Segment 1:

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

-- Type your code below:

```
16
17 •  SELECT table_name, table_rows
18    FROM information_schema.tables
19    WHERE table_schema = 'imdb';
20
```

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

TABLE_NAME	TABLE_ROWS
director_mapping	3867
genre	14662
movie	9926
names	28220
ratings	7927
role_mapping	14818

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

-- Type your code below:

```
30 •  SELECT
31      COUNT(*) AS total_rows,
32      SUM(CASE WHEN title IS NULL THEN 1 ELSE 0 END) AS null_title,
33      SUM(CASE WHEN year IS NULL THEN 1 ELSE 0 END) AS null_year,
34      SUM(CASE WHEN date_published IS NULL THEN 1 ELSE 0 END) AS null_date_published,
35      SUM(CASE WHEN duration IS NULL THEN 1 ELSE 0 END) AS null_duration,
36      SUM(CASE WHEN country IS NULL THEN 1 ELSE 0 END) AS null_country,
37      SUM(CASE WHEN worlwide_gross_income IS NULL THEN 1 ELSE 0 END) AS null_worlwide_gross_income,
38      SUM(CASE WHEN languages IS NULL THEN 1 ELSE 0 END) AS null_languages,
39      SUM(CASE WHEN production_company IS NULL THEN 1 ELSE 0 END) AS null_production_company
40    FROM movie;
41
42
```

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

total_rows	null_title	null_year	null_date_published	null_duration	null_country	null_worlwide_gross_income	null_languages	null_production_company
7997	0	0	0	0	20	3724	194	528

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

```
71      -- Part 1: Movies released each year
72 •    SELECT year, COUNT(*) AS number_of_movies
73      FROM movie
74      GROUP BY year
75      ORDER BY year;
76
```

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

year	number_of_movies
2017	3052
2018	2944
2019	2001

```
77      -- Part 2: Month-wise trend
78 •    SELECT MONTH(date_published) AS month_num, COUNT(*) AS number_of_movies
79      FROM movie
80      GROUP BY MONTH(date_published)
81      ORDER BY month_num;
82
```

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

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

Q4. How many movies were produced in the USA or India in the year 2019?

```
95    SELECT COUNT(*) AS movies_count
96    FROM imdb.movie
97    WHERE country IN ('USA', 'India') AND year = 2019;
98
99
```

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

movies_count
887

-- Q5. Find the unique list of the genres present in the data set?

```
116   SELECT DISTINCT genre
117   FROM genre
118   ORDER BY genre;
```

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

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

Q6. Which genre had the highest number of movies produced overall?

```sql
SELECT genre, COUNT(*) AS movie_count
FROM genre
GROUP BY genre
ORDER BY movie_count DESC
LIMIT 1;
```

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

genre	movie_count
Drama	4285

Q7. How many movies belong to only one genre?

```sql
SELECT COUNT(*) AS single_genre_movies
FROM (
  SELECT movie_id
  FROM genre
  GROUP BY movie_id
  HAVING COUNT(genre) = 1
) AS single_genre_list;
```

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

single_genre_movies
3289

Q8. What is the average duration of movies in each genre?

```
191 •   SELECT g.genre, ROUND(AVG(m.duration), 2) AS avg_duration
192     FROM genre g
193     JOIN movie m ON g.movie_id = m.id
194     GROUP BY g.genre
195     ORDER BY avg_duration DESC;
196
197
```

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

genre	avg_duration
Action	112.88
Romance	109.53
Crime	107.05
Drama	106.77
Fantasy	105.14
Comedy	102.62
Adventure	101.87
Mystery	101.80
Thriller	101.58
Family	100.97
Others	100.16
Sci-Fi	97.94
Horror	92.72

Q9. What is the rank of the 'thriller' genre of movies among all the genres in terms of number of movies produced?

```
216 •   SELECT genre, movie_count, genre_rank
217     FROM (
218       SELECT genre,
219              COUNT(*) AS movie_count,
220              RANK() OVER (ORDER BY COUNT(*) DESC) AS genre_rank
221       FROM genre
222       GROUP BY genre
223     ) ranked_genres
224     WHERE genre = 'Thriller';
225
226
```

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

genre	movie_count	genre_rank
Thriller	1484	3

Segment 2:

Q10. Find the minimum and maximum values in each column of the ratings table except the movie_id column?

```
255 •   SELECT
256     MIN(avg_rating) AS min_avg_rating,
257     MAX(avg_rating) AS max_avg_rating,
258     MIN(total_votes) AS min_total_votes,
259     MAX(total_votes) AS max_total_votes,
260     MIN(median_rating) AS min_median_rating,
261     MAX(median_rating) AS max_median_rating
262     FROM imdb.ratings;
263
264
265
```

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

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

Q11. Which are the top 10 movies based on average rating?

