```
USE imdb;

-- 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';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

TABLE_NAME	TABLE_ROWS
director_mapping	3867
genre	14662
movie	7258
names	24499
ratings	8230
role_mapping	16551

```
-- 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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```
-- Now as you can see four columns of the movie table has null values.
Let's look 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:
```

```sql
SELECT year, COUNT(year) AS number_of_movies
FROM movie
GROUP BY year;

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;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

month_num	number_of_movies
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 the USA and India produce a huge number of movies each year. Let's find the number of movies produced by the 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:

```
11 • select year,country,count(country) as Movie_count from movie
12   where country in ('USA','India')
13   group by country
14
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	country	Movie_count
2019	India	1007
2018	USA	2260

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

```
15 • select distinct(genre) from genre;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

/* 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:

```
17  select m.year,g.genre,count(g.genre) from movie m inner join genre g
18  on (m.id=g.movie_id)
19  group by genre
20  order by year;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

year	genre	count(g.genre)
2017	Drama	4285
2017	Fantasy	342
2017	Thriller	1484
2017	Comedy	2412
2017	Family	302
2017	Romance	906
2017	Adventure	591
2018	Horror	1208
2018	Action	1289
2018	Sci-Fi	375
2019	Crime	813
2019	Mystery	555
2019	Others	100

/* 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:
```

```
280
281 • select count(id) from movie where id in (select movie_id from genre group by movie_id having count(genre)=1)
282
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

count(id)
3289

```
/* 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:
```

```
25 • select avg(m.duration),g.genre from movie m inner join genre g
26   on (m.id=g.movie_id)
27   group by genre;
28
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

avg(m.duration)	genre
106.7746	Drama
105.1404	Fantasy
101.5761	Thriller
102.6227	Comedy
92.7243	Horror
100.9669	Family
109.5342	Romance
101.8714	Adventure
112.8829	Action
97.9413	Sci-Fi

```
/* 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:
```

```
36 • with a as(select genre,count(genre),rank() over(order by count(genre) desc )
37   from genre
38   group by genre)
39   select * from a
40   where genre='thriller'
41
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	count(genre)	rank() over(order by count(genre) desc)
Thriller	1484	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?
/* 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:
```

```
92 • select min(avg_rating),min(total_votes),min(median_rating),
93   max(avg_rating),max(total_votes),max(median_rating) from ratings;
94
```

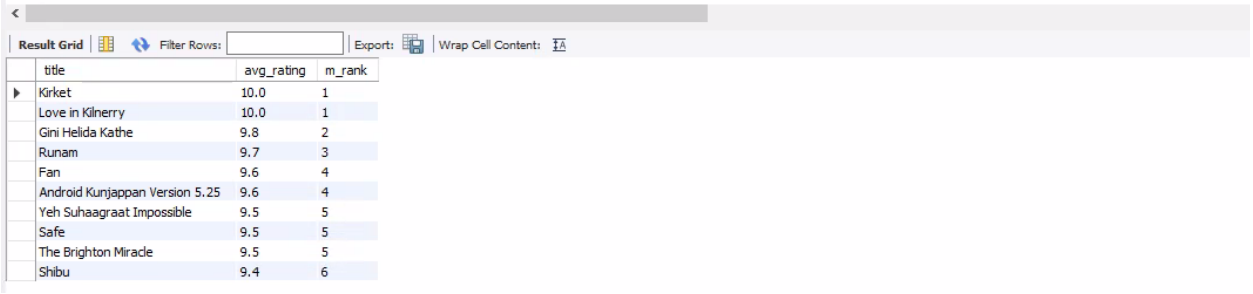
Result Grid | Filter Rows: | Export: | Wrap Cell Content:

min(avg_rating)	min(total_votes)	min(median_rating)	max(avg_rating)	max(total_votes)	max(median_rating)
1.0	100	1	10.0	725138	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?
/* 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
```

```
87 • select m.title,r.avg_rating,
88   dense_rank() over(order by r.avg_rating desc) as m_rank from movie m
89   inner join ratings r on (r.movie_id=m.id)
90   limit 10
91
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

