DATA-DRIVEN SQL ANALYSIS PROJECT FOR RSVP MOVIES

PRESENTER : DR. SAMUEL ISRAEL

PRODUCT MANAGER || BUSINESS ANALYST || DATA ANALYST || CSPO

DATE : 03 - 03 - 2025

```
USE imdb;

-- QUESTION 1 : Find the total number of rows in each table

SELECT 'genre' AS table_name, COUNT(*) AS row_count FROM genre
UNION ALL
SELECT 'movie' AS table_name, COUNT(*) AS row_count FROM movie
UNION ALL
SELECT 'names' AS table_name, COUNT(*) AS row_count FROM names
UNION ALL
SELECT 'ratings' AS table_name, COUNT(*) AS row_count FROM ratings
UNION ALL
SELECT 'director_mapping' AS table_name, COUNT(*) AS row_count FROM director_mapping
UNION ALL
SELECT 'role_mapping' AS table_name, COUNT(*) AS row_count FROM role_mapping;
```

table_name	row_count
genre	14662
movie	7997
names	25735
ratings	7997
director_mapping	3867
role_mapping	15615

```
-- QUESTION 2 : Which columns in the movie table have null values?

SELECT column_name, null_count
FROM (
   SELECT 'country' AS column_name, COUNT(*) AS null_count FROM movie WHERE country IS NULL
   UNION ALL
```

```
   SELECT 'worlwide_gross_income' AS column_name, COUNT(*) AS null_count FROM movie WHERE
worlwide_gross_income IS NULL
   UNION ALL
   SELECT 'languages' AS column_name, COUNT(*) AS null_count FROM movie WHERE languages IS NULL
   UNION ALL
   SELECT 'production_company' AS column_name, COUNT(*) AS null_count FROM movie WHERE
production_company IS NULL
) AS temp_table
WHERE null_count > 0;
```

column_name	null_count
country	20
worlwide_gross_income	3724
languages	194
production_company	528

```
-- QUESTION 3 : Find the total number of movies released each year? How does the trend look month_wise?

-- a: Total number of movies released each year
SELECT
   YEAR(date_published) AS release_year,
   COUNT(*) AS total_movies
FROM movie
GROUP BY YEAR(date_published)  -- Grouping by Year only
ORDER BY release_year;
```

release_year	total_movies
2017	3052
2018	2944
2019	2001

```
-- b : Monthly trend

-- Count the total number of movies released each month
SELECT
   MONTH(date_published) AS release_month,
   COUNT(*) AS total_movies
FROM movie
```

GROUP BY release_month
ORDER BY release_month;

release_month	total_movies
1	804
2	640
3	824
4	680
5	625
6	580
7	493
8	678
9	809
10	801
11	625
12	438

-- QUESTION 4: How many movies were produced in the USA or India in the year 2019?

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

Total_movies
1059

-- QUESTION 5: Find the unique list of the genres present in the data set?

```
SELECT DISTINCT genre AS unique_genres
```

FROM genre;

unique_genres
Drama
Fantasy
Thriller
Comedy
Horror
Family
Romance
Adventure
Action
Sci-Fi
Crime
Mystery
Others

-- QUESTION 6: Which genre had the highest number of movies produced overall?

```
SELECT genre, COUNT(*) AS total_movies
FROM genre
GROUP BY genre
ORDER BY total_movies DESC
LIMIT 1;
```

genre	total_movies
Drama	4285

-- QUESTION 7: How many movies belong to only one genre?

```
SELECT COUNT(movie_id) AS total_movies
```

```
FROM (
    SELECT movie_id
    FROM genre
    GROUP BY movie_id
    HAVING COUNT(genre) = 1
) AS single_genre_movies;
```

total_movies
3289

```
-- QUESTION 8: What is the average duration of movies in each genre?

SELECT g.genre, AVG(m.duration) AS avg_duration
FROM genre g
JOIN movie m ON g.movie_id = m.id
GROUP BY g.genre
ORDER BY avg_duration DESC;
```

genre	avg_duration
Action	112.8829
Romance	109.5342
Crime	107.0517
Drama	106.7746
Fantasy	105.1404
Comedy	102.6227
Adventure	101.8714
Mystery	101.8000
Thriller	101.5761
Family	100.9669
Others	100.1600
Sci-Fi	97.9413
Horror	92.7243

