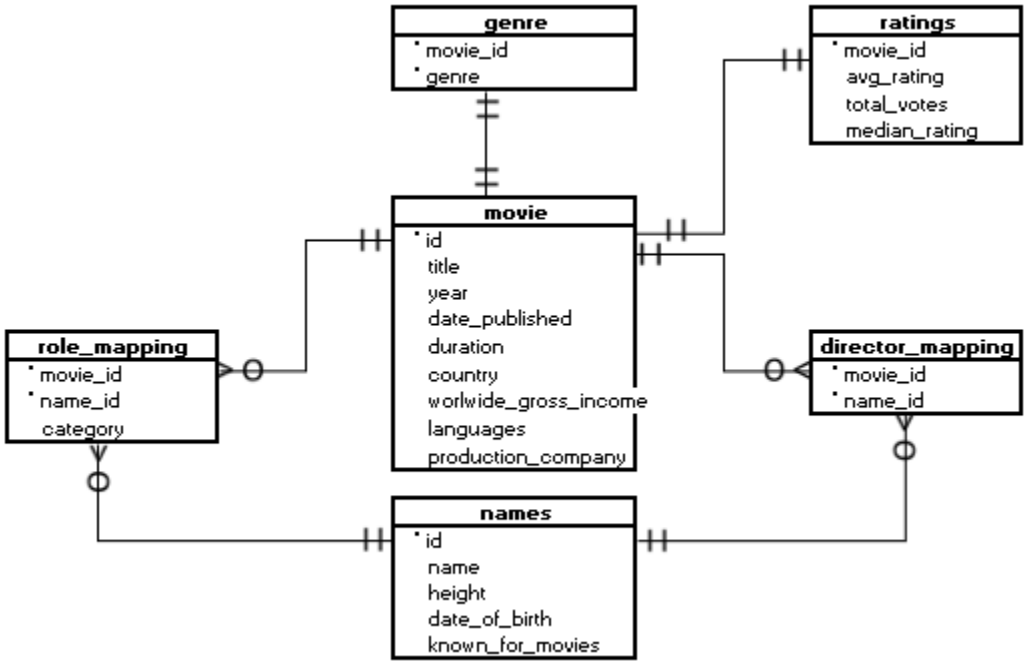
IMDb Movies Analysis using SQL

Segment 1: Database - Tables, Columns, Relationships

Q1. What are the different tables in the database and how are they connected to each other in the database?



Q2. Find the total number of rows in each table of the schema.

```sql
select table_name, table_rows
from information_schema.tables
where table_schema = 'imdb'
```

Q3. Identify which columns in the movie table have null values.

```sql
SELECT
    COUNT(country) AS country_null_count,
    COUNT(date_published) AS date_published_null_count,
    COUNT(duration) AS duration_null_count,
    COUNT(id) AS id_null_count,
    COUNT(languages) AS languages_null_count,
    COUNT(production_company) AS production_company_null_count,
```

```
    COUNT(title) AS title_null_count,
    COUNT(worlwide_gross_income) AS worlwide_gross_income_null_count,
    COUNT(year) AS year_null_count
FROM
    movies
WHERE
    country IS NULL
    OR date_published IS NULL
    OR duration IS NULL
    OR id IS NULL
    OR languages IS NULL
    OR production_company IS NULL
    OR title IS NULL
    OR worlwide_gross_income IS NULL
    OR year IS NULL;
```

Segment 2: Movie Release Trends

Q1. Determine the total number of movies released each year and analyse the month-wise trend.

```
select year,
       count(title) as 'number_of_movies'
from movies
group by year
```

```
select MONTH(STR_TO_DATE(date_published, '%Y-%m-%d')) as 'month',
       count(title) as 'number_of_movies'
from movies
group by 1
order by 1;
```

Q2. Calculate the number of movies produced in the USA or India in the year 2019.

```
select year,
       count(title) as 'number_of_movies'
from movies
```

```sql
where year = 2019 and (country like '%USA%' or country like '%India%')
group by 1
```

Segment 3: Production Statistics and Genre Analysis

Q1. Retrieve the unique list of genres present in the dataset.

```sql
select distinct genre
from genre
```

Q2. Identify the genre with the highest number of movies produced overall.

```sql
select genre,
       count(movies.title) as total_movies
from movies
join genre on movies.id = genre.movie_id
group by 1
order by 2 desc
limit 1
```

Q3. Determine the count of movies that belong to only one genre.

```sql
select count(movie_id)
from (
select movies.id as movie_id,
       count(genre.genre) as genre_count
from movies
join genre on movies.id = genre.movie_id
group by 1
having count(genre.genre) = 1
)as a
```

Q4. Calculate the average duration of movies in each genre.

```sql
select genre,
       avg(duration) as avg_duration
```

```
from movies
join genre on movies.id = genre.movie_id
group by 1
```

Q5. Find the rank of the 'thriller' genre among all genres in terms of the number of movies produced.

```
select *
from (
select genre,
       count(movies.id) as total_movies,
       rank() over(order by count(movies.id) desc) as genre_rank
from movies
join genre on movies.id = genre.movie_id
group by 1
) as a
where genre = 'Thriller'
```

Segment 4: Ratings Analysis and Crew Members

Q1. 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
```

Q2. Identify the top 10 movies based on average rating.