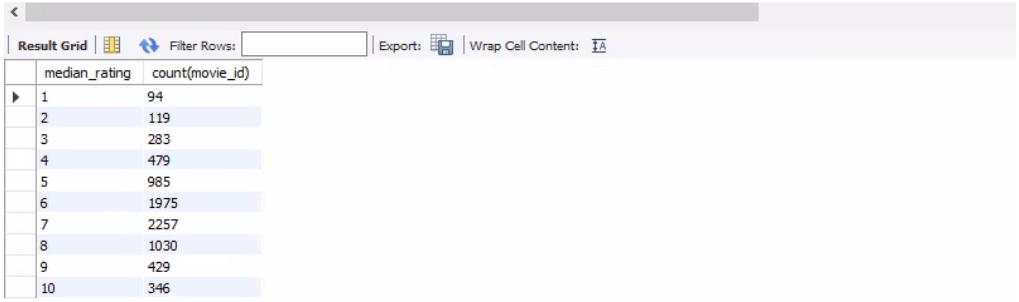
title	avg_rating	m_rank
Kirket	10.0	1
Love in Kilnerry	10.0	1
Gini Helida Kathe	9.8	2
Runam	9.7	3
Fan	9.6	4
Android Kunjappan Version 5.25	9.6	4
Yeh Suhaagraat Impossible	9.5	5
Safe	9.5	5
The Brighton Miracle	9.5	5
Shibu	9.4	6

```
/* 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
```

```
95 • select median_rating,count(movie_id) from ratings
96   group by median_rating
97   order by median_rating
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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

```
/* 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 m.production_company,count(m.title),
rank() over(order by count(m.title) desc) from movie m
inner join ratings r on (m.id=r.movie_id)
where r.avg_rating > 8;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

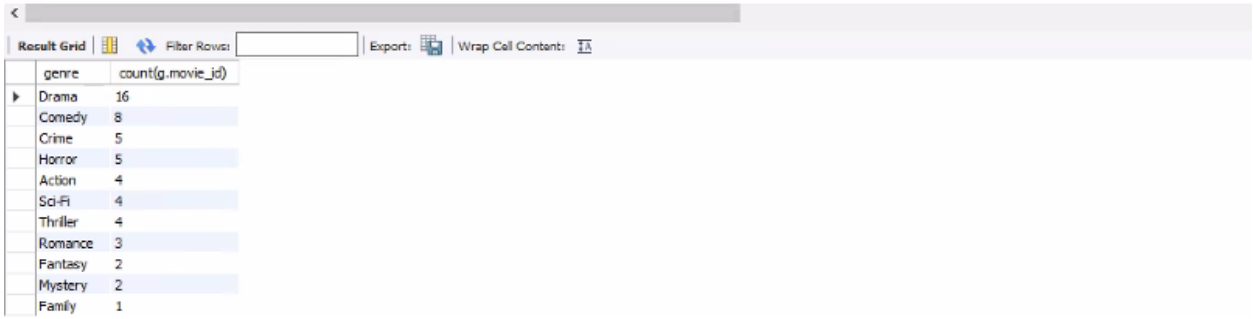
production_company	count(m.title)	rank() over(order by count(m.title) desc)
NULL	21	1
Dream Warrior Pictures	3	2
National Theatre Live	3	2
Lietuvos Kinostudija	2	4
Swadharm Entertainment	2	4
Panorama Studios	2	4
Marvel Studios	2	4
Central Base Productions	2	4
Painted Creek Productions	2	4
National Theatre	2	4
Colour Yellow Productions	2	4

```
-- 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:
```

```
select g.genre,count(g.movie_id) from genre as g
inner join movie as m on (g.movie_id=m.id)
inner join ratings as r on(m.id=r.movie_id)
where (m.date_published between '2017-03-01' and '2017-03-31')
and (m.country='USA') and (r.total_votes>1000)
group by g.genre
order by count(g.movie_id) desc ;
```



