DATA-DRIVEN SQL ANALYSIS PROJECT FOR RSVP MOVIES

PRESENTER: POOJA THAKUR

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.*/

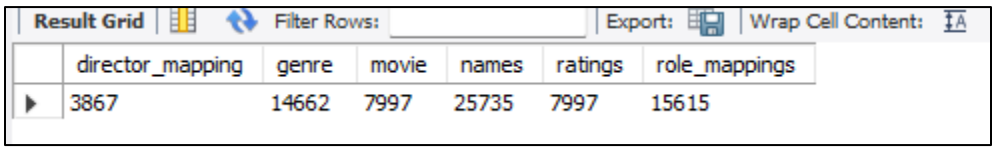
-- SEGMENT 1:

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

-- Type your code below:

```
SELECT
        (SELECT COUNT(*) FROM director_mapping) AS director_mapping,
         (SELECT COUNT(*) FROM genre) AS genre ,
        (SELECT COUNT(*) FROM movie) AS movie ,
        (SELECT COUNT(*)  FROM names) AS names,
        (SELECT COUNT(*) FROM ratings) AS ratings ,
        (SELECT COUNT(*)  FROM role_mapping) AS role_mappings
  ;
```

OUTPUT:



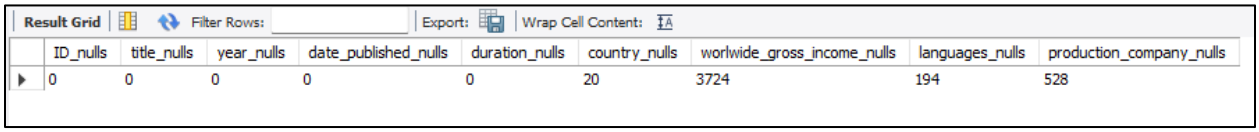
director_mapping	genre	movie	names	ratings	role_mappings
3867	14662	7997	25735	7997	15615

-- 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;
```

OUTPUT:



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

ID_nulls	title_nulls	year_nulls	date_published_nulls	duration_nulls	country_nulls	worlwide_gross_income_nulls	languages_nulls	production_company_nulls
0	0	0	0	0	20	3724	194	528

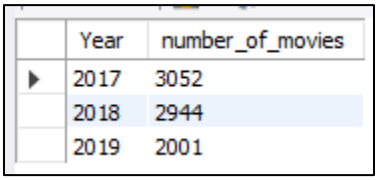
-- Now as you can see four columns of the movie table has null values. Let's look at the at the movies released each year.

-- Q3. Find the total number of movies released each year? How does the trend look month wise?

- **Part a) total number of movies released each year**

```
SELECT year AS Year, COUNT(id) AS number_of_movies FROM imdb.movie
GROUP BY Year
ORDER BY Year;
```

OUTPUT:

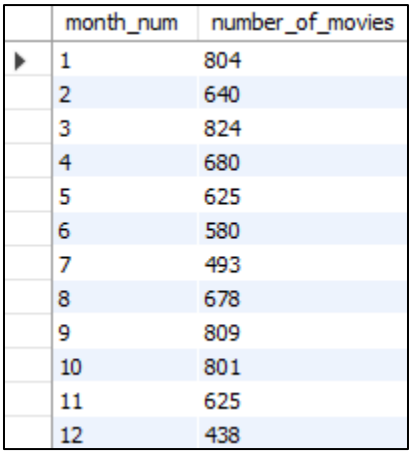


Year	number_of_movies
2017	3052
2018	2944
2019	2001

- **Part b) monthly trend year**

```
SELECT MONTH(date_published) AS month_num, COUNT(id) AS number_of_movies
FROM imdb.movie
GROUP BY month_num
ORDER BY month_num;
```

OUTPUT:



month_num	number_of_movies
1	804
2	640
3	824
4	680
5	625
6	580
7	493
8	678
9	809
10	801
11	625
12	438

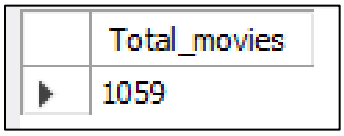
/*The highest number of movies is produced in the month of March. So, now that you have understood the month-wise trend of movies, let's take a look at the other details in the movies table. We know USA and India produces huge number of movies each year. Lets find the number of movies produced by USA 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 COUNT(*) AS Total_movies
FROM movie
WHERE year = 2019 AND (country LIKE '%USA%' OR country LIKE '%India%');
```

OUTPUT:



Total_movies
1059

/* 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?

```
SELECT DISTINCT genre
FROM genre;
```

OUTPUT:

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?