```
select  movies.title,
        ratings.avg_rating,
        rank() over(order by avg_rating desc) as movie_rank
from ratings
```

```
join movies on movies.id = ratings.movie_id
order by avg_rating desc
limit 10
```

Q3. Summarise the ratings table based on movie counts by median ratings.

```
select median_rating,
       count(movie_id) as movie_count
from ratings
group by median_rating
order by 2 desc
```

Q4. Identify the production house that has produced the most number of hit movies (average rating > 8).

```
select  production_company,
        count(id) as movie_count,
        rank() over(order by count(id) desc) as prod_company_rank
from (
    select  production_company,
            movies.id,
            avg_rating
    from movies
    join ratings on movies.id = movie_id
    where avg_rating > 8 and production_company is not null
    order by avg_rating desc
) as a

group by 1
order by 2 desc
limit 1
```

Q5. Determine the number of movies released in each genre during March 2017 in the USA with more than 1,000 votes.

```
select  genre,
        count(movie_id) as movie_count
from (
```

```
    select  genre,
            ratings.movie_id
    from ratings
    join genre on genre.movie_id = ratings.movie_id
    join movies on movies.id = genre.movie_id
    where total_votes > 1000
    and month(STR_TO_DATE(date_published, '%Y-%m-%d')) = 3
    and year(STR_TO_DATE(date_published, '%Y-%m-%d')) = 2017
    and country in ('USA')
) as a
group by 1
order by 2 desc
```

Q6. Retrieve movies of each genre starting with the word 'The' and having an average rating > 8.

```
select  title,
        avg_rating,
        genre
from movies
join ratings on ratings.movie_id = movies.id
join genre on genre.movie_id = movies.id
where avg_rating > 8 and lower(title) like 'the%'
```

Segment 5: Crew Analysis

Q1. Identify the columns in the names table that have null values.

```
select  sum(case when id is null then 1 else 0 end) as id,
        sum(case when name is null then 1 else 0 end) as name,
        sum(case when height is null then 1 else 0 end) as height,
        sum(case when date_of_birth is null then 1 else 0 end) as
date_of_birth,
        sum(case when known_for_movies is null then 1 else 0 end) as
known_for_movies
from names
```

Q2. Determine the top three directors in the top three genres with movies having an average rating > 8.

```
with top_3_genres as (
select genre
from (
select genre, count(movies.id) as total_movies
from ratings
join movies on movies.id = ratings.movie_id
join genre on genre.movie_id = ratings.movie_id
where avg_rating > 8
group by 1
order by 2 desc
limit 3
) a)
select  name as director_name,
        count(movies.id) as total_movies
from ratings
join movies on movies.id = ratings.movie_id
join genre on genre.movie_id = ratings.movie_id
join director_mapping on director_mapping.movie_id = ratings.movie_id
join names on names.id = director_mapping.name_id
where genre in (select * from top_3_genres)
and avg_rating > 8
group by 1
order by 2
limit 3
```

Q3. Find the top two actors whose movies have a median rating >= 8.

```
select  name as actor_name,
        count(ratings.movie_id) as total_movies
from names
join role_mapping on role_mapping.name_id = names.id
join ratings on ratings.movie_id = role_mapping.movie_id
where median_rating > 8 and category = 'actor'
group by 1
order by 2 desc
limit 2
```

Q4. Identify the top three production houses based on the number of votes received by their movies.

```
select  production_company,
        sum(total_votes) as total_votes
from movies
join ratings on ratings.movie_id = movies.id
group by production_company
order by 2 desc
limit 3;
```

Q5. Rank actors based on their average ratings in Indian movies released in India.

```
select  name,
        avg(avg_rating),
        dense_rank() over(order by avg(avg_rating) desc) as actor_rank
from movies
join ratings on ratings.movie_id = movies.id
join role_mapping on role_mapping.movie_id = movies.id
join names on names.id = role_mapping.name_id
where category in ('actor') and country in ('India')
group by 1
order by 2 desc;
```

Q6. Identify the top five actresses in Hindi movies released in India based on their average ratings.