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	count(g.movie_id)
Drama	16
Comedy	8
Crime	5
Horror	5
Action	4
Sci-Fi	4
Thriller	4
Romance	3
Fantasy	2
Mystery	2
Family	1

```
-- 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:
```

```
14 • select m.title,r.avg_rating,g.genre from genre as g
15   inner join movie as m on (g.movie_id=m.id)
16   inner join ratings as r on(m.id=r.movie_id)
17   where (m.title like 'The%') and (r.avg_rating>8)
18   group by genre;
19
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

title	avg_rating	genre
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
Theeran Adhigaaram Ondru	8.3	Action
Theeran Adhigaaram Ondru	8.3	Thriller
The King and I	8.2	Romance

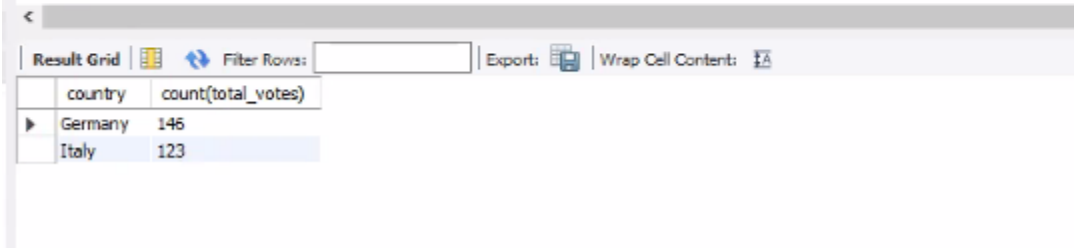
```
-- You should also try your hand at median rating and check whether the
'median rating' column gives any significant insights.
-- Q16. Of the movies released between 1 April 2018 and 1 April 2019, how
many were given a median rating of 8?
-- Type your code below:
```

```
select count(g.movie_id),g.movie_id from movie m
inner join genre g on (g.movie_id=m.id)
inner join ratings r on (r.movie_id=g.movie_id)
where (m.date_published between '2018-04-01' and '2019-04-01' )
and (r.median_rating=8)
group by genre
```

count(g.movie_id)	movie_id
89	tt0060908
253	tt0060908
31	tt0083907
31	tt0352314
15	tt0352314
59	tt0437086
20	tt0437086
18	tt0437086
44	tt0862930
13	tt10034272
58	tt1289403
7	tt2967856

```
-- 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 country, count(total_votes) from movie m
inner join ratings r on (m.id=r.movie_id)
where country in ('Germany','Italy')
group by country;
```



country	count(total_votes)
Germany	146
Italy	123

```
-- 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:
```

```
79  select count(*)-count(name) as name_nulls,
80   count(*)-count(height) as height_nulls,
81   count(*)-count(date_of_birth) as date_of_birth_nulls,
82   count(*)-count(known_for_movies) as known_for_movies_nulls
83   from names
84
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

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:

```
284 • with top3genre as (select g.genre from movie m join ratings r on m.id= r.movie_id
285   join genre g on m.id=g.movie_id where avg_rating >8
286   group by genre order by count(title) desc limit 3)
287   select name as "Director_name",count(g2.movie_id) as "movie_count"  from director_mapping dm
288   join names n on n.id=dm.name_id join genre g2 on g2.movie_id=dm.movie_id join ratings r on r.movie_id=dm.movie_id
289   join top3genre tg on g2.genre=tg.genre where r.avg_rating>8 group by name order by count(dm.movie_id)desc limit 3;
290
291
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Director_name	movie_count
James Mangold	4
Anthony Russo	3
Soubin Shahir	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:
```

```
38 • SELECT DISTINCT name, COUNT(r.movie_id) FROM ratings AS r
39   INNER JOIN role_mapping AS rm ON (rm.movie_id = r.movie_id)
40   INNER JOIN names AS n ON (rm.name_id = n.id)
41   where (r.median_rating>=8) and rm.category='actor'
42   group by name
43   order by r.movie_id
44   limit 2;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

name	COUNT(r.movie_id)
David Niven	1
Robert Coote	1

