****

FINAL SQL CAPSTONE PROJECT

****

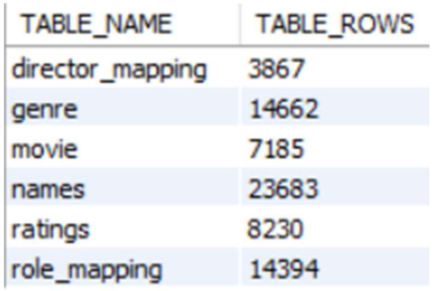
-- Segment 1:

-- Q1. Find the total number of rows in each table of the schema?

-- Type your code below:

```
SELECT
	table_name,
	 table_rows
FROM
	information_schema.tables
WHERE
	table_schema='imdb';
```

Output Q1:



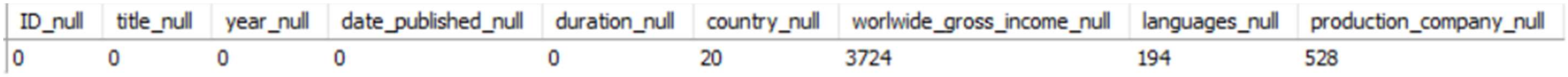
TABLE_NAME	TABLE_ROWS
director_mapping	3867
genre	14662
movie	7185
names	23683
ratings	8230
role_mapping	14394

-- Q2. Which columns in the movie table have null values?

-- Type your code below:

```
SELECT
        SUM(CASE WHEN id IS NULL THEN 1 ELSE 0 END) AS ID_null,
        SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS title_null,
        SUM(CASE WHEN year IS NULL THEN 1 ELSE 0 END) AS year_null,
        SUM(CASE WHEN date_published IS NULL THEN 1 ELSE 0 END) AS date_published_null,
        SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS duration_null,
        SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS country_null,
        SUM(CASE WHEN worlwide_gross_income IS NULL THEN 1 ELSE 0 END) AS worlwide_gross_income_null,
        SUM(CASE WHEN languages IS NULL THEN 1 ELSE 0 END) AS languages_null,
        SUM(CASE WHEN production_company IS NULL THEN 1 ELSE 0 END) AS production_company_null
FROM movie;
```

Output Q2:



```
ID_null  title_null  year_null  date_published_null  duration_null  country_null  worlwide_gross_income_null  languages_null  production_company_null
0        0           0          0                    0              20            3724                        194             528
```

In the movie table there are 4 columns that have null values:

country, worlwide_gross_income, languages, production_company

-- Now as you can see four columns of the movie table has null values. Let's look at the at the movies released each year.

-- Q3. Find the total number of movies released each year? How does the trend look month wise? (Output expected)

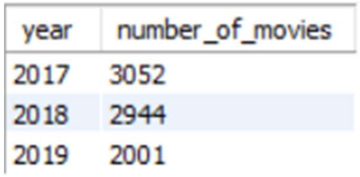
/* Output format for the first part:

```
+-------------------------+-------------------------+
|          Year           | number_of_movies |
+-------------------------+-------------------------+
|          2017           |         2134            |
|          2018           |         -               |
|          2019           |         -               |
+-------------------------+-------------------------+
```

Output format for the second part of the question:

```
+-------------------------+-------------------------+
|  month_num              | number_of_movies |
+-------------------------+-------------------------+
|          1              |          134            |
|          2              |          231            |
|          -              |         -               |
+-------------------------+-------------------------+*/
```

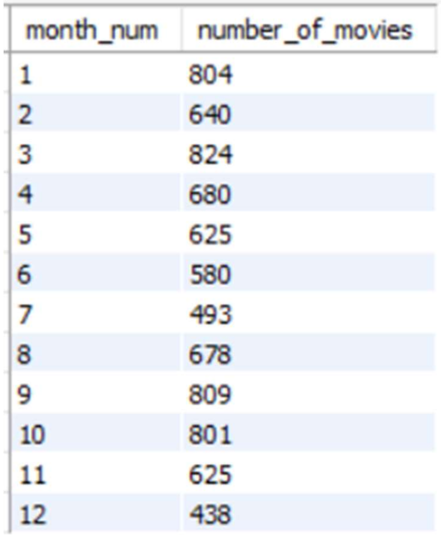
```
-- Type your code below:

SELECT
        year,
        COUNT(*) as number_of_movies
FROM movie
GROUP by year
ORDER by year;


/* Output 1st part Q3:
```



year	number_of_movies
2017	3052
2018	2944
2019	2001

```
 SELECT
        month(date_published) AS month_num,
        COUNT(month(date_published)) AS number_of_movies
FROM movie
GROUP BY month_num
ORDER BY month_num;
```

/* Output 2nd part Q3:



month_num	number_of_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

/*The highest number of movies is produced in the month of March.

So, now that you have understood the month-wise trend of movies, let’s take a look at the other details in the movies table.

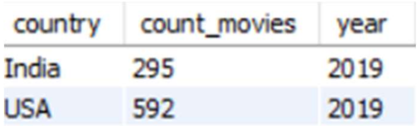
We know USA and India produces huge number of movies each year. Lets find the number of movies produced by USA or India for the last year.*/

-- Q4. How many movies were produced in the USA or India in the year 2019?

-- Type your code below:

```
SELECT country, COUNT(country) AS count_movies, year
FROM movie
WHERE country='USA' OR country='India'
GROUP BY country, year
HAVING year = 2019;
```

Output Q4:



country	count_movies	year
India	295	2019
USA	592	2019

-- In 2019, 1007 movies were produced in either the United States or India.

/* USA and India produced more than a thousand movies(you know the exact number!) in the year 2019.

Exploring table Genre would be fun!!

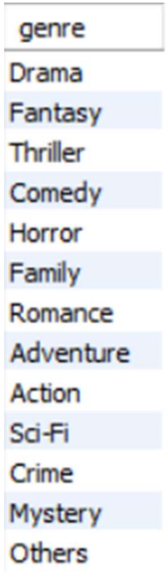
Let’s find out the different genres in the dataset.*/

-- Q5. Find the unique list of the genres present in the data set?

-- Type your code below:

```
SELECT DISTINCT genre
FROM genre;
```

Output Q5:



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

-- The dataset includes movies from 13 genre.

/* So, RSVP Movies plans to make a movie of one of these genres.

Now, wouldn’t you want to know which genre had the highest number of movies produced in the last year?

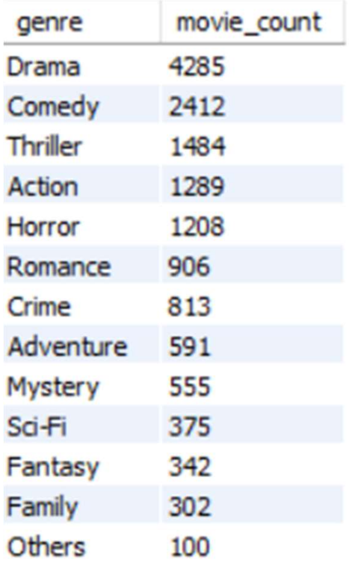
Combining both the movie and genres table can give more interesting insights. */

-- Q6. Which genre had the highest number of movies produced overall?