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

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

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

Q12. Summarise the ratings table based on the movie counts by median ratings.

```
316 •   SELECT median_rating, COUNT(*) AS movie_count
317     FROM ratings
318     GROUP BY median_rating
319     ORDER BY median_rating;
320
```

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

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

Q13. Which production house has produced the most number of hit movies (average rating > 8)??

```
344 •  SELECT production_company, COUNT(*) AS movie_count,
345           RANK() OVER (ORDER BY COUNT(*) DESC) AS prod_company_rank
346    FROM movie m
347    JOIN ratings r ON m.id = r.movie_id
348    WHERE r.avg_rating > 8 AND m.production_company IS NOT NULL
349    GROUP BY production_company;
```

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

production_company	movie_count	prod_company_rank
Dream Warrior Pictures	3	1
National Theatre Live	3	1
Lietuvos Kinostudija	2	3
Swadharm Entertainment	2	3
Panorama Studios	2	3
Marvel Studios	2	3
Central Base Productions	2	3
Painted Creek Productions	2	3
National Theatre	2	3
Colour Yellow Productions	2	3
The Archers	1	11
Blaze Film Enterprises	1	11
Bradeway Pictures	1	11
Bert Marcus Productions	1	11
A Studios	1	11
Ronk Film	1	11
Benaras Mediaworks	1	11
Bioscope Film Framers	1	11
Bestwin Production	1	11
Studio Green	1	11
AKS Film Studio	1	11
Kaargo Cinemas	1	11
Animonsta Studios	1	11
O3 Turkey Medya	1	11
StarVision	1	11
Synergy Films	1	11
PVP Cinema	1	11
Plan J Studios	1	11
20 Steps Productions	1	11
Prime Zero Productions	1	11
Shreya Films International	1	11
SLN Cinemas	1	11
Epiphany Entertainments	1	11
3 Ng Film	1	11
Eastpool Films	1	11
A square productions	1	11

Result 28

Q14. How many movies released in each genre during March 2017 in the USA had more than 1,000 votes?

```
372 •   SELECT g.genre, COUNT(*) AS movie_count
373     FROM movie m
374     JOIN genre g ON m.id = g.movie_id
375     JOIN ratings r ON m.id = r.movie_id
376     WHERE MONTH(m.date_published) = 3
377       AND YEAR(m.date_published) = 2017
378       AND m.country LIKE '%USA%'
379       AND r.total_votes > 1000
380     GROUP BY g.genre;
381
```

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

genre	movie_count
Action	8
Comedy	9
Crime	6
Drama	24
Sci-Fi	7
Fantasy	3
Mystery	4
Romance	4
Thriller	8
Adventure	3
Horror	6
Family	1

Q15. Find movies of each genre that start with the word 'The' and which have an average rating > 8?

```
402 •   SELECT m.title, r.avg_rating, g.genre
403     FROM movie m
404     JOIN ratings r ON m.id = r.movie_id
405     JOIN genre g ON m.id = g.movie_id
406     WHERE m.title LIKE 'The %' AND r.avg_rating > 8;
407
408
```

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 Brighton Miracle	9.5	Drama
The Irishman	8.7	Crime
The Irishman	8.7	Drama
The Colour of Darkness	9.1	Drama
The Mystery of Godliness: The Sequel	8.5	Drama
The Gambinos	8.4	Crime
The Gambinos	8.4	Drama
The King and I	8.2	Drama
The King and I	8.2	Romance

Q16. Number Of the movies released between 1 April 2018 and 1 April 2019, how many were given a median rating of 8?

```
419 •   SELECT COUNT(*) AS movie_count
420     FROM movie m
421     JOIN ratings r ON m.id = r.movie_id
422     WHERE m.date_published BETWEEN '2018-04-01' AND '2019-04-01'
423       AND r.median_rating = 8;
424
425
426
```

