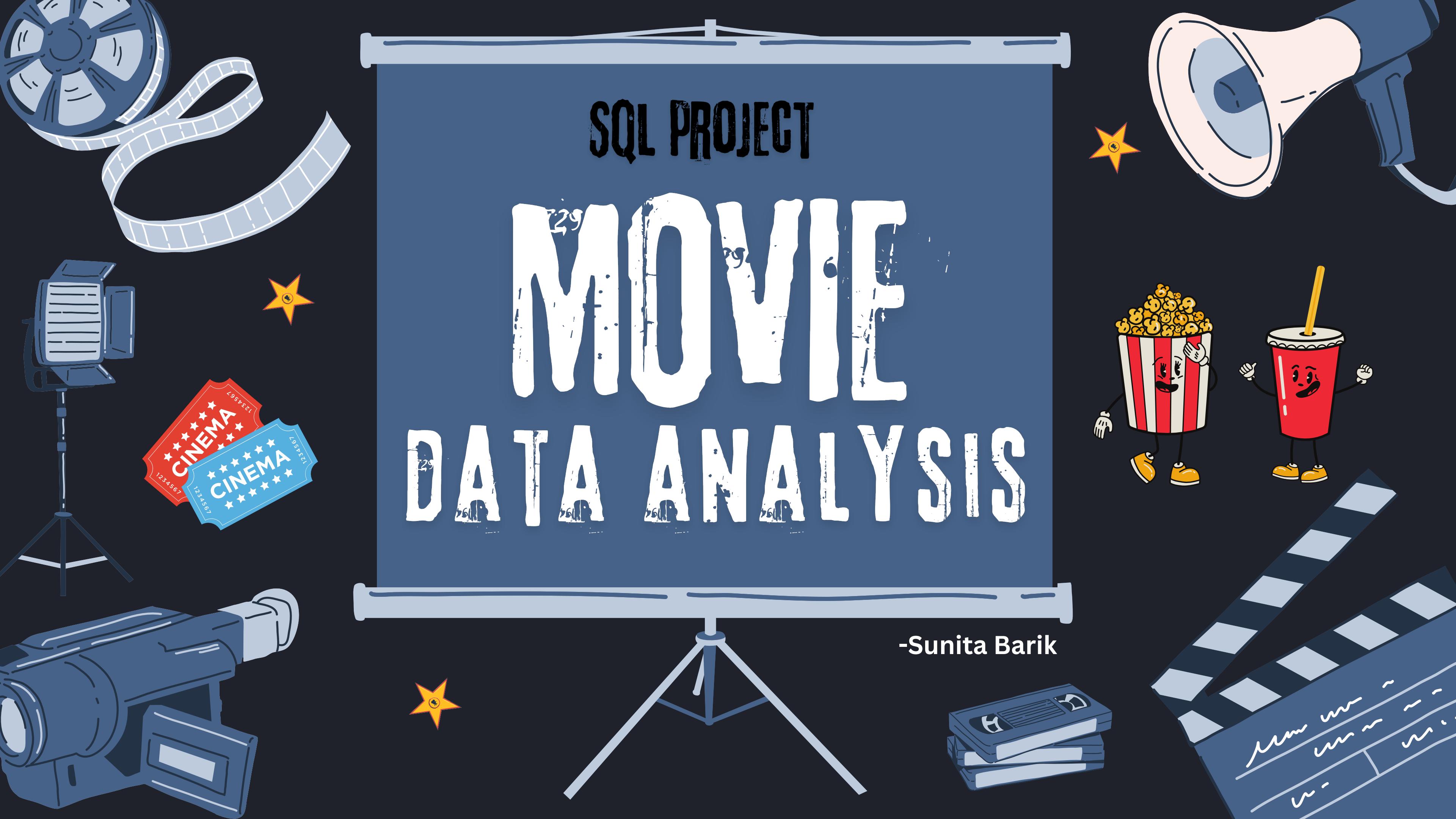


SQL PROJECT

MOVIE

DATA ANALYSIS

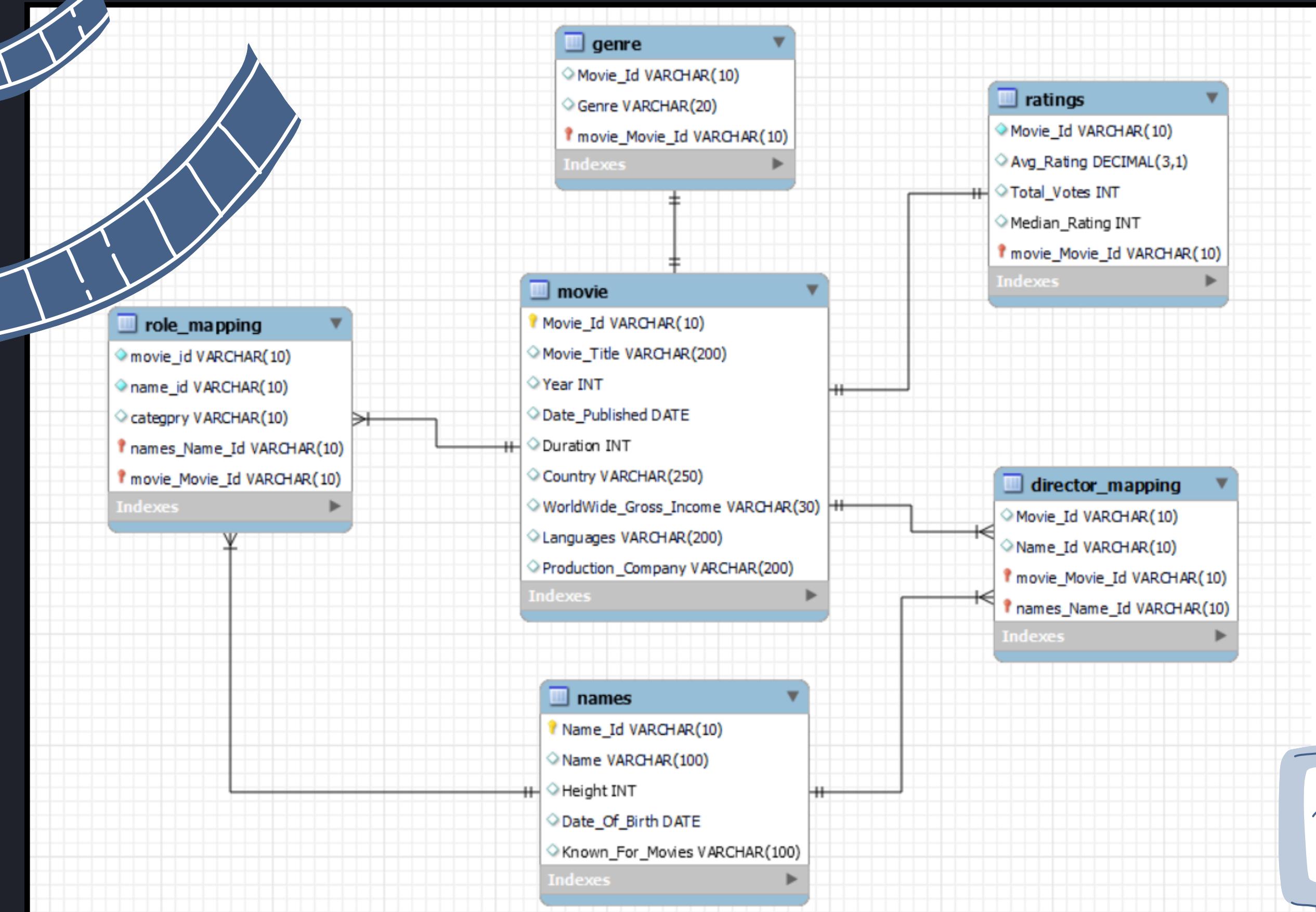
-Sunita Barik



OBJECTIVE

The objective of this project is to find the basic information about movies produced in the year 2017,2018,2019 like - **Hit Movies, Popular actresses and actors, Ratings of Movies, Popular Genres, Top Production Houses, etc.**