-- Type your code below:

```
SELECT genre, COUNT(*) as movie_count
FROM genre
GROUP BY genre
ORDER BY movie_count DESC;
```

Output Q6:



genre	movie_count
Drama	4285
Comedy	2412
Thriller	1484
Action	1289
Horror	1208
Romance	906
Crime	813
Adventure	591
Mystery	555
Sci-Fi	375
Fantasy	342
Family	302
Others	100

-- The overall number of drama films produced was 4285, the most of any genre.

/* So, based on the insight that you just drew, RSVP Movies should focus on the ‘Drama’ genre.

But wait, it is too early to decide. A movie can belong to two or more genres.

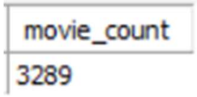
So, let’s find out the count of movies that belong to only one genre.*/

-- Q7. How many movies belong to only one genre?

-- Type your code below:

```
SELECT COUNT(*) as movie_count
FROM(
        SELECT movie_id
        FROM genre
        GROUP BY movie_id
        HAVING COUNT(*) =1
) AS single_genre_movie;
```

Output Q7:



```
movie_count
3289
```

-- 3289 movies adhere to only one genre.

/* There are more than three thousand movies which has only one genre associated with them.

So, this figure appears significant.

Now, let's find out the possible duration of RSVP Movies’ next project.*/

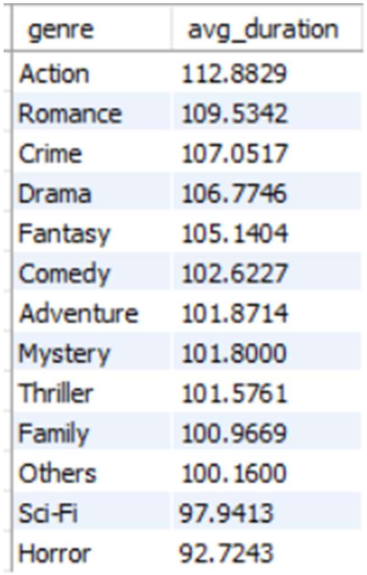
-- Q8. What is the average duration of movies in each genre?

-- (Note: The same movie can belong to multiple genres.)

-- Type your code below:

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

Output Q8:



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

-- The action genre has the longest duration (112.88 seconds), followed by the romance and crime genres.

/* Now you know, movies of genre 'Drama' (produced highest in number in 2019) has the average duration of 106.77 mins.

Lets find where the movies of genre 'thriller' on the basis of number of movies.*/

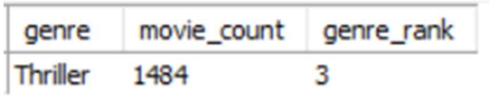
-- Q9. What is the rank of the ‘thriller’ genre of movies among all the genres in terms of number of movies produced?

-- (Hint: Use the Rank function)

-- Type your code below:

```
WITH genre_rank AS
(
        SELECT
                genre, COUNT(movie_id) AS movie_count,
                RANK() OVER(ORDER BY COUNT(movie_id) DESC) AS genre_rank
        FROM genre
        GROUP BY genre
)
SELECT *
FROM genre_rank
WHERE genre='thriller';
```

Output Q9:



genre	movie_count	genre_rank
Thriller	1484	3

-- Thriller has a 1484 movie count with a rank of 3.

```
/*Thriller movies is in top 3 among all genres in terms of number of movies
 In the previous segment, you analysed the movies and genres tables.
 In this segment, you will analyse the ratings table as well.
To start with lets get the min and max values of different columns in the table*/
```

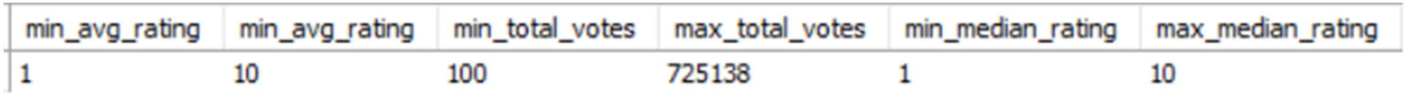
-- Segment 2:

-- Q10. Find the minimum and maximum values in each column of the ratings table except the movie_id column?

-- Type your code below:

```
SELECT
	 ROUND(MIN(avg_rating)) AS min_avg_rating
	,ROUND(MAX(avg_rating)) AS min_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
```

Output Q10:



min_avg_rating	min_avg_rating	min_total_votes	max_total_votes	min_median_rating	max_median_rating
1	10	100	725138	1	10

/* So, the minimum and maximum values in each column of the ratings table are in the expected range.

This implies there are no outliers in the table.

Now, let’s find out the top 10 movies based on average rating.*/

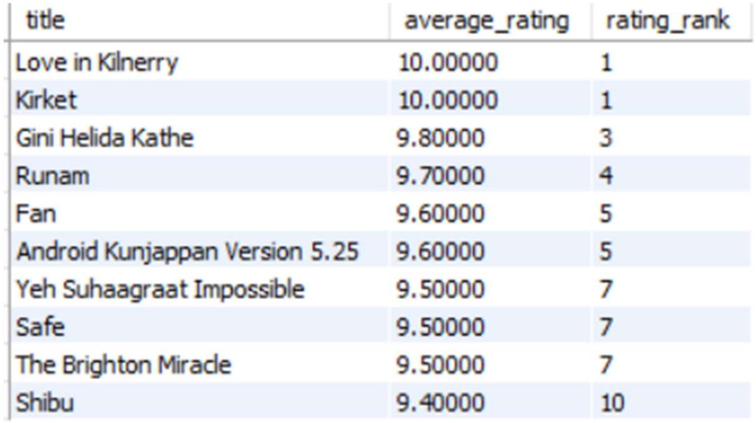
-- Q11. Which are the top 10 movies based on average rating?

-- Type your code below:

-- It's ok if RANK() or DENSE_RANK() is used too

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

Output Q11:



title	average_rating	rating_rank
Love in Kilnerry	10.00000	1
Kirket	10.00000	1
Gini Helida Kathe	9.80000	3
Runam	9.70000	4
Fan	9.60000	5
Android Kunjappan Version 5.25	9.60000	5
Yeh Suhaagraat Impossible	9.50000	7
Safe	9.50000	7
The Brighton Miracle	9.50000	7
Shibu	9.40000	10

-- The top three movies have an average rating of more than 9.8.

/* Do you find you favourite movie FAN in the top 10 movies with an average rating of 9.6? If not, please check your code again!!

