# **SQL IMDB Movies Analysis for RSVP**

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

SELECT table\_name, table\_rows FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_SCHEMA = 'imdb';

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

describe MOVIE;

describe GENRE;

SELECT 'ID', COUNT(\*) AS Null\_Count FROM MOVIE WHERE ID IS NULL UNION SELECT 'Title', COUNT(\*) AS Null\_Count FROM MOVIE WHERE TITLE IS NULL UNION SELECT 'Year', COUNT(\*) AS Null\_Count FROM MOVIE WHERE YEAR IS NULL UNION SELECT 'Date Published', COUNT(\*) AS Null\_Count FROM MOVIE WHERE DATE\_PUBLISHED IS NULL UNION SELECT 'Movie', COUNT(\*) AS Null\_Count FROM MOVIE WHERE DURATION IS NULL UNION SELECT 'Country', COUNT(\*) AS Null\_Count FROM MOVIE WHERE COUNTRY IS NULL UNION SELECT 'WorldWide\_Gross', COUNT(\*) AS Null\_Count FROM MOVIE WHERE WORLWIDE\_GROSS\_INCOME IS NULL UNION SELECT 'Languages', COUNT(\*) AS Null\_Count FROM MOVIE WHERE LANGUAGES IS NULL UNION SELECT 'Prod Company', COUNT(\*) AS Null\_Count FROM MOVIE WHERE PRODUCTION\_COMPANY IS NULL;

-- country, worlwide\_gross\_income, languages and production\_company columns have NULL values. -- Now as we 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? .

-- Number of movies released anually.

SELECT Year, COUNT(TITLE) AS 'number\_of\_movies' FROM MOVIE GROUP BY YEAR;

-- Number of movies released every month.

SELECT MONTH(DATE\_PUBLISHED) AS'month\_num', COUNT(TITLE) AS 'number\_of\_movies' FROM MOVIE GROUP BY MONTH(DATE\_PUBLISHED) ORDER BY COUNT(TITLE) DESC;

