```
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:
select count(*) from director_mapping
select count(*) from genre
select count(*) from movie
select count(*) from names
select count(*) from ratings
select count(*) from role_mapping;

-- Q2. Which columns in the movie table have null values?
-- Type your code below:
select title,
	year ,
	date_published,
	duration,
	country,
	worlwide_gross_income,
```

```
	languages,
	production_company
from movie
where year is null
or title is null
or date_published is null
or country is null
or worlwide_gross_income is null
or languages is null
or production_company is null;
```

-- 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:
-- 1st part
select year, sum(title) as number_of_movies
from movie
group by year;

-- 2nd part
select month(date_published) as month_num,count(title) as number_of_movies
from movie
group by month_num
order by month_num;
```

/*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(date_published) as years ,count(title) as number_of_movies
from movie
where country = 'India' or country = 'USA'
group by years;

-- as USA was not in the result in 2019 to make sure about that i wrote this code
```

/* 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 genre

from genre

group by 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:

--- join is starting from

```
select count(id), genre

from movie m inner join genre g

on g.movie_id=m.id

group by 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(*) ,genre
from genre
group by 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.)

/* Output format:

+---------------+-------------------+
| genre                    |         avg_duration     |
+-------------------+----------------
|          thriller  |                  105                        |
|          .                          |                  .                           |
|          .                          |                  .                           |
+---------------+-------------------+ */
-- Type your code below:

select genre,avg(duration) as avg_duration
from movie m inner join genre g
on g.movie_id=m.id
group by genre;

/* 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)

/* Output format:

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

| genre | movie_count | genre_rank |

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

|drama | 2312 | 2 |

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

-- Type your code below:

```
select  genre, count(movie_id) as movie_count,
  row_number() over (order by count(movie_id) desc) as genre_rank
from movie m inner join genre g
on g.movie_id=m.id
group by genre;
```

/*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 as avg_rating,

 rank() over (order by avg_rating desc) as movie_rank

from movie m inner join ratings a

on a.movie_id=m.id

group by avg_rating;



/* 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.*/

-- FYI fan is not my favourite movie

-- 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(median_rating) as movie_count

from ratings
```

```
group by median_rating
order by median_rating;

/* 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(distinct id) as movie_count,
 rank() over (order by count(distinct id) desc) as prod_company_rank
from movie m inner join ratings r
on r.movie_id=m.id
where avg_rating > 8 and production_company is not null
group by production_company;

-- It's ok if RANK() or DENSE_RANK() is used too
-- Answer can be Dream Warrior Pictures or National Theatre Live or both
 -- No there are no dream warrior pictures or national theare or both in full list
-- 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:

select g.genre,count(id) as movie_count

from movie m inner join genre g on g.movie_id=m.id

inner join ratings r on g.movie_id=r.movie_id

where country = 'USA' and month(date_published) = '3' and year(date_published) = '2017' and total_votes > '1000'

group by genre

order by movie_count;

-- 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,r.avg_rating,g.genre

from movie m inner join genre g on g.movie_id=m.id

inner join ratings r on g.movie_id=r.movie_id

```
where title like 'The%' and avg_rating > 8

order by avg_rating desc;


-- 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 count(title) as movie_count

from movie m inner join genre g on g.movie_id=m.id

inner join ratings r on g.movie_id=r.movie_id

where median_rating = '8' and date_published between '2018-04-01' and '2019-04-01'


-- 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:


select languages,count(total_votes) as total_votes

from movie m inner join genre g on g.movie_id=m.id

inner join ratings r on g.movie_id=r.movie_id

where languages='german' or 'italian'

group by total_votes;


-- 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 name
from names
where name is null;

select id
from names
where id is null;

select count(height)
from names
where height is null;

select count(date_of_birth)
from names
where date_of_birth is null;

select count(known_for_movies)
from names
where known_for_movies is null;

/* 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 name as director_name ,count(known_for_movies) as movie_count
from names n inner join genre g on g.movie_id=n.known_for_movies
inner join ratings r on g.movie_id=r.movie_id
where avg_rating > '8'
group by director_name
order by movie_count desc;
```

/* 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 as actor_name,count(median_rating) as movie_count

from names n inner join role_mapping d on n.id = d.name_id

inner join ratings r  on r.movie_id = d.movie_id

where median_rating >= '8'

group by actor_name

order by movie_count desc;


/* 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 as production_company , total_votes as vote_count ,

rank() over (order by total_votes desc) as prod_comp_rank

from movie m inner join ratings r on r.movie_id = m.id

group by production_company

order by vote_count 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:

```
select name as actor_name ,
sum(total_votes) as total_votes ,
 count(m.id) as movie_count,
sum(avg_rating*total_votes)/sum(total_votes)  AS actor_avg_rating,
```

```
rank() over (order by sum(avg_rating*total_votes)/sum(total_votes) desc) as actor_rank
from movie m inner join
  ratings r on r.movie_id=m.id
inner join role_mapping t  on t.movie_id = m.id
inner join names n on t.name_id=n.id
where country = 'India'
group by actor_name
having movie_count >4;

-- 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:

select name as actress_name ,
```

```
sum(total_votes) as total_votes ,
 count(m.id) as movie_count,
sum(avg_rating*total_votes)/sum(total_votes)  AS actress_avg_rating,
rank() over (order by sum(avg_rating*total_votes)/sum(total_votes) desc) as actress_rank
from movie m inner join
  ratings r on r.movie_id=m.id
inner join role_mapping t  on t.movie_id = m.id
inner join names n on t.name_id=n.id
where country = 'India' and category = 'actress'
group by actress_name
having movie_count > 4;

/* 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
  genre,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'
        else 'Flop Movies'
          end as film_type
```