DATABASE SCHEMA



LEVEL OF QUERIES

1

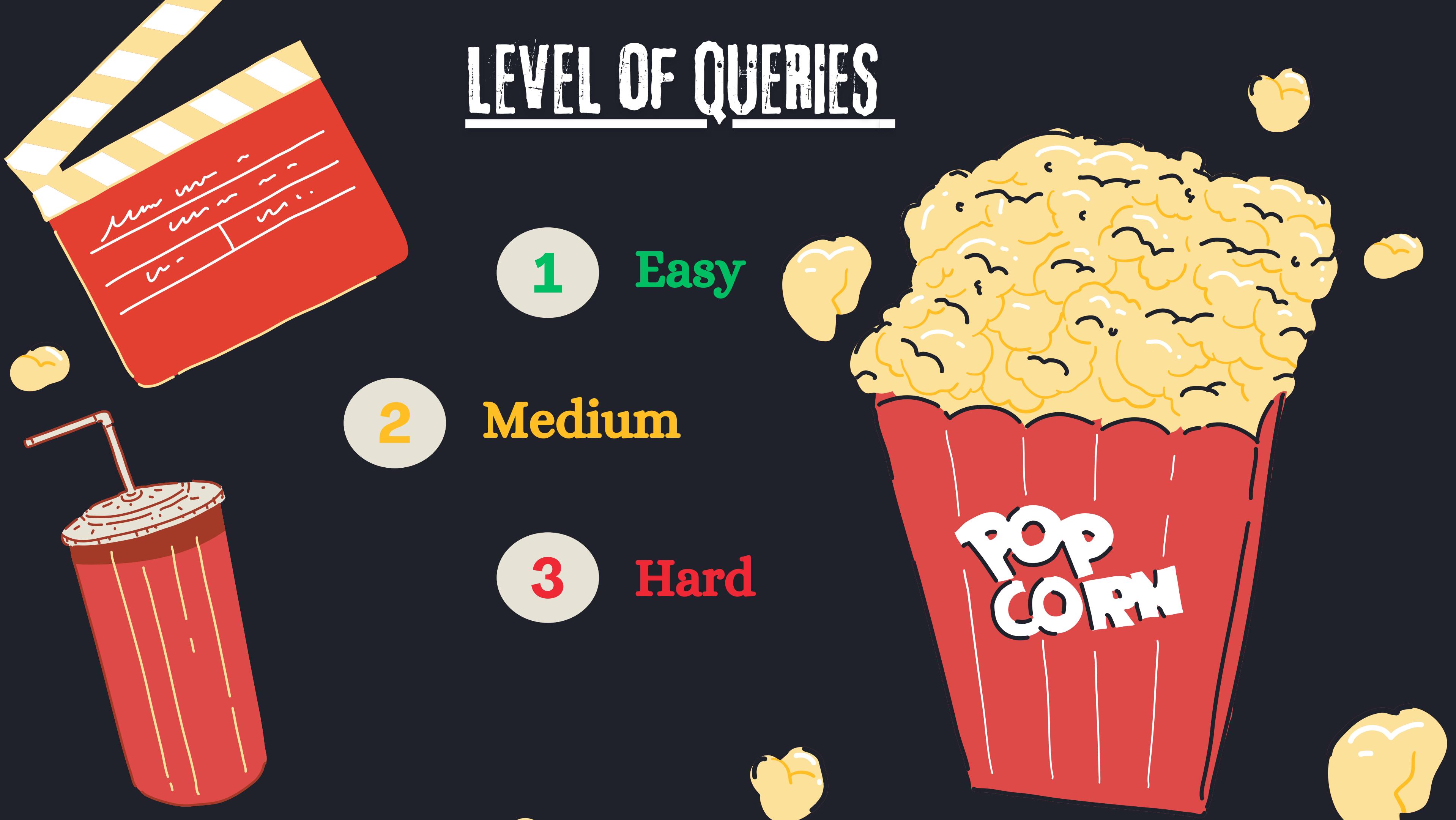
Easy

2

Medium

3

Hard



QUESTION :1

Q1: Write a SQL query to find the name and year of the movies.
Return movie title and movie release year.

```
4 • Select movie_title,year  
5   from movie;
```

	movie_title	year
▶	Khandaani Shafakhana	2019
	Gully Boy	2019
	Raees	2017
	Baadshaho	2017
	Fraud Saiyyan	2019
	Hate Story IV	2018
	OK Jaanu	2017
	Secret Superstar	2017
	Zero	2018
	Fukrey Returns	2017
	Baaghi 2	2018
	Bareilly Ki Barfi	2017
	Blackmail	2018
	Student of the Year 2	2019
	Welcome to New York	2018

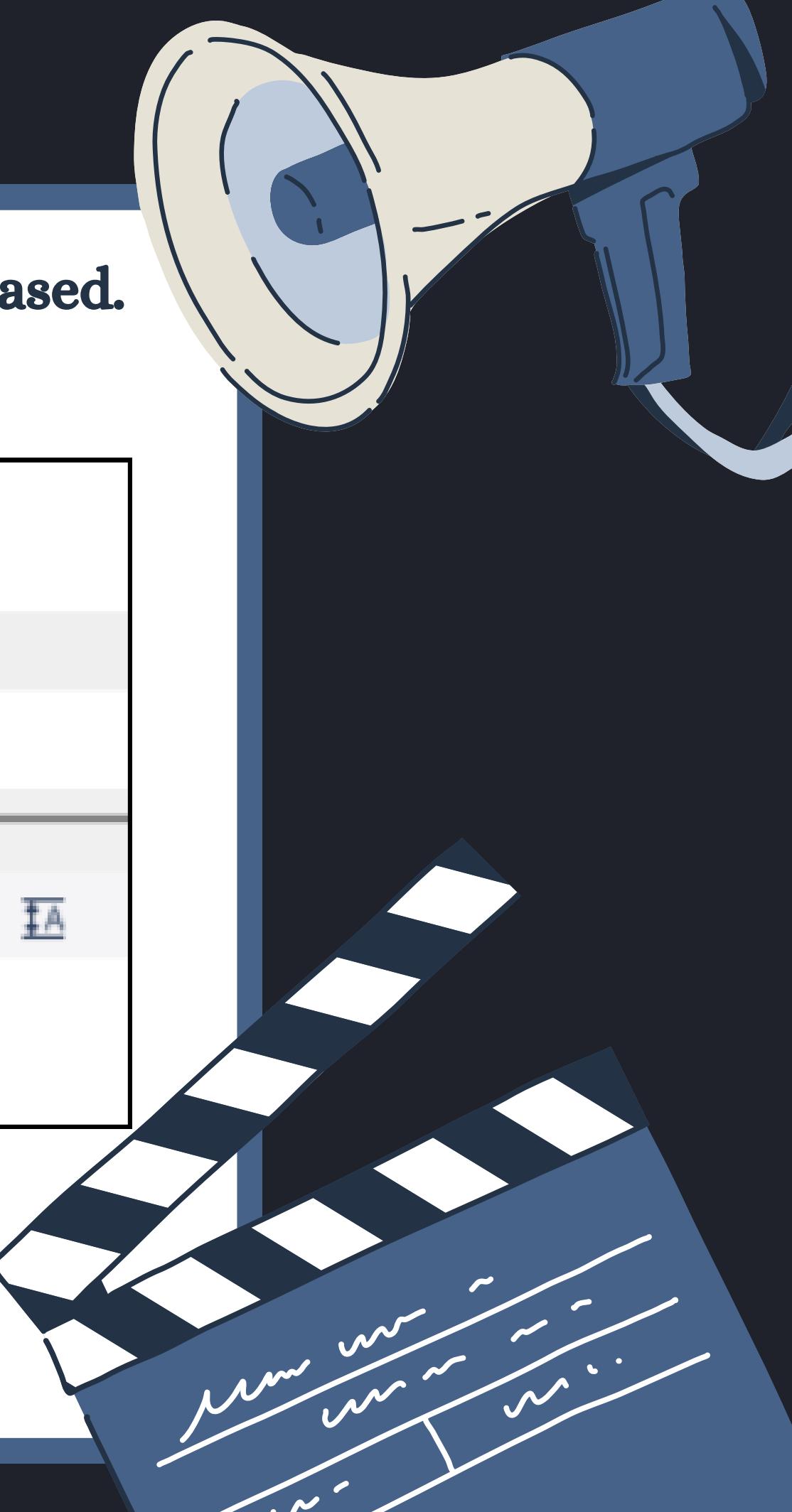
QUESTION : 2

Q2: Write a SQL query to find when the movie 'Secret Superstar' released.
Return movie release year.

```
9 •      select Movie_Title,year  
10      from movie  
11      where Movie_title = "Secret Superstar";  
12
```

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

	Movie_Title	year
▶	Secret Superstar	2017



QUESTION : 3

Q3: Write a SQL query to find the movie that was released in 2017. Return movie title.

```
16 • select movie_title,year  
17   from movie  
18   where year = 2017;  
19
```

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

	movie_title	year
▶	Raees	2017
	Baadshaho	2017
	OK Jaanu	2017
	Secret Superstar	2017
	Fukrey Returns	2017
	Bareilly Ki Barfi	2017
	Firangi	2017