/\*The highest number of movies is produced in the month of March. So, now that we 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?? SELECT year, COUNT(TITLE) AS number\_of\_movies FROM MOVIE WHERE YEAR = 2019 AND ( COUNTRY LIKE '%USA%' OR COUNTRY LIKE '%India%' ) GROUP BY YEAR; /\* USA and India produced more than a thousand movies(we 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? SELECT DISTINCT genre FROM GENRE; /\* So, RSVP Movies plans to make a movie of one of these genres. Now, wouldn’t we 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? SELECT g.genre, COUNT(m.TITLE) AS no\_of\_movies FROM MOVIE m INNER JOIN GENRE g ON g.MOVIE\_ID = m.ID GROUP BY g.GENRE ORDER BY COUNT(m.TITLE) DESC LIMIT 1; /\* So, based on the insight that we 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? WITH AGG AS (SELECT m.ID, Count(g.GENRE) AS Genre FROM MOVIE m INNER JOIN GENRE g ON g.MOVIE\_ID = m.ID GROUP BY ID HAVING Count(g.GENRE) = 1) SELECT Count(ID) AS movie\_count FROM AGG; /\* 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 | | . | . | | . | . | +---------------+-------------------+ \*/ SELECT g.genre, ROUND(AVG(m.DURATION), 2) AS avg\_duration FROM MOVIE m INNER JOIN GENRE g ON g.MOVIE\_ID = m.ID GROUP BY g.GENRE ORDER BY ROUND(AVG(m.DURATION), 2) DESC; /\* Now we 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 | +---------------+-------------------+---------------------+\*/ WITH GENRE\_RANKS 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\_RANKS WHERE GENRE = 'thriller'; /\*Thriller movies is in top 3 among all genres in terms of number of movies In the previous segment, we analysed the movies and genres tables. In this segment, we 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 | +---------------+-------------------+---------------------+----------------------+-----------------+-----------------+\*/ SELECT ROUND(MIN(AVG\_RATING), 1) AS min\_avg\_rating, ROUND(MAX(AVG\_RATING), 1) 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 | | . | . | . | | . | . | . | | . | . | . | +---------------+-------------------+---------------------+\*/ SELECT M.title, R.avg\_rating, RANK() OVER(ORDER BY R.AVG\_RATING DESC) AS movie\_rank FROM RATINGS R INNER JOIN MOVIE M ON R.MOVIE\_ID=M.ID ORDER BY R.AVG\_RATING DESC LIMIT 10; /\* Do we find our 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 we know the top 10 movies, do we 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 | | . | . | | . | . | +---------------+-------------------+ \*/ -- Order by is good to have SELECT median\_rating, COUNT(MOVIE\_ID) AS movie\_count FROM RATINGS GROUP BY MEDIAN\_RATING ORDER BY COUNT(MOVIE\_ID) 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 AGG AS ( SELECT M.production\_company, M.ID, R.AVG\_RATING FROM MOVIE M INNER JOIN RATINGS R ON M.ID=R.MOVIE\_ID WHERE AVG\_RATING>8 ORDER BY R.AVG\_RATING DESC ) SELECT production\_company, COUNT(ID) AS movie\_count, RANK() OVER (ORDER BY COUNT(ID) DESC) AS prod\_company\_rank FROM AGG WHERE PRODUCTION\_COMPANY IS NOT NULL GROUP BY PRODUCTION\_COMPANY ORDER BY MOVIE\_COUNT DESC LIMIT 2; -- 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 | | . | . | | . | . | +---------------+-------------------+ \*/ WITH AGG AS (SELECT g.genre, r.MOVIE\_ID, m.DATE\_PUBLISHED, m.COUNTRY FROM RATINGS r INNER JOIN GENRE g ON r.MOVIE\_ID = g.MOVIE\_ID INNER JOIN MOVIE m ON g.MOVIE\_ID = m.ID WHERE r.TOTAL\_VOTES > 1000 AND Month(DATE\_PUBLISHED) = 3 AND Year(DATE\_PUBLISHED) = 2017 AND m.COUNTRY IN ( 'USA' )) SELECT GENRE, Count(MOVIE\_ID) AS movie\_count FROM AGG GROUP BY GENRE ORDER BY Count(MOVIE\_ID) DESC; -- 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 | | . | . | . | | . | . | . | | . | . | . | +---------------+-------------------+---------------------+\*/ SELECT m.title, r.avg\_rating, g.genre FROM GENRE g INNER JOIN RATINGS