```
/* 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:
```

```
11 • select m.production_company,r.total_votes,
12   dense_rank() over(order by r.total_votes desc) as pro_rank from movie m
13   inner join ratings r on (m.id=r.movie_id)
14   group by m.production_company
15   limit 3
16
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

production_company	total_votes	pro_rank
Marvel Studios	551245	1
Syncopy	487517	2
New Line Cinema	408221	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:
```

```
34 •  select name ,total_votes ,count(m.id) as m_count,
35    SUM(r.avg_rating*r.total_votes)/SUM(r.total_votes) AS a_avg_rating,
36    dense_rank() over(order by avg_rating desc, total_votes desc) as actor_rank
37    from movie m join ratings r on m.id=r.movie_id join role_mapping rm on rm.movie_id= m.id
38    join names n on rm.name_id=n.id
39    where m.country='India' and rm.category='actor'
40    group by name
41    having count(m.title)>=5;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	total_votes	m_count	a_avg_rating	actor_rank
Vijay Sethupathi	20364	5	8.41673	1
Kunchacko Boban	3684	6	7.48351	2
Fahadh Faasil	3684	5	7.98604	2
Nassar	508	5	7.03312	3
Jimmy Sheirgill	821	6	6.28772	4
Pankaj Tripathi	13723	5	7.43706	5
Rajkummar Rao	8320	6	7.36701	6
Mohanlal	5471	6	6.50840	7
Hareesh Kanaran	817	5	6.57747	8
Amit Sadh	476	5	7.21306	9
Ammy Virk	169	6	7.55383	10
Dulquer Salmaan	2432	5	7.30087	11

```
-- 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:
```

```sql
select name ,total_votes ,count(m.id) as m_count,
SUM(r.avg_rating*r.total_votes)/SUM(r.total_votes) AS a_avg_rating,
dense_rank() over(order by avg_rating desc, total_votes desc) as actress_rank
from movie m join ratings r on m.id=r.movie_id join role_mapping rm on rm.movie_id= m.id
join names n on rm.name_id=n.id
where m.country='India' and rm.category='actress'
group by name
having count(m.title)>=3;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

name	total_votes	m_count	a_avg_rating	actress_rank
Anusree	1017	4	7.56253	1
Anupama Parameshwaran	613	3	7.41190	2
Regina Cassandra	1793	4	7.74640	3
Nirmal Rishi	552	4	7.29145	4
Kajal Aggarwal	22704	3	7.13767	5
Aparna Balamurali	1197	3	7.31803	6
Kriti Sanon	14978	3	7.04911	7
Manju Warrier	5471	5	6.76135	8
Rachita Ram	242	3	6.95269	9
Raashi Khanna	3746	5	7.00515	10
Nivetha Pethuraj	365	3	7.58202	11
Neeru Bajwa	117	3	6.34000	12
Sohini Sarkar	272	3	6.23900	13
Varalaxmi Sarathkumar	181	3	6.70343	14

/* 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:
```

```sql
select m.title,
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
from movie m
inner join genre g on (m.id=g.movie_id)
inner join ratings r on (g.movie_id=r.movie_id)
where g.genre='thriller';
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

title	avg_Rating
Der müde Tod	Hit movies
Fahrenheit 451	Flop movies
Pet Sematary	One-time-watch movies
Dukun	One-time-watch movies
Back Roads	Hit movies
Countdown	One-time-watch movies
Staged Killer	Flop movies
Vellaipookal	Hit movies
Uriyadi 2	Hit movies
Incitement	Hit movies
Rakshasudu	Superhit movies
Trois jours et ...	One-time-watch movies