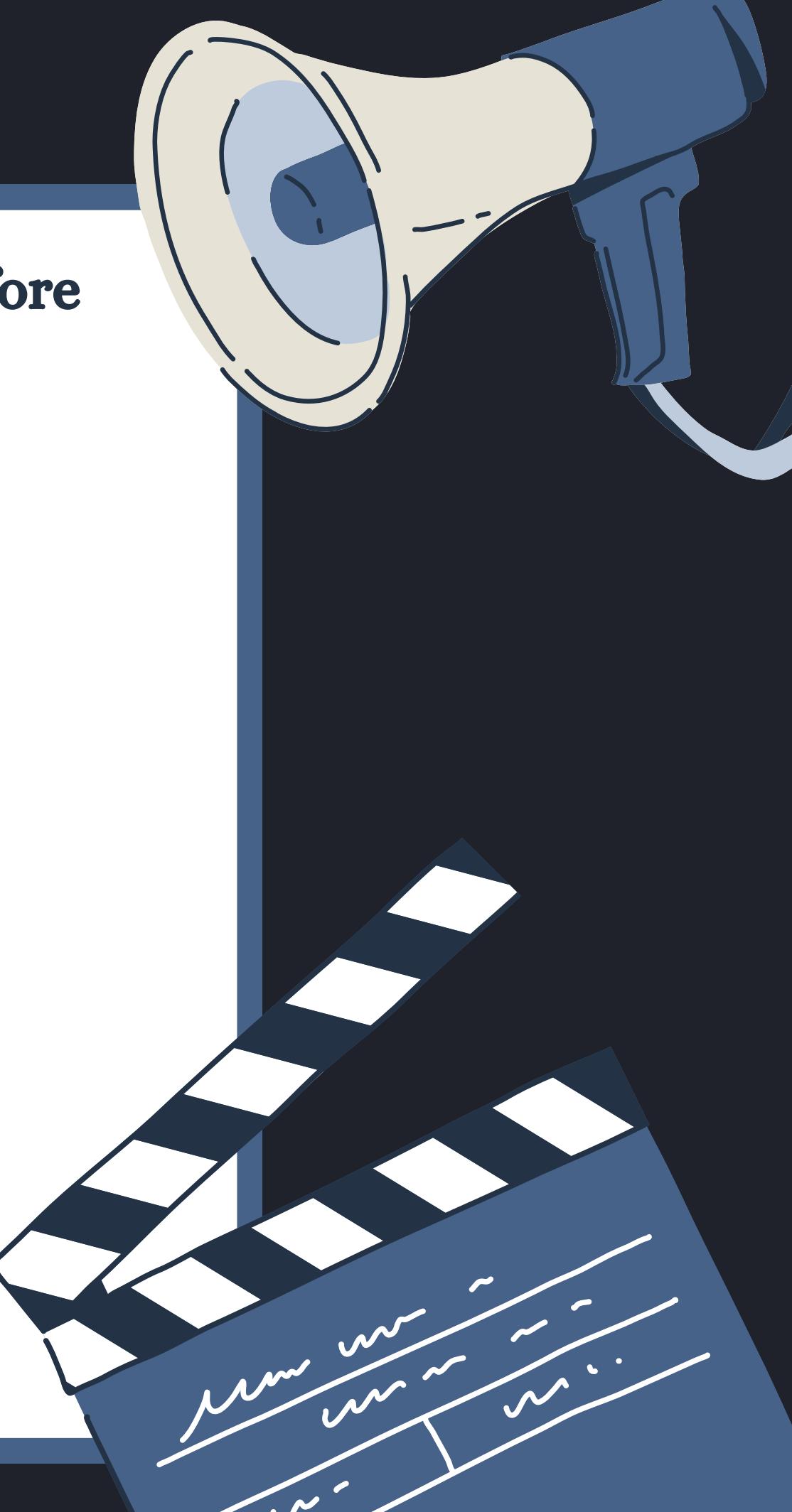
QUESTION : 4

Q4: Write a SQL query to find those movies, which were released before 2019. Return movie title.

```
21 •   select movie_title,year  
22     from movie  
23    where year < 2019  
24   order by year;
```

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

	movie_title	year
▶	Raees	2017
	Baadshaho	2017
	OK Jaanu	2017
	Secret Superstar	2017
	Fukrey Returns	2017
	Bareilly Ki Barfi	2017
	Firangi	2017
	Hate Story IV	2018
	Zero	2018
	Baaghi 2	2018
	Blackmail	2018
	Welcome to New York	2018
	Pari	2018



QUESTION : 5

Q5: Write a SQL query to find all Movie_id who have an Avg_rating seven or more stars to their rating.

```
27 •     select movie_id, Avg_rating  
28         from ratings  
29     where Avg_Rating>=7  
30     order by Avg_Rating DESC;
```

	movie_id	Avg_rating
▶	tt8291224	8.4
	tt9052870	8.3
	tt2395469	8.2
	tt7485048	8.2
	tt8130968	7.9
	tt9420648	7.9
	tt6108090	7.9
	tt6967980	7.5
	tt9248940	7.5
	tt6972140	7.0

QUESTION : 6

Q6: Write a SQL query to find the movie titles that contain the letters 'll'. Sort the result-set in ascending order by movie year. Return movie ID, movie title and movie release year.

```
34 •      select movie_id,movie_title,year  
35          from movie  
36         where Movie_Title like "%ll%"  
37         order by year asc;
```



The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:

```
34 •      select movie_id,movie_title,year  
35          from movie  
36         where Movie_Title like "%ll%"  
37         order by year asc;
```
- Result Grid:

	movie_id	movie_title	year
▶	tt6967980	Bareilly Ki Barfi	2017
	tt2395469	Gully Boy	2019
	tt8108196	Judgementall Hai Kya	2019



QUESTION : 7

Q7: Write a SQL query to find those actors with the first name 'A' and the last name 'A' and the Movies_id they are known for. Return actor ID.

```
42 • select name_id, name, Known_For_Movies  
43   from names  
44   where name like "A%A";
```

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

	name_id	name	Known_For_Movies
▶	nm0788855	Abhishek Sharma	tt8304386
	nm3189957	Anurag Arora	tt7431594, tt3405236
	nm6269780	Aparshakti Khurana	tt8908002
	nm9526395	Aayush Sharma	tt7820846

QUESTION : 8

08: Write a SQL query to find the movies with ID 'tt3405236' or 'tt5764024' or 'tt7820846'. Return movie title.

```
47 • select movie_title  
48   from movie  
49 where movie_id in("tt3405236","tt5764024","tt7820846");
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	movie_title			
▶	Raees			
	OK Jaanu			
	Loveyatri			



QUESTION : 9

Q-9 What genres of movies are their in the data set?

```
54 • select distinct(genre) as Name_of_Genre  
55   from Genre;  
56
```

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

	Name_of_Genre
▶	Thriller
▶	Mystery
▶	Comedy
▶	Romance
▶	Drama

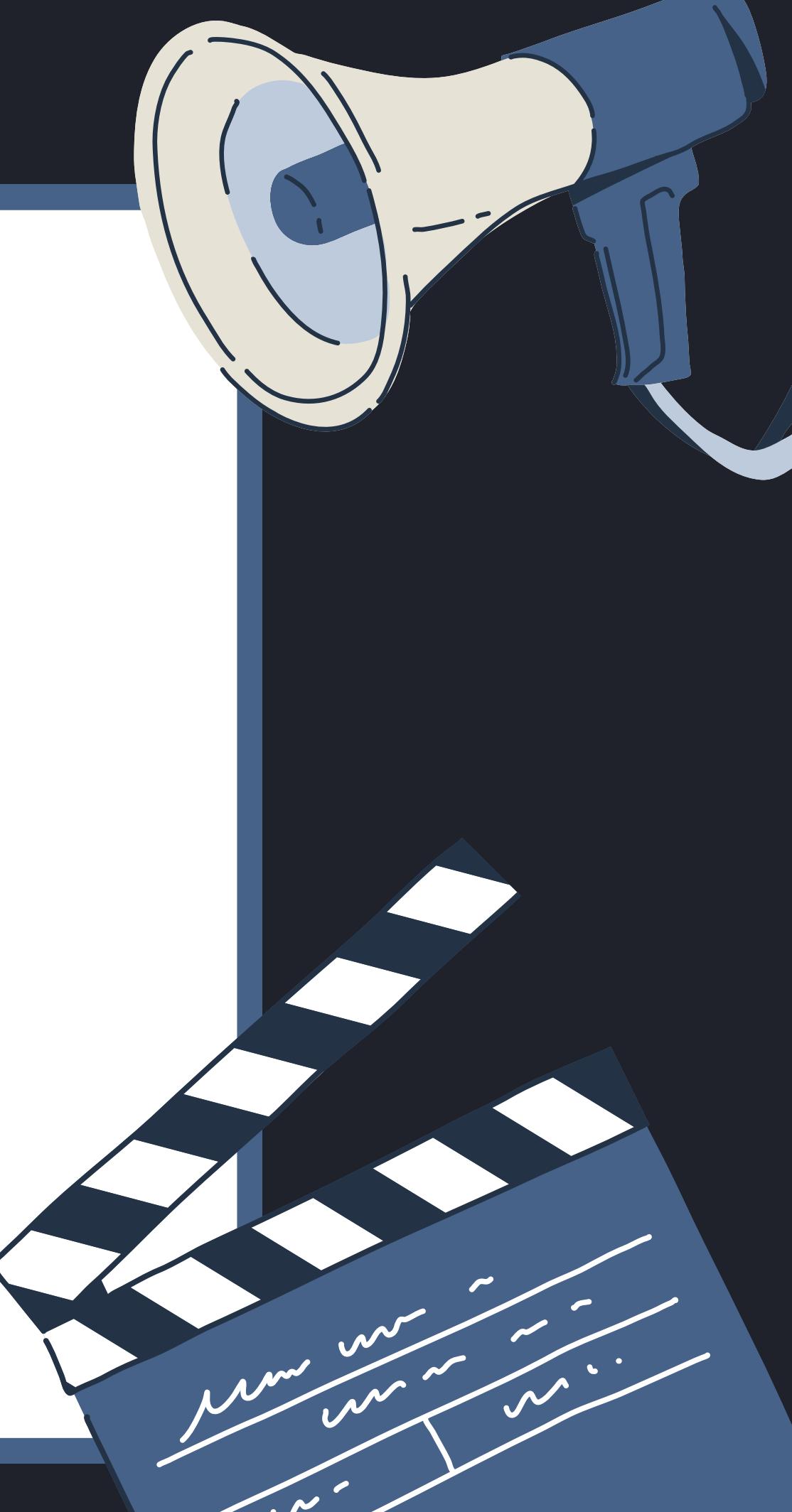
QUESTION : 10

Q-10 Show the avg_rating of each movie. Sort the result-set in descending order by avg_rating.

