# IMDB Dataset Analysis using Advanced SQL

## Approach

The approach to analyzing the IMDb dataset involves several key SQL operations:

- Understanding the dataset structure and relationships between tables.

- Identifying missing data and handling NULL values where necessary.

- Using aggregate functions and grouping to derive meaningful statistics.

- Applying filters and conditions to extract specific insights.

- Sorting and ranking movies based on different parameters such as ratings and votes.

- Joining multiple tables to correlate information from different sources.

- Comparing trends over time to identify changes in the film industry.

## Detailed SQL Queries and Explanations

### Count of Records in Each Table

Objective: Counts the number of records in each table to understand dataset size.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Identifying NULL Values in the Movie Table

Objective: Checks for columns in the `movie` table that contain NULL values.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Movies Released per Year and Month-wise Trends

Objective: Groups movies by year and month to analyze trends.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Movies Produced in USA or India in 2019

Objective: Filters movies released in 2019 in either the USA or India.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Unique Genres and Their Movie Counts

Objective: Lists unique genres and the count of movies belonging exclusively to each.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Most Popular Genre by Movie Count

Objective: Identifies the genre with the highest number of movies.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Average Movie Duration per Genre

Objective: Calculates average duration of movies per genre.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Actors in More than Three Low-rated Movies

Objective: Identifies actors appearing in more than three movies with an average rating below 5.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Minimum and Maximum Values in Ratings Table

Objective: Retrieves min and max values for each numeric column in the `ratings` table (excluding movie\_id).

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Top 10 Movies by Average Rating

Objective: Ranks the top 10 movies based on average rating.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in trend analysis, ranking, or comparison.

### Grouping Movies by Median Ratings

Objective: Summarizes the `ratings` table by grouping movies based on median rating.

SQL Logic: This query typically involves using aggregation, filtering, or joining.

Expected Result: The query returns a result set containing aggregated data per category.

Use Case: This helps in filtering and identifying specific data points.

### Movies Released in March 2017 in the USA with Over 1000 Votes

Objective: Filters movies meeting the specific conditions.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Highly-rated Movies Starting with 'The'

Objective: Finds movies from each genre that start with 'The' and have an average rating greater than 8.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Movies Released Between April 1, 2018 - April 1, 2019 with Median Rating 8

Objective: Counts movies released in this timeframe with a median rating of 8.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Comparing Average Votes for German and Italian Movies

Objective: Compares the average votes received by German and Italian movies.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Identifying NULL Values in the Names Table

Objective: Checks for columns with NULL values in the `names` table.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Top Two Actors with Highly-rated Movies

Objective: Identifies the top two actors with movies having a median rating of 8 or higher.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in trend analysis, ranking, or comparison.

### Top Three Production Companies by Total Votes

Objective: Find the top three production companies based on total votes received by their movies.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in trend analysis, ranking, or comparison.

### Directors Who Worked on More than Three Movies

Objective: Identifies directors associated with more than three movies.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Average Height of Actors and Actresses

Objective: Calculates average height separately for male and female actors.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### 10 Oldest Movies with Director and Country

Objective: Retrieves the 10 oldest movies along with their country and director.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Top 5 Movies by Total Votes with Genres

Objective: Find the top 5 movies with the highest votes along with their genres.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in trend analysis, ranking, or comparison.

### Longest Duration Movie with Genre and Production Company

Objective: Identifies the movie with the longest duration along with genre and production company.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Total Votes for Movies Released in 2018

Objective: Computes total votes for movies released in 2018.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

### Most Common Language in Movie Production

Objective: Determines the most frequently used language in the dataset.

SQL Logic: This query typically involves using basic selection and filtering.

Expected Result: The query returns a result set containing filtered records matching the criteria.

Use Case: This helps in filtering and identifying specific data points.

## Conclusion

This enhanced SQL analysis provides a structured approach to extracting insights from the IMDb dataset. By using advanced SQL techniques, we can uncover hidden patterns, trends, and key performance indicators in the film industry.