```
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 table_name, table_rows

   FROM INFORMATION_SCHEMA.TABLES

   WHERE TABLE_SCHEMA = 'imdb';



-- 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_nulls,
                    SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS title_nulls,
                    SUM(CASE WHEN year IS NULL THEN 1 ELSE 0 END) AS year_nulls,
                    SUM(CASE WHEN date_published IS NULL THEN 1 ELSE 0 END) AS
date_published_nulls,
                    SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS duration_nulls,
                    SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS country_nulls,
                    SUM(CASE WHEN worlwide_gross_income IS NULL THEN 1 ELSE 0 END) AS
worlwide_gross_income_nulls,
                    SUM(CASE WHEN languages IS NULL THEN 1 ELSE 0 END) AS languages_nulls,
                    SUM(CASE WHEN production_company IS NULL THEN 1 ELSE 0 END) AS
production_company_nulls


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)

-- Number of movies released each year.

SELECT year, COUNT(id) as number_of_movies

FROM movie

GROUP BY year

ORDER BY year;


SELECT year,COUNT(title)
```

```
from movie
group by year
ORDER BY year;


/* Output format for the first part:

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

SELECT *
FROM movie;

Output format for the second part of the question:
+---------------+-------------------+
|           month_num      |           number_of_movies|
+---------------+----------------
|           1                              |           134                       |
|           2                              |           231                       |
|           .                              |                     .                              |
+---------------+-------------------+ */
-- Type your code below:

SELECT *
FROM movie;
```

```
-- Number of movies released each month.

SELECT MONTH(date_published) AS month_num, COUNT(id) AS number_of_movies

FROM movie

GROUP BY MONTH(date_published)

ORDER BY MONTH(date_published);


SELECT DATE_FORMAT(date_published, '%m') AS month_num,count(title)

FROM movie

GROUP 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 *

FROM movie;


SELECT year,COUNT(title) AS TOTAL_NO_MOVIE

FROM movie

WHERE country='USA' OR country='India'

GROUP by country
```

```sql
HAVING year=2019;


SELECT COUNT(id) AS number_of_movies, year

FROM movie

WHERE country = 'USA' OR country = 'India'

GROUP BY country

HAVING year=2019;
```

```sql
/* 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

ORDER 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:

SELECT *

FROM genre;

SELECT *

FROM movie;


SELECT genre,COUNT(id) AS number_of_movie

FROM movie AS m

INNER JOIN genre AS g

ON m.id=g.movie_id

GROUP BY genre

ORDER BY count(id) DESC

LIMIT 1;


SELECT genre, year, COUNT(movie_id) AS number_of_movies

FROM genre AS g

INNER JOIN movie AS m

ON g.movie_id = m.id

WHERE year = 2019

GROUP BY genre

ORDER BY number_of_movies DESC

LIMIT 1;
```

/* 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:


WITH NUMBER_genre AS

(

SELECT count(genre)AS number_of_genre, title, movie_id

FROM genre AS g

INNER JOIN movie AS m

ON g.movie_id = m.id

GROUP BY movie_id

)

SELECT count(number_of_genre)

FROM NUMBER_genre

WHERE number_of_genre=1

;


WITH ct_genre AS

(

        SELECT movie_id,
                    COUNT(genre) AS number_of_movies
        FROM genre
        GROUP BY movie_id
        HAVING number_of_movies=1
)

SELECT COUNT(movie_id) AS number_of_movies
FROM ct_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.)
SELECT genre,AVG(duration)
FROM movie AS m
INNER JOIN genre AS g
ON m.id=g.movie_id
GROUP BY  genre
ORDER BY genre DESC;


/* Output format:

+---------------+-------------------+
| genre                    |           avg_duration     |
+-------------------+----------------
|          thriller  |                      105                        |
|          .                              |                    .                              |
|          .                              |                    .                              |
+---------------+-------------------+ */
```

-- Type your code below:

/* 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),
     RANK() OVER(ORDER BY COUNT(movie_id) DESC) AS rank_m

FROM movie as m
INNER JOIN genre AS g
ON m.id=g.movie_id
GROUP BY genre;



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';
```

```
/*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 min_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,

     DENSE_RANK() OVER(ORDER BY avg_rating DESC) AS rank_Avg,

     ROW_NUMBER() OVER(ORDER BY avg_rating DESC) AS ROW_no

FROM ratings AS r

INNER JOIN movie AS m
```

```
ON r.movie_id=m.id

LIMIT 10;
```

/* 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(title) AS movie_count
FROM movie AS m
INNER jOIN ratings AS r
ON m.id=r.movie_id
GROUP BY median_rating
ORDER BY COUNT(title) DESC;
```

```
/* 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:
WITH highest_prod_company AS
(
SELECT avg_rating,production_company,title
FROM movie AS m
INNER JOIN ratings AS r
ON m.id=r.movie_id
WHERE avg_rating>8
ORDER BY avg_rating
)
SELECT production_company,COUNT(title),
                        DENSE_RANK() OVER(ORDER BY COUNT(title) DESC) AS no_pro_comp
FROM highest_prod_company
WHERE production_company IS NOT NULL
GROUP BY production_company;