```
59 •     select movie_id,avg_rating  
60         from ratings  
61     order by avg_rating desc;
```

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

	movie_id	avg_rating
▶	tt8291224	8.4
	tt9052870	8.3
	tt2395469	8.2
	tt7485048	8.2
	tt8130968	7.9
	tt9420648	7.9
	tt6108090	7.9
	tt6967980	7.5
	tt9248940	7.5



QUESTION :11

Q-11 Find the total number of movies released each year?

```
4 • select count(movie_id) as Total_No_of_Movies,year  
5   from Movie  
6   group by year;
```



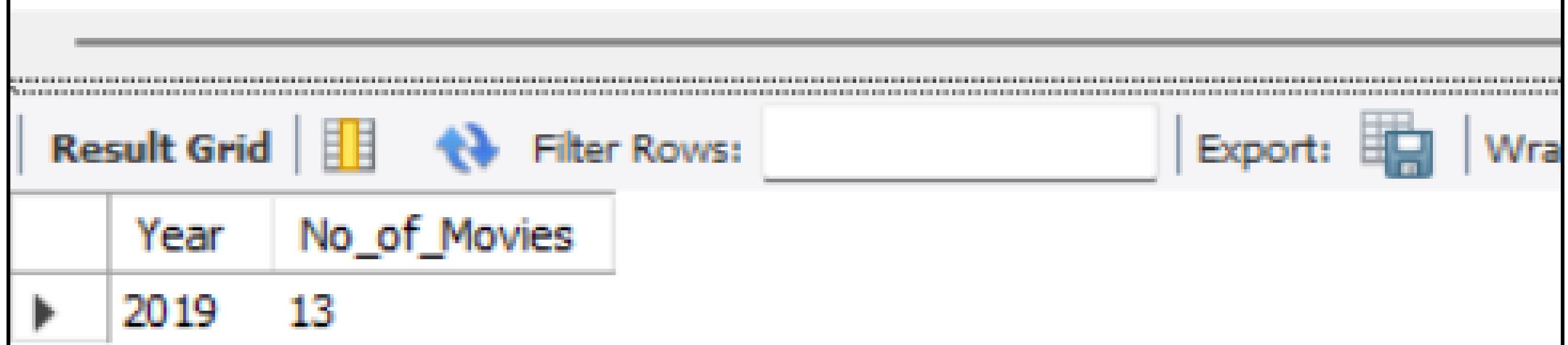
The screenshot shows a MySQL query results interface. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', 'Export:' (with a CSV icon), and 'Wrap Cell Content:'. The result grid displays the following data:

	Total_No_of_Movies	year
▶	13	2019
	7	2017
	11	2018

QUESTION : 12

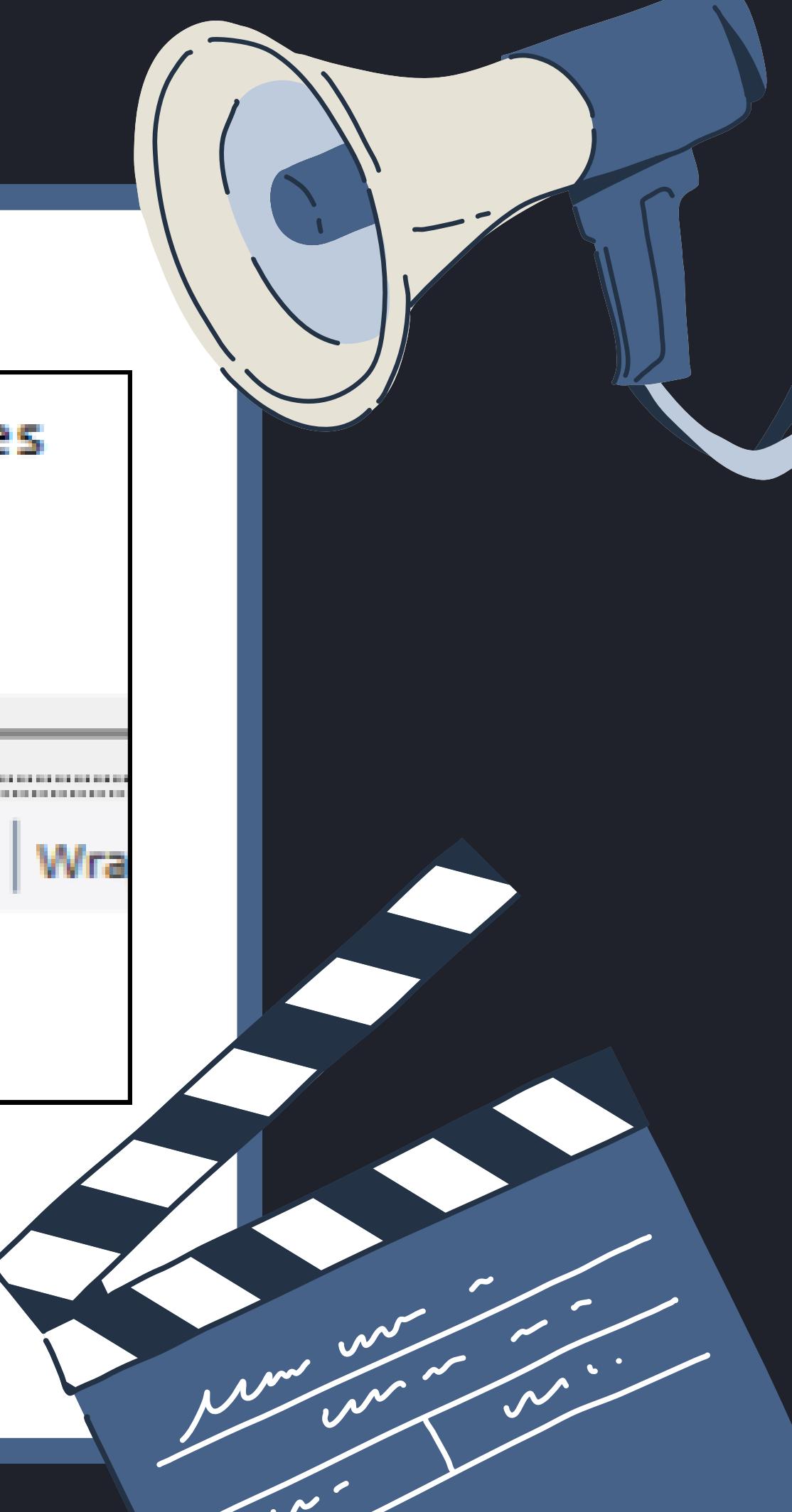
Q-12 How many movies were produced in the year 2019?

```
10 •      select Year,count(movie_id) as No_of_Movies  
11       from movie  
12       where year = 2019;
```



The screenshot shows a MySQL query results interface. At the top, there are buttons for 'Result Grid' (selected), 'Filter Rows:', and 'Export:'.

Year	No_of_Movies
2019	13



QUESTION : 13

Q-13 Which genre had the highest number of movies produced overall?

```
17 •     select count(movie_id) as Total_No_of_Movies,Genre  
18      from Genre  
19      group by genre  
20      order by Total_No_of_Movies desc  
21      limit 1;
```

Result Grid | Filter Rows: Export: Wrap Cell Content

	Total_No_of_Movies	Genre
▶	11	Drama

QUESTION : 14

Q-14 What is the average duration of movies in each genre?

```
34 • select Genre,round(avg(duration),2) as Avg_Duration_of_Movies  
35   from Movie as M  
36   Join Genre as G  
37   on M.Movie_Id = G.Movie_Id  
38   group by Genre;
```



Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	Genre	Avg_Duration_of_Movies		
▶	Thriller	137.57		
	Mystery	127.00		
	Comedy	137.00		
	Romance	136.78		
	Drama	140.18		

QUESTION : 15

Q-15 Find the minimum and maximum values in each column of the ratings table except the movie_id column.

```
41 • select max(Avg_Rating) as Max_AvgRating, Min(Avg_Rating) as Min_AvgRating,  
42   max(Total_Votes) as Max_Votes, min(Total_Votes) as Min_Votes,  
43   Max(Median_Rating) as Max_MedianRating, Min(Median_Rating) as Min_MedianRating  
44   from ratings;
```

	Max_AvgRating	Min_AvgRating	Max_Votes	Min_Votes	Max_MedianRating	Min_MedianRating
▶	8.4	1.9	37644	388	10	1

QUESTION : 16

Q- 16 How many movies were released in each genre during March 2018 and had votes of more than 1,000?

```
52 • Select G.Genre,count(M.Movie_Id) as No_of_Movies  
53     from Movie as M  
54     Join Genre as G  
55     on M.Movie_Id = G.Movie_Id  
56     Join ratings as R  
57     on M.Movie_Id = R.Movie_Id  
58     where M.year = 2018 and month(M.date_published)=3 and R.Total_Votes>1000  
59     group by G.genre;
```

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

	Genre	No_of_Movies
▶	Thriller	2
	Mystery	1



QUESTION :17

Q-17 Find movies of each genre that start with the word 'Ba' and which have an average rating > 7.

