPTION: - The **Movies DB** database is designed to store, manage, and analyse data related to movies, including their financials, languages, actors, and other associated information. This structure enables efficient querying and data analysis for insights into the movie industry. Below is a description of each table and its purpose.

TABLES IN THE DATABSE

Movies: - Stores information about each movie, such as title, industry, release year, IMDb rating, studio, language_id, movie_id.

Financials: - Stores information about every movie_id, budget, revenue, unit, currency.

Movie actor: - Stores Information about movie_id, actor_id .

Languages: - Stores Information about language id, Name.

Actors: - Store information about actor id, name, birth year.

QUERIES

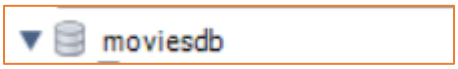
CREATING DATABASE

Question: -Creating Database

INPUT QUERY

```
CREATE DATABASE MoviesDB;
```

OUTPUT



INSIGHTS: - The command creates a new database named MOVIESDB

CREATE TABLE

Question: - Create a table of languages where there are 2 columns name language id and Name and also give appropriate constraints

INPUT QUERY

```
CREATE TABLE languages
( language_id INT auto_increment Primary key,
Name VARCHAR(50));
```

OUTPUT

language_id	Name
1	Hindi
2	Telugu
3	Kannada
4	Tamil
5	English
6	French
7	Bengali
8	Gujrati
NULL	NULL

INSIGHTS: - The query creates the languages table to store unique language identifiers and names, and populates it with common languages to support multilingual data in the database.

CONSTRAINS

Question: -Add a constraint to ensure no movie has an IMDb rating above 10.0.

INPUT QUERY

```
ALTER TABLE movies
ADD CONSTRAINT chk_imdb_rating CHECK (imdb_rating <= 10);
```

OUTPUT

```
3  20:00:27  ALTER TABLE movies ADD CONSTRAINT chk_imdb_rating CHECK (imdb_rating <= 10)   37 row(s) affected Records: 37  Duplicates: 0  Warnings: 0
```

DISABLE SAFE MODE AND TRIM

Question: - Remove Leading spaces in the studio's column of movies table.