r ON g.MOVIE\_ID = r.MOVIE\_ID INNER JOIN MOVIE m ON g.MOVIE\_ID = m.ID WHERE r.AVG\_RATING > 8 AND LOWER(m.TITLE) LIKE 'the%' ORDER BY r.AVG\_RATING DESC; -- We should also try our 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? SELECT r.median\_rating, COUNT(m.TITLE) AS movie\_count FROM RATINGS r INNER JOIN MOVIE m ON m.ID = r.MOVIE\_ID WHERE r.MEDIAN\_RATING = 8 AND m.DATE\_PUBLISHED BETWEEN '2018-04-01' AND '2019-04-01' GROUP BY r.MEDIAN\_RATING; -- Once again, try to solve the problem given below. -- Q17. Do German movies get more votes than Italian movies? -- Hint: Here we have to find the total number of votes for both German and Italian movies. WITH LANGUAGES\_GROUPED AS ( SELECT languages, total\_votes, CASE WHEN LANGUAGES REGEXP 'German' THEN 'German' WHEN LANGUAGES REGEXP 'Italian' THEN 'Italian' ELSE 'Others' END AS languages\_grouped FROM MOVIE M INNER JOIN RATINGS R ON M.ID=R.MOVIE\_ID ) SELECT LANGUAGES\_GROUPED AS 'languages', SUM(TOTAL\_VOTES) AS total\_votes FROM LANGUAGES\_GROUPED WHERE LANGUAGES\_GROUPED IN ('German', 'Italian') GROUP BY LANGUAGES\_GROUPED ORDER BY TOTAL\_VOTES DESC ; -- Answer is Yes /\* Now that we 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: we 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 | +---------------+-------------------+---------------------+----------------------+\*/ SELECT COUNT(\*) - COUNT(ID) AS id\_nulls, COUNT(\*) - COUNT(NAME) AS name\_nulls, COUNT(\*) - COUNT(HEIGHT) AS height\_nulls, COUNT(\*) - COUNT(DATE\_OF\_BIRTH) AS date\_of\_birth\_nulls, COUNT(\*) - COUNT(KNOWN\_FOR\_MOVIES) AS known\_for\_movies\_nulls FROM NAMES; /\* 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 | | . | . | | . | . | +---------------+-------------------+ \*/ WITH TOP\_3\_GENRE AS ( SELECT GENRE FROM RATINGS R INNER JOIN MOVIE M ON R.MOVIE\_ID=M.ID INNER JOIN GENRE USING (MOVIE\_ID) WHERE AVG\_RATING > 8 GROUP BY GENRE ORDER BY COUNT(GENRE) DESC LIMIT 3 ) SELECT NAME AS director\_name, COUNT(NAME) AS movie\_count FROM RATINGS R INNER JOIN MOVIE M ON R.MOVIE\_ID=M.ID INNER JOIN GENRE USING (MOVIE\_ID) INNER JOIN DIRECTOR\_MAPPING D USING (MOVIE\_ID) INNER JOIN NAMES N ON D.NAME\_ID=N.ID WHERE GENRE IN ( SELECT \* FROM TOP\_3\_GENRE) AND AVG\_RATING>8 GROUP BY NAME ORDER BY COUNT(NAME) DESC LIMIT 3 ; /\* James Mangold can be hired as the director for RSVP's next project. Do we 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 | | . | . | +---------------+-------------------+ \*/ SELECT NAME AS actor\_name, COUNT(NAME) AS movie\_count FROM NAMES N INNER JOIN ROLE\_MAPPING RO ON N.ID = RO.NAME\_ID INNER JOIN RATINGS RA ON RO.MOVIE\_ID = RA.MOVIE\_ID WHERE MEDIAN\_RATING >= 8 AND CATEGORY = 'actor' GROUP BY NAME ORDER BY COUNT(NAME) DESC LIMIT 2; /\* Have we find our favourite actor 'Mohanlal' in the list. If no, please check our 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 | | . | . | . | | . | . | . | +-------------------+-------------------+---------------------+\*/ SELECT production\_company, SUM(TOTAL\_VOTES) AS vote\_count, DENSE\_RANK() OVER(ORDER BY SUM(TOTAL\_VOTES) DESC) AS prod\_comp\_rank FROM MOVIE M INNER JOIN RATINGS RA ON M.ID=RA.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: We 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 | | . | . | . | . | . | | . | . | . | . | . | | . | . | . | . | . | +---------------+-------------------+---------------------+----------------------+-----------------+\*/ WITH ACTORS AS ( SELECT NAME AS actor\_name , SUM(TOTAL\_VOTES) AS total\_votes, COUNT(NAME) AS movie\_count, ROUND(SUM(AVG\_RATING \* TOTAL\_VOTES) / SUM(TOTAL\_VOTES), 2) AS actor\_avg\_rating FROM NAMES N INNER JOIN ROLE\_MAPPING RO ON N.ID = RO.NAME\_ID INNER JOIN MOVIE M ON RO.MOVIE\_ID = M.ID INNER JOIN RATINGS RA ON M.ID = RA.MOVIE\_ID WHERE COUNTRY REGEXP 'india' AND CATEGORY = 'actor' GROUP