So, now that you know the top 10 movies, do you think character actors and filler actors can be from these movies?

Summarising the ratings table based on the movie counts by median rating can give an excellent insight.*/

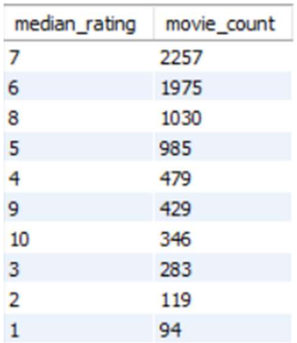
-- Q12. Summarise the ratings table based on the movie counts by median ratings.

-- Type your code below:

-- Order by is good to have

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

Output Q12:



median_rating	movie_count
7	2257
6	1975
8	1030
5	985
4	479
9	429
10	346
3	283
2	119
1	94

-- The highest movie count of 2257 is found in the median rating of 7.

/* Movies with a median rating of 7 is highest in number.

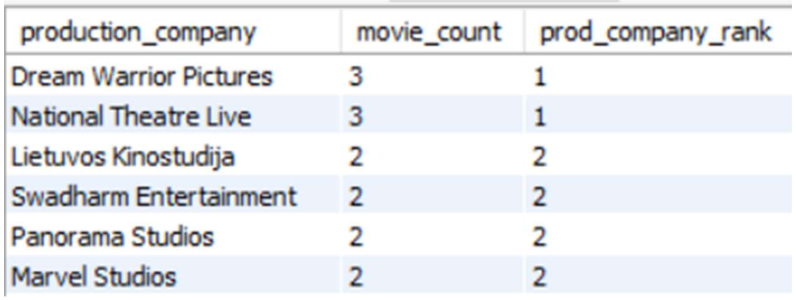
Now, let's find out the production house with which RSVP Movies can partner for its next project.*/

-- Q13. Which production house has produced the most number of hit movies (average rating > 8)?

-- Type your code below:

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

Output Q13:



production_company	movie_count	prod_company_rank
Dream Warrior Pictures	3	1
National Theatre Live	3	1
Lietuvos Kinostudija	2	2
Swadharm Entertainment	2	2
Panorama Studios	2	2
Marvel Studios	2	2

- The production companies Dream Warrior Pictures and National Theatre Live have made the most amount of highly rated movies (average rating > 8). Their rank is 1 and their movie count is 3.

-- It's ok if RANK() or DENSE_RANK() is used too

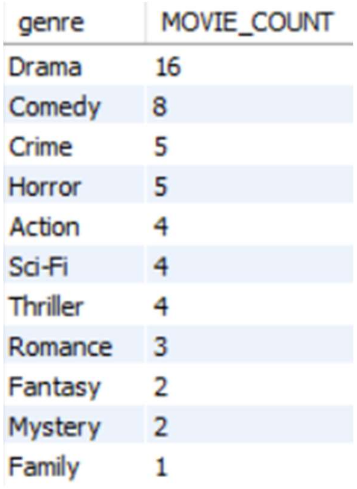
-- Answer can be Dream Warrior Pictures or National Theatre Live or both

-- Q14. How many movies released in each genre during March 2017 in the USA had more than 1,000 votes?

-- Type your code below:

```
SELECT
        genre,
        Count(M.id) AS MOVIE_COUNT
FROM   movie AS M
     INNER JOIN genre AS G
           ON G.movie_id = M.id
     INNER JOIN ratings AS R
           ON R.movie_id = M.id
WHERE  year = 2017
     AND Month(date_published) = 3
     AND country = 'USA'
     AND total_votes > 1000
GROUP  BY genre
ORDER  BY movie_count DESC;
```

Output Q14:



genre	MOVIE_COUNT
Drama	16
Comedy	8
Crime	5
Horror	5
Action	4
Sci-Fi	4
Thriller	4
Romance	3
Fantasy	2
Mystery	2
Family	1

-- In March 2017, 24 drama movies were released in the USA and received over 1,000 votes.

-- The top three genres in March 2017 in the United States were drama, comedy, and action, with over 1,000 votes.

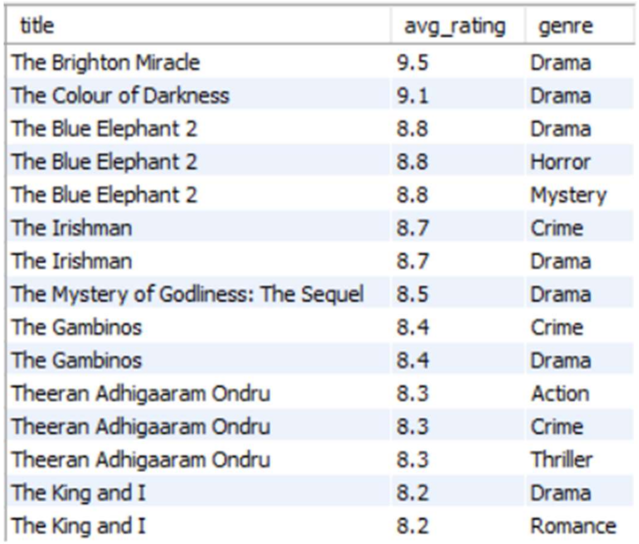
-- Lets try to analyse with a unique problem statement.

-- Q15. Find movies of each genre that start with the word ‘The’ and which have an average rating > 8?

-- Type your code below:

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

Output Q15:



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
Theeran Adhigaaram Ondru	8.3	Action
Theeran Adhigaaram Ondru	8.3	Crime
Theeran Adhigaaram Ondru	8.3	Thriller
The King and I	8.2	Drama
The King and I	8.2	Romance

-- There are 15 movies whose titles begin with 'The'.

-- The Brighton Miracle gets the highest average rating of 9.5, followed by The Colour of Darkness with 9.1.

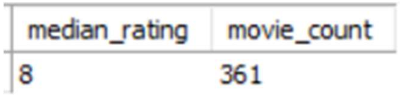
-- You should also try your hand at median rating and check whether the ‘median rating’ column gives any significant insights.

-- Q16. Of the movies released between 1 April 2018 and 1 April 2019, how many were given a median rating of 8?

-- Type your code below:

```
SELECT
        median_rating,
        COUNT(movie_id) AS movie_count
FROM movie AS m
INNER JOIN ratings AS r
ON m.id = r.movie_id
WHERE median_rating = 8 AND date_published BETWEEN '2018-04-01' AND '2019-04-01'
GROUP BY median_rating;
```

Output Q16:



median_rating	movie_count
8	361

-- Between 1 April 2018 and 1 April 2019, 361 movies were released with a median rating of 8.

-- Once again, try to solve the problem given below.