```
61 • select M.Movie_Title,R.Avg_Rating  
62   from movie as M  
63   Join ratings as R  
64   on M.Movie_Id=R.Movie_Id  
65   where M.Movie_Title Like "Ba%" and R.Avg_rating>7;
```

Result Grid | Filter Rows: Export: Wrap Cell Content

	Movie_Title	Avg_Rating
▶	Badla	7.9
▶	Bala	7.9
▶	Bareilly Ki Barfi	7.5

QUESTION : 18

Q-18 No. of movies released between 1 April 2018 and 1 April 2019, how many were given a median rating of 8?

```
73 • select count(M.Movie_Id) as No_of_Movies_Released,R.Median_Rating  
74     from movie as m  
75     join ratings as R  
76     where M.date_published between "2018-04-01" and "2019-04-01" and Median_Rating=8  
77     group by R.Median_Rating;
```

Result Grid		Filter Rows:	Export:	Wrap Cell Content:
	No_of_Movies_Released	Median_Rating		
▶	72	8		

QUESTION : 19

Q-19 Summarise the ratings table based on the movie counts by median ratings.

```
81 • select Median_Rating, count(movie_id) as Movie_Count  
82   from ratings  
83   group by median_rating  
84   order by Median_Rating Desc;
```

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

	Median_Rating	Movie_Count
▶	10	1
	9	4
	8	6
	7	7
	6	4
	5	2
	4	1
	3	2
	1	4

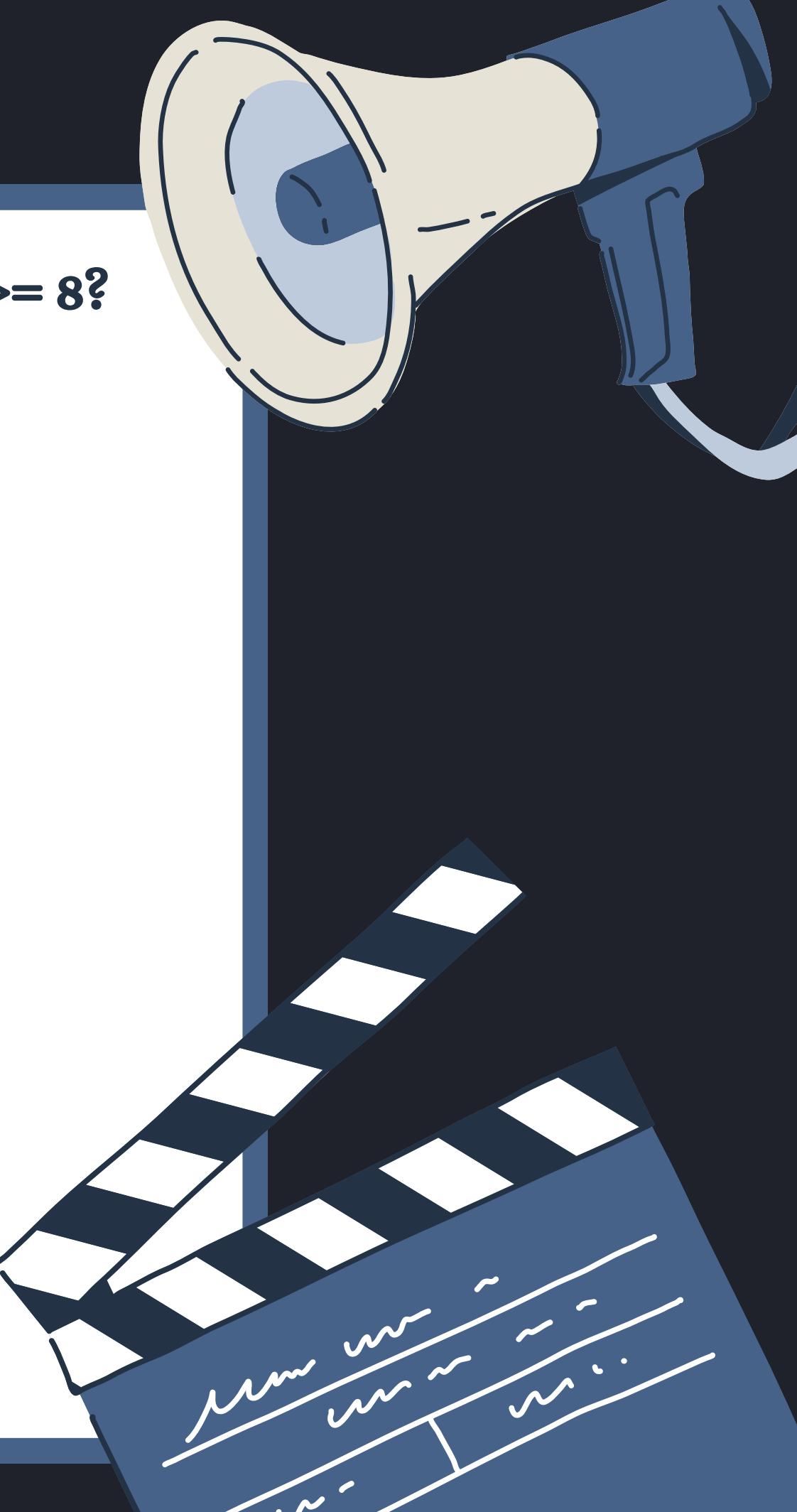
QUESTION : 20

Q-20 Who are the top two actors whose movies have a median rating ≥ 8 ?

```
92 •      select n.name as Actor_Name, count(rm.movie_id) as Movie_Count
93       from role_mapping as rm
94       join names as n
95       on rm.movie_id = n.Known_For_Movies
96       join ratings as r
97       on r.movie_id=rm.movie_id
98       where category = "actor" and median_rating >=8
99       group by n.name
100      order by movie_count desc
101      limit 2;
```

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

	Actor_Name	Movie_Count
▶	Uday Tikekar	4
	Yug	4

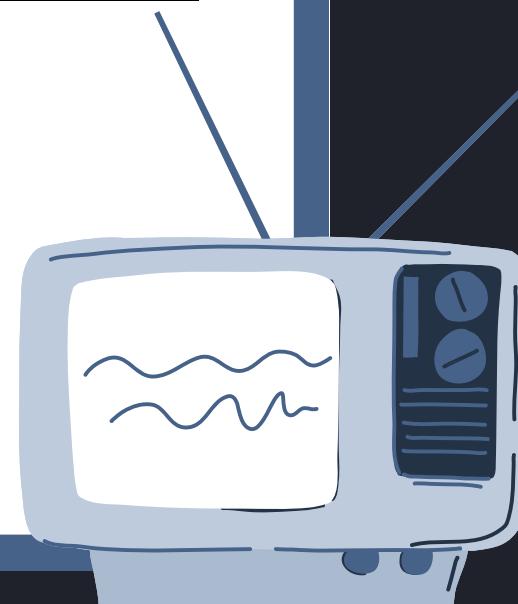


QUESTION : 21

Q-21 What is the rank of the 'thriller' genre of movies among all the genres in terms of number of movies produced?

```
3 • With Genre_Thriller_Rank as
4   (Select Genre,Count(Movie_Id)as No_of_Movies,Dense_rank() over(order by count(Movie_Id) desc) as Genre_Rank
5     from genre
6     group by Genre)
7   select* from Genre_Thriller_Rank
8   where Genre = "Thriller";
```

Result Grid			
	Genre	No_of_Movies	Genre_Rank
▶	Thriller	7	3



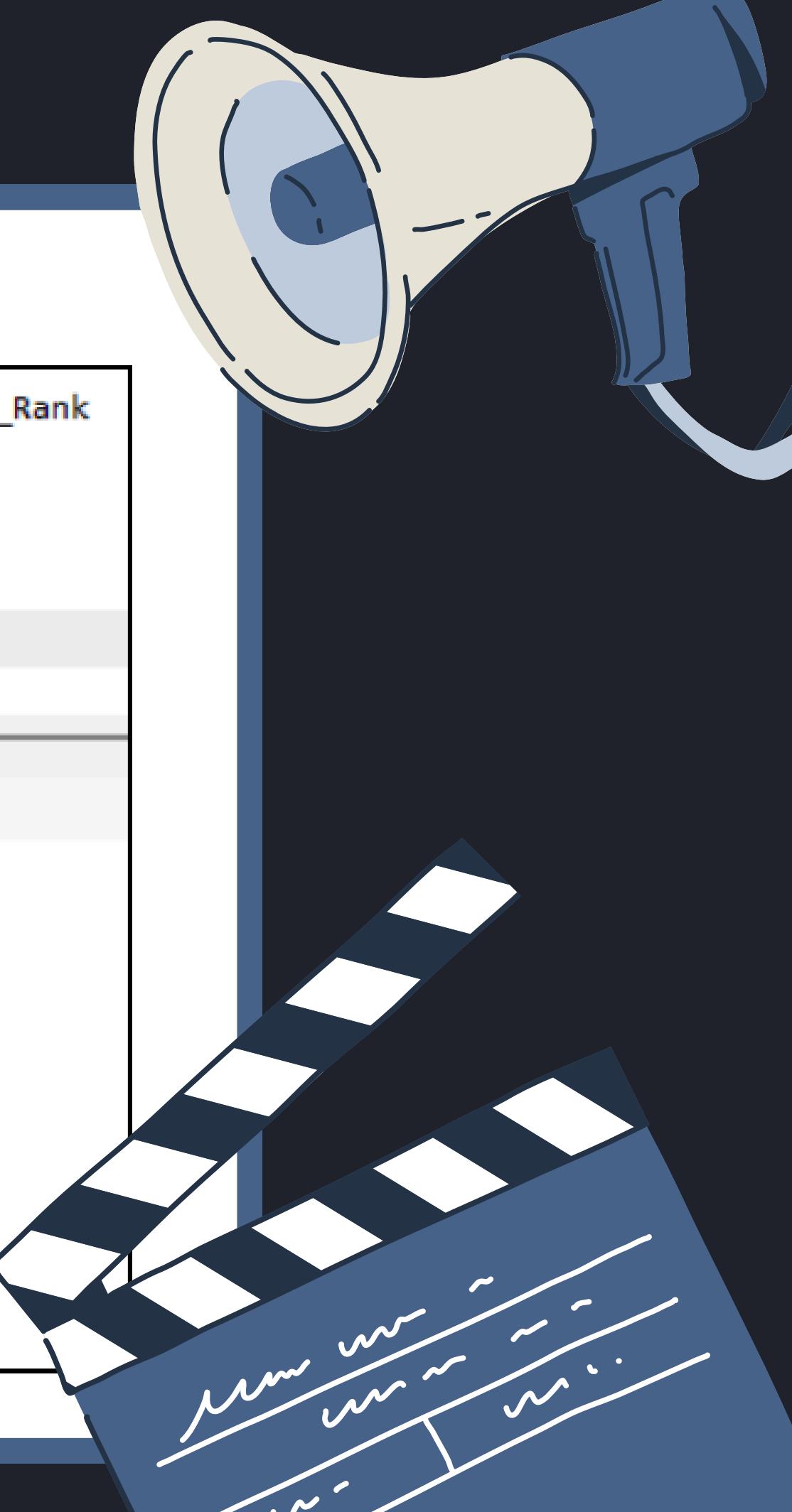
QUESTION : 22

Q-22 Which are the top 10 movies based on average rating?