BY NAME HAVING MOVIE\_COUNT >= 5) SELECT \*, DENSE\_RANK() OVER ( ORDER BY ACTOR\_AVG\_RATING DESC, TOTAL\_VOTES DESC) AS actor\_rank FROM ACTORS; -- 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: We 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 | | . | . | . | . | . | | . | . | . | . | . | | . | . | . | . | . | +---------------+-------------------+---------------------+----------------------+-----------------+\*/ WITH ACTRESS AS ( SELECT NAME AS actress\_name, total\_votes, COUNT(NAME) AS movie\_count, ROUND(SUM(AVG\_RATING\*TOTAL\_VOTES)/SUM(TOTAL\_VOTES),2) AS actress\_avg\_rating FROM NAMES N INNER JOIN ROLE\_MAPPING RO ON N.ID=RO.NAME\_ID INNER JOIN MOVIE M ON RO.MOVIE\_ID=M.ID INNER JOIN RATINGS RA USING (MOVIE\_ID) WHERE LANGUAGES REGEXP 'hindi' AND COUNTRY REGEXP 'india' AND CATEGORY = 'actress' GROUP BY ACTRESS\_NAME HAVING COUNT(ACTRESS\_NAME)>=3 ) SELECT \*, DENSE\_RANK() OVER(ORDER BY ACTRESS\_AVG\_RATING DESC, TOTAL\_VOTES DESC) AS actress\_rank FROM ACTRESS LIMIT 5; /\* Taapsee Pannu tops with average rating 7.74. Now let us divide all the thriller movies in the following categories and find out their numbers.\*/ /\* Q24. Select thriller movies as per avg rating and classify them in the following category: Rating > 8: Superhit movies Rating between 7 and 8: Hit movies Rating between 5 and 7: One-time-watch movies Rating < 5: Flop movies --------------------------------------------------------------------------------------------\*/ SELECT TITLE AS movie, AVG\_RATING, CASE WHEN AVG\_RATING > 8 THEN 'Superhit movies' WHEN AVG\_RATING BETWEEN 7 AND 8 THEN 'Hit movies' WHEN AVG\_RATING BETWEEN 5 AND 7 THEN 'One-time-watch movies' WHEN AVG\_RATING < 5 THEN 'Flop movies' END AS 'avg\_rating\_category' FROM GENRE g INNER JOIN RATINGS ra USING(MOVIE\_ID) INNER JOIN MOVIE m ON ra.MOVIE\_ID = m.ID WHERE GENRE = 'thriller'; /\* Until now, we have analysed various tables of the data set. Now, we will perform some tasks that will give we 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: We 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 | | . | . | . | . | | . | . | . | . | | . | . | . | . | +---------------+-------------------+---------------------+----------------------+\*/ WITH GENRE AS (SELECT GENRE, ROUND(AVG(DURATION), 2) AS avg\_duration, SUM(AVG(DURATION)) OVER ( ORDER BY GENRE ROWS UNBOUNDED PRECEDING) AS running\_total\_duration, AVG(AVG(DURATION)) OVER ( ORDER BY GENRE ROWS UNBOUNDED PRECEDING) AS moving\_avg\_duration FROM MOVIE m INNER JOIN GENRE g ON m.ID = g.MOVIE\_ID GROUP BY GENRE) SELECT genre, avg\_duration, ROUND(RUNNING\_TOTAL\_DURATION, 2) AS running\_total\_duration, ROUND(MOVING\_AVG\_DURATION, 2) AS moving\_avg\_duration FROM GENRE; -- 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 | $103244842 | 1 | | . | . | . | . | . | | . | . | . | . | . | | . | . | . | . | . | +---------------+-------------------+---------------------+----------------------+-----------------+\*/ -- Top 3 Genres based on most number of movies WITH TOP\_3\_GENRE AS ( SELECT GENRE FROM GENRE GROUP BY GENRE ORDER BY COUNT(GENRE) DESC LIMIT 3 ), TOP\_MOVIES AS ( SELECT genre, year, TITLE AS movie\_name, CAST(REPLACE(IFNULL(WORLWIDE\_GROSS\_INCOME,0),'$ ','') AS DECIMAL(10)) AS worldwide\_gross\_income\_$, ROW\_NUMBER() OVER (PARTITION BY YEAR ORDER BY CAST(REPLACE(IFNULL(WORLWIDE\_GROSS\_INCOME,0),'$ ','') AS DECIMAL(10)) DESC) AS movie\_rank FROM MOVIE M INNER JOIN GENRE G ON M.ID = G.MOVIE\_ID WHERE GENRE IN ( SELECT \* FROM TOP\_3\_GENRE) ) SELECT \* FROM TOP\_MOVIES WHERE MOVIE\_RANK<=5; -- Finally, let’s find out the names of the top two production houses that have produced the highest number of hits among multilingual movies. -- Q27. Which are the top two production houses that have produced the highest number of hits (median rating >= 8) among multilingual