-- QUESTION 9: What is the rank of the ‘thriller’ genre of movies among all the genres in terms of number of movies produced?

```
WITH GenreMovieCount AS (
   SELECT genre, COUNT(movie_id) AS total_movies
   FROM genre
   GROUP BY genre
)
SELECT genre, total_movies,
      RANK() OVER (ORDER BY total_movies DESC) AS genre_rank
FROM GenreMovieCount
WHERE genre = 'Thriller';
```

genre	total_movies	genre_rank
Thriller	1484	1

-- QUESTION 10 : Find the minimum and maximum values in each column of the ratings table except the movie_id column?

```
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;
```

min_avg_rating	max_avg_rating	min_total_votes	max_total_votes	min_median_rating	max_median_rating
1.0	10.0	100	725138	1	10

-- QUESTION 11 : Which are the top 10 movies based on average rating?

```
SELECT m.title, r.avg_rating,
      RANK() OVER (ORDER BY r.avg_rating DESC) AS movie_rank
FROM movie m
JOIN ratings r ON m.id = r.movie_id
ORDER BY r.avg_rating DESC
LIMIT 10;
```

title	avg_rating	movie_rank
Kirket	10.0	1
Love in Kilnerry	10.0	1
Gini Helida Kathe	9.8	3
Runam	9.7	4
Fan	9.6	5
Android Kunjappan Version 5.25	9.6	5
Yeh Suhaagraat Impossible	9.5	7
Safe	9.5	7
The Brighton Miracle	9.5	7
Shibu	9.4	10

-- QUESTION 12 : Summarise the ratings table based on the movie counts by median ratings.

```
SELECT median_rating, COUNT(movie_id) AS movie_count
FROM ratings
GROUP BY median_rating
ORDER BY median_rating;
```

median_rating	movie_count
1	94
2	119
3	283
4	479
5	985
6	1975
7	2257
8	1030

9	429
10	346

-- QUESTION 13 : Which production house has produced the most number of hit movies (average rating > 8)?

```
WITH HitMovies AS (
   SELECT m.production_company, COUNT(m.id) AS movie_count
   FROM movie m
   JOIN ratings r ON m.id = r.movie_id
   WHERE r.avg_rating > 8
   GROUP BY m.production_company
)
SELECT production_company, movie_count,
     RANK() OVER (ORDER BY movie_count DESC) AS prod_company_rank
FROM HitMovies
WHERE production_company IS NOT NULL
ORDER BY movie_count DESC
LIMIT 1;
```

Production_company	movie_count	prod_company_rank
Dream Warrior Pictures	3	1

-- QUESTION 14 : How many movies released in each genre during March 2017 in the USA had more than 1,000 votes?

```
SELECT g.genre, COUNT(DISTINCT m.id) AS movie_count
FROM movie m
JOIN genre g ON m.id = g.movie_id
JOIN ratings r ON m.id = r.movie_id
WHERE m.country LIKE '%USA%'
AND MONTH(m.date_published) = 3
AND YEAR(m.date_published) = 2017
AND r.total_votes > 1000
GROUP BY g.genre
ORDER BY movie_count DESC;
```

genre	movie_count
Drama	24
Comedy	9
Action	8

Thriller	**8**
Sci-Fi	**7**
Crime	**6**
Horror	**6**
Mystery	**4**
Romance	**4**
Adventure	**3**
Fantasy	**3**
Family	**1**