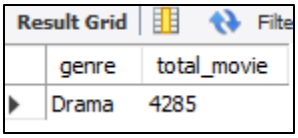
```
SELECT g.genre, COUNT(m.id) AS total_movie
FROM movie m JOIN genre g ON m.id = g.movie_id
GROUP BY g.genre
ORDER BY total_movie DESC;
```

-- OR without joining two tables

```
SELECT genre, COUNT(*) AS total_movie
FROM genre
GROUP BY genre
```

```
ORDER BY total_movie DESC
LIMIT 1;
```

OUTPUT:



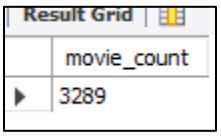
genre	total_movie
Drama	4285

/* 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?

```
SELECT COUNT(*) AS movie_count
FROM ( SELECT movie_id
    FROM imdb.genre
    GROUP BY movie_id
    HAVING COUNT(*) = 1
   ) AS single_genre_movie;
```

OUTPUT:



movie_count
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.)

```
SELECT genre, AVG(movie.duration) AS avg_duration
FROM imdb.genre
INNER JOIN imdb.movie ON movie.id = genre.movie_id
GROUP BY genre
ORDER BY avg_duration DESC;
```

OUTPUT:

	genre	avg_duration
▶	Action	112.8829
	Romance	109.5342
	Crime	107.0517
	Drama	106.7746
	Fantasy	105.1404
	Comedy	102.6227
	Adventure	101.8714
	Mystery	101.8000
	Thriller	101.5761
	Family	100.9669
	Others	100.1600
	Sci-Fi	97.9413
	Horror	92.7243

/* 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)

```
WITH GenreMovieCount AS (
        SELECT genre, COUNT(movie_id) AS total_movies
        FROM genre GROUP BY genre
        )
SELECT genre, total_movies,
        RANK() OVER (ORDER BY total_movies DESC) AS genre_rank
FROM GenreMovieCount
WHERE genre = 'Thriller';
```

OUTPUT:

	genre	total_movies	genre_rank
▶	Thriller	1484	1

/*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?

```
SELECT MIN(avg_rating) AS min_avg_rating,
       MAX(avg_rating) AS max_avg_rating,
       MIN(total_votes) AS min_total_votes,
       MAX(total_votes) AS max_total_votes,
       MIN(median_rating) AS min_median_rating,
       MAX(median_rating) AS max_median_rating
FROM ratings;
```

OUTPUT:

min_avg_rating	max_avg_rating	min_total_votes	max_total_votes	min_median_rating	max_median_rating
1.0	10.0	100	725138	1	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?

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

```
SELECT m.title, r.avg_rating,
       RANK() OVER (ORDER BY r.avg_rating DESC) AS movie_rank
FROM movie m JOIN ratings r ON m.id = r.movie_id
ORDER BY r.avg_rating DESC
LIMIT 10;
```

OUTPUT:

title	avg_rating	movie_rank
Kirket	10.0	1
Love in Kilnerry	10.0	1
Gini Helida Kathe	9.8	3
Runam	9.7	4
Fan	9.6	5
Android Kunjappan Version 5.25	9.6	5
Yeh Suhaagraat Impossible	9.5	7
Safe	9.5	7
The Brighton Miracle	9.5	7
Shibu	9.4	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.