movies? /\* Output format: +-------------------+-------------------+---------------------+ |production\_company |movie\_count | prod\_comp\_rank| +-------------------+-------------------+---------------------+ | The Archers | 830 | 1 | | . | . | . | | . | . | . | +-------------------+-------------------+---------------------+\*/ SELECT production\_company, COUNT(PRODUCTION\_COMPANY) AS movie\_count , DENSE\_RANK() OVER(ORDER BY COUNT(PRODUCTION\_COMPANY) DESC) AS prod\_comp\_rank FROM MOVIE M INNER JOIN RATINGS RA ON M.ID=RA.MOVIE\_ID WHERE MEDIAN\_RATING>=8 AND LANGUAGES REGEXP ',' GROUP BY PRODUCTION\_COMPANY LIMIT 2; -- Multilingual is the important piece in the above question. It was created using POSITION(',' IN languages)>0 logic -- If there is a comma, that means the movie is of more than one language -- Q28. Who are the top 3 actresses based on number of Super Hit movies (average rating >8) in drama genre? /\* Output format: +---------------+-------------------+---------------------+----------------------+-----------------+ | actress\_name | total\_votes | movie\_count |actress\_avg\_rating |actress\_rank | +---------------+-------------------+---------------------+----------------------+-----------------+ | Laura Dern | 1016 | 1 | 9.60 | 1 | | . | . | . | . | . | | . | . | . | . | . | +---------------+-------------------+---------------------+----------------------+-----------------+\*/ SELECT NAME AS actress\_name, SUM(TOTAL\_VOTES) AS total\_votes, COUNT(NAME) AS movie\_count, ROUND(SUM(AVG\_RATING\*TOTAL\_VOTES)/SUM(TOTAL\_VOTES),2) AS actress\_avg\_rating, ROW\_NUMBER() OVER (ORDER BY COUNT(NAME) DESC) AS actress\_rank FROM GENRE G INNER JOIN MOVIE M ON G.MOVIE\_ID=M.ID INNER JOIN RATINGS RA USING (MOVIE\_ID) INNER JOIN ROLE\_MAPPING RO USING (MOVIE\_ID) INNER JOIN NAMES N ON RO.NAME\_ID=N.ID WHERE AVG\_RATING >8 AND GENRE = 'drama' AND CATEGORY = 'actress' GROUP BY NAME LIMIT 3; /\* Q29. Get the following details for top 9 directors (based on number of movies) Director id Name Number of movies Average inter movie duration in days Average movie ratings Total votes Min rating Max rating total movie durations Format: +---------------+-------------------+---------------------+----------------------+--------------+--------------+------------+------------+----------------+ | director\_id | director\_name | number\_of\_movies | avg\_inter\_movie\_days | avg\_rating | total\_votes | min\_rating | max\_rating | total\_duration | +---------------+-------------------+---------------------+----------------------+--------------+--------------+------------+------------+----------------+ |nm1777967 | A.L. Vijay | 5 | 177 | 5.65 | 1754 | 3.7 | 6.9 | 613 | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | | . | . | . | . | . | . | . | . | . | +---------------+-------------------+---------------------+----------------------+--------------+--------------+------------+------------+----------------+ --------------------------------------------------------------------------------------------\*/ WITH NEXT\_DATE\_PUBLISH AS ( SELECT NAME\_ID AS DIRECTOR\_ID, NAME AS DIRECTOR\_NAME, DATE\_PUBLISHED, AVG\_RATING, TOTAL\_VOTES, DURATION, LEAD(DATE\_PUBLISHED,1) OVER(PARTITION BY NAME\_ID ORDER BY DATE\_PUBLISHED) AS NEXT\_DATE\_PUBLISHED FROM DIRECTOR\_MAPPING D INNER JOIN NAMES N ON D.NAME\_ID=N.ID INNER JOIN MOVIE M ON D.MOVIE\_ID=M.ID INNER JOIN RATINGS RA USING (MOVIE\_ID) ) SELECT director\_id, director\_name, COUNT(DIRECTOR\_NAME) AS number\_of\_movies, ROUND(AVG(DATEDIFF(NEXT\_DATE\_PUBLISHED,DATE\_PUBLISHED)),0) AS avg\_inter\_movie\_days, ROUND(AVG(AVG\_RATING),2) AS avg\_rating, SUM(TOTAL\_VOTES) AS total\_votes, MIN(AVG\_RATING) AS min\_rating, MAX(AVG\_RATING) AS max\_rating, SUM(DURATION) AS total\_duration FROM NEXT\_DATE\_PUBLISH GROUP BY DIRECTOR\_ID ORDER BY NUMBER\_OF\_MOVIES DESC LIMIT 9;