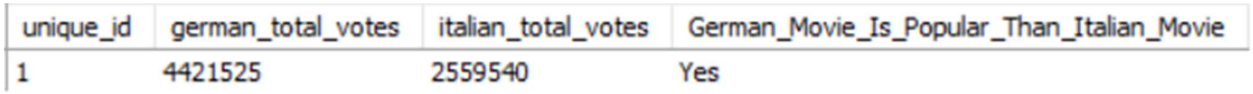
-- Q17. Do German movies get more votes than Italian movies?

-- Hint: Here you have to find the total number of votes for both German and Italian movies.

-- Type your code below:

```
with german_summary AS (
SELECT SUM(r.total_votes) AS german_total_votes,
RANK() OVER(ORDER BY SUM(r.total_votes)) AS unique_id
FROM movie AS m
INNER JOIN ratings AS r
ON m.id=r.movie_id
WHERE m.languages LIKE '%German%'
), italian_summary AS (
SELECT SUM(r.total_votes) AS italian_total_votes,
RANK() OVER(ORDER BY sum(r.total_votes)) AS unique_id
FROM movie AS m
INNER JOIN ratings AS r
ON m.id=r.movie_id
WHERE m.languages LIKE '%Italian%'
) SELECT *,
CASE
        WHEN german_total_votes > italian_total_votes THEN 'Yes' ELSE 'No'
   END AS 'German_Movie_Is_Popular_Than_Italian_Movie'
FROM german_summary
INNER JOIN
italian_summary
USING(unique_id);
```

Output Q17:



unique_id	german_total_votes	italian_total_votes	German_Movie_Is_Popular_Than_Italian_Movie
1	4421525	2559540	Yes

-- Based on observation, German movies received the most votes when compared to language and country columns.

-- The answer is yes.

/* Now that you have analysed the movies, genres and ratings tables, let us now analyse another table, the names table.

Let’s begin by searching for null values in the tables.*/

-- Segment 3:

-- Q18. Which columns in the names table have null values?

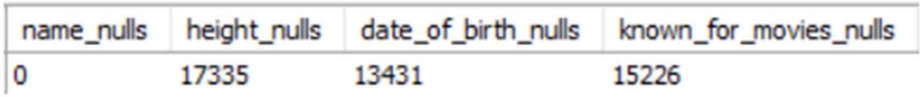
/*Hint: You can find null values for individual columns or follow below output format

```
+----------------+----------------+-------------------------+-------------------------------+
| name_nulls | height_nulls | date_of_birth_nulls  |known_for_movies_nulls|
+----------------+----------------+-------------------------+-------------------------------+
|        0        |      123        |          1234             |         12345                  |
+----------------+----------------+-------------------------+-------------------------------+*/
```

-- Type your code below:

```
SELECT
        COUNT(*)-COUNT(name) AS name_nulls
        ,COUNT(*)-COUNT(height) AS height_nulls
        ,COUNT(*)-COUNT(date_of_birth) AS date_of_birth_nulls
        ,COUNT(*)-COUNT(known_for_movies) AS known_for_movies_nulls
FROM names;
```

Output Q18:



name_nulls	height_nulls	date_of_birth_nulls	known_for_movies_nulls
0	17335	13431	15226

/* There are no Null value in the column 'name'.

The director is the most important person in a movie crew.

Let’s find out the top three directors in the top three genres who can be hired by RSVP Movies.*/

-- Q19. Who are the top three directors in the top three genres whose movies have an average rating > 8?

-- (Hint: The top three genres would have the most number of movies with an average rating > 8.)

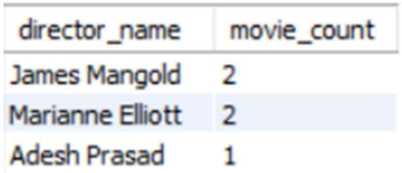
```
/* Output format:
+---------------------------------+----------------------------------|
| director_name                   |          movie_count             |
+---------------------------------+----------------------------------|
| James Mangold                   |               4                  |
|         .                       |               .                  |
|         .                       |               .                  |
+---------------------------------+----------------------------------| */
```

-- Type your code below:

```
WITH genre_top3 AS
(
    SELECT
        g.genre
        ,COUNT(g.movie_id) AS movie_count
        ,r.avg_rating
    FROM movie AS m
    INNER JOIN genre AS g
    ON m.id=g.movie_id
    INNER JOIN ratings AS r
    ON m.id=r.movie_id
    WHERE r.avg_rating>8
    GROUP BY g.genre, r.avg_rating
    ORDER BY movie_count DESC
    LIMIT 3
)
SELECT
```

```
          n.name as director_name
          ,COUNT(m.id) as movie_count
FROM names AS n
INNER JOIN director_mapping AS d
ON n.id=d.name_id
INNER JOIN movie AS m
ON d.movie_id=m.id
INNER JOIN genre AS g
ON m.id=g.movie_id
INNER JOIN ratings AS r
ON m.id=r.movie_id
WHERE r.avg_rating>8 AND g.genre IN (SELECT genre FROM genre_top3)
GROUP BY director_name
ORDER BY movie_count DESC, director_name ASC
LIMIT 3;
```

Output Q19:



director_name	movie_count
James Mangold	2
Marianne Elliott	2
Adesh Prasad	1

/* James Mangold can be hired as the director for RSVP's next project. Do you remember his movies, 'Logan' and 'The Wolverine'.

Now, let’s find out the top two actors.*/

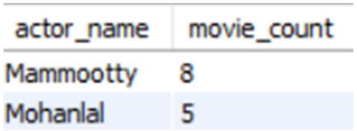
-- Q20. Who are the top two actors whose movies have a median rating >= 8?

```
/* Output format:
+---------------------------------+-----------------+
| actor_name                      | movie_count |
+---------------------------------+-----------------+
|Christain Bale                   | 10              |
+---------------------------------+-----------------+ */
```

-- Type your code below:

```
SELECT
        DISTINCT name AS actor_name
        ,COUNT(r.movie_id) AS movie_count
FROM ratings AS r
INNER JOIN role_mapping AS rm
ON rm.movie_id = r.movie_id
INNER JOIN names AS n
ON rm.name_id = n.id
WHERE median_rating >= 8 AND category = 'actor'
GROUP BY name
ORDER BY movie_count DESC
LIMIT 2;
```

Output Q20:



actor_name	movie_count
Mammootty	8
Mohanlal	5

/* Have you find your favourite actor 'Mohanlal' in the list. If no, please check your code again.

RSVP Movies plans to partner with other global production houses.

Let’s find out the top three production houses in the world.*/

-- Q21. Which are the top three production houses based on the number of votes received by their movies?