```
14 • select Movie_Title,Avg_Rating,Dense_Rank() Over(order by Avg_Rating desc) as Movie_Rank  
15   from ratings as R  
16   Join Movie as M  
17   on M.Movie_Id = R.Movie_Id  
18   limit 10;
```

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

	Movie_Title	Avg_Rating	Movie_Rank
	Uri: The Surgical Strike	8.4	1
	Chhichhore	8.3	2
	Gully Boy	8.2	3
	Super 30	8.2	3
	Badla	7.9	4
	Bala	7.9	4
	Secret Superstar	7.9	4
	Bareilly Ki Barfi	7.5	5
	Dream Girl	7.5	5
▶	Blackmail	7.0	6



QUESTION : 23

Q-23 Which production house has produced the most number of hit movies (average rating > 7)?

```
23 • select Production_Company,Avg_Rating,count(M.Movie_Id) as No_Of_Movies,  
24   dense_rank() over(order by count(M.Movie_id) desc) as Production_Company_Rank  
25   from movie as M  
26   join ratings as R  
27   on M.Movie_Id = R.Movie_Id  
28   where R.Avg_Rating>7  
29   group by M.Production_Company,Avg_Rating;
```

Result Grid				
	Production_Company	Avg_Rating	No_Of_Movies	Production_Company_Rank
▶	Azure Entertainment	7.9	1	1
	Maddock Films	7.9	1	1
	B.R. Studios	7.5	1	1
	Fox STAR Studios	8.3	1	1
	Balaji Motion Pictures	7.5	1	1
	Excel Entertainment	8.2	1	1
	Aamir Khan Productions	7.9	1	1
	HRX Films	8.2	1	1
	Bulb Chamka	8.4	1	1

QUESTION : 24

Q-24 Which are the top three production houses based on the number of votes received by their movies?

```
62 • select M.Production_Company,sum(Total_Votes) as No_of_Votes,  
63 Dense_rank() Over(Order by sum(Total_Votes) desc) as Production_Company_Rank  
64 from movie as M  
65 Join ratings as R  
66 on M.Movie_Id = R.Movie_Id  
67 group by Production_Company  
68 order by No_of_Votes desc  
69 limit 3;
```

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

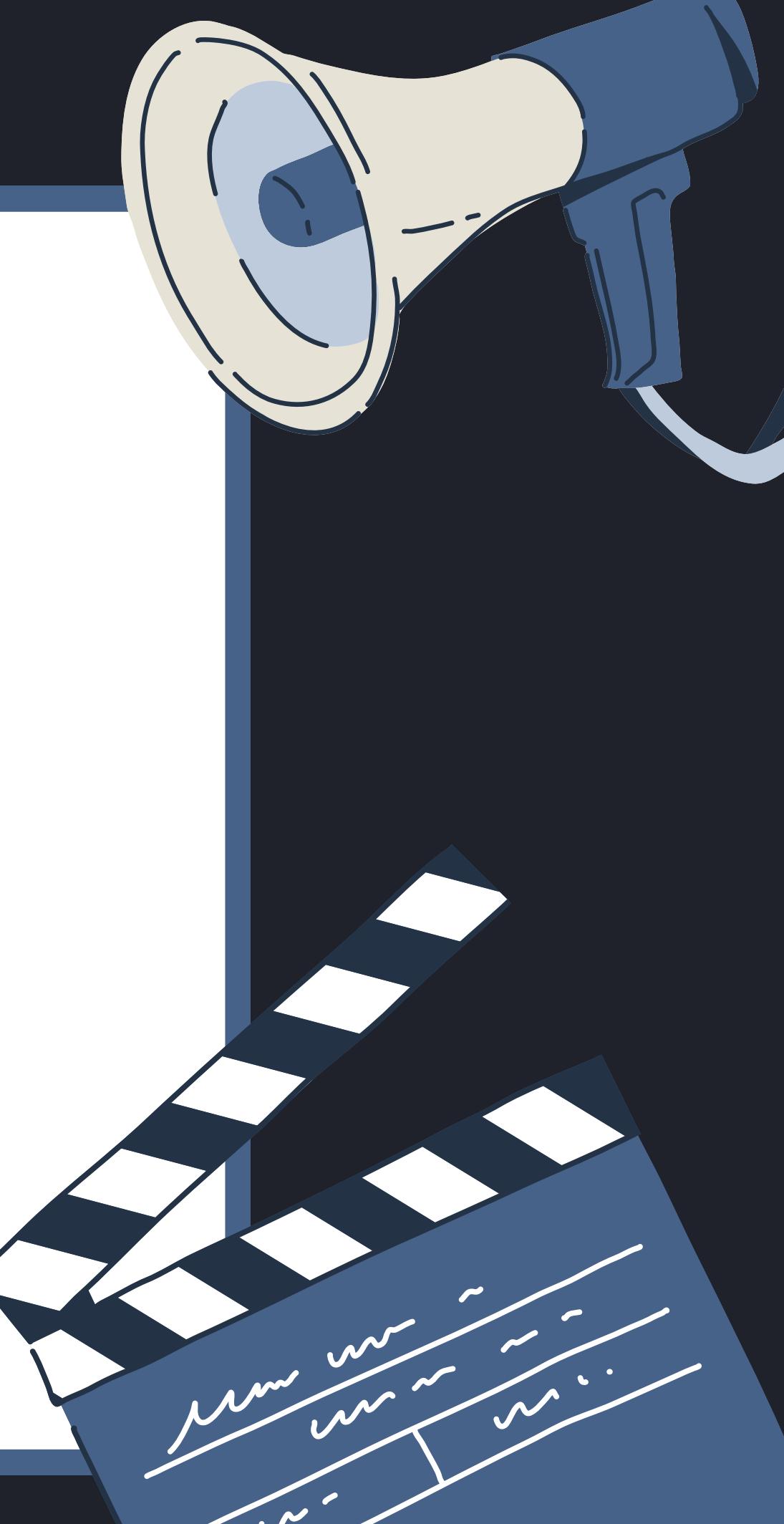
	Production_Company	No_of_Votes	Production_Company_Rank
▶	Excel Entertainment	64352	1
	Bulb Chamka	35278	2
	Salman Khan Films	31831	3

QUESTION : 25

Q-25 Rank actors based on their average ratings.

```
78 • WITH rank_actors AS ( SELECT NAME AS actor_name, Sum(total_votes) AS total_votes, Count(RM.movie_id) AS movie_count,
79     Round(Sum(avg_rating * total_votes) / Sum(total_votes), 2) AS actor_avg_rating
80     FROM role_mapping RM
81     JOIN names N
82     ON RM.name_id = N.name_id
83     JOIN ratings R
84     ON RM.movie_id = R.movie_id
85     JOIN movie M
86     ON RM.movie_id = M.movie_id
87     where categpry = "actor"
88     group by actor_name)
89     SELECT *, DENSE_RANK() OVER (ORDER BY actor_avg_rating DESC) AS actor_rank FROM rank_actors;
```

Result Grid					
	actor_name	total_votes	movie_count	actor_avg_rating	actor_rank
▶	Mohit Raina	35278	1	8.40	1
	Paresh Rawal	35278	1	8.40	1
	Vicky Kaushal	35278	1	8.40	1
	Siddhant Chaturvedi	22593	1	8.20	2
	Nandish Singh	14207	1	8.20	2
	Hrithik Roshan	14207	1	8.20	2
	Raj Bhansali	4145	1	7.50	3
	Raj Arjun	55388	2	7.15	4
	Yug	37644	1	6.80	5
	Shubham Tukaram	37644	1	6.80	5
	Shubham Chintamani	37644	1	6.80	5



QUESTION : 26

Q-26 Find out the top five actress's movies based on their average ratings.