```
/* 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:
```

```
63 •  select genre,avg(duration), sum(avg(duration)) over (partition by genre order by genre),
64    avg(avg(duration)) over(order by genre) from movie m
65    inner join genre g on (m.id=g.movie_id)
66    group by genre
67
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	avg(duration)	sum(avg(duration)) over (partition by genre order by genre)	avg(avg(duration)) over(order by genre)
Action	112.8829	112.8829	112.88290000
Adventure	101.8714	101.8714	107.37715000
Comedy	102.6227	102.6227	105.79233333
Crime	107.0517	107.0517	106.10717500
Drama	106.7746	106.7746	106.24066000
Family	100.9669	100.9669	105.36170000
Fantasy	105.1404	105.1404	105.33008571
Horror	92.7243	92.7243	103.75436250
Mystery	101.8000	101.8000	103.53721111
Others	100.1600	100.1600	103.19949000
Romance	109.5342	109.5342	103.77537273

-- 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:
```

```
with top_3 as(
select g.genre,count(r.movie_id) from genre g
inner join ratings r on (g.movie_id=r.movie_id)
group by g.genre
order by count(r.movie_id) desc
limit 3),
top5 as (
select g.genre,m.title,m.year,m.worlwide_gross_income,
dense_rank() over(partition by m.year order by m.worlwide_gross_income desc) as movie_rank from movie m
inner join genre g on (g.movie_id=m.id)
where g.genre in (select genre from top_3)
) select * from top5 where movie_rank<=5
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

genre	title	year	worlwide_gross_income	movie_rank
Drama	Shatamanam Bhavati	2017	INR 530500000	1
Drama	Winner	2017	INR 250000000	2
Drama	Thank You for Your Service	2017	$ 9995692	3
Comedy	The Healer	2017	$ 9979800	4
Drama	The Healer	2017	$ 9979800	4
Thriller	Gi-eok-ui bam	2017	$ 9968972	5
Thriller	The Villain	2018	INR 1300000000	1
Drama	Antony & Cleopatra	2018	$ 998079	2

```
-- Top 3 Genres based on most number of movies
```