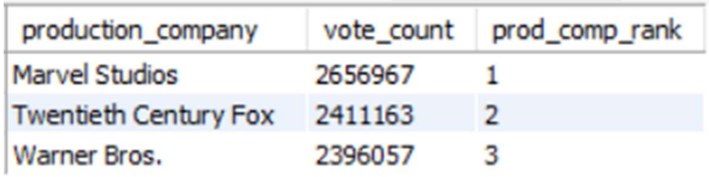
/* Output format:

```
+---------------------------------+----------------+------------------------+
|   production_company      | vote_count |  prod_comp_rank  |
+---------------------------------+----------------+------------------------+
| The Archers                     |        830     |                  1          |
|         -                           |        -         |                  -          |
+---------------------------------+----------------+------------------------+*/
```

-- Type your code below:

```
SELECT
	production_company
	,SUM(r.total_votes) AS vote_count
	,DENSE_RANK() OVER(ORDER BY sum(r.total_votes)DESC) AS prod_comp_rank
FROM movie AS m
INNER JOIN
ratings AS r
ON m.id= r.movie_id
GROUP BY production_company
LIMIT 3;
```

Output Q21:



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

/*Yes Marvel Studios rules the movie world.

So, these are the top three production houses based on the number of votes received by the movies they have produced.

Since RSVP Movies is based out of Mumbai, India also wants to woo its local audience.

RSVP Movies also wants to hire a few Indian actors for its upcoming project to give a regional feel.

Let’s find who these actors could be.*/

-- Q22. Rank actors with movies released in India based on their average ratings. Which actor is at the top of the list?

-- Note: The actor should have acted in at least five Indian movies.

-- (Hint: You should use the weighted average based on votes. If the ratings clash, then the total number of votes should act as the tie breaker.)

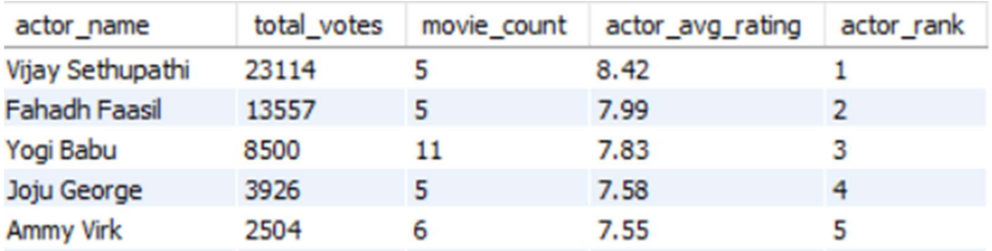
/* Output format:

```
+-------------------------+----------------+------------------------+-------------------------+----------------+
| actor_name              | total_votes    | movie_count            | actor_avg_rating        | actor_rank     |
+-------------------------+----------------+------------------------+-------------------------+----------------+
| Yogi Babu               |   345          |          11            |        8.42             |        1       |
+-------------------------+----------------+------------------------+-------------------------+----------------+*/
```

```
-- Type your code below:
WITH top_actor AS
(
    SELECT b.NAME
            AS
            actor_name,
            SUM(c.total_votes)
            AS
              total_votes,
            COUNT(DISTINCT a.movie_id)
            AS
              movie_count,
            ROUND(Sum(c.avg_rating * c.total_votes) / Sum(c.total_votes), 2)
            AS
            actor_avg_rating
       FROM   role_mapping a
            INNER JOIN names b
                  ON a.name_id = b.id
            INNER JOIN ratings c
                  ON a.movie_id = c.movie_id
            INNER JOIN movie d
                  ON a.movie_id = d.id
       WHERE  a.category = 'actor'
            AND d.country LIKE '%India%'
       GROUP  BY a.name_id,
              b.NAME
       HAVING COUNT(DISTINCT a.movie_id) >= 5
)
```

```
SELECT *,
       RANK()
         OVER (
           ORDER BY actor_avg_rating DESC) AS actor_rank
FROM   top_actor;
```

Output Q22:



actor_name	total_votes	movie_count	actor_avg_rating	actor_rank
Vijay Sethupathi	23114	5	8.42	1
Fahadh Faasil	13557	5	7.99	2
Yogi Babu	8500	11	7.83	3
Joju George	3926	5	7.58	4
Ammy Virk	2504	6	7.55	5

-- Top actor is Vijay Sethupathi

-- Q23.Find out the top five actresses in Hindi movies released in India based on their average ratings?

-- Note: The actresses should have acted in at least five Indian movies.

-- (Hint: You should use the weighted average based on votes. If the ratings clash, then the total number of votes should act as the tie breaker.)

/* Output format:

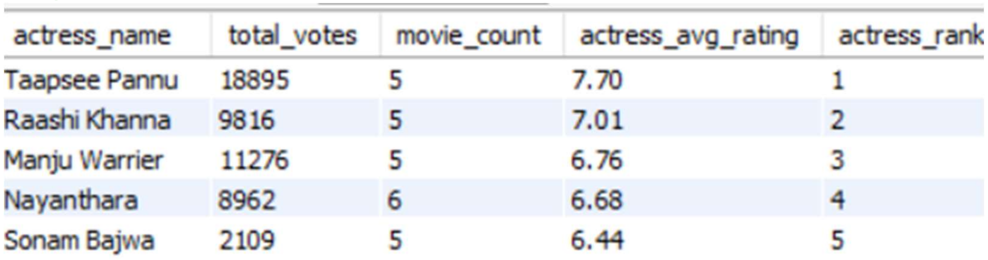
```
+-------------------------+----------------+-------------------------+---------------------------+------------------------+
| actress_name        | total_votes  | movie_count          | actress_avg_rating  | actress_rank        |
+-------------------------+----------------+-------------------------+---------------------------+------------------------+
|Tabu                      |     3455      |              11            |         8.42               |              1            |
|  -                      |   -            |          -              |       -                   |          -            |
+-------------------------+----------------+-------------------------+---------------------------+-----------------------+
```

-- Type your code below:

```
WITH top_actress AS
(
   SELECT b.NAME
         AS
         actress_name,
         SUM(c.total_votes)
         AS
           total_votes,
         COUNT(DISTINCT a.movie_id)
         AS
           movie_count,
         ROUND(Sum(c.avg_rating * c.total_votes) / Sum(c.total_votes), 2)
         AS
         actress_avg_rating
     FROM   role_mapping a
         INNER JOIN names b
               ON a.name_id = b.id
         INNER JOIN ratings c
               ON a.movie_id = c.movie_id
         INNER JOIN movie d
               ON a.movie_id = d.id
     WHERE  a.category = 'actress'
         AND d.country LIKE '%India%'
     GROUP  BY a.name_id,
           b.NAME
```

```
        HAVING COUNT(DISTINCT a.movie_id) >= 5
)
SELECT *,
      RANK()
        OVER (
          ORDER BY actress_avg_rating DESC) AS actress_rank
FROM   top_actress
LIMIT 5;
```

Output Q23:



actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Taapsee Pannu	18895	5	7.70	1
Raashi Khanna	9816	5	7.01	2
Manju Warrier	11276	5	6.76	3
Nayanthara	8962	6	6.68	4
Sonam Bajwa	2109	5	6.44	5

/* Taapsee Pannu tops with average rating 7.74.

