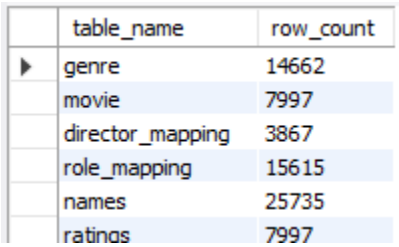
SQL imdb project query insights Axton

```
use imdb;
```

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

```
use imdb;
SELECT 'genre' AS table_name, COUNT(*) AS row_count FROM genre
UNION
SELECT 'movie', COUNT(*) FROM movie
UNION
SELECT 'director_mapping', COUNT(*) FROM director_mapping
UNION
SELECT 'role_mapping', COUNT(*) FROM role_mapping
UNION
SELECT 'names', COUNT(*) FROM names
UNION
SELECT 'ratings', COUNT(*) FROM ratings;
```



table_name	row_count