-- QUESTION 15: Find movies of each genre that start with the word 'The' and which have an average rating > 8?

```
SELECT m.title, r.avg_rating, g.genre
FROM movie m
JOIN ratings r ON m.id = r.movie_id
JOIN genre g ON m.id = g.movie_id
WHERE m.title LIKE 'The %'
AND r.avg_rating > 8
ORDER BY r.avg_rating DESC;
```

title	avg_rating	genre
The Brighton Miracle	**9.5**	**Drama**
The Colour of Darkness	**9.1**	**Drama**
The Blue Elephant 2	**8.8**	**Drama**
The Blue Elephant 2	**8.8**	**Horror**
The Blue Elephant 2	**8.8**	**Mystery**
The Irishman	**8.7**	**Crime**
The Irishman	**8.7**	**Drama**
The Mystery of Godliness: The Sequel	**8.5**	**Drama**

The Gambinos	8.4	Crime
The Gambinos	8.4	Drama
The King and I	8.2	Drama
The King and I	8.2	Romance

-- QUESTION 16: Of the movies released between 1 April 2018 and 1 April 2019, how many were given a median rating of 8?

```
-- Count movies released between 1 April 2018 and 1 April 2019 with median rating of 8
SELECT COUNT(m.id) AS movie_count
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE m.date_published BETWEEN '2018-04-01' AND '2019-04-01'
AND r.median_rating = 8;
```

movie_count
361

QUESTION 17 : Do German movies get more votes than Italian movies?

```
SELECT
   'Germany' AS country, SUM(r.total_votes) AS total_votes
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE m.country LIKE '%Germany%'
  UNION ALL
   SELECT
   'Italy' AS country, SUM(r.total_votes) AS total_votes
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE m.country LIKE '%Italy%'
ORDER BY total_votes DESC;
```

country	total_votes
Germany	2026223
Italy	703024

Yes, German movies have more votes than Italian movies.

```
-- QUESTION 18 :  Which columns in the names table have null values?

SELECT column_name, null_count
FROM (
   SELECT 'height' AS column_name, SUM(CASE WHEN height IS NULL THEN 1 ELSE 0 END) AS null_count
FROM names
   UNION ALL
   SELECT 'date_of_birth' AS column_name, SUM(CASE WHEN date_of_birth IS NULL THEN 1 ELSE 0 END)
AS null_count FROM names
   UNION ALL
   SELECT 'known_for_movies' AS column_name, SUM(CASE WHEN known_for_movies IS NULL THEN 1
ELSE 0 END) AS null_count FROM names
) AS temp_table
WHERE null_count > 0;
```

olumn_name	null_count
height	17335
date_of_birth	13431
known_for_movies	15226

```
-- QUESTION 19 :  Who are the top three directors in the top three genres whose movies have an average
rating > 8?

-- Step 1: Identify the top 3 genres with the most movies having avg_rating > 8
WITH TopGenres AS (
   SELECT g.genre, COUNT(m.id) AS movie_count
   FROM genre g
   JOIN movie m ON g.movie_id = m.id
   JOIN ratings r ON m.id = r.movie_id
   WHERE r.avg_rating > 8
   GROUP BY g.genre
   ORDER BY movie_count DESC
   LIMIT 3
),

-- Step 2: Find the top directors within the top 3 genres
TopDirectors AS (
   SELECT dm.name_id AS director_id, n.name AS director_name,
        COUNT(DISTINCT dm.movie_id) AS movie_count
   FROM director_mapping dm
   JOIN names n ON dm.name_id = n.id
   JOIN movie m ON dm.movie_id = m.id
   JOIN ratings r ON m.id = r.movie_id
   JOIN genre g ON m.id = g.movie_id
   JOIN TopGenres tg ON g.genre = tg.genre
   WHERE r.avg_rating > 8
   GROUP BY dm.name_id, n.name
```

```
    ORDER BY movie_count DESC
    LIMIT 3
)

-- Step 3: Display the final result
SELECT director_id, director_name, movie_count
FROM TopDirectors;
```