## **Data Explorer**

Version 5 (6.43 MB)

* insert\_drive\_file

Case Study - Analysis.sql

* drive\_pdf\_outline

Executive Summary.pdf

* insert\_drive\_file

IMDBdatasetimport.sql

* arrow\_right

calendar\_view\_week

IMDbmoviesDataandERDfinal.xlsx

## **Summary**

arrow\_right

folder

4 files

arrow\_right

calendar\_view\_week

38 columns

text\_snippet

## **Metadata**

unfold\_more**Expand all**

### **Collaborators**

keyboard\_arrow\_down

### **Authors**

keyboard\_arrow\_down

### **Coverage**

keyboard\_arrow\_down

### **DOI Citation**

keyboard\_arrow\_down

### **Provenance**

keyboard\_arrow\_down

### **License**

keyboard\_arrow\_down

### **Expected Update Frequency**

keyboard\_arrow\_down

insights

## **Activity Overview**

visibility

###### **Views**

##### **923**

|  |  |
| --- | --- |
|  |  |
| Dec 22, 2023 | 11 |
| Dec 23, 2023 | 4 |
| Dec 24, 2023 | 2 |
| Dec 25, 2023 | 1 |
| Dec 26, 2023 | 5 |
| Dec 27, 2023 | 8 |
| Dec 28, 2023 | 3 |
| Dec 29, 2023 | 7 |
| Dec 30, 2023 | 4 |
| Dec 31, 2023 | 6 |
| Jan 1, 2024 | 2 |
| Jan 2, 2024 | 5 |
| Jan 3, 2024 | 3 |
| Jan 4, 2024 | 2 |
| Jan 5, 2024 | 2 |
| Jan 7, 2024 | 1 |
| Jan 8, 2024 | 5 |
| Jan 9, 2024 | 5 |
| Jan 10, 2024 | 9 |
| Jan 11, 2024 | 6 |
| Jan 12, 2024 | 2 |
| Jan 13, 2024 | 4 |
| Jan 14, 2024 | 2 |
| Jan 15, 2024 | 1 |
| Jan 16, 2024 | 1 |
| Jan 17, 2024 | 8 |
| Jan 18, 2024 | 6 |

|  |  |
| --- | --- |
|  |  |
| Dec 22, 2023 | 1 |
| Dec 25, 2023 | 1 |
| Dec 27, 2023 | 1 |
| Dec 29, 2023 | 2 |
| Dec 30, 2023 | 1 |
| Dec 31, 2023 | 1 |
| Jan 9, 2024 | 1 |
| Jan 10, 2024 | 2 |
| Jan 13, 2024 | 1 |
| Jan 15, 2024 | 1 |