**Project 1: IMDb Movies Analysis using SQL**

Bolly Movies, an Indian film production company, has a successful track record of producing numerous blockbuster films. While primarily catering to the Indian audience, they have decided to venture into the global market with their upcoming project scheduled for release in 2022.

Objective:

Recognizing the value of data-driven decision-making, Bolly Movies has enlisted your expertise as a data analyst and SQL specialist. The objective of this case study is to analyse the movie dataset using SQL queries and extract valuable insights to guide Bolly Movies in planning their new project. The analysis will cover various aspects such as table exploration, movie release trends, production statistics, genre popularity, ratings analysis, crew members, and more.

Segment 1: Database - Tables, Columns, Relationships

* **What are the different tables in the database and how are they connected to each other in the database?**
* Ans- movie,rating,genre,names,roll\_maping,director\_maping,
* Movies table is connected to genre table by the column of movie.id = genre.movie\_id ,
* Ratings table is connected to movies table by the column of ratings.movie\_id = movies.id,
* Director\_mapping table is connected to names table by the column of director\_mapping.name\_id = names.id,
* Role\_mapping table is connected to names table by the column of role\_mapping.name\_id = names.id,
* Role\_mapping table is connected to movies table by the column of movies.id = role\_mapping.movie\_id,
* Director\_mapping table is connected to movies table by the column of movies.id = director\_mapping.movie\_id
* **Find the total number of rows in each table of the schema.**
* Ans- select count(\*) from movies; 7997
* select count(\*) from genres1; 14662
* select count(\*) from name; 8400
* select count(\*) from direct\_mapping; 3867
* select count(\*) from rating; 7997
* select count(\*) from role\_map; 15615
* **Identify which columns in the movie table have null values.**
* Ans-production\_company,country, worlwide\_gross\_income, languages columns have null values.

Segment 2: Movie Release Trends

* **Determine the total number of movies released each year and analyse the month-wise trend.**

|  |  |
| --- | --- |
|  |  |
|  |  |
| select year,count(title) as movies,month(str\_to\_date(date\_published,'%d/%m/%Y')) as month from movies  group by year,month(str\_to\_date(date\_published,'%d/%m/%Y'))  order by 1,3 |  |

* **Calculate the number of movies produced in the USA or India in the year 2019?**
* Ans -select count(id) from movies where (country = 'USA' OR country = 'India') and year = 2019; -887

Segment 3: Production Statistics and Genre Analysis

* **Retrieve the unique list of genres present in the dataset.**
* select distinct(genre) from genres1;
* **Identify the genre with the highest number of movies produced overall.**
* select genre,count(movie\_id) from genres1
* group by genre order by count(movie\_id) desc limit 1;

|  |  |
| --- | --- |
| Drama | 4285 |

* **Determine the count of movies that belong to only one genre.**

select count(\*) from (

select movie\_id,count(genre) from genres1

group by movie\_id having count(genre) = 1) as a; 3289

* **Calculate the average duration of movies in each genre.**
* select genre,avg(duration) from movies M
* inner join genres1 G
* on M.id = G.movie\_id group by genre;
* **Find the rank of the 'thriller' genre among all genres in terms of the number of movies produced.**
* select genre\_rank from (
* select genre,movies,rank() over(order by movies desc) as genre\_rank from (
* select genre,count(movie\_id) as movies from genres1
* group by genre) A) B where genre = 'Thriller';

Segment 4: Ratings Analysis and Crew Members

* **Retrieve the minimum and maximum values in each column of the ratings table (except movie\_id).**
* select

min(avg\_rating) as min\_avg\_rating,

* max(avg\_rating) as max\_avg\_rating,
* min(total\_votes) as min\_total\_votes,
* max(total\_votes) as max\_total\_votes,
* min(median\_rating) as min\_median\_rating,
* max(median\_rating) as max\_median\_rating
* from
* ratings;
* **Identify the top 10 movies based on average rating.**
* select title,avg\_rating from movies M left join ratings R
* on M.id = R.movie\_id
* order by avg\_rating desc
* limit 10;

**or**

select \* from

(select title,avg\_rating,

rank() over(order by avg\_rating desc) as rnk

from movies M left join ratings R

on M.id = R.movie\_id

order by avg\_rating desc) t

where rnk < 11;