```
89 • select name as Actress_Name,Sum(Total_Votes) as Total_Votes,count(M.Movie_Id) as Movie_count,  
90 round(sum(avg_rating*total_votes)/sum(total_votes),2) as Actress_Avg_Rating,  
91 Dense_rank() over(order by round(sum(avg_rating*total_votes)/sum(total_votes),2) desc) as Actress_Rank  
92 from role_mapping as RM  
93 join names as N  
94 on RM.name_id = N.name_id  
95 join ratings as R  
96 on RM.Movie_id = R.Movie_id  
97 join movie as M  
98 on RM.Movie_id = M.Movie_id  
99 where categpry = "Actress"  
100 group by Actress_Name  
101 LIMIT 5;
```

	Actress_Name	Total_Votes	Movie_count	Actress_Avg_Rating	Actress_Rank
▶	Yami Gautam	37794	2	8.37	1
	Meher Vij	17744	1	7.90	2
	Zaira Wasim	17744	1	7.90	2
	Kriti Sanon	21277	2	7.14	3
	Kirti Kulhari	7210	1	7.00	4

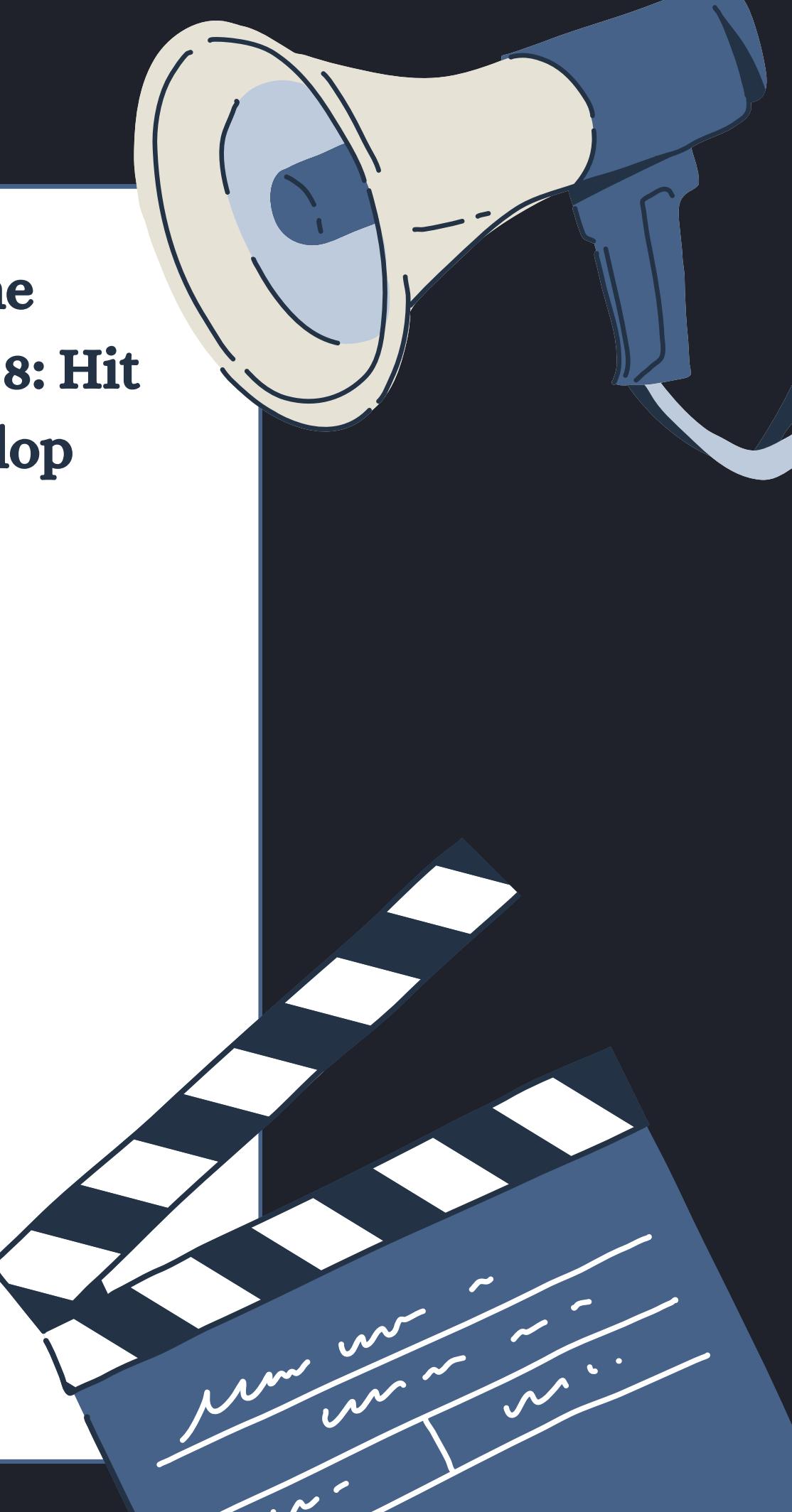
QUESTION : 27

Q-27 Select drama movies as per average 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

```
111 •      select Movie_Title,Avg_rating,  
112   case  
113     when avg_rating > 8 then "Superhit Movies"  
114     when avg_rating between 7 and 8 then "Hit Movies"  
115     when avg_rating between 5 and 7 then "One-Time-Watch Movies"  
116     when avg_rating < 5 then "Flop Movies"  
117   end as "Movies Status"  
118   from ratings as R  
119   join movie as M  
120   on R.Movie_Id = M.Movie_id  
121   join genre as G  
122   on R.Movie_Id = G.Movie_Id  
123   where genre = "Drama"  
124   order by Avg_rating desc;  
125
```

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

Movie_Title	Avg_rating	Movies Status
Uri: The Surgical Strike	8.4	Superhit Movies
Chhichhore	8.3	Superhit Movies
Gully Boy	8.2	Superhit Movies
Super 30	8.2	Superhit Movies
Secret Superstar	7.9	Hit Movies
Raees	6.8	One-Time-Watch Movies
Khandaani Shafakhana	4.7	Flop Movies

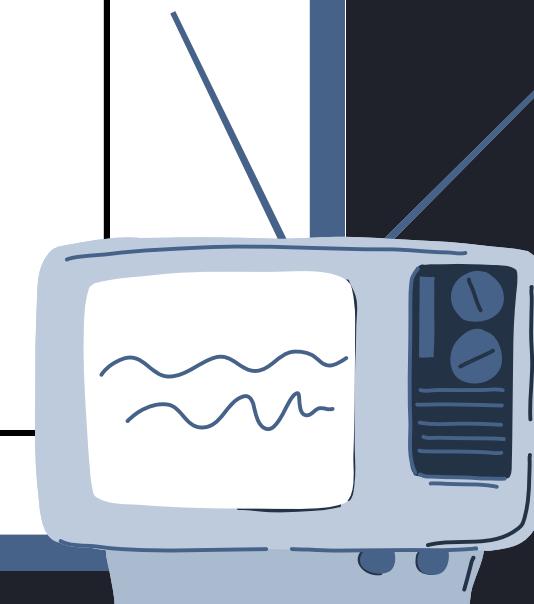


QUESTION : 28

Q-28 What is the genre-wise running total and moving average of the average movie duration?