INPUT QUERY

DISABLE SAFE MODE

```
SET SQL_SAFE_UPDATES=0;
```

TRIM

```
UPDATE movies
SET studio = TRIM(studio);
```

OUTPUT

DISABLE SAFE MODE

4	20:10:01	SET SQL_SAFE_UPDATES=0	0 row(s) affected

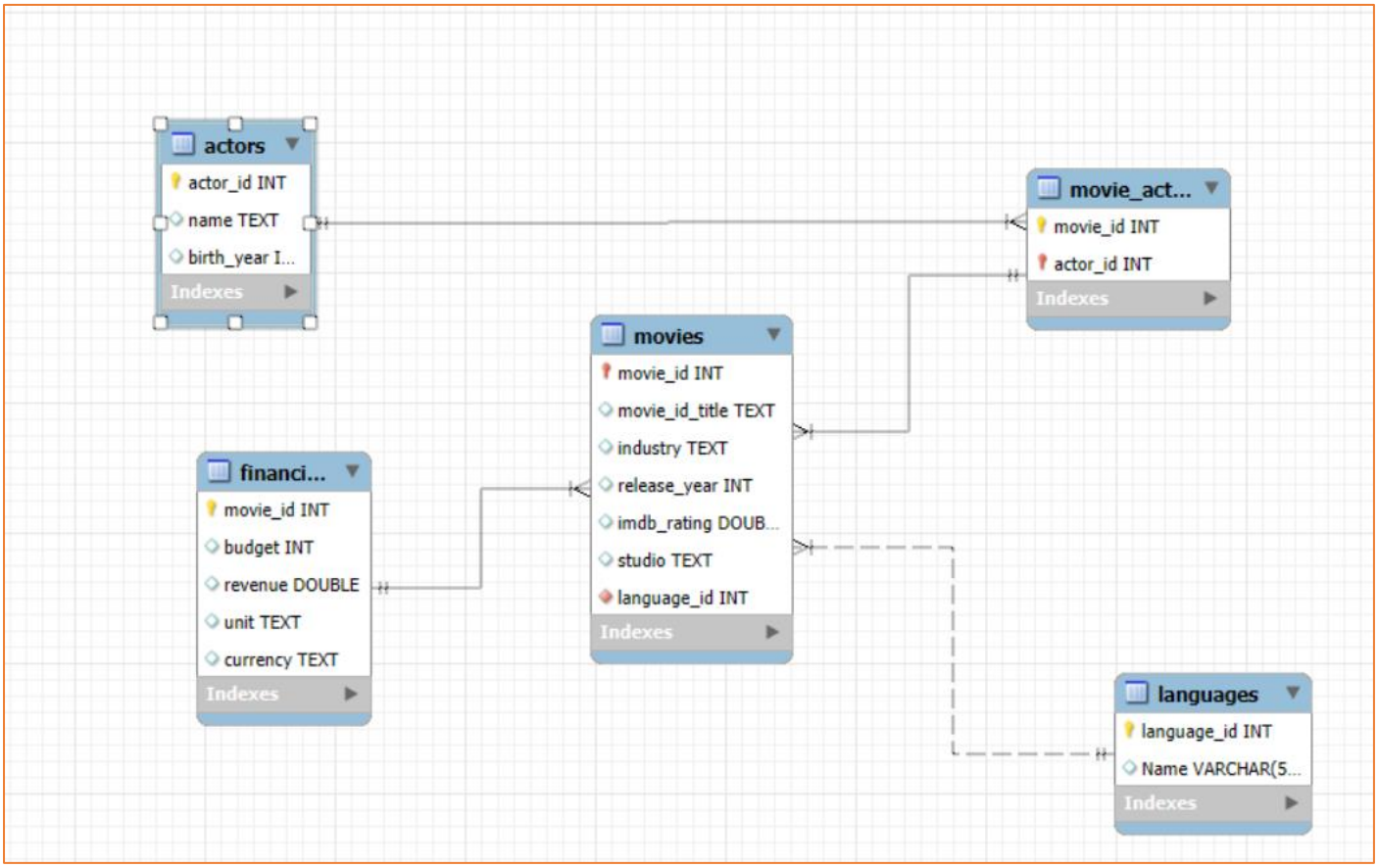
TRIM

movie_id	movie_id_title	industry	release_year	imdb_rating	studio	language_id
101	K.G.F: Chapter 2	Bollywood	2022	8.4	Hombale Films	3
102	Doctor Strange in the Multiverse of Mad...	Hollywood	2022	7	Marvel Studios	5
103	Thor: The Dark World	Hollywood	2013	6.8	Marvel Studios	5
104	Thor: Ragnarok	Hollywood	2017	7.9	Marvel Studios	5
105	Thor: Love and Thunder	Hollywood	2022	6.8	Marvel Studios	5
107	Dilwale Dulhania Le Jayenge	Bollywood	1995	8	Yash Raj Films	1
108	3 Idiots	Bollywood	2009	8.4	Vinod Chopra Films	1
109	Kabhi Khushi Kabhie Gham	Bollywood	2001	7.4	Dharma Productions	1

INSIGHTS: -This query removes unwanted spaces from the studio column in the movies table, ensuring consistent and clean data formatting.

RELATION BETWEEN ALL THE TABLE

Enhanced-Entity-Relationship Diagram (EER)



This diagram represents relationship between all the tables.

WHERE CLAUSE

Question: Retrieve all movies released after 2000 with an IMDb rating above 8.0.

INPUT QUERY

```
SELECT * FROM movies
WHERE release_year > 2000 AND imdb_rating > 8.0;
```

OUTPUT

movie_id	movie_id_title	industry	release_year	imdb_rating	studio	language_id
101	K.G.F: Chapter 2	Bollywood	2022	8.4	Hombale Films	3
108	3 Idiots	Bollywood	2009	8.4	Vinod Chopra Films	1
113	Interstellar	Hollywood	2014	8.6	Warner Bros. Pictures	5
121	The Dark Knight	Hollywood	2008	9	Syncopy	5
124	Parasite	Hollywood	2019	8.5	Not Available	5
125	Avengers: Endgame	Hollywood	2019	8.4	Marvel Studios	5
126	Avengers: Infinity War	Hollywood	2018	8.4	Marvel Studios	5
128	Taare Zameen Par	Bollywood	2007	8.3	Not Available	1
129	Munna Bhai M.B.B.S.	Bollywood	2003	8.1	Vinod Chopra Productions	1
130	PK	Bollywood	2014	8.1	Vinod Chopra Films	1

INSIGHT: Filtering based on year and rating gives insights into high-quality modern releases.

IN LIST

Question: - Retrieve movies that belong to the Studios "Marvel studios", "Hombale Films", "Columbia Pictures".

INPUT QUERY

```
SELECT * FROM movies
WHERE Studio IN ("Marvel studios", "Hombale Films", "Columbia Pictures");
```

OUTPUT

movie_id	movie_id_title	industry	release_year	imdb_rating	studio	language_id
101	K.G.F: Chapter 2	Bollywood	2022	8.4	Hombale Films	3
102	Doctor Strange in the Multiverse of Mad...	Hollywood	2022	7	Marvel Studios	5
103	Thor: The Dark World	Hollywood	2013	6.8	Marvel Studios	5
104	Thor: Ragnarok	Hollywood	2017	7.9	Marvel Studios	5
105	Thor: Love and Thunder	Hollywood	2022	6.8	Marvel Studios	5
115	The Pursuit of Happyness	Hollywood	2006	8	Columbia Pictures	5
125	Avengers: Endgame	Hollywood	2019	8.4	Marvel Studios	5
126	Avengers: Infinity War	Hollywood	2018	8.4	Marvel Studios	5
137	Captain America: The First Avenger	Hollywood	2011	6.9	Marvel Studios	5
138	Captain America: The Winter Soldier	Hollywood	2014	7.8	Marvel Studios	5

INSIGHTS: - Analysing specific Studios helps to focus on certain studio type.