-- Order by is good to have

```
SELECT median_rating, COUNT(movie_id) AS movie_count
FROM ratings
GROUP BY median_rating
ORDER BY median_rating;
```

OUTPUT:

median_rating	movie_count
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 with movie_count of 2257. 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)??

```
WITH HitMovies AS (
    SELECT m.production_company, COUNT(m.id) AS movie_count
    FROM movie m
    JOIN ratings r ON m.id = r.movie_id
    WHERE r.avg_rating > 8
    GROUP BY m.production_company
)
```

```
SELECT production_company, movie_count,
 RANK() OVER (ORDER BY movie_count DESC) AS prod_company_rank
FROM HitMovies
WHERE production_company IS NOT NULL
ORDER BY movie_count DESC
LIMIT 1;
```

OUTPUT:

	production_company	movie_count	prod_company_rank
▸	Dream Warrior Pictures	3	1

-- 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?

```
SELECT g.genre, COUNT(DISTINCT m.id) AS movie_count
FROM movie AS m JOIN ratings AS r ON m.id = r.movie_id
                        JOIN genre AS g ON  m.id = g.movie_id
WHERE ( m.country LIKE '%USA%' ) AND (MONTH(m.date_published) = 3 )
  AND ( YEAR(m.date_published) = 2017)  AND ( r.total_votes > 1000 )
GROUP BY g.genre
ORDER BY movie_count DESC;
```

-- OR -- We can use (m.date_published BETWEEN '2017-03-01' AND '2017-03-31')

OUTPUT:

	genre	movie_count
▸	Drama	24
	Comedy	9
	Action	8
	Thriller	8
	Sci-Fi	7
	Crime	6
	Horror	6
	Mystery	4
	Romance	4
	Adventure	3
	Fantasy	3
	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?

```
SELECT m.title AS title, AVG(r.avg_rating) AS avg_rating, g.genre AS genre
FROM movie AS m JOIN genre AS g ON m.id=g.movie_id
                JOIN ratings AS r ON m.id=r.movie_id
WHERE m.title LIKE 'THE%' AND r.avg_rating > 8
GROUP BY title, genre
ORDER BY avg_rating DESC;
```

OUTPUT:

title	avg_rating	genre
The Brighton Miracle	9.5	Drama
The Colour of Darkness	9.1	Drama
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
The Irishman	8.7	Drama
The Mystery of Godliness: The Sequel	8.5	Drama
The Gambinos	8.4	Crime
The Gambinos	8.4	Drama
Theeran Adhigaaram Ondru	8.3	Action
Theeran Adhigaaram Ondru	8.3	Crime
Theeran Adhigaaram Ondru	8.3	Thriller
The King and I	8.2	Drama
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?

```
SELECT COUNT(m.id) AS movie_count
FROM movie m JOIN ratings r ON m.id = r.movie_id
WHERE m.date_published BETWEEN '2018-04-01' AND '2019-04-01'
AND r.median_rating = 8
GROUP BY median_rating;
```

OUTPUT:

movie_count
361

-- 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 'Germany' AS country, SUM(r.total_votes) AS total_votes
FROM movie m JOIN ratings r ON m.id = r.movie_id
WHERE m.country LIKE '%Germany%'
UNION ALL
SELECT 'Italy' AS country, SUM(r.total_votes) AS total_votes
FROM movie m JOIN ratings r ON m.id = r.movie_id
WHERE m.country LIKE '%Italy%'
ORDER BY total_votes DESC;
```

OUTPUT:

country	total_votes
Germany	106710
Italy	77965

-- 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

```
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;
```

OUTPUT:

	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.)

```
WITH TopGenres AS (
                    SELECT g.genre, COUNT(m.id) AS movie_count
                    FROM genre g  JOIN movie m ON g.movie_id = m.id
                                  JOIN ratings r ON m.id = r.movie_id
                    WHERE r.avg_rating > 8
                    GROUP BY g.genre
                    ORDER BY movie_count DESC
                    LIMIT 3
)
SELECT dm.name_id AS director_id, n.name AS director_name,
        COUNT(DISTINCT dm.movie_id) AS movie_count
```

```
FROM director_mapping dm JOIN names n ON dm.name_id = n.id
                         JOIN movie m ON dm.movie_id = m.id
                         JOIN ratings r ON m.id = r.movie_id
                         JOIN genre g ON m.id = g.movie_id
                         JOIN TopGenres tg ON g.genre = tg.genre
WHERE r.avg_rating > 8
GROUP BY dm.name_id, n.name
ORDER BY movie_count DESC
LIMIT 3;
```

OUTPUT:

director_id	director_name	movie_count
nm0751648	Joe Russo	2
nm0003506	James Mangold	2
nm0751577	Anthony Russo	2

/* 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?