* **Summarise the ratings table based on movie counts by median ratings.**
* select median\_rating,count(movie\_id) from ratings group by median\_rating order by median\_ rating;
* **Identify the production house that has produced the most number of hit movies (average rating > 8)**.
* select production\_company,count(movie\_id) from ratings R inner join movies M
* on R.movie\_id = M.id where avg\_rating > 8 and (production\_company is not null
* and production\_company != '') group by production\_company
* order by count(movie\_id) desc limit 1 ;

**or**

select \* from (

select production\_company,count(movie\_id),

rank() over(order by count(movie\_id) desc) as cnt

from ratings R inner join movies M

on R.movie\_id = M.id where avg\_rating > 8 and (production\_company is not null

and production\_company != '') group by production\_company

order by count(movie\_id) desc limit 1 ) t

where cnt = 1;

* **Determine the number of movies released in each genre during March 2017 in the USA with more than 1,000 votes.**
* select genre, count(id) as no\_of\_movies from movies M inner join ratings R on M.id = R.movie\_id inner join genre G
* on M.id = G.movie\_id where country = 'USA' and total\_votes > 1000 and date\_published like '%03/2017'
* group by genre order by no\_of\_movies desc;
* **OR**

select genre,count(G. movie\_id) as no\_of\_movies from genre G

inner join movies M on M.id=G.movie\_id

inner join ratings R on R.movie\_id=G.movie\_id

where year = 2017 and

month(date\_published) = 3 and

lower(country) like "%USA%" and

total\_votes > 1000

group by genre

order by no\_of\_movies desc;

* **Retrieve movies of each genre starting with the word 'The' and having an average rating > 8.**
* select title,genre from movies M inner join ratings R on M.id = R.movie\_id inner join genre G
* on M.id = G.movie\_id where title like 'The %'and avg\_rating > 8 ;

**OR**

select genre,title,avg\_rating as no\_of\_movies from genre G

inner join movies M on M.id=G.movie\_id

inner join ratings R on R.movie\_id=G.movie\_id

where lower(title) like "the%" and

avg\_rating > 8;

Segment 5: Crew Analysis

* **Identify the columns in the names table that have null values.**
* select \* from names where known\_for\_movies is null or known\_for\_movies = '';
* select \* from names where date\_of\_birth is null or date\_of\_birth = '';
* select \* from names where height is null or height = '';
* select \* from names where name is null or name = '';
* select \* from names where id is null or id = '';

**OR**

select count(\*) from names where known\_for\_movies is null or known\_for\_movies = '';

select count(\*) from names where date\_of\_birth is null or date\_of\_birth = '';

select count(\*) from names where height is null or height = '';

select count(\*) from names where name is null or name = '';

select count(\*) from names where id is null or id = '';

**“Height,date\_of\_birth and known\_for\_movies are the column which has null values.”**

* **Determine the top three directors in the top three genres with movies having an average rating > 8.**

with top\_genre as

(select genre,count(G.movie\_id) as total\_movies from genre G inner join ratings R

on G.movie\_id = R.movie\_id

where avg\_rating > 8

group by genre

order by total\_movies desc limit 3)

select

n.name as top\_director, count(m.id) as movie\_count

from names n inner join director\_mapping dm on dm.name\_id = n.id

inner join monies m on m.id = dm.movies\_id

inner join genre g on g.movie\_id = m.id

inner join ratings r on r.movie\_id = m.id

where avg\_rating > 8 and genre in

(select genre from top\_genre limit 3)

group by 1

order by movie\_count

limit 3;

* **Find the top two actors whose movies have a median rating >= 8.**

select name,count(R.movie\_id)

from role\_maping RM inner join ratings R

on RM.movie\_id = R.movie\_id

inner join names N on N.id = RM.name\_id

where median\_rating >= 8 and category = 'actor'

group by name

order by count(R.movie\_id) desc

limit 2

**OR**

select name,movies\_count,

rank() over(order by movies\_count desc) as rnk from (

select name,count(R.movie\_id) as movies\_count

from role\_maping RM inner join ratings R

on RM.movie\_id = R.movie\_id

inner join names N on N.id = RM.name\_id

where median\_rating >= 8 and category = 'actor'

group by name

) as A

limit 2

* **Identify the top three production houses based on the number of votes received by their movies.**
* select production\_company as production\_house,sum(total\_votes) from movies
* M inner join ratings R
* on M.id=R.movie\_id
* group by production\_company
* order by sum(total\_votes) desc limit 3;

