SQL Portfolio Project: IMDB Movie Analysis

Project Overview

This project analyzes the IMDB movie dataset using SQL to extract meaningful insights about movies, genres, directors, actors, production houses, and viewer ratings. The goal is to demonstrate SQL skills such as joins, aggregations, window functions, CTEs, ranking functions, and data cleaning.

Objectives

- Understand dataset structure and identify missing values.

- Analyze movie trends year-wise and month-wise.

- Identify top genres, directors, actors, and production houses.

- Calculate average durations, rankings, and weighted averages.

- Perform advanced SQL operations using CTEs and window functions.

Dataset Description

The dataset includes 6 major tables:

1. movie – movie details such as title, year, duration, country, income.

2. genre – movie genres.

3. ratings – rating details including avg_rating, median_rating, total_votes.

4. names – actors, actresses, directors' details.

5. role_mapping – mapping of actors/actresses to movies.

6. director_mapping – mapping of directors to movies.

Key SQL Skills Use

- SELECT, WHERE, GROUP BY, ORDER BY

- DISTINCT, COUNT, SUM, AVG, MIN, MAX

- CASE WHEN (Conditional Logic)

- INNER JOIN / LEFT JOIN / Multiple Joins

- Window Functions: RANK, DENSE_RANK, ROW_NUMBER

- Aggregate Window Functions: SUM OVER(), AVG OVER()

- CTEs (WITH clause)

- Handling NULL values

Important Insights (Summary)

1. Identified total row counts for all tables.

2. Found columns containing NULL values in movie and names tables.

```
SELECT
  COUNT(*) AS movie_total_rows,
  SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS null_id,
  SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS null_title,
  SUM(CASE WHEN year IS NULL THEN 1 ELSE 0 END) AS null_year,
  SUM(CASE WHEN date_published IS NULL THEN 1 ELSE 0 END) AS

null_date_published,
  SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS null_duration,
  SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS null_country,
  SUM(CASE WHEN worlwide_gross_income IS NULL THEN 1 ELSE 0 END) AS
null_worlwide_gross_income,
  SUM(CASE WHEN languages IS NULL THEN 1 ELSE 0 END) AS null_languages,
  SUM(CASE WHEN production_company IS NULL THEN 1 ELSE 0 END) AS
null_production_company
FROM movie;
```

movie_total_rows	null_id	null_title	null_year	null_date_published	null_duration	null_country	null_worlwide_gross_income	null_languages	null_production_company
7997	0	0	0	0	0	20	3724	194	528

3. Analyzed year-wise and month-wise movie trends.

4. USA and India produced highest movies in 2019.

```
SELECT
        COUNT(*) AS num_of_movies
FROM movie
WHERE year = 2019
        AND(LOWER(country) LIKE '%USA%' OR LOWER(country) LIKE '%India%');
```

num_of_movies
1059

5. Drama genre had highest number of movies.

6. Calculated movies with only one genre.

7. Computed average duration per genre.

8. Ranked genres based on movie count.

9. Found top 10 movies based on avg_rating.

10. Identified production houses with highest hit movies.

11. Ranked directors, actors, and actresses based on weighted ratings.

```
WITH actress_ratings AS
(
SELECT
        n.name as actress_name,
   SUM(r.total_votes) AS total_votes,
   COUNT(m.id) as movie_count,
        ROUND(
                SUM(r.avg_rating*r.total_votes)
    /
                SUM(r.total_votes)
                        ,2) AS actress_avg_rating
FROM
        names AS n
INNER JOIN
        role_mapping rm ON n.id=rm.name_id
INNER JOIN
        movie m ON rm.movie_id = m.id
INNER JOIN
        ratings r ON m.id=r.movie_id
WHERE category = 'actress' AND LOWER(languages) like '%hindi%'
GROUP BY actress_name
)
SELECT *,
        ROW_NUMBER() OVER (ORDER BY actress_avg_rating DESC, total_votes DESC) AS
actress_rank
FROM
        actress_ratings
WHERE movie_count>=3
LIMIT 5;
```

actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Taapsee Pannu	18061	3	7.74	1
Kriti Sanon	21967	3	7.05	2
Divya Dutta	8579	3	6.88	3
Shraddha Kapoor	26779	3	6.63	4
Kriti Kharbanda	2549	3	4.80	5

12. Classified thriller movies using rating categories.

13. Calculated running total and moving average of durations.

14. Extracted top-grossing movies per year and genre.

15. Identified multilingual hit production houses.

Project Conclusion

This SQL project demonstrates advanced querying and analytical capabilities using a real-world dataset. Insights generated can help movie production companies like RSVP Movies make data-driven decisions regarding actors, directors, genres, and partnerships.

Project Summary (SQL – IMDB Movie Analysis)

This project demonstrates my strong ability to write complex SQL queries and extract meaningful insights purely through SQL. By applying advanced techniques such as joins, window functions, CTEs, ranking functions, aggregations, and conditional logic, I was able to transform raw IMDB movie data into clear, actionable insights.

Although the project was completed entirely using SQL (without Power BI or Python), it still showcases my ability to think like a business analyst by interpreting meaningful patterns such as genre popularity, year-wise movie trends, hit movie analysis, top-performing actors/directors, and production house performance.

I completed this project as part of my Upgrade learning program, where the dataset and problem statements were provided, but all SQL query writing, analysis, and insight generation were done by me with guided support. This experience strengthened my analytical thinking and helped me apply SQL to solve real-world business questions.