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

movie_count
361

Q17. Do German movies get more votes than Italian movies?

```
SELECT country_group, SUM(total_votes) AS total_votes
FROM (
  SELECT
    CASE
      WHEN country LIKE '%Germany%' THEN 'Germany'
      WHEN country LIKE '%Italy%' THEN 'Italy'
    END AS country_group,
    r.total_votes
  FROM movie m
  JOIN ratings r ON m.id = r.movie_id
  WHERE country LIKE '%Germany%' OR country LIKE '%Italy%'
) AS country_votes
GROUP BY country_group;
```

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

country_group	total_votes
Germany	2026223
Italy	613535

Segment 3:

Q18. Which columns in the names table have null values??

```
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;
```

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

name_nulls	height_nulls	date_of_birth_nulls	known_for_movies_nulls
0	17335	13431	15226

Result Grid

Q19. Who are the top three directors in the top three genres whose movies have an average rating > 8?

```sql
-- Step 1: top 3 genres with most high-rated movies (avg_rating > 8)
SELECT genre, COUNT(*) AS high_rated_movies
FROM genre g
JOIN ratings r ON g.movie_id = r.movie_id
WHERE r.avg_rating > 8
GROUP BY genre
ORDER BY high_rated_movies DESC
LIMIT 3;

-- Step 2: top 3 directors who directed movies in those top 3 genres with avg_rating > 8
SELECT n.name AS director_name, COUNT(*) AS movie_count
FROM genre g
JOIN ratings r ON g.movie_id = r.movie_id
JOIN director_mapping d ON g.movie_id = d.movie_id
JOIN names n ON d.name_id = n.id
WHERE r.avg_rating > 8
  AND g.genre IN ('Drama', 'Comedy', 'Action')  -- replace with your top 3 genres
GROUP BY n.name
ORDER BY movie_count DESC
LIMIT 3;
```

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

director_name	movie_count
James Mangold	4
Soubin Shahir	3
Joe Russo	3

Result Grid

Q20. Who are the top two actors whose movies have a median rating >= 8?

```
553 •   SELECT n.name AS actor_name, COUNT(*) AS movie_count
554     FROM role_mapping rm
555     JOIN names n ON rm.name_id = n.id
556     JOIN ratings r ON rm.movie_id = r.movie_id
557     WHERE r.median_rating >= 8 AND rm.category = 'actor'
558     GROUP BY n.name
559     ORDER BY movie_count DESC
560     LIMIT 2;
561
```

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

actor_name	movie_count
Mammootty	8
Mohanlal	5

Q21. Which are the top three production houses based on the number of votes received by their movies?

```
582 •   SELECT m.production_company, SUM(r.total_votes) AS vote_count,
583             RANK() OVER (ORDER BY SUM(r.total_votes) DESC) AS prod_comp_rank
584     FROM movie m
585     JOIN ratings r ON m.id = r.movie_id
586     WHERE m.production_company IS NOT NULL
587     GROUP BY m.production_company
588     ORDER BY vote_count DESC
589     LIMIT 3;
590
591
```

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

production_company	vote_count	prod_comp_rank
Marvel Studios	2656967	1
Twentieth Century Fox	2411163	2
Warner Bros.	2396057	3

Q22. Rank actors with movies released in India based on their average ratings. Which actor is at the top of the list?

```sql
WITH indian_actors AS (
  SELECT rm.name_id, n.name AS actor_name, r.avg_rating, r.total_votes
  FROM role_mapping rm
  JOIN names n ON rm.name_id = n.id
  JOIN ratings r ON rm.movie_id = r.movie_id
  JOIN movie m ON rm.movie_id = m.id
  WHERE rm.category = 'actor' AND m.country LIKE '%India%'
),
actor_stats AS (
  SELECT actor_name,
         COUNT(*) AS movie_count,
         SUM(total_votes) AS total_votes,
         ROUND(SUM(avg_rating * total_votes) / SUM(total_votes), 2) AS actor_avg_rating
  FROM indian_actors
  GROUP BY actor_name
  HAVING COUNT(*) >= 5
)
SELECT *, RANK() OVER (ORDER BY actor_avg_rating DESC, total_votes DESC) AS actor_rank
FROM actor_stats;
```

