```
USE imdb;

/* Now that you have imported the data sets, let’s explore some of the tables.
 To begin with, it is beneficial to know the shape of the tables and whether any column has null values.
 Further in this segment, you will take a look at 'movies' and 'genre' tables.*/

-- Segment 1:

-- Q1. Find the total number of rows in each table of the schema?
-- Type your code below:

-- 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,
        SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS title,
```

```
   SUM(CASE WHEN year IS NULL THEN 1 ELSE 0 END) AS year,
   SUM(CASE WHEN date_published IS NULL THEN 1 ELSE 0 END) AS date_published,
        SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS duration,
   SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS country,
   SUM(CASE WHEN worlwide_gross_income IS NULL THEN 1 ELSE 0 END) AS
worlwide_gross_income,
        SUM(CASE WHEN languages IS NULL THEN 1 ELSE 0 END) AS languages,
   SUM(CASE WHEN production_company IS NULL THEN 1 ELSE 0 END) AS production_company
FROM movie;
```

-- 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:
Select count(*) as number_of_movies,year from movie
group by year

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

| 2018 | . |

| 2019 | . |

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

Output format for the second part of the question:

SELECT

EXTRACT(MONTH FROM date_published) AS month,

COUNT(*) AS total_movies

FROM movie

GROUP BY EXTRACT(MONTH FROM date_published)

ORDER BY month;

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

| month_num | number_of_movies|

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

| 1 | 134 |

| 2 | 231 |

| . | . |

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

-- Type your code below:

/*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,year,count(*) as no_of_movie from movie

where year=2019 and

country in('USA','India')

group by country,year
```

India 2019 295

USA 2019 592

/* 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 * from 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 no_of_genre

from genre

group by genre

order by no_of_genre desc
```

Drama has the highest number of movie produced

/* 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.*/ -3289

```
-- Q7. How many movies belong to only one genre?
-- Type your code below:
SELECT genre, COUNT(*) AS single_genre_count
FROM (
   SELECT movie_id
   FROM genre
   GROUP BY movie_id
   HAVING COUNT(genre) = 1
) AS single_genre_movie;
```

/* 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.)
```

/* Output format:

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

| genre | avg_duration |

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

| thriller | 105 |

| . | . |

| . | . |

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

-- Type your code below:

```
Select genre,ROUND(avg(duration)) as duration
from movie m
join genre g
on m.id=g.movie_id
group by genre
order by duration DESC
```

/* 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)

11 is the rank

/* Output format:

```
+---------------+-------------------+---------------------+
| genre                    |                movie_count     |                genre_rank    |
+---------------+-------------------+---------------------+
|drama                    |         2312                    |                        2                  |
+---------------+-------------------+---------------------+*/
-- Type your code below:
Select genre,
RANK() over (order by no_of_genre) as RANK
from
(Select genre,count(*) as no_of_genre
from genre
group by genre)
as Genre_counts
order by rank
```

/*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?

/* Output format:

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

| min_avg_rating| max_avg_rating| min_total_votes | max_total_votes
|min_median_rating|min_median_rating|

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

| 0 | 5 | 177 |
          2000 | 0 | 8 |

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

-- Type your code below:

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

/* 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?

/* Output format:

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

| title | avg_rating | movie_rank |

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

| Fan | 9.6 | 5 |

| . | . | . |

| . | . | . |

| . | . | . |

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

-- Type your code below:

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

Select title,avg_rating,rank

FROM(

Select m.title,r.avg_rating,

DENSE_RANK() over (order by r.avg_rating DESC) as rank

from movie m

join ratings r

on m.id=r.movie_id) as movie_ratings

where rank<=10

ORDER BY rank asc

/* 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.

/* Output format:

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

| median_rating | movie_count |

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

| 1 | 105 |

| . | . |

| . | . |

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

-- Type your code below:

-- Order by is good to have

select median_rating,count(*) as movie_counts

from ratings

group by median_rating,movie_id

order by movie_counts

/* 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)??

/* Output format:

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

|production_company|movie_count | prod_company_rank|

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

| The Archers | 1 | 1 |

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

-- Type your code below:

```
SELECT production_company,
      COUNT(*) AS hit_movie_count
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE r.avg_rating > 8
GROUP BY production_company
ORDER BY hit_movie_count DESC
```

-- 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?

/* Output format:

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

| genre | movie_count |

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

| thriller | 105 |

| . | . |

| . | . |

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

-- Type your code below:

-- 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?

/* Output format:

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

| title | avg_rating | genre |

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

| Theeran | 8.3 | Thriller |

| . | . | . |

| . | . | .
|

| . | . | .
|

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

-- Type your code below:

```
select m.title, g.genre,r.avg_rating
from movie m
join genre g on m.id=g.movie_id
join ratings r on m.id=r.movie_id
where avg_rating>8
and m.title like 'Th%'
and g.genre like 'Th%'
```

-- 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:

20

```
Select 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.avg_rating=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:

Italian got more votes than German

```
select m.languages,sum(r.total_votes) as total_votes
from movie m
join ratings r
on m.id=r.movie_id
where languages in('German','Italian')
group by m.languages
```

-- 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
        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_birth,
        sum(case when known_for_movies is null then 1 else 0 end)as known_for_movies
from names;
```

/* 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:

```
Select d.name_id,g.genre,AVG(r.avg_rating) as director_avg_rating
        from director_mapping d
        join genre g on d.movie_id=g.movie_id
        join ratings r on d.movie_id=r.movie_id
        where g.genre in(SELECT
   g.genre
   FROM genre g
   JOIN ratings r ON g.movie_id = r.movie_id
   WHERE r.avg_rating > 8
   GROUP BY g.genre
   LIMIT 3)
GROUP BY g.genre, d.name_id
ORDER BY g.genre,director_avg_rating
LIMIT 3;
```

/* James Mangold can be hired as the director for RSVP's next project. Do you remeber 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 name,name_id,avg_median_rating,row_num

from (Select n.name,ro.name_id,AVG(r.median_rating) as avg_median_rating,

ROW_NUMBER() OVER (ORDER BY AVG(r.median_rating) DESC) AS row_num

from names n

join role_mapping ro

on n.id=ro.name_id

join ratings r

on r.movie_id=ro.movie_id

where r.median_rating>=8

```
group by n.name,ro.name_id

) as ranked_actors

where row_num<=2
```

I dint find Mohanlal in the list

```
/* 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 m.production_company,count(distinct m.id),sum(r.total_votes) as Total_votes

from movie m

join ratings r

on m.id=r.movie_id

group by production_company

order by Total_votes desc
```

/*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 | 3455 | 11 | 8.42 | 1 |

| . | . | . | . | . | . | . |

| . | . | . | . | . | . | . |

```
|                    .                    |                            .                    |              .               |
            .                            |                   .              |

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

-- Type your code below:


WITH ActorMovieCounts AS (
   SELECT
      ro.name_id,
      COUNT(ro.movie_id) AS movie_count
   FROM
      role_mapping ro
   GROUP BY
      ro.name_id
   HAVING
      COUNT(ro.movie_id) > 5
)
SELECT
   n.name,
   m.country,
   r.avg_rating,
   ro.category,
   amc.movie_count
FROM
   role_mapping ro
JOIN
   names n ON ro.name_id = n.id
JOIN
   ratings r ON ro.movie_id = r.movie_id
JOIN
   movie m ON r.movie_id = m.id
JOIN
```

```
    ActorMovieCounts amc ON ro.name_id = amc.name_id
WHERE
    m.country = 'India'
    AND r.avg_rating >= 8
        AND ro.category='actor'
ORDER BY
    r.avg_rating DESC;
```

"nm0001375"

-- 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 three 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 ActressMoviecount as(
	select ro.name_id,ro.category,count(ro.movie_id) as movie_count
	from role_mapping ro
	group by name_id,ro.category
	having count(ro.movie_id)>=3
	and ro.category='actress'
)
Select m.country,m.languages,n.name,r.avg_rating,amc.movie_count,
ROW_NUMBER() OVER (ORDER BY r.avg_rating DESC) AS rank
from movie m
join ratings r
on m.id=r.movie_id
join role_mapping ro
on ro.movie_id=r.movie_id
join names n
on n.id=ro.name_id
join ActressMoviecount amc
on amc.name_id=ro.name_id
where m.country = 'India'
   and  m.languages='Hindi'
order by r.avg_rating DESC
limit 3;
```

/* 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:

```
Select m.title,g.movie_id,genre,r.avg_rating,
        case
            when r.avg_rating > 8 then 'Superhit movies'
                    when r.avg_rating between 7 and 8 then 'Hit movies'
                    when r.avg_rating between 5 and 7 then 'One time watch movies'
                    else  'Flop_movie'
        end as movie_category
from genre g
join ratings r
on g.movie_id=r.movie_id
join movie m
on m.id=r.movie_id
where genre='Thriller'
order by r.avg_rating desc
```

/* 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 |

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

| comdy | 145 | 106.2 | 128.42 |

| . | . | . | . |

| . | . | . | . |

| . | . | . | . |

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

-- Type your code below:

-- 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 |worldwide_gross_income|movie_rank |

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

| comedy | 2017 | indian | $103244842 | 1 |

| . | . | . | . |

| . | . | . | . |

| . | . | . | . |

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

-- Type your code below:

-- Top 3 Genres based on most number of movies

WITH TopGenres AS (

SELECT

```
    g.genre,
    SUM(CAST(REGEXP_REPLACE(m.worlwide_gross_income, '[^\d.]', '', 'g') AS NUMERIC)) AS
total_gross_income
  FROM
    genre g
  JOIN
    movie m ON m.id = g.movie_id
  WHERE
    m.worlwide_gross_income IS NOT NULL
  GROUP BY
    g.genre
  ORDER BY
    total_gross_income DESC
  LIMIT 3
),
TopMovies AS (
  SELECT
    m.title,
    EXTRACT(year FROM m.date_published) AS year_published,
    CAST(REGEXP_REPLACE(m.worlwide_gross_income, '[^\d.]', '', 'g') AS NUMERIC) AS
worlwide_gross_income,
    g.genre,
    ROW_NUMBER() OVER (
      PARTITION BY EXTRACT(year FROM m.date_published), g.genre
      ORDER BY CAST(REGEXP_REPLACE(m.worlwide_gross_income, '[^\d.]', '', 'g') AS NUMERIC)
DESC
    ) AS rank_within_year_genre
  FROM
    movie m
  JOIN
    genre g ON m.id = g.movie_id
  WHERE
```

```
    g.genre IN (SELECT genre FROM TopGenres)
    AND m.worlwide_gross_income IS NOT NULL
)
SELECT
  title,
  year_published,
  genre,
  worlwide_gross_income,
  rank_within_year_genre
FROM
  TopMovies
WHERE
  rank_within_year_genre <= 5
ORDER BY
  year_published, genre, rank_within_year_genre;
```

-- 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:



with Mutilingual as
            (select production_company,median_rating
from movie m
join ratings r
on m.id=r.movie_id
where production_company is not null
and median_rating>=8
and length(languages)-length(replace(languages,',',''))>0
order by median_rating desc
),
ProductionHits as
(
            select production_company,count(*) as hit_count
            from Mutilingual
            group by production_company
)
Select production_company,hit_count
from ProductionHits
order by hit_count desc
limit 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 Dern | 1016 | 1 | 9.60 | 1 |

| . | . | . | . | . | . |

| . | . | . | . | . | . |

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

-- Type your code below:

```
with Tophit as(
        select name_id,
        count(*) as hitcount
        from ratings r
   join role_mapping ro
   on r.movie_id=ro.movie_id
        join genre g
        on g.movie_id=ro.movie_id
        where avg_rating>8
```

```
and category like 'actress'
and genre like 'Drama'
group by name_id
order by hitcount
),
TopActress as(
        select n.name,t.hitcount
        from names n
        join Tophit t
        on n.id=t.name_id
)
Select name,hitcount
from TopActress
order by hitcount desc
limit 3
```

/* 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 DirectorMovies AS (
    SELECT
        dm.name_id,
        m.id AS movie_id,
        m.date_published,
        LAG(m.date_published) OVER (PARTITION BY dm.name_id ORDER BY m.date_published)
            AS prev_date_published
    FROM director_mapping dm
    JOIN movie m ON dm.movie_id = m.id
),
InterMovieDurations AS (
    SELECT
        name_id,
        EXTRACT(DAY FROM (date_published - prev_date_published)) AS inter_movie_duration
    FROM DirectorMovies
    WHERE prev_date_published IS NOT NULL
),
DirectorStats AS (
    SELECT
        dm.name_id,
        COUNT(dm.movie_id) AS movie_count,
        AVG(imd.inter_movie_duration) AS avg_inter_movie_duration,
        AVG(r.avg_rating) AS avg_movie_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_movie_duration
    FROM director_mapping dm
    JOIN movie m ON dm.movie_id = m.id
    JOIN ratings r ON m.id = r.movie_id
```

```
    LEFT JOIN InterMovieDurations imd ON dm.director_id = imd.director_id
    GROUP BY dm.director_id
    ORDER BY movie_count DESC
    LIMIT 9
)
SELECT
    ds.director_id,
    n.name,
    ds.movie_count,
    ds.avg_inter_movie_duration,
    ds.avg_movie_rating,
    ds.total_votes,
    ds.min_rating,
    ds.max_rating,
    ds.total_movie_duration
FROM DirectorStats ds
JOIN names n ON ds.director_id = n.id;
```