director_id	director_name	movie_count
nm0751648	Joe Russo	2
nm0003506	James Mangold	2
nm0751577	Anthony Russo	2

```
-- QUESTION 20 : Who are the top two actors whose movies have a median rating >= 8?

WITH TopActors AS (
    SELECT a.name AS actor_name, COUNT(DISTINCT rm.movie_id) AS movie_count
    FROM role_mapping rm
    JOIN names a ON rm.name_id = a.id
    JOIN ratings r ON rm.movie_id = r.movie_id
    WHERE r.median_rating >= 8
    GROUP BY a.name
    ORDER BY movie_count DESC
    LIMIT 2
)

-- Display final result
SELECT actor_name, movie_count
FROM TopActors;
```

actor_name	movie_count
Mammootty	8
Mohanlal	5

```
-- QUESTION 21 : Which are the top three production houses based on the number of votes received by their movies?

WITH ProductionVotes AS (
    SELECT m.production_company, SUM(r.total_votes) AS vote_count
    FROM movie m
    JOIN ratings r ON m.id = r.movie_id
    WHERE m.production_company IS NOT NULL
```

```
    GROUP BY m.production_company
)
SELECT production_company, vote_count,
       RANK() OVER (ORDER BY vote_count DESC) AS prod_comp_rank
FROM ProductionVotes
ORDER BY vote_count DESC
LIMIT 3;
```

production_company	vote_count	prod_comp_rank
Marvel Studios	2656967	1
Twentieth Century Fox	2411163	2
Warner Bros.	2396057	3

```
-- QUESTION 22 : Rank actors with movies released in India based on their average ratings. Which actor is at the top of the list?

-- Rank actors based on weighted average ratings in Indian movies
WITH ActorMovie AS (
    -- Filter Indian movies and count movies per actor
    SELECT n.name AS actor_name,
           COUNT(DISTINCT rm.movie_id) AS movie_count,
           SUM(r.total_votes) AS total_votes,
           SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes) AS actor_avg_rating
    FROM role_mapping rm
    JOIN names n ON rm.name_id = n.id
    JOIN movie m ON rm.movie_id = m.id
    JOIN ratings r ON m.id = r.movie_id
    WHERE m.country LIKE '%India%'
    GROUP BY n.name
    HAVING COUNT(DISTINCT rm.movie_id) >= 5
)

-- Assign ranking based on weighted average rating and total votes (tie-breaker)
SELECT actor_name, total_votes, movie_count, actor_avg_rating,
       RANK() OVER (ORDER BY actor_avg_rating DESC, total_votes DESC) AS actor_rank
FROM ActorMovie
ORDER BY actor_rank
LIMIT 1;
```

actor_name	total_votes	movie_count	actor_avg_rating	actor_rank
Vijay Sethupathi	23114	5	8.41673	1

-- QUESTION 23 : Find out the top five actresses in Hindi movies released in India based on their average ratings?

```
-- Find the top five actresses in Hindi movies released in India based on average ratings
WITH ActressMovie AS (
   SELECT n.name AS actress_name,
        COUNT(DISTINCT rm.movie_id) AS movie_count,
        SUM(r.total_votes) AS total_votes,
        SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes) AS actress_avg_rating
   FROM role_mapping rm
   JOIN names n ON rm.name_id = n.id
   JOIN movie m ON rm.movie_id = m.id
   JOIN ratings r ON m.id = r.movie_id
   WHERE m.country LIKE '%India%'
   AND m.languages LIKE '%Hindi%'
   AND rm.category = 'actress' -- Ensure filtering for actresses
   GROUP BY n.name
   HAVING COUNT(DISTINCT rm.movie_id) >= 3
)

-- Assign ranking based on weighted average rating and total votes (tie-breaker)
SELECT actress_name, total_votes, movie_count, actress_avg_rating,
     RANK() OVER (ORDER BY actress_avg_rating DESC, total_votes DESC) AS actress_rank
FROM ActressMovie
ORDER BY actress_rank
LIMIT 5;
```

actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Taapsee Pannu	18061	3	7.73692	1
Kriti Sanon	21967	3	7.04911	2
Divya Dutta	8579	3	6.88440	3
Shraddha Kapoor	26779	3	6.63024	4
Kriti Kharbanda	2549	3	4.80314	5