```
128 •   SELECT genre, ROUND(AVG(duration),2) AS avg_duration,  
129     SUM(AVG(duration)) OVER(ORDER BY genre) AS running_total_duration,  
130     AVG(AVG(duration)) OVER(ORDER BY genre) AS moving_avg_duration  
131   FROM movie m  
132   JOIN genre g  
133   ON m.movie_id = g.movie_id  
134   GROUP BY genre;
```

	genre	avg_duration	running_total_duration	moving_avg_duration
▶	Comedy	137.00	137.0000	137.00000000
	Drama	140.18	277.1818	138.59090000
	Mystery	127.00	404.1818	134.72726667
	Romance	136.78	540.9596	135.23990000
	Thriller	137.57	678.5310	135.70620000

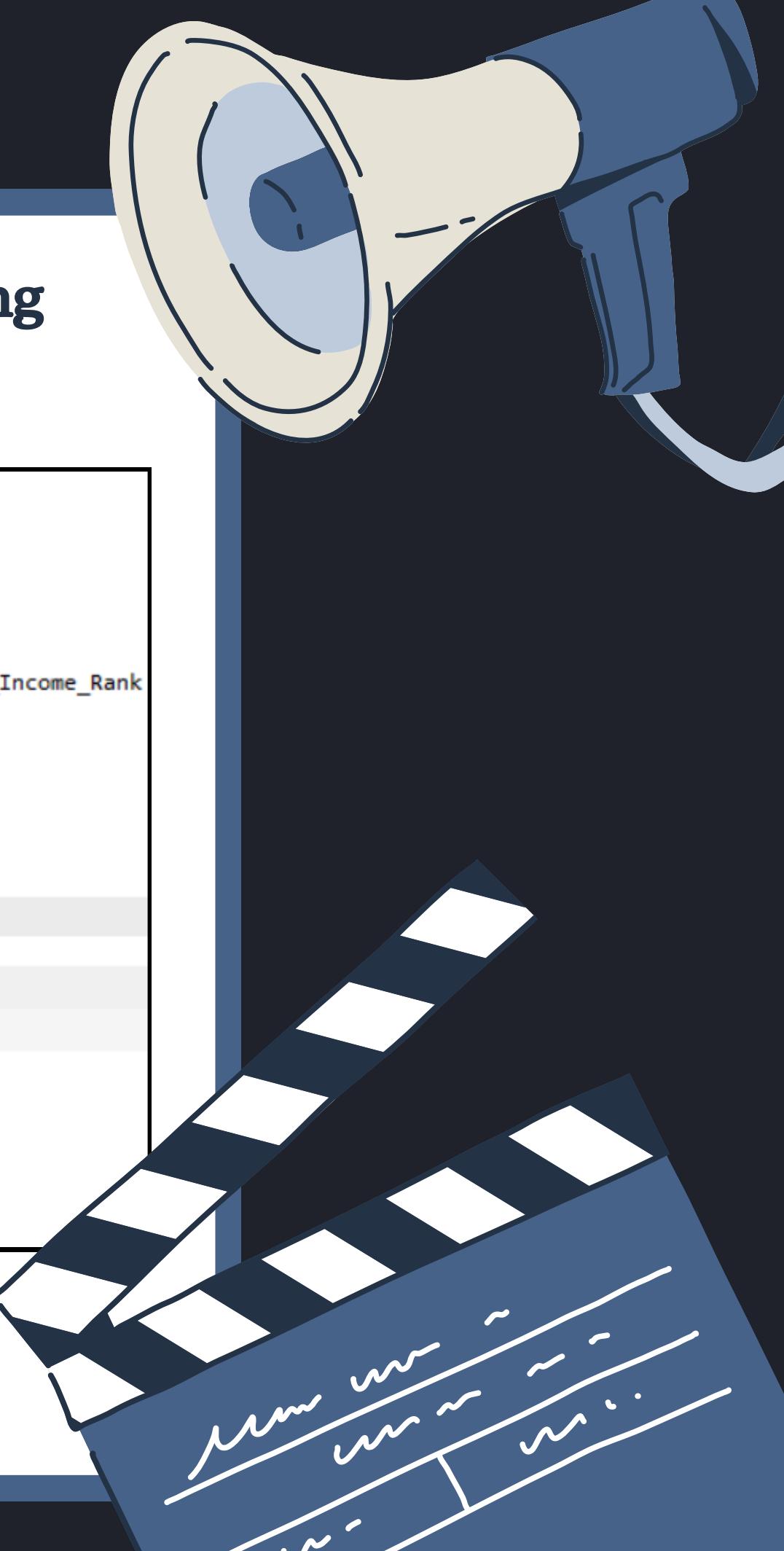


QUESTION : 29

Q-29 Which are the five highest-grossing movies of each year that belong to the top two genres?

```
140 • With Top_2_Genre as (
141     Select genre,count(Movie_id) as Movie_Count,rank() over (order by count(Movie_Id) desc) as genre_rank
142     from genre
143     group by genre
144     limit 2),
145     Movies_Rank As (Select Genre,Movie_Title,Year,WorldWide_Gross_Income,rank() over(order by worldwide_gross_income desc) as Worldwide_Income_Rank
146     from movie as M
147     Join genre as G
148     on M.movie_Id = G.movie_Id
149     where genre in (select genre from Top_2_Genre))
150
151     SELECT * FROM Movies_Rank WHERE Worldwide_Income_Rank<=5;
152
```

Result Grid				
Genre	Movie_Title	Year	WorldWide_Gross_Income	WorldWide_Income_Rank
Drama	Raees	2017	\$ 8428190	1
Drama	Khandaani Shafakhana	2019	\$ 67534	2
Romance	Dhadak	2018	\$ 549628	3
Romance	OK Jaanu	2017	\$ 5405901	4
Drama	Super 30	2019	\$ 3953337	5



QUESTION : 30

Q-30 Which are the top two production houses that have produced the highest number of hits (median rating ≥ 8) ?

```
156 • select Production_Company,Count(M.Movie_Id) as Movie_Count,  
157     Row_Number() over(order by count(M.Movie_Id) desc) as Prod_Comp_Rank,Median_Rating  
158     from Movie as M  
159     Join Ratings as R  
160     on M.movie_id = R.movie_id  
161     where median_rating >=8  
162     group by Production_Company,Median_Rating  
163     limit 3;  
164
```

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

	Production_Company	Movie_Count	Prod_Comp_Rank	Median_Rating
▶	Azure Entertainment	1	1	9
	Maddock Films	1	2	8
	B.R. Studios	1	3	8

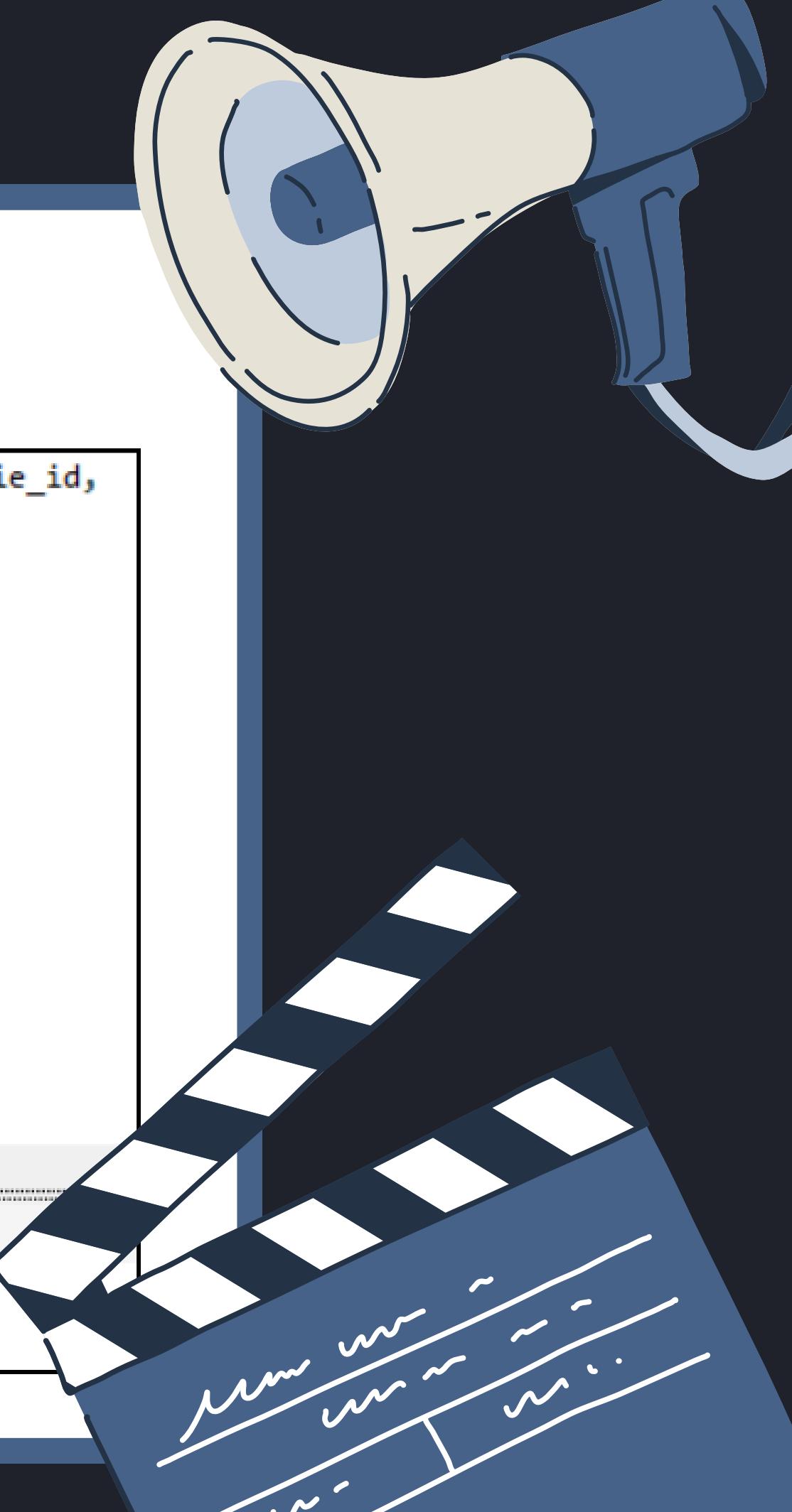
QUESTION : 31

Q-31 Who are the top actress based on a number of Super Hit movies (average rating >=8) in the drama genre?

```
168 •   SELECT name as actress_name, SUM(total_votes) AS total_votes, COUNT(rm.movie_id) as movie_id,
169     Round(Sum(avg_rating * total_votes)/Sum(total_votes),2) AS actress_avg_rating,
170     DENSE_RANK() OVER(ORDER BY COUNT(rm.movie_id) DESC) AS actress_rank
171     FROM names as n
172     JOIN role_mapping rm
173     ON n.name_id = rm.name_id
174     JOIN ratings r
175     ON r.movie_id = rm.movie_id
176     JOIN genre g
177     ON g.movie_id = r.movie_id
178     WHERE categpry="actress" AND avg_rating>=8 AND g.genre="Drama"
179     GROUP BY name;
```

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

	actress_name	total_votes	movie_id	actress_avg_rating	actress_rank
▶	Yami Gautam	35278	1	8.40	1





THANK YOU