```
SELECT n.name AS actor_name, COUNT(n.id) AS movie_count
FROM names AS n JOIN role_mapping AS rm ON n.id = rm.name_id
                JOIN movie AS m ON rm.movie_id = m.id
          JOIN ratings AS r ON m.id = r.movie_id
WHERE median_rating >= 8
GROUP BY n.name
ORDER BY movie_count DESC
LIMIT 2;
```

-- Alternate way

```
WITH TopActors AS (
 SELECT a.name AS actor_name, COUNT(DISTINCT rm.movie_id) AS movie_count
 FROM role_mapping rm
 JOIN names a ON rm.name_id = a.id
 JOIN ratings r ON rm.movie_id = r.movie_id
 WHERE r.median_rating >= 8
```

```
 GROUP BY a.name
 ORDER BY movie_count DESC
 LIMIT 2
)
-- Display final result
SELECT actor_name, movie_count
FROM TopActors;
```

OUTPUT:

	actor_name	movie_count
▶	Mammootty	8
	Mohanlal	5

/* 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?

```
WITH TopProductionHouse AS (
 SELECT m.production_company AS production_company, SUM(r.total_votes) AS
vote_count
 FROM movie AS m JOIN ratings r ON m.id = r.movie_id
 WHERE m.production_company IS NOT NULL
 GROUP BY production_company
)
-- Display final result
SELECT production_company, vote_count, RANK() OVER (ORDER BY vote_count
DESC) AS prod_comp_rank
FROM TopProductionHouse
ORDER BY vote_count DESC
LIMIT 3;
```

OUTPUT:

	production_company	vote_count	prod_comp_rank
▶	Marvel Studios	2656967	1
	Twentieth Century Fox	2411163	2
	Warner Bros.	2396057	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.)

```
WITH Actor AS (
 -- Filter Indian movies and count movies per actor
 SELECT n.name AS actor_name, COUNT(DISTINCT rm.movie_id) AS movie_count,
        SUM(r.total_votes) AS total_votes,
        SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes) AS actor_avg_rating
 FROM role_mapping rm JOIN names n ON rm.name_id = n.id
                      JOIN movie m ON rm.movie_id = m.id
                      JOIN ratings r ON m.id = r.movie_id
 WHERE m.country LIKE '%India%'
 GROUP BY n.name
 HAVING COUNT(DISTINCT rm.movie_id) >= 5
)
SELECT actor_name, total_votes, movie_count, actor_avg_rating,
        RANK() OVER (ORDER BY actor_avg_rating DESC,
        total_votes DESC) AS actor_rank
FROM Actor
ORDER BY actor_rank
LIMIT 1;
```

OUTPUT:

	actor_name	total_votes	movie_count	actor_avg_rating	actor_rank
▶	Vijay Sethupathi	23114	5	8.41673	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.)

```
WITH Actresses AS (
 -- Filter Indian movies and count movies per actor
 SELECT n.name AS actress_name,
                COUNT(DISTINCT rm.movie_id) AS movie_count,
                SUM(r.total_votes) AS total_votes,
                SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes) AS
actress_avg_rating
 FROM role_mapping rm JOIN names n ON rm.name_id = n.id
                                        JOIN movie m ON rm.movie_id = m.id
          JOIN ratings r ON m.id = r.movie_id
 WHERE m.country LIKE '%India%'
          AND rm.category = 'actress'
          AND m.languages LIKE '%Hindi%'
 GROUP BY n.name
 HAVING COUNT(DISTINCT rm.movie_id) >= 3
)
-- Assign ranking based on weighted average rating and total votes (tie-breaker)
SELECT actress_name, total_votes, movie_count, actress_avg_rating,
                RANK() OVER (ORDER BY actress_avg_rating DESC,
    total_votes DESC) AS actress_rank
FROM Actresses
ORDER BY actress_rank
LIMIT 5;
```

OUTPUT:

actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Taapsee Pannu	18061	3	7.73692	1
Kriti Sanon	21967	3	7.04911	2
Divya Dutta	8579	3	6.88440	3
Shraddha Kapoor	26779	3	6.63024	4
Kriti Kharbanda	2549	3	4.80314	5

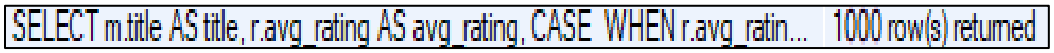
/* Taapsee Pannu tops with average rating 7.74. Now let us divide all the thriller movies in the following categories and find out their numbers.*/

/* 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 ----------*/

-- Type your code below:

```
SELECT m.title AS title, r.avg_rating AS avg_rating,
CASE
        WHEN r.avg_rating>8 THEN 'Superhit movies'
  WHEN r.avg_rating BETWEEN 7 AND 8 THEN 'Hit movies'
  WHEN r.avg_rating BETWEEN 5 AND 7 THEN 'One-time-watch movies'
  ELSE 'Flop movies'
END as movie_category
FROM movie AS m INNER JOIN ratings AS r ON m.id=r.movie_id
INNER JOIN genre AS g ON m.id=g.movie_id
WHERE g.genre='thriller'
ORDER BY avg_rating DESC;
```

OUTPUT:



title	avg_rating	movie_category
Safe	9.5	Superhit movies
Digbhayam	9.2	Superhit movies
Dokyala Shot	9.2	Superhit movies
Abstruse	9.0	Superhit movies
Kaithi	8.9	Superhit movies
Raju Gari Gadhi 3	8.8	Superhit movies
Lost Angelas	8.8	Superhit movies
Enigma	8.8	Superhit movies
Joker	8.8	Superhit movies
Birbal Trilogy	8.8	Superhit movies
Vikram Vedha	8.7	Superhit movies
Ratsasan	8.7	Superhit movies
Ghost	8.7	Superhit movies
Bell Bottom	8.7	Superhit movies
Rakshasudu	8.4	Superhit movies
Adios Vaya Con ...	8.4	Superhit movies
Andhadhun	8.4	Superhit 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.)

```
SELECT g.genre AS genre, ROUND(AVG(m.duration)) AS avg_duration,
        ROUND(SUM(AVG(m.duration)) OVER(ORDER BY g.genre ROWS
UNBOUNDED PRECEDING),2) AS running_total_duration,
        ROUND(AVG(AVG(m.duration)) OVER(ORDER BY g.genre ROWS 12
PRECEDING),2) AS moving_avg_duration
FROM movie AS m JOIN genre AS g ON m.id=g.movie_id
GROUP BY genre
ORDER BY genre;
```

OUTPUT:

genre	avg_duration	running_total_duration	moving_avg_duration
Action	113	112.88	112.88
Adventure	102	214.75	107.38
Comedy	103	317.38	105.79
Crime	107	424.43	106.11
Drama	107	531.20	106.24
Family	101	632.17	105.36
Fantasy	105	737.31	105.33
Horror	93	830.03	103.75
Mystery	102	931.83	103.54
Others	100	1031.99	103.20
Romance	110	1141.53	103.78
Sci-Fi	98	1239.47	103.29
Thriller	102	1341.05	103.16

-- 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.)

-- Top 3 Genres based on most number of movies

-- Step 1: Identify the top 3 genres with the most number of movies
	WITH TopGenres AS (
			SELECT g.genre, COUNT(g.movie_id) AS movie_count
			FROM genre g
			GROUP BY g.genre
			ORDER BY movie_count DESC
			LIMIT 3
	 ),
-- Step 2: Get the five highest-grossing movies for each year within the top 3 genres
	RankedMovie AS (
			SELECT g.genre, m.year, m.title AS movie_name,
			CAST(REPLACE(REPLACE(m.worlwide_gross_income, '$', ''), ',', '') AS
	DECIMAL) AS
				worldwide_gross_income,
			RANK() OVER (PARTITION BY g.genre, m.year ORDER BY
				CAST(REPLACE(REPLACE(m.worlwide_gross_income, '$', ''), ',',
	'') AS DECIMAL) DESC) AS movie_rank
			FROM genre g
			JOIN movie m ON g.movie_id = m.id
			JOIN TopGenres tg ON g.genre = tg.genre
			WHERE m.worlwide_gross_income IS NOT NULL
			AND TRIM(m.worlwide_gross_income) != ''
	)
-- Step 3: Select the top 5 highest-grossing movies for each year
	SELECT genre, year, movie_name, worldwide_gross_income, movie_rank
	FROM RankedMovie
	WHERE movie_rank <= 5
	ORDER BY year DESC, genre, movie_rank;
```