-- QUESTION 25 : What is the genre-wise running total and moving average of the average movie duration?

```
WITH GenreDuration AS (
   SELECT g.genre, AVG(m.duration) AS avg_duration
   FROM genre g
   JOIN movie m ON g.movie_id = m.id
   WHERE m.duration IS NOT NULL
   GROUP BY g.genre
)
```

```
SELECT gd.genre,
    gd.avg_duration,
    SUM(gd.avg_duration) OVER (ORDER BY gd.avg_duration) AS running_total_duration,
    AVG(gd.avg_duration) OVER (ORDER BY gd.avg_duration ROWS BETWEEN 2 PRECEDING AND
CURRENT ROW) AS moving_avg_duration
FROM GenreDuration gd
ORDER BY gd.avg_duration DESC;
```

genre	avg_duration	running_total_duration	moving_avg_duration
Action	112.8829	1341.0465	109.82293333
Romance	109.5342	1228.1636	107.78683333
Crime	107.0517	1118.6294	106.32223333
Drama	106.7746	1011.5777	104.84590000
Fantasy	105.1404	904.8031	103.21150000
Comedy	102.6227	799.6627	102.09803333
Adventure	101.8714	697.0400	101.74916667
Mystery	101.8000	595.1686	101.44766667
Thriller	101.5761	493.3686	100.90100000
Family	100.9669	391.7925	99.68940000
Others	100.1600	290.8256	96.94186667
Sci-Fi	97.9413	190.6656	95.33280000
Horror	92.7243	92.7243	92.72430000

```
-- QUESTION 26 :  Which are the five highest-grossing movies of each year that belong to the top three
genres?

-- Step 1: Identify the top 3 genres with the most number of movies
WITH TopGenres AS (
   SELECT g.genre, COUNT(g.movie_id) AS movie_count
   FROM genre g
   GROUP BY g.genre
   ORDER BY movie_count DESC
   LIMIT 3
),
```

```
-- Step 2: Get the five highest-grossing movies for each year within the top 3 genres
RankedMovie AS (
   SELECT g.genre, m.year, m.title AS movie_name,
        CAST(REPLACE(REPLACE(m.worlwide_gross_income, '$', ''), ',', '') AS DECIMAL) AS
worldwide_gross_income,
        RANK() OVER (PARTITION BY g.genre, m.year ORDER BY
CAST(REPLACE(REPLACE(m.worlwide_gross_income, '$', ''), ',', '') AS DECIMAL) DESC) AS movie_rank
   FROM genre g
   JOIN movie m ON g.movie_id = m.id  -- Ensure correct join on movie_id
   JOIN TopGenres tg ON g.genre = tg.genre  -- Only consider top 3 genres
   WHERE m.worlwide_gross_income IS NOT NULL
     AND TRIM(m.worlwide_gross_income) != '' -- Ensure it's not empty
)

-- Step 3: Select the top 5 highest-grossing movies for each year
SELECT genre, year, movie_name, worldwide_gross_income, movie_rank
FROM RankedMovie
WHERE movie_rank <= 5
ORDER BY year DESC, genre, movie_rank;
```

genre	year	movie_name	worldwide_gross_income	movie_rank
Comedy	2019	Toy Story 4	1073168585	1
Comedy	2019	Pokémon Detective Pikachu	431705346	2
Comedy	2019	The Secret Life of Pets 2	429434163	3
Comedy	2019	Once Upon a Time... in Hollywood	371207970	4
Comedy	2019	Shazam!	364571656	5
Drama	2019	Avengers: Endgame	2797800564	1
Drama	2019	The Lion King	1655156910	2
Drama	2019	Joker	995064593	3
Drama	2019	Liu lang di qiu	699760773	4
Drama	2019	It Chapter Two	463326885	5
Thriller	2019	Joker	995064593	1
Thriller	2019	Ne Zha zhi mo tong jiang shi	700547754	2
Thriller	2019	John Wick: Chapter 3 - Parabellum	326667460	3
Thriller	2019	Us	255105930	4