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

actor_name	movie_count	total_votes	actor_avg_rating	actor_rank
Vijay Sethupathi	5	23114	8.42	1
Fahadh Faasil	5	13557	7.99	2
Yogi Babu	11	8500	7.83	3
Joju George	5	3926	7.58	4
Ammy Virk	6	2504	7.55	5
Dileesh Pothan	5	6235	7.52	6
Kunchacko Boban	6	5628	7.48	7
Pankaj Tripathi	5	40728	7.44	8
Rajkummar Rao	6	42560	7.37	9
Dulquer Salmaan	5	17666	7.30	10
Amit Sadh	5	13355	7.21	11
Tovino Thomas	8	11596	7.15	12
Mammootty	8	12613	7.04	13
Nassar	5	4016	7.03	14
Karamjit Anmol	6	1970	6.91	15
Hareesh Kanaran	5	3196	6.58	16
Naseeruddin Shah	5	12604	6.54	17
Anandraj	6	2750	6.54	18
Mohanlal	7	17622	6.47	19
Siddique	7	5953	6.43	20
Aju Varghese	5	2237	6.43	21
Prakash Raj	6	8548	6.37	22
Jimmy Sheirgill	6	3826	6.29	23
Mahesh Achanta	6	2716	6.21	24
Biju Menon	5	1916	6.21	25
Suraj Venjaramo...	6	4284	6.19	26
Abir Chatterjee	5	1413	5.80	27
Sunny Deol	5	4594	5.71	28
Radha Ravi	5	1483	5.70	29
Prabhu Deva	5	2044	5.68	30
Atul Sharma	5	9604	4.78	31

Q23.Find out the top five actresses in Hindi movies released in India based on their average ratings?

```sql
WITH hindi_actresses AS (
  SELECT rm.name_id, n.name AS actress_name, r.avg_rating, r.total_votes
  FROM role_mapping rm
  JOIN names n ON rm.name_id = n.id
  JOIN ratings r ON rm.movie_id = r.movie_id
  JOIN movie m ON rm.movie_id = m.id
  WHERE rm.category = 'actress'
    AND m.country LIKE '%India%'
    AND m.languages LIKE '%Hindi%'
),
actress_stats AS (
  SELECT actress_name,
         COUNT(*) AS movie_count,
         SUM(total_votes) AS total_votes,
         ROUND(SUM(avg_rating * total_votes) / SUM(total_votes), 2) AS actress_avg_rating
  FROM hindi_actresses
  GROUP BY actress_name
  HAVING COUNT(*) >= 3
)
SELECT *, RANK() OVER (ORDER BY actress_avg_rating DESC, total_votes DESC) AS actress_rank
FROM actress_stats
LIMIT 5;
```

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

actress_name	movie_count	total_votes	actress_avg_rating	actress_rank
Taapsee Pannu	3	18061	7.74	1
Kriti Sanon	3	21967	7.05	2
Divya Dutta	3	8579	6.88	3
Shraddha Kapoor	3	26779	6.63	4
Kriti Kharbanda	3	2549	4.80	5

Result Grid

Form Editor

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

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

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

title	avg_rating	movie_category
Der müde Tod	7.7	Hit
Fahrenheit 451	4.9	Flop
Pet Sematary	5.8	One-time-watch
Dukun	6.9	One-time-watch
Back Roads	7.0	Hit
Countdown	5.4	One-time-watch
Staged Killer	3.3	Flop
Vellaipookal	7.3	Hit
Uriyadi 2	7.3	Hit
Incitement	7.5	Hit
Rakshasudu	8.4	Superhit
Trois jours et ...	6.6	One-time-watch
Killer in Law	5.1	One-time-watch
Kalki	7.3	Hit
Milliard	2.7	Flop
Vinci Da	7.2	Hit
Gunned Down	5.1	One-time-watch
Deviant Love	3.5	Flop
Storozh	6.3	One-time-watch
Sivappu Manja...	7.2	Hit
Magamuni	8.1	Superhit
Hometown Killer	5.8	One-time-watch
ECCO	5.0	One-time-watch
Baaji	7.1	Hit
Kasablanka	6.7	One-time-watch
Annabellum: T...	2.9	Flop
Zuo jia de hua...	5.8	One-time-watch
Evaru	8.3	Superhit
Saja	6.2	One-time-watch
Jiivi	7.9	Hit
Ai-naki Mori de...	6.4	One-time-watch
Ne Zha zhi mo ...	7.7	Hit
Bornoporichoy...	5.1	One-time-watch
Ratu Ilmu Hitam	7.4	Hit
Barot House	7.2	Hit
La llorona	7.1	Hit

Segment 4:

Q25. What is the genre-wise running total and moving average of the average movie duration?

```
748 • WITH genre_avg_duration AS (
749     SELECT g.genre, ROUND(AVG(m.duration), 2) AS avg_duration
750     FROM genre g
751     JOIN movie m ON g.movie_id = m.id
752     GROUP BY g.genre
753   ),
754   duration_with_running_avg AS (
755     SELECT
756       genre,
757       avg_duration,
758       SUM(avg_duration) OVER (ORDER BY genre) AS running_total_duration,
759       ROUND(AVG(avg_duration) OVER (ORDER BY genre ROWS BETWEEN 2 PRECEDING AND CURRENT ROW), 2)
760     FROM genre_avg_duration
761   )
762   SELECT * FROM duration_with_running_avg;
763
```

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

genre	avg_duration	running_total_duration	moving_avg_duration
Action	112.88	112.88	112.88
Adventure	101.87	214.75	107.38
Comedy	102.62	317.37	105.79
Crime	107.05	424.42	103.85
Drama	106.77	531.19	105.48
Family	100.97	632.16	104.93
Fantasy	105.14	737.30	104.29
Horror	92.72	830.02	99.61
Mystery	101.80	931.82	99.89
Others	100.16	1031.98	98.23
Romance	109.53	1141.51	103.83
Sci-Fi	97.94	1239.45	102.54
Thriller	101.58	1341.03	103.02

Result Grid | Form Editor | Field Types | Query Stats

Q26. Which are the five highest-grossing movies of each year that belong to the top three genres?

```sql
WITH top_genres AS (
  SELECT genre
  FROM genre
  GROUP BY genre
  ORDER BY COUNT(*) DESC
  LIMIT 3
),
genre_movies AS (
  SELECT g.genre, m.year, m.title AS movie_name, m.worlwide_gross_income
  FROM genre g
  JOIN top_genres tg ON g.genre = tg.genre
  JOIN movie m ON g.movie_id = m.id
  WHERE m.worlwide_gross_income IS NOT NULL
),
ranked_movies AS (
  SELECT *,
         RANK() OVER (PARTITION BY genre, year ORDER BY worlwide_gross_income DESC) AS movie_rank
  FROM genre_movies
)

-- Step 4: Final output
SELECT * FROM ranked_movies
WHERE movie_rank <= 5;
```

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

genre	year	movie_name	worlwide_gross_income	movie_rank
Comedy	2017	The Healer	$ 9979800	1
Comedy	2017	Tim Timmerman, Hope of America	$ 97727	2
Comedy	2017	Il a déjà tes yeux	$ 9755458	3
Comedy	2017	Jumanji: Welcome to the Jungle	$ 962102237	4
Comedy	2017	All Nighter	$ 96162	5
Comedy	2018	La fuitina sbagliata	$ 992070	1
Comedy	2018	Gung-hab	$ 9899017	2
Comedy	2018	Simmba	$ 9865268	3
Comedy	2018	Aleksi	$ 9791	4
Comedy	2018	Os Farofeiros	$ 9786399	5
Comedy	2019	Eaten by Lions	$ 99276	1
Comedy	2019	Friend Zone	$ 9894885	2
Comedy	2019	Organize Isler: Sazan Sarmali	$ 9831515	3
Comedy	2019	Benjamin	$ 97521	4
Comedy	2019	Brochevarevarura	$ 9737	5
Drama	2017	Shatamanam Bhavati	INR 530500000	1
Drama	2017	Winner	INR 250000000	2
Drama	2017	Thank You for Your Service	$ 9995692	3
Drama	2017	The Healer	$ 9979800	4
Drama	2017	Shan guang shao nu	$ 9949926	5
Drama	2018	Antony & Cleopatra	$ 998079	1
Drama	2018	Zaba	$ 991	2
Drama	2018	Canary	$ 98665	3
Drama	2018	Simmba	$ 9865268	4
Drama	2018	Une saison en France	$ 98390	5
Drama	2019	Joker	$ 995064593	1
Drama	2019	Nur eine Frau	$ 9884	2
Drama	2019	Running with the Devil	$ 98682	3
Drama	2019	Charlie Says	$ 98240	4
Drama	2019	Transit	$ 982372	5
Thriller	2017	Gi-eok-ui bam	$ 9968972	1
Thriller	2017	V.I.P.	$ 9710283	2
Thriller	2017	Fixeur	$ 9669	3
Thriller	2017	Overdrive	$ 9650552	4
Thriller	2017	Den 12. mann	$ 9567121	5
Thriller	2018	The Villain	INR 1300000000	1
Thriller	2018	Shéhérazade	$ 966225	2
Thriller	2018	Truth or Dare	$ 95330493	3
Thriller	2018	La nuit a dévoré le monde	$ 95208	4
Thriller	2018	Replicas	$ 9206925	5
Thriller	2019	Prescience	$ 9956	1
Thriller	2019	Joker	$ 995064593	2
Thriller	2019	Running with the Devil	$ 98682	3
Thriller	2019	The Boat	$ 98559	4
Thriller	2019	Division 19	$ 981	5