```
select  name,
        avg(avg_rating),
        dense_rank() over(order by avg(avg_rating) desc) as actor_rank
from movies
join ratings on ratings.movie_id = movies.id
join role_mapping on role_mapping.movie_id = movies.id
join names on names.id = role_mapping.name_id
where category in ('actress') and country in ('India') and languages in
('hindi')
group by 1
order by 2 desc
```

Segment 6: Broader Understanding of Data

Q1. Classify thriller movies based on average ratings into different categories.

```
select   title,
         CASE WHEN AVG_RATING > 8 THEN 'Superhit'
         WHEN AVG_RATING BETWEEN 7 AND 8 THEN 'Hit'
         WHEN AVG_RATING BETWEEN 5 AND 7 THEN 'Average'
         WHEN AVG_RATING < 5 THEN 'Below Average'
         END AS 'rating_category'
from movies
join ratings on ratings.movie_id = movies.id
join genre on genre.movie_id = movies.id and genre = 'thriller';
```

Q2. Analyse the genre-wise running total and moving average of the average movie duration.

```
with genreAvgDuration as (
    select  genre,
            avg(duration) as avg_duration
    from movies
    join genre on genre.movie_id = movies.id
    group by 1
)

select  genre,
        avg_duration,
        sum(avg_duration) over(partition by genre order by genre) as
running_total_duration,
        avg(avg_duration) over(partition by genre order by genre) as
moving_avg_duration
from genreAvgDuration
```

Q3. Identify the five highest-grossing movies of each year that belong to the top three genres.

```
with top_3_genre as (
    select genre
```

```
    from genre
    group by genre
    order by count(genre) desc
    limit 3
)


select *
from (
select  genre,
        year,
        title,
        cast(replace(ifnull(worlwide_gross_income, 0), '$', '') as
decimal(10)) as gross_income,
        row_number() over(partition by year order by
cast(replace(ifnull(worlwide_gross_income, 0), '$', '') as decimal(10))
desc) as movie_rank
from movies
join genre on genre.movie_id = movies.id
where genre in (select * from top_3_genre)
) as a
where movie_rank <= 5
```

Q4. Determine the top two production houses that have produced the highest number of hits among multilingual movies.

```
select  production_company,
        count(*) as movie_count,
        dense_rank() over(order by count(*) desc) as
production_company_rank
from movies
join ratings on ratings.movie_id = movies.id
where median_rating >= 8
and production_company != ''
and languages regexp ','
group by 1
limit 2
```

Q5. Identify the top three actresses based on the number of Super Hit movies (average rating > 8) in the drama genre.

```
select  name as actress_name,
        sum(total_votes) as total_votes,
        count(movies.id) as movie_count,
        sum(avg_rating*total_votes)/sum(total_votes) as avg_rating
from genre
join movies on movies.id = genre.movie_id
join ratings on ratings.movie_id = movies.id
join role_mapping on role_mapping.movie_id = movies.id
join names on names.id = role_mapping.name_id
where category = 'actress'
and avg_rating > 8
and genre = 'drama'
group by 1
order by 4 desc, 2 desc
limit 3
```

Q6. Retrieve details for the top nine directors based on the number of movies, including average inter-movie duration, ratings, and more.

```
with director_info as (
select  name as director_name,
        date_published,
        avg_rating,
        total_votes,
        duration,
        lead(date_published, 1) over(partition by name_id order by
date_published) as movie_published_duration
from genre
join movies on movies.id = genre.movie_id
join ratings on ratings.movie_id = movies.id
join director_mapping on director_mapping.movie_id = movies.id
join names on names.id = director_mapping.name_id
)

select  director_name,
        count(*) as total_movies,
```

```
        avg(avg_rating) as avg_rating,
        sum(total_votes) as total_votes,
        sum(avg_rating*total_votes)/sum(total_votes) as
weighted_avg_rating,
        sum(duration),
        avg(movie_published_duration) as avg_time_between_movies
from director_info
group by 1
order by 5 desc, 2 desc, 6 desc, 7 desc
limit 9
```

Segment 7: Recommendations

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

1. Bolly Movies should focus more on genre like ‘Drama’, ‘Comedy’ and ‘Thriller’ as these are the top 3 grossing movie genres as per the analysis.
2. Bolly Movies can use Russo Brothers as directors because according to the analysis they are maintaining high standards in terms of weighted avg rating and total vote count.
3. ‘Dafne Keen’ or ‘Teresa Palmer’ can be considered as the movie actress as they have good global reach according to the votes and avg ratings.
4. ‘Robert Downey Jr.’ or ‘Chris Evans’ can be considered as the movie actors as they have good global reach according to the votes and avg ratings.
5. Also, if Bolly Movies want to team up with some production company to make another blockbuster then 'Twentieth Century Fox' productions.