Thriller	2019	Glass	246985576	5
Comedy	2018	Deadpool 2	785046920	1
Comedy	2018	Ant-Man and the Wasp	622674139	2
Comedy	2018	Tang ren jie tan an 2	544061916	3
Comedy	2018	Ralph Breaks the Internet	529323962	4
Comedy	2018	Hotel Transylvania 3: Summer Vacation	528583774	5
Drama	2018	Bohemian Rhapsody	903655259	1
Drama	2018	Hong hai xing dong	579220560	2
Drama	2018	Wo bu shi yao shen	451183391	3
Drama	2018	A Star Is Born	434888866	4
Drama	2018	Fifty Shades Freed	371985018	5
Thriller	2018	Venom	856085151	1
Thriller	2018	Mission: Impossible - Fallout	791115104	2
Thriller	2018	Hong hai xing dong	579220560	3
Thriller	2018	Fifty Shades Freed	371985018	4
Thriller	2018	The Nun	365550119	5
Comedy	2017	Despicable Me 3	1034799409	1
Comedy	2017	Jumanji: Welcome to the Jungle	962102237	2
Comedy	2017	Guardians of the Galaxy Vol. 2	863756051	3
Comedy	2017	Thor: Ragnarok	853977126	4
Comedy	2017	Sing	634151679	5
Drama	2017	Zhan lang II	870325439	1
Drama	2017	Logan	619021436	2
Drama	2017	Dunkirk	526940665	3

Drama	2017	War for the Planet of the Apes	490719763	4
Drama	2017	La La Land	446092357	5
Thriller	2017	The Fate of the Furious	1236005118	1
Thriller	2017	Zhan lang II	870325439	2
Thriller	2017	xXx: Return of Xander Cage	346118277	3
Thriller	2017	Annabelle: Creation	306515884	4
Thriller	2017	Split	278454358	5

-- QUESTION 27 : Which are the top two production houses that have produced the highest number of hits (median rating >= 8) among multilingual movies?

```
-- Step 1: Identify multilingual movies with a median rating >= 8
WITH MultilingualHits AS (
   SELECT m.production_company, COUNT(m.id) AS movie_count
   FROM movie m
   JOIN ratings r ON m.id = r.movie_id
   WHERE r.median_rating >= 8
   AND LENGTH(m.languages) - LENGTH(REPLACE(m.languages, ',', '')) + 1 > 1 -- Identifies multilingual
movies
    AND m.production_company IS NOT NULL
   GROUP BY m.production_company
)

-- Step 2: Rank production houses based on the number of hit multilingual movies
SELECT production_company, movie_count,
      RANK() OVER (ORDER BY movie_count DESC) AS prod_comp_rank
FROM MultilingualHits
ORDER BY prod_comp_rank
LIMIT 2;
```

production_company	movie_count	prod_comp_rank
Star Cinema	7	1
Twentieth Century Fox	4	2

-- QUESTION 28 : Who are the top 3 actresses based on number of Super Hit movies (average rating >8) in the drama genre?

```
-- Step 1: Identify Super Hit movies in the Drama genre
WITH DramaSuperHits AS (
```

```
    SELECT rm.name_id, COUNT(DISTINCT rm.movie_id) AS movie_count,
        SUM(r.total_votes) AS total_votes,
        SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes) AS actress_avg_rating
    FROM role_mapping rm
    JOIN names n ON rm.name_id = n.id
    JOIN movie m ON rm.movie_id = m.id
    JOIN ratings r ON m.id = r.movie_id
    JOIN genre g ON m.id = g.movie_id
    WHERE g.genre = 'Drama'
    AND r.avg_rating > 8
    AND rm.category = 'actress' -- Ensure filtering for actresses
    GROUP BY rm.name_id
)

-- Step 2: Rank actresses based on the number of Super Hit movies and total votes as tie-breaker
SELECT n.name AS actress_name, dsh.total_votes, dsh.movie_count, dsh.actress_avg_rating,
       RANK() OVER (ORDER BY dsh.movie_count DESC, dsh.total_votes DESC) AS actress_rank
FROM DramaSuperHits dsh
JOIN names n ON dsh.name_id = n.id
ORDER BY actress_rank
LIMIT 3;
```

actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Parvathy Thiruvothu	4974	2	8.24813	1
Susan Brown	656	2	8.94436	2
Amanda Lawrence	656	2	8.94436	2

```
-- QUESTION 29 :  Get the following details for top 9 directors (based on number of movies)
-- Director id
-- Name
-- Number of movies
-- Average inter movie duration in days
-- Average movie ratings
-- Total votes
-- Min rating
-- Max rating
-- total movie durations

-- Step 1: Aggregate director-wise movie data
WITH DirectorMovies AS (
    SELECT dm.name_id AS director_id, n.name AS director_name,
        COUNT(DISTINCT dm.movie_id) AS number_of_movies,
        AVG(r.avg_rating) AS avg_rating,
        SUM(r.total_votes) AS total_votes,
        MIN(r.avg_rating) AS min_rating,
        MAX(r.avg_rating) AS max_rating,
```

```
        SUM(m.duration) AS total_duration
    FROM director_mapping dm
    JOIN names n ON dm.name_id = n.id
    JOIN movie m ON dm.movie_id = m.id
    JOIN ratings r ON m.id = r.movie_id
    WHERE m.duration IS NOT NULL
    GROUP BY dm.name_id, n.name
),

-- Step 2: Calculate inter-movie duration for each director
DirectorInterMovieDays AS (
    SELECT dm.name_id AS director_id,
        m.date_published AS movie_date,
        LAG(m.date_published) OVER (PARTITION BY dm.name_id ORDER BY m.date_published) AS
prev_movie_date
    FROM director_mapping dm
    JOIN movie m ON dm.movie_id = m.id
    WHERE m.date_published IS NOT NULL
),

-- Step 3: Compute the average inter-movie duration for each director
DirectorAvgInterMovieDays AS (
    SELECT director_id,
        AVG(DATEDIFF(movie_date, prev_movie_date)) AS avg_inter_movie_days
    FROM DirectorInterMovieDays
    WHERE prev_movie_date IS NOT NULL
    GROUP BY director_id
)

-- Step 4: Combine results and rank directors based on the number of movies
SELECT dm.director_id, dm.director_name, dm.number_of_movies,
      daimd.avg_inter_movie_days, dm.avg_rating, dm.total_votes,
      dm.min_rating, dm.max_rating, dm.total_duration
FROM DirectorMovies dm
LEFT JOIN DirectorAvgInterMovieDays daimd ON dm.director_id = daimd.director_id
ORDER BY dm.number_of_movies DESC
LIMIT 9;
```

director_id	director_name	number_of_movies	avg_inter_movie_days	avg_rating	total_votes	min_rating	max_rating	total_duration
nm2096009	Andrew Jones	5	190.7500	3.02000	1989	2.7	3.2	432
nm1777967	A.L. Vijay	5	176.7500	5.42000	1754	3.7	6.9	613
nm0814469	Sion Sono	4	331.0000	6.02500	2972	5.4	6.4	502
nm0831321	Chris Stokes	4	198.3333	4.32500	3664	4.0	4.6	352

nm0515005	Sam Liu	4	260.3333	6.22500	28557	5.8	6.7	312
nm0001752	Steven Soderbergh	4	254.3333	6.47500	171684	6.2	7.0	401
nm0425364	Jesse V. Johnson	4	299.0000	5.45000	14778	4.2	6.5	383
nm2691863	Justin Price	4	315.0000	4.50000	5343	3.0	5.8	346
nm6356309	Özgür Bakar	4	112.0000	3.75000	1092	3.1	4.9	374

Thank You!