Now let us divide all the thriller movies in the following categories and find out their numbers.*/

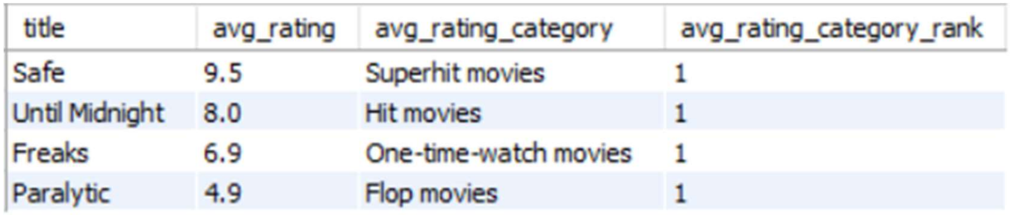
-- Q24. Select thriller movies as per avg rating and classify them in the following category:

Rating > 8: Superhit movies
Rating between 7 and 8: Hit movies
Rating between 5 and 7: One-time-watch movies
Rating < 5: Flop movies

-- Type your code below:

```
WITH cte_avg_rating_category AS
(
    SELECT
        title
        ,r.avg_rating
        ,CASE
            WHEN avg_rating > 8 THEN 'Superhit movies'
            WHEN avg_rating BETWEEN 7 AND 8 THEN 'Hit movies'
            WHEN avg_rating BETWEEN 5 AND 7 THEN 'One-time-watch movies'
            WHEN avg_rating < 5 THEN 'Flop movies'
        END AS avg_rating_category
    FROM movie AS m
    INNER JOIN genre AS g
    ON m.id=g.movie_id
    INNER JOIN ratings AS r
    ON m.id=r.movie_id
    WHERE genre='thriller'
    ORDER BY r.avg_rating DESC
),
cte_avg_rating_category_rank AS
(
    SELECT *
        ,ROW_NUMBER() OVER (PARTITION BY avg_rating_category ORDER BY avg_rating DESC)
        AS avg_rating_category_rank
    FROM cte_avg_rating_category
)
SELECT * FROM cte_avg_rating_category_rank
WHERE avg_rating_category_rank =1
```

Output Q24:



title	avg_rating	avg_rating_category	avg_rating_category_rank
Safe	9.5	Superhit movies	1
Until Midnight	8.0	Hit movies	1
Freaks	6.9	One-time-watch movies	1
Paralytic	4.9	Flop movies	1

/* Until now, you have analysed various tables of the data set.

Now, you will perform some tasks that will give you a broader understanding of the data in this segment.*/

-- Segment 4:

-- Q25. What is the genre-wise running total and moving average of the average movie duration?

-- (Note: You need to show the output table in the question.)

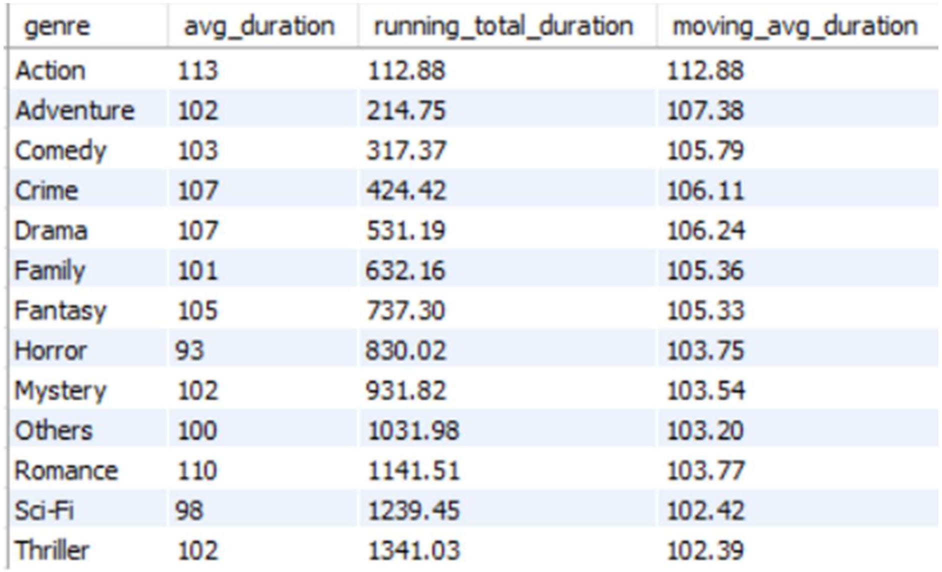
/* Output format:

```
+-------------------------+-------------------------+---------------------------------+----------------------------------+
| genre                   | avg_duration            | running_total_duration          | moving_avg_duration              |
+-------------------------+-------------------------+---------------------------------+----------------------------------+
| Comedy                  | 145                     | 106.2                           | 128.42                           |
| -                       | -                       | -                               | -                       |
+-------------------------+-------------------------+---------------------------------+----------------------------------+*/
```

-- Type your code below:

```
SELECT genre,
        ROUND(AVG(duration)) AS avg_duration,
        SUM(ROUND(AVG(duration),2)) OVER(ORDER BY genre ROWS UNBOUNDED PRECEDING) AS running_total_duration,
        ROUND(AVG(ROUND(AVG(duration),2)) OVER(ORDER BY genre ROWS 10 PRECEDING),2) AS moving_avg_duration
FROM movie AS m
INNER JOIN genre AS g
ON m.id= g.movie_id
GROUP BY genre
ORDER BY genre;
```

Output Q25:



genre	avg_duration	running_total_duration	moving_avg_duration
Action	113	112.88	112.88
Adventure	102	214.75	107.38
Comedy	103	317.37	105.79
Crime	107	424.42	106.11
Drama	107	531.19	106.24
Family	101	632.16	105.36
Fantasy	105	737.30	105.33
Horror	93	830.02	103.75
Mystery	102	931.82	103.54
Others	100	1031.98	103.20
Romance	110	1141.51	103.77
Sci-Fi	98	1239.45	102.42
Thriller	102	1341.03	102.39

-- Round is good to have and not a must have; Same thing applies to sorting

-- Let us find top 5 movies of each year with top 3 genres.

Q26. Which are the five highest-grossing movies of each year that belong to the top three genres?