OR

select production\_house,count\_votes,

rank() over(order by count\_votes desc) as votes\_rank from (

select production\_company as production\_house,sum(total\_votes) as count\_votes from movies

M inner join ratings R

on M.id=R.movie\_id

group by production\_company) as A

limit 3;

* **Rank actors based on their average ratings in Indian movies released in India.**

select name, actor\_rating ,

rank() over(order by actor\_rating desc) as avg\_rating\_rank from (

select name,avg(avg\_rating) as actor\_rating

from ratings R inner join movies M

on M.id = R.movie\_id inner join role\_maping RM

on M.id = RM.movie\_id inner join names N on N.id = RM.name\_id

where country = 'India' and category = 'actor'

group by name) A;

* **Identify the top five actresses in Hindi movies released in India based on their average ratings.**

select name, avg(avg\_rating) from names N inner join role\_maping RM

on n.id = RM.name\_id inner join movies M

on M.id = Rm.movie\_id inner join ratings R

on R.movie\_id = M.id

where category = 'actress' and languages = 'Hindi' and country = 'India'

group by name

order by avg(avg\_rating) desc

limit 5;

Segment 6: Broader Understanding of Data

* **Classify thriller movies based on average ratings into different categories.**
* select title,avg\_rating,
* case when avg\_rating > 6 then 'high' else 'low' end as rating\_category
* from movies M inner join ratings R
* on R.movie\_id = M.id inner join genre G
* on G.movie\_id = M.id
* where genre = 'Thriller'
* order by avg\_rating desc;
* **analyse the genre-wise running total and moving average of the average movie duration**.

select genre,avg(duration),sum(duration) from genre G inner join ratings R

on G.movie\_id = R.movie\_id inner join movies M

on M.id = G.movie\_id

group by genre;

* **Identify the five highest-grossing movies of each year that belong to the top three genres**.
* select \* from (
* select title,year,worlwide\_gross\_income,genre,
* rank() over(partition by genre,year order by worlwide\_gross\_income desc) as movie\_rank
* from movies M inner join genre G
* on M.id = G.movie\_id
* where genre in (
* select genre from (
* select genre,count(movie\_id) ,rank() over(order by count(movie\_id) desc) as genre\_rank
* from movies M inner join genre G
* on M.id = G.movie\_id
* group by genre) as A
* where genre\_rank <= 3)) as B
* where movie\_rank <= 5;
* **Determine the top two production houses that have produced the highest number of hits among multilingual movies.**
* select production\_company as production\_house,count(title) from movies M inner join ratings R
* on M.id = R.movie\_id
* where languages like '%,%' and production\_company != '' and production\_company is not null and
* avg\_rating > 8
* group by production\_company
* order by count(title) desc
* limit 2;
* **Identify the top three actresses based on the number of Super Hit movies (average rating > 8) in the drama genre.**

select name,count(title) from names N inner join role\_maping RM

on N.id = RM.name\_id inner join genre G

on G.movie\_id = RM.movie\_id inner join movies M

on M.id = RM.movie\_id inner join ratings R

on R.movie\_id = M.id

where avg\_rating > 8 and genre = 'drama' and category = 'actress'

group by name

order by count(title) desc

limit 3;

* **Retrieve details for the top nine directors based on the number of movies, including average inter-movie duration, ratings, and more**.

select name as directors,count(title) as movies,

avg(avg\_rating),avg(median\_rating) from names N

inner join director\_maping DM

on N.id = DM.name\_id inner join ratings R

on R.movie\_id = DM.movie\_id inner join movies M

on M.id = R.movie\_id

group by name

order by count(title) desc

limit 9;

Segment 7: Recommendations

* **Based on the analysis, provide recommendations for the types of content Bolly movies should focus on producing.**

select genre,avg(avg\_rating) from genre G inner join ratings R

on G.movie\_id = R.movie\_id inner join movies M

on M.id = G.movie\_id

where country = 'India'

group by genre

order by avg(avg\_rating) desc;

**movies should focus on producing the Crime movies.**