LIKE

Question: Find all movies where the title contains "Star".

INPUT QUERY

```
SELECT * FROM movies
WHERE movie_id_title LIKE "Avenger%";
```

OUTPUT

movie_id	movie_id_title	industry	release_year	imdb_rating	studio	language_id
125	Avengers: Endgame	Hollywood	2019	8.4	Marvel Studios	5
126	Avengers: Infinity War	Hollywood	2018	8.4	Marvel Studios	5

INSIGHTS: - This query can be used to identify popular franchises or recurring titles.

ORDER BY

Question: Display all movies ordered by imdb-rating in descending order.

INPUT QUERY

```
SELECT movie_id_title as Title,imdb_rating as Rating FROM movies
ORDER BY  Rating DESC;
```

OUTPUT

Title	Rating
The Shawshank Redemption	9.3
The Godfather	9.2
Schindler's List	9
The Dark Knight	9
Interstellar	8.6
It's a Wonderful Life	8.6
Parasite	8.5
Gladiator	8.5
3 Idiots	8.4
K.G.F: Chapter 2	8.4

INSIGHTS: - Sorting by imdb-rating helps identify top-grossing movies for analysis.

DISTINCT

Question: List all unique industries in the movies dataset.

INPUT QUERY

```
SELECT DISTINCT industry FROM movies;
```

OUTPUT

	industry
▶	Bollywood
	Hollywood

Insight: Finding unique industries allows for analyses.

LIMIT

Question: Retrieve the 5 movies with the AVG imdb-rating.

INPUT QUERY

```
SELECT movie_id_title as Title,imdb_rating as Rating FROM movies
Limit  5 ;
```

OUTPUT

	Title	Rating
▶	K.G.F: Chapter 2	8.4
	Doctor Strange in the Multiverse of Mad...	7
	Thor: The Dark World	6.8
	Thor: Ragnarok	7.9
	Thor: Love and Thunder	6.8

INSIGHTS: - Identifying high-average movies can help understand us the movies with the highest ratings.

IS NULL / IS NOT NULL

Question: Find all movies where the studio information is missing.

INPUT QUERY

```
SELECT * FROM movies
WHERE imdb_rating IS NULL;
```

OUTPUT

	movie_id	movie_id_title	industry	release_year	imdb_rating	studio	language_id
▶	131	Sanju	Bollywood	2018	NULL	Vinod Chopra Films	1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

INSIGHTS: - Missing imdb-raiting data indicates gaps in information, useful for data cleaning.

CASE

Question: Categorize movies based on IMDb rating: "Excellent" (8+), "Good" (5-7.9), or "Average" (<5).

INPUT QUERY

```
SELECT movie_id_title as Title ,imdb_rating as Rating ,
CASE
        WHEN imdb_rating > 8 THEN "Excellent"
        WHEN imdb_rating BETWEEN 5 AND 7.9 THEN "Good"
        ELSE "Average"
        END as Rating_category
        FROM movies;
```

OUTPUT

Title	Rating	Rating_category
K.G.F: Chapter 2	8.4	Excellent
Doctor Strange in the Multiverse of Mad...	7	Good
Thor: The Dark World	6.8	Good
Thor: Ragnarok	7.9	Good
Thor: Love and Thunder	6.8	Good
Dilwale Dulhania Le Jayenge	8	Average
3 Idiots	8.4	Excellent
Kabhi Khushi Kabhie Gham	7.4	Good
Bajirao Mastani	7.2	Good
The Shawshank Redemption	9.3	Excellent
Interstellar	8.6	Excellent
The Pursuit of Happyness	8	Average
Gladiator	8.5	Excellent

INSIGHTS: - Rating categories allow for qualitative assessment in audience preference.

GROUP BY with HAVING

Question: List studios with an average movie rating above 7.

INPUT QUERY

```
SELECT studio, ROUND(AVG(imdb_rating),1) as AVG_RATING FROM movies
GROUP BY studio
HAVING AVG_RATING > 7;
```

OUTPUT

	studio	AVG_RATING
▶	Hombale Films	8.4
	Marvel Studios	7.5
	Yash Raj Films	8
	Vinod Chopra Films	8.2
	Dharma Productions	7.9
	Not Available	8
	Castle Rock Entertainment	9.3
	Warner Bros. Pictures	8.6
	Columbia Pictures	8
	Universal Pictures	8.6

INSIGHTS: - High-average-rating studios can be seen as consistent producers of quality content.