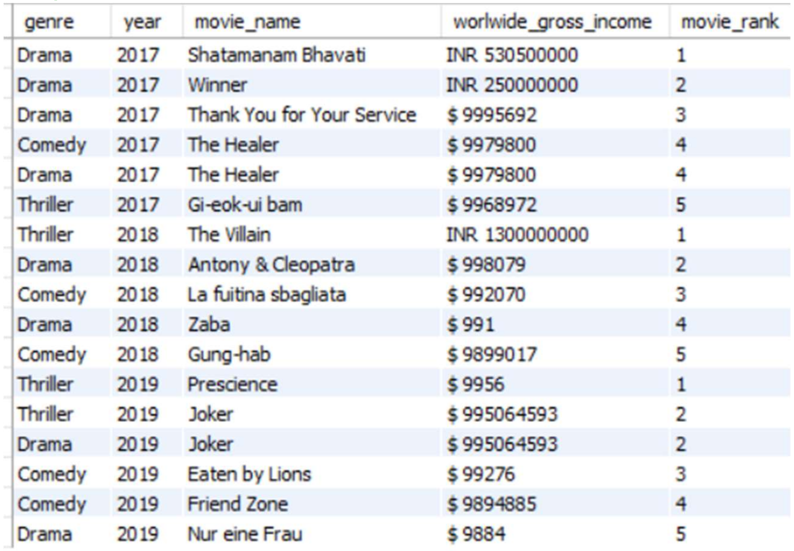
-- (Note: The top 3 genres would have the most number of movies.)

```
/* Output format:

+-------------------------+----------------+-----------------------------------------+-----------------------------------------+----------------+
| genre                   | year           | movie_name                              | worlwide_gross_income                   | movie_rank |
+-------------------------+----------------+-----------------------------------------+-----------------------------------------+----------------+
| Comedy                  |  2017          | indian                                  | $103244842                              |        1       |
| -                       |  -             | -                                       | -                                       |        -       |
+-------------------------+----------------+-----------------------------------------+-----------------------------------------+----------------+
```

-- Type your code below:

```
WITH top_3_genre AS -- Top 3 Genres based on most number of movies
(
        SELECT genre, COUNT(movie_id) AS number_of_movies
   FROM genre AS g
   INNER JOIN movie AS m
   ON g.movie_id = m.id
   GROUP BY genre
   ORDER BY COUNT(movie_id) DESC
   LIMIT 3
),
top_5 AS
(
        SELECT genre,
                        year,
                        title AS movie_name,
                        worlwide_gross_income,
                        DENSE_RANK() OVER(PARTITION BY year ORDER BY worlwide_gross_income DESC) AS movie_rank
              FROM movie AS m
         INNER JOIN genre AS g
         ON m.id= g.movie_id
              WHERE genre IN (SELECT genre FROM top_3_genre)
)
SELECT *
FROM top_5
WHERE movie_rank<=5;
```

Output Q26:



genre	year	movie_name	worlwide_gross_income	movie_rank
Drama	2017	Shatamanam Bhavati	INR 530500000	1
Drama	2017	Winner	INR 250000000	2
Drama	2017	Thank You for Your Service	$ 9995692	3
Comedy	2017	The Healer	$ 9979800	4
Drama	2017	The Healer	$ 9979800	4
Thriller	2017	Gi-eok-ui bam	$ 9968972	5
Thriller	2018	The Villain	INR 1300000000	1
Drama	2018	Antony & Cleopatra	$ 998079	2
Comedy	2018	La fuitina sbagliata	$ 992070	3
Drama	2018	Zaba	$ 991	4
Comedy	2018	Gung-hab	$ 9899017	5
Thriller	2019	Prescience	$ 9956	1
Thriller	2019	Joker	$ 995064593	2
Drama	2019	Joker	$ 995064593	2
Comedy	2019	Eaten by Lions	$ 99276	3
Comedy	2019	Friend Zone	$ 9894885	4
Drama	2019	Nur eine Frau	$ 9884	5

-- Finally, let’s find out the names of the top two production houses that have produced the highest number of hits among multilingual movies.

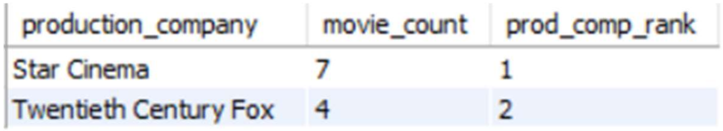
Q27. Which are the top two production houses that have produced the highest number of hits (median rating >= 8) among multilingual movies?

```
/* Output format:

+-------------------------------------------+------------------------+--------------------------+
|production_company                         | movie_count            | prod_comp_rank    |
+-------------------------------------------+------------------------+--------------------------+
| The Archers                               | 830                    |            1             |
| -                                         | -                      |            -             |
+-------------------------------------------+------------------------+--------------------------+*/
```

-- Type your code below:

```
SELECT production_company ,count(m.id)AS movie_count,
RANK() OVER(ORDER BY count(id) DESC) AS prod_comp_rank
FROM movie AS m
INNER JOIN ratings AS r
ON m.id=r.movie_id
WHERE median_rating>=8 AND production_company IS NOT NULL AND position(',' IN languages)>0
GROUP BY production_company
LIMIT 2;
```

Output Q27:



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

-- Multilingual is the important piece in the above question. It was created using POSITION(',' IN languages)>0 logic

-- If there is a comma, that means the movie is of more than one language

---

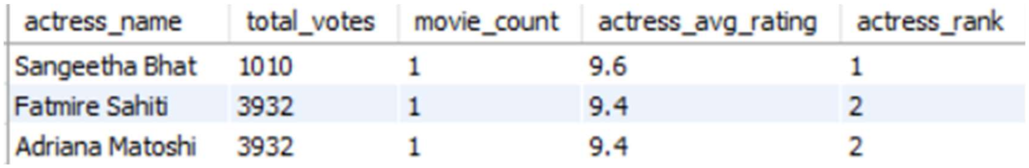
-- Q28. Who are the top 3 actresses based on number of Super Hit movies (average rating >8) in drama genre?

```
/* Output format:

+---------------------------------+---------------+-----------------------+---------------------------------+-----------------------+
| actress_name                    | total_votes   | movie_count           | actress_avg_rating              | actress_rank          |
+---------------------------------+---------------+-----------------------+---------------------------------+-----------------------+
| Laura Dem                       | 1016          | 1                     |        9.6                      |        1              |
| -                               | -             | -                     |        -                        |        -              |
+---------------------------------+---------------+-----------------------+---------------------------------+-----------------------+*/
```

-- Type your code below:

```
SELECT
        name AS actress_name
        ,SUM(total_votes) AS total_votes
         ,COUNT(rm.movie_id) AS movie_count
        ,avg_rating AS actress_avg_rating
        ,DENSE_RANK() OVER(ORDER BY avg_rating DESC) AS actress_rank
FROM names AS n
INNER JOIN role_mapping AS rm
ON n.id = rm.name_id
INNER JOIN ratings AS r
ON r.movie_id = rm.movie_id
INNER JOIN genre AS g
ON r.movie_id = g.movie_id
WHERE category = 'actress' AND avg_rating > 8 AND genre = 'drama'
GROUP BY name ,avg_rating
LIMIT 3;
```