```
  from genre g inner join ratings r on g.movie_id = r.movie_id
  where genre = 'Thriller'
```

/* 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:

```
with duration as
(
select genre as genre,
 sum(duration)/count(duration) as avg_duration
from genre g
inner join
movie m on g.movie_id=m.id
group by genre
```

```
 )
 select * ,
  sum(avg_duration) over w1 as running_total_duration,
  avg(avg_duration) over w2 as moving_avg_duration
from duration
window w1 as (order by avg_duration rows unbounded preceding),
w2 as (order by avg_duration rows 13 preceding)
;

-- 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 grossing as
(
```

```
select genre, year(date_published) as year,
        title as movie_name ,
            worlwide_gross_income as worldwide_gross_income,
        rank() over (partition by year order by worlwide_gross_income  desc) as movie_rank
                        from genre g inner join movie m on g.movie_id = m.id
                        where genre in ('Drama','comedy','thriller')
 )
 select * from grossing
 where movie_rank < 6;
```

-- 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 production as
(
select production_company as production_company , count(id) as movie_count,
row_number() over (order by count(id) desc) as prod_comp_rank
from movie m inner join ratings r on r.movie_id = m.id
where languages like '%,%' and median_rating > 8 and production_company is not null
group by production_company
```

```
)
select * from production
where prod_comp_rank < 3;
-- 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 actress as (
select name as actress_name ,
sum(total_votes) as total_votes ,
 count(m.id) as movie_count,
sum(avg_rating*total_votes)/sum(total_votes)  AS actress_avg_rating,
row_number() over (order by sum(avg_rating*total_votes)/sum(total_votes) desc) as actress_rank
from movie m inner join
   ratings r on r.movie_id=m.id
inner join role_mapping t  on t.movie_id = m.id
inner join names n on t.name_id=n.id
inner join genre g on g.movie_id = t.movie_id
where category = 'actress' and avg_rating > 8 and genre = 'drama'
```

```
group by actress_name
)
select * from actress
where actress_rank < 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

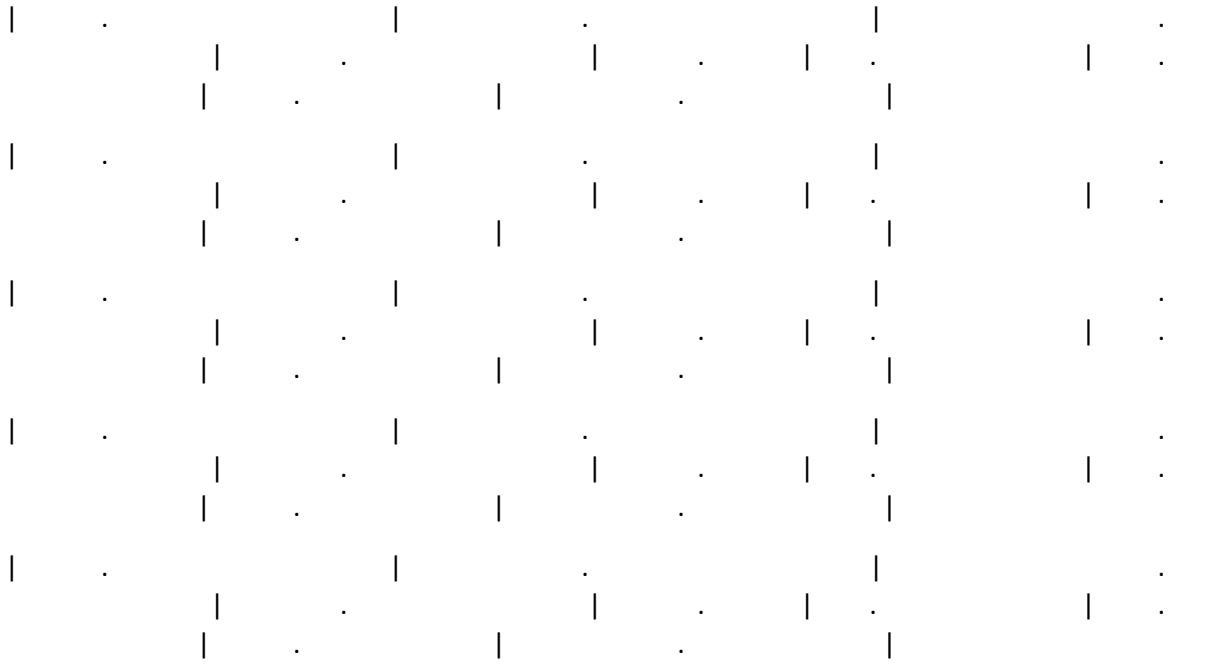
Min rating

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:

drop view cad;

create view cad as

select

n.name,date_published

from movie m inner join director_mapping dm on dm.movie_id = m.id

inner join names n on dm.name_id=n.id

order by n.name;

drop view next_cad1;

create view next_cad1 as

select *,

lag(date_published,1,'2017-06-01') over (partition by name order by date_published) as
previous_movie

from cad;

```
drop view final2;

create view final2 as
select *,
    datediff(date_published,previous_movie) as nextmovdiff
from next_cad1;

select dm.name_id as director_id,
n.name as director_name,
count(distinct(m.id)) as number_of_movies,
avg(nextmovdiff) as avg_inter_movie_days,
avg(avg_rating) 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

from movie m inner join genre g
on g.movie_id = m.id
inner join ratings r on r.movie_id = m.id
inner join director_mapping dm on dm.movie_id=m.id
inner join names n on dm.name_id = n.id
inner join final2 f1 on f1.name=n.name

group by n.name
order by count(distinct(m.id)) desc
limit 9;
```