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;
```

```
-- 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 title,avg_rating,genre

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 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:

```
USE imdb;



SELECT COUNT(id) as movie_count,median_rating

FROM movie as m

INNER JOIN ratings AS r

ON m.id=r.movie_id

WHERE date_published between '2018-04-01' AND '2019-04-01'

GROUP BY median_rating

HAVING median_rating=8;
```

--

```
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;

SELECT title;
```

```
-- 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,SUM(total_votes) AS total

FROM movie AS m

INNER JOIN ratings AS r

ON m.id=r.movie_id

WHERE languages='German' OR languages='Italian'

GROUP BY languages

ORDER BY SUM(total_votes) DESC;
```

-- 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 name IS NULL THEN 1 ELSE 0 END) AS name_nulls,
		SUM(CASE WHEN height IS NULL THEN 1 ELSE 0 END) AS height_nulls,
		SUM(CASE WHEN date_of_birth IS NULL THEN 1 ELSE 0 END) AS date_of_birth_nulls,
		SUM(CASE WHEN known_for_movies IS NULL THEN 1 ELSE 0 END) AS
known_for_movies_nulls
```

```
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:

```
WITH top_genre AS
(
        SELECT g.genre, COUNT(g.movie_id) AS movie_count
        FROM genre AS g
```

```
            INNER JOIN ratings AS r
            ON g.movie_id = r.movie_id
            WHERE avg_rating > 8
    GROUP BY genre
    ORDER BY movie_count
    LIMIT 3
),

top_director AS
(
SELECT n.name AS director_name,
                        COUNT(g.movie_id) AS movie_count,
       ROW_NUMBER() OVER(ORDER BY COUNT(g.movie_id) DESC) AS director_row_rank
FROM names AS n
INNER JOIN director_mapping AS dm
ON n.id = dm.name_id
INNER JOIN genre AS g
ON dm.movie_id = g.movie_id
INNER JOIN ratings AS r
ON r.movie_id = g.movie_id,
top_genre
WHERE g.genre in (top_genre.genre) AND avg_rating>8
GROUP BY director_name
ORDER BY movie_count DESC
)

SELECT *
FROM top_director
WHERE director_row_rank <= 3
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 median_rating, category, name,count(r.movie_id) AS total
FROM ratings AS r
INNER JOIN role_mapping AS rm
ON r.movie_id=rm.movie_id
INNER JOIN names AS n
ON n.id=rm.name_id
WHERE median_rating>=8 AND category='actor'
GROUP BY name
ORDER BY count(r.movie_id) DESC
limit 2;
--------------------
SELECT * FROM role_mapping;
SELECT * FROM movie;
SELECT * FROM names;

SELECT * FROM ratings;

-------------------------------


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;


/* 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 * FROM ratings;

SELECT * FROM movie;

SELECT  production_company,

                    SUM(total_votes) AS vote_count,

                    DENSE_RANK() OVER(ORDER BY SUM(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;

------------------------------------------------------------------------------------------

SELECT production_company, SUM(total_votes) AS vote_count,

                    DENSE_RANK() OVER(ORDER BY SUM(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;

----------------------------------------------------------------------
```

/*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 ,AVG(avg_rating) AS actor_avg_rating,
                      count(m.id) AS movie_count,
                      total_votes,
       RANK() OVER(ORDER BY AVG(avg_rating)  DESC) AS actor_rank
FROM movie AS m
INNER JOIN ratings AS r
ON m.id=r.movie_id
```

```
INNER JOIN role_mapping AS rm
ON  m.id=rm.movie_id
INNER JOIN names AS n
ON rm.name_id=n.id
WHERE category='actor' AND country='India'
GROUP BY name
HAVING count(m.id)>=5;

-------
SELECT * FROM movie;
SELECT * FROM ratings;
SELECT * FROM role_mapping;
SELECT * FROM names;



SELECT name AS actor_name, total_votes,
        COUNT(m.id) as movie_count,
        ROUND(SUM(avg_rating*total_votes)/SUM(total_votes),2) AS actor_avg_rating,
        RANK() OVER(ORDER BY avg_rating DESC) AS actor_rank

FROM movie AS m
INNER JOIN ratings AS r
ON m.id = r.movie_id
INNER JOIN role_mapping AS rm
ON m.id=rm.movie_id
INNER JOIN names AS nm
ON rm.name_id=nm.id
WHERE category='actor' AND country= 'india'
GROUP BY name
HAVING COUNT(m.id)>=5
LIMIT 1;
```

-- 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 actor_name, total_votes,
        COUNT(m.id) as movie_count,
        ROUND(SUM(avg_rating*total_votes)/SUM(total_votes),2) AS actor_avg_rating,
        RANK() OVER(ORDER BY avg_rating DESC) AS actor_rank
```

```
FROM movie AS m

INNER JOIN ratings AS r

ON m.id = r.movie_id

INNER JOIN role_mapping AS rm

ON m.id=rm.movie_id

INNER JOIN names AS nm

ON rm.name_id=nm.id

WHERE category='actoress' AND country= 'india'

GROUP BY name

HAVING COUNT(m.id)>=3

LIMIT 1;
```