Output Q28:



actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Sangeetha Bhat	1010	1	9.6	1
Fatmire Sahiti	3932	1	9.4	2
Adriana Matoshi	3932	1	9.4	2

Q29. Get the following details for top 9 directors (based on number of movies)

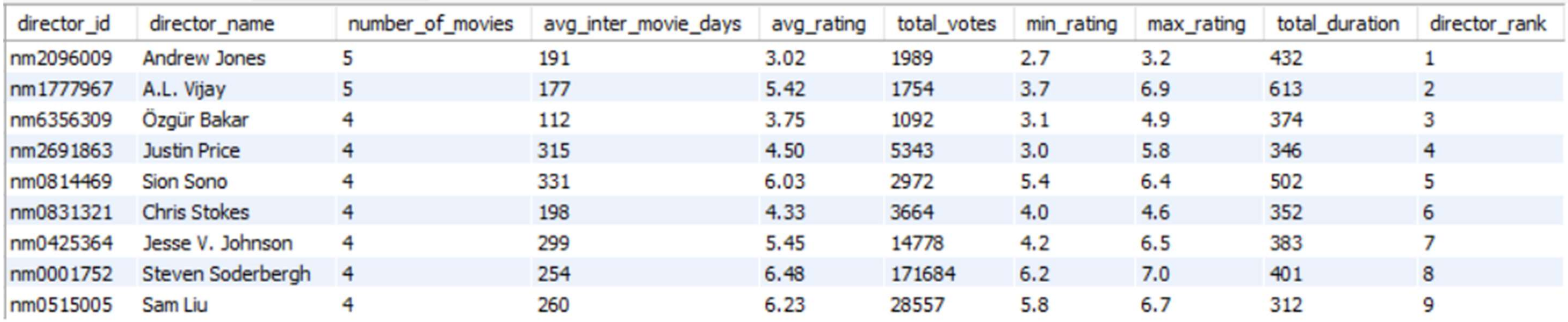
Director id
Name
Number of movies
Average inter movie duration in days
Average movie ratings
Total votes
Min rating
Max rating
total movie durations

Format:

```
+---------------+-------------------+---------------------+---------------------+--------------+--------------+------------+------------+----------------+

| director_id   |        director_name        |        number_of_movies  |        avg_inter_movie_days |     avg_rating     | total_votes  |
min_rating     | max_rating | total_duration |

+---------------+-------------------+---------------------+---------------------+--------------+--------------+------------+------------+----------------+

|nm1777967           |        A.L. Vijay               |                          5                |            177                          |           5.65
|        1754     |      3.7                |        6.9                |                    613                |

+---------------+-------------------+---------------------+---------------------+--------------+--------------+------------+------------+----------------+*/
```

```
-- Type you code below:

WITH movie_date_information AS
(
SELECT d.name_id, name, d.movie_id,
        m.date_published,
    LEAD(date_published, 1) OVER(PARTITION BY d.name_id ORDER BY date_published, d.movie_id) AS next_movie_date
FROM director_mapping d
       JOIN names AS n
   ON d.name_id=n.id
       JOIN movie AS m
   ON d.movie_id=m.id
),
date_diff AS
(
       SELECT *, DATEDIFF(next_movie_date, date_published) AS diff
       FROM movie_date_information
 ),
 avg_inter_days AS
 (
       SELECT name_id, AVG(diff) AS avg_inter_movie_days
       FROM date_diff
       GROUP BY name_id
 ),
 final_output AS
 (
       SELECT d.name_id AS director_id,
             name AS director_name,
             COUNT(d.movie_id) AS number_of_movies,
             ROUND(avg_inter_movie_days) AS avg_inter_movie_days,
             ROUND(AVG(avg_rating),2) AS avg_rating,
             SUM(total_votes) AS total_votes,
             MIN(avg_rating) AS min_rating,
             MAX(avg_rating) AS max_rating,
             SUM(duration) AS total_duration,
             ROW_NUMBER() OVER(ORDER BY COUNT(d.movie_id) DESC) AS director_rank
       FROM
             names AS n
   JOIN director_mapping AS d
   ON n.id=d.name_id
             JOIN ratings AS r
   ON d.movie_id=r.movie_id
             JOIN movie AS m
   ON m.id=r.movie_id
             JOIN avg_inter_days AS a
   ON a.name_id=d.name_id
       GROUP BY director_id
 )
 SELECT *
 FROM final_output
 LIMIT 9;
```

/* output Q29:



director_id	director_name	number_of_movies	avg_inter_movie_days	avg_rating	total_votes	min_rating	max_rating	total_duration	director_rank
nm2096009	Andrew Jones	5	191	3.02	1989	2.7	3.2	432	1
nm1777967	A.L. Vijay	5	177	5.42	1754	3.7	6.9	613	2
nm6356309	Özgür Bakar	4	112	3.75	1092	3.1	4.9	374	3
nm2691863	Justin Price	4	315	4.50	5343	3.0	5.8	346	4
nm0814469	Sion Sono	4	331	6.03	2972	5.4	6.4	502	5
nm0831321	Chris Stokes	4	198	4.33	3664	4.0	4.6	352	6
nm0425364	Jesse V. Johnson	4	299	5.45	14778	4.2	6.5	383	7
nm0001752	Steven Soderbergh	4	254	6.48	171684	6.2	7.0	401	8
nm0515005	Sam Liu	4	260	6.23	28557	5.8	6.7	312	9

****

EXECUTIVE SUMMARY

****

- Major number of movies were released in the month of March, thus releasing in **March** might not be much profitable due to strong competition.
- Studying yearly and monthly movie release data helps grasp global market trends, anticipate prime release times, and smartly roll out Indian films for optimal impact.
- Finding top-rated movies by average ratings assists producers in comparing quality, gauging performance against industry benchmarks, and making informed production decisions.
- The company should focus more on releasing/making movies for **Drama** genre as it is the most popular genre and most produced genre in terms of number of movies produced.
- Perhaps considering the **Drama** as the key genre to focus upon, the **average duration** should be approximately **107 minutes**.
- RSVP Movies should partner with **Dream Warrior Pictures** or **National Theatre Live** production houses as they have produced the most number of hit movies with an average rating greater than 8.
- The most popular director **James Mangold**, acclaimed in the top 3 genres with ratings over 8, is the prime directorial candidate for RSVP's film project.
- **Mammootty** or **Mohanlal** could be hired as actors based on the median ratings and number of movies. As they were the top actors from the desired insights.

- The actor **Vija Sethupathi** and the actress **Tapsee Pannu** could be hired for the next movie project as they were the most popular based on the average rating and total votes.
- RSVP can opt to partner with production house **Marvel Studios** as their global partners based on highest votes received.
- RSVP should consider hiring **Andrew Jones** as he is the top ranked director.
- For multilingual movies, RSVP can partner with **Star Cinema** or **20th Century Fox** as they are top 2 production houses in terms of multilingual movies.