-- 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:

```sql
select m.production_company, count(m.id) ,dense_rank() over(order by r.movie_id desc) as pro_rank from movie m
inner join ratings r on(r.movie_id=m.id)
where r.median_rating>=8 and position(',' in languages) >0 and production_company is not null
group by production_company
limit 2
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

production_company	count(m.id)	pro_rank
Jorkwang Films	1	1
MF Production	1	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:
```

```
253 •  select name as "actress_name",sum(total_votes),count(r.movie_id) as movie_count,
254    avg_rating as "actress_avg_rating", dense_rank() over(order by count(r.movie_id) desc) as actress_rank
255    from names n join role_mapping rm on n.id=rm.name_id
256    inner join ratings r on r.movie_id=rm.movie_id
257    inner join genre g on r.movie_id=g.movie_id where category="actress"
258    and avg_rating>8 and genre='drama' group by name limit 3;
259
260
261
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

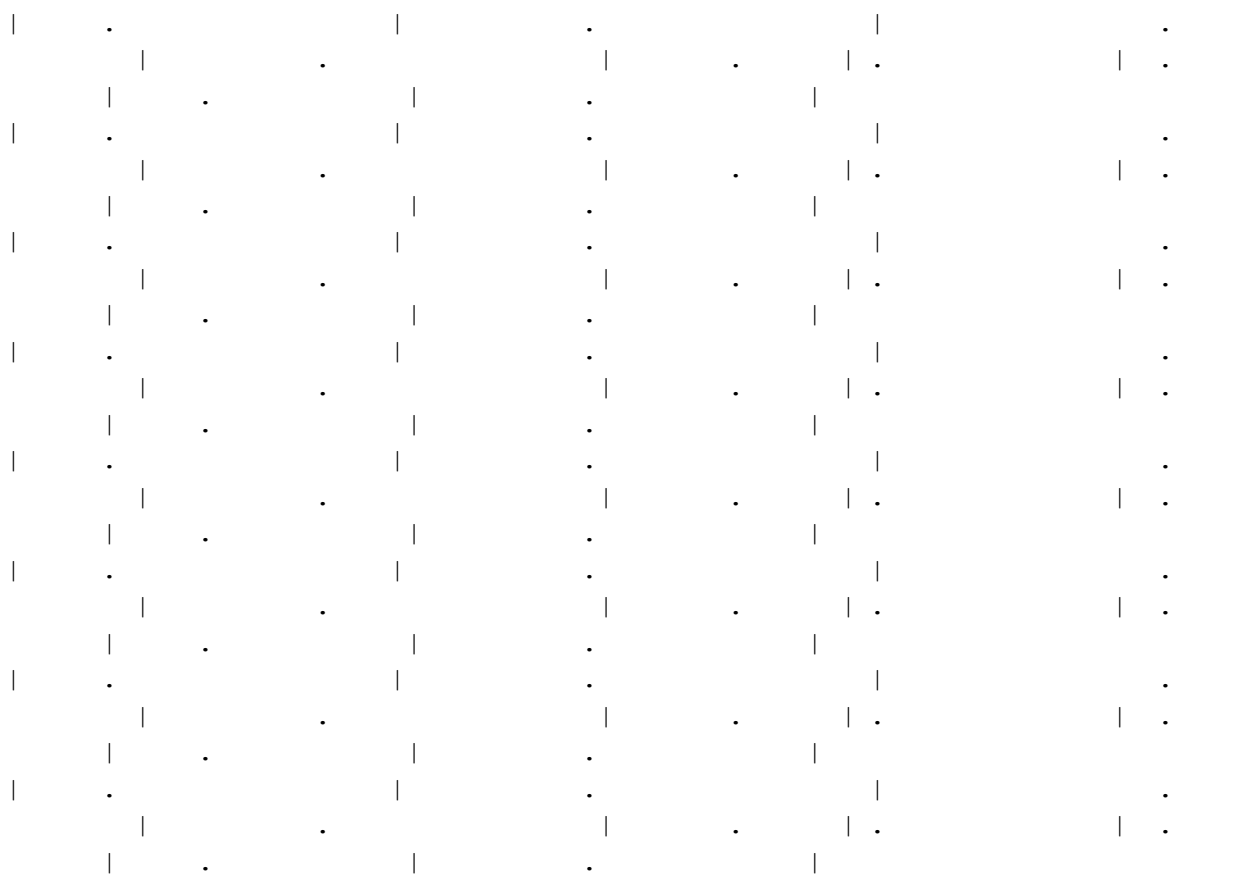
actress_name	sum(total_votes)	movie_count	actress_avg_rating	actress_rank
Parvathy Thiruvothu	4974	2	8.3	1
Susan Brown	656	2	8.9	1
Amanda Lawrence	656	2	8.9	1

/* 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:
```

```
261 • select name_id as Director_id ,name,count(r.movie_id) as Number_of_movies,
262   avg_rating as Average_movie_ratings,total_votes as Total_votes ,min(avg_rating) Min_rating,
263   max(avg_rating) Max_rating,duration total_movie_durations from movie m join ratings r on m.id=r.movie_id
264    join director_mapping dm on dm.movie_id=r.movie_id join names n on n.id=dm.name_id group by name
265    order by count(r.movie_id) desc limit 9;
266
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: | Fetch rows:

Director_id	name	Number_of_movies	Average_movie_ratings	Total_votes	Min_rating	Max_rating	total_movie_durations
nm1777967	A.L. Vijay	5	5.5	555	3.7	6.9	141
nm2096009	Andrew Jones	5	3.0	508	2.7	3.2	84
nm0831321	Chris Stokes	4	4.2	1787	4.0	4.6	81
nm2691863	Justin Price	4	4.6	586	3.0	5.8	80
nm0425364	Jesse V. Johnson	4	4.2	129	4.2	6.5	94
nm0001752	Steven Soderbergh	4	7.0	110979	6.2	7.0	118
nm0814469	Sion Sono	4	6.4	561	5.4	6.4	151
nm6356309	Özgür Bakar	4	4.9	371	3.1	4.9	93