STRING FUNCTIONS

Question: Convert movie titles to uppercase.

INPUT QUERY

```
SELECT UPPER(industry) AS industry_upper FROM movies;
```

OUTPUT

	industry_upper
▶	BOLLYWOOD
	HOLLYWOOD
	HOLLYWOOD
	HOLLYWOOD
	HOLLYWOOD
	BOLLYWOOD

INSIGHTS: - Consistent formatting simplifies textual analysis and data presentation.

MATH FUNCTION

Question: - Listed some MATH FUNCTION

INPUT QUERY

```
SELECT abs(-1000) as Absolute_result,
floor(5.3) as floor_result,
ceil(5.3) as ceeil_result,
floor(-5.3) as floor_result,
ceil(-5.3) as ceil_result,
power(2,3) as  power_result,
Sqrt(225) as Square_root;
```

OUTPUT

Absolute_result	floor_result	ceeil_result	floor_result	ceil_result	power_result	Square_root
1000	5	6	-6	-5	8	15

INSIGHTS: - Here we have done some maths function for understanding better function regarding to mathematical analysis.

DATE FUNCTION

Question: Calculate the age of actors based on their birth year.

INPUT QUERY

```
SELECT name, (YEAR(CURDATE()) - birth_year) AS age FROM actors;
```

OUTPUT

name	age
Yash	38
Sanjay Dutt	65
Benedict Cumberbatch	48
Elizabeth Olsen	35
Chris Hemsworth	41

INSIGHTS: - Age data helps analyse the actor demographic across movies.

AGGREGATE FUNCTIONS

Question: Find movies with a budget higher than the average budget.

INPUT QUERY

```
SELECT ROUND(SUM(revenue),2) as TOTAL_REVENUE,
ROUND(AVG(budget),2) as AVG_REVENUE,
MAX(revenue) as MAX_REVENUE,
MIN(revenue) as MIN_REVENUE
FROM financials;
```

OUTPUT

TOTAL_REVENUE	AVG_REVENUE	MAX_REVENUE	MIN_REVENUE
52434	195.00	11690	3.1

INSIGHTS: - It gives us a quick performance metrics to analyse overall trends and budget efficiency.

JOIN

Question: - Retrieve each movie with its language name.

INPUT QUERY

```
SELECT m.movie_id_title, l.name AS language
FROM movies m
JOIN languages l
ON m.language_id = l.language_id;
```

OUTPUT

Title	Rating	Rating_category
K.G.F: Chapter 2	8.4	Excellent
Doctor Strange in the Multiverse of Mad...	7	Good
Thor: The Dark World	6.8	Good
Thor: Ragnarok	7.9	Good
Thor: Love and Thunder	6.8	Good
Dilwale Dulhania Le Jayenge	8	Average

INSIGHTS: - Combining movies and language data helps analyse multilingual trends.

SUBQUERY

Question: Find actors who have appeared in movies with a revenue above the average.

INPUT QUERY: -

```
SELECT Name
FROM actors  as a
JOIN movie_actor as ma
ON a.actor_id=ma.actor_id
JOIN financials as f
ON ma.movie_id=f.movie_id
WHERE f.revenue > (SELECT (AVG(revenue)) from financials);
```

OUTPUT

Name
Shah Rukh Khan
Kajol
Aamir Khan
R. Madhavan
Sharman Joshi
Leonardo DiCaprio
Kate Winslet
Sam Worthington
Zoe Saldana

INSIGHTS: - Identifying high-grossing actors helps target popular talents.

VIEWS

Question: - Create a view for all movies with budgets above $100 million.

INPUT QUERY

```
CREATE VIEW high_revenue_movies AS
SELECT m.movie_id_title as TITLE, f.revenue AS REVENUE
FROM movies m
JOIN financials f ON m.movie_id = f.movie_id
WHERE f.revenue > 100;
SELECT * FROM high_revenue_movies;
```

OUTPUT

movie_id_title	revenue
Doctor Strange in the Multiverse of Mad...	954.8
Thor: The Dark World	644.8
Thor: Ragnarok	854
Thor: Love and Thunder	670
Dilwale Dulhania Le Jayenge	2000
3 Idiots	4000
Kabhi Khushi Kabhie Gham	1360
Interstellar	701.8
The Pursuit of Happyness	307.1
Gladiator	460.5
Titanic	2202
Avatar	2847
The Godfather	291
The Dark Knight	1006

INSIGHTS: - A view helps quickly access and analyse movies with substantial financial backing.