Q27. Which are the top two production houses that have produced the highest number of hits (median rating >= 8) among multilingual movies?

```
838 •   SELECT
839       m.production_company,
840       COUNT(*) AS movie_count,
841       RANK() OVER (ORDER BY COUNT(*) DESC) AS prod_comp_rank
842     FROM movie m
843     JOIN ratings r ON m.id = r.movie_id
844     WHERE POSITION(',' IN m.languages) > 0
845       AND r.median_rating >= 8
846       AND m.production_company IS NOT NULL
847     GROUP BY m.production_company
848     LIMIT 2;
849
850
```

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

production_company	movie_count	prod_comp_rank
Star Cinema	7	1
Twentieth Century Fox	4	2

Q28. Who are the top 3 actresses based on number of Super Hit movies (average rating >8) in drama genre?

```
870 •   SELECT
871       n.name AS actress_name,
872       COUNT(*) AS movie_count,
873       SUM(r.total_votes) AS total_votes,
874       ROUND(AVG(r.avg_rating), 2) AS actress_avg_rating,
875       RANK() OVER (ORDER BY COUNT(*) DESC) AS actress_rank
876     FROM role_mapping rm
877     JOIN names n ON rm.name_id = n.id
878     JOIN ratings r ON rm.movie_id = r.movie_id
879     JOIN genre g ON rm.movie_id = g.movie_id
880     WHERE rm.category = 'actress'
881       AND g.genre = 'Drama'
882       AND r.avg_rating > 8
883     GROUP BY n.name
884     ORDER BY movie_count DESC
885     LIMIT 3;
886
887
```

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

actress_name	movie_count	total_votes	actress_avg_rating	actress_rank
Parvathy Thiruvothu	2	4974	8.20	1
Susan Brown	2	656	8.95	1
Amanda Lawrence	2	656	8.95	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

```sql
WITH director_movies AS (
  SELECT d.name_id, COUNT(*) AS number_of_movies
  FROM director_mapping d
  GROUP BY d.name_id
  ORDER BY number_of_movies DESC
  LIMIT 9
),
movie_data AS (
  SELECT d.name_id, m.id AS movie_id, m.date_published, m.duration, r.avg_rating, r.total_votes, r.median_rating
  FROM director_mapping d
  JOIN movie m ON d.movie_id = m.id
  JOIN ratings r ON m.id = r.movie_id
),
joined AS (
  SELECT dm.name_id, n.name AS director_name, COUNT(md.movie_id) AS number_of_movies,
         ROUND(AVG(md.avg_rating), 2) AS avg_rating,
         SUM(md.total_votes) AS total_votes,
         MIN(md.avg_rating) AS min_rating,
         MAX(md.avg_rating) AS max_rating,
         SUM(md.duration) AS total_duration
  FROM director_movies dm
  JOIN movie_data md ON dm.name_id = md.name_id
  JOIN names n ON dm.name_id = n.id
  GROUP BY dm.name_id, n.name
)
SELECT * FROM joined;
```

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

name_id	director_name	number_of_movies	avg_rating	total_votes	min_rating	max_rating	total_duration
nm6356309	Özgür Bakar	4	3.75	1092	3.1	4.9	374
nm2691863	Justin Price	4	4.50	5343	3.0	5.8	346
nm0814469	Sion Sono	4	6.03	2972	5.4	6.4	502
nm0831321	Chris Stokes	4	4.33	3664	4.0	4.6	352
nm2096009	Andrew Jones	5	3.02	1989	2.7	3.2	432
nm0425364	Jesse V. Johnson	4	5.45	14778	4.2	6.5	383
nm0001752	Steven Soderbergh	4	6.48	171684	6.2	7.0	401
nm0515005	Sam Liu	4	6.23	28557	5.8	6.7	312
nm1777967	A.L. Vijay	5	5.42	1754	3.7	6.9	613