OUTPUT:

genre	year	movie_name	worldwide_gross_income	movie_rank
Comedy	2019	Toy Story 4	1073168585	1
Comedy	2019	Pokémon Detective Pikachu	431705346	2
Comedy	2019	The Secret Life of Pets 2	429434163	3
Comedy	2019	Once Upon a Time... in Hollywood	371207970	4
Comedy	2019	Shazam!	364571656	5
Drama	2019	Avengers: Endgame	2797800564	1
Drama	2019	The Lion King	1655156910	2
Drama	2019	Joker	995064593	3
Drama	2019	Liu lang di qiu	699760773	4
Drama	2019	It Chapter Two	463326885	5
Thriller	2019	Joker	995064593	1
Thriller	2019	Ne Zha zhi mo tong jiang shi	700547754	2
Thriller	2019	John Wick: Chapter 3 - Parabellum	326667460	3
Thriller	2019	Us	255105930	4
Thriller	2019	Glass	246985576	5
Comedy	2018	Deadpool 2	785046920	1
Comedy	2018	Ant-Man and the Wasp	622674139	2
Comedy	2018	Tang ren jie tan an 2	544061916	3
Comedy	2018	Ralph Breaks the Internet	529323962	4
Comedy	2018	Hotel Transylvania 3: Summer Va...	528583774	5
Drama	2018	Bohemian Rhapsody	903655259	1
Drama	2018	Hong hai xing dong	579220560	2
Drama	2018	Wo bu shi yao shen	451183391	3
Drama	2018	A Star Is Born	434888866	4
Drama	2018	Fifty Shades Freed	371985018	5
Thriller	2018	Venom	856085151	1
Thriller	2018	Mission: Impossible - Fallout	791115104	2
Thriller	2018	Hong hai xing dong	579220560	3
Thriller	2018	Fifty Shades Freed	371985018	4
Thriller	2018	The Nun	365550119	5
Comedy	2017	Despicable Me 3	1034799409	1
Comedy	2017	Jumanji: Welcome to the Jungle	962102237	2
Comedy	2017	Guardians of the Galaxy Vol. 2	863756051	3
Comedy	2017	Thor: Ragnarok	853977126	4
Comedy	2017	Sing	634151679	5
Drama	2017	Zhan lang II	870325439	1
Drama	2017	Logan	619021436	2
Drama	2017	Dunkirk	526940665	3
Drama	2017	War for the Planet of the Apes	490719763	4
Drama	2017	La La Land	446092357	5
Thriller	2017	The Fate of the Furious	1236005118	1
Thriller	2017	Zhan lang II	870325439	2
Thriller	2017	xXx: Return of Xander Cage	346118277	3
Thriller	2017	Annabelle: Creation	306515884	4
Thriller	2017	Split	278454358	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?

```
SELECT m.production_company AS production_company,
                COUNT(m.id) AS movie_count,
                DENSE_RANK() OVER(ORDER BY COUNT(m.id) DESC) AS
prod_comp_rank
FROM movie m JOIN ratings r ON m.id=r.movie_id
WHERE r.median_rating>=8
                AND POSITION(',' IN m.languages)>0
    AND m.production_company IS NOT NULL
GROUP BY production_company
LIMIT 2;
```

OUTPUT:

	production_company	movie_count	prod_comp_rank
▶	Star Cinema	7	1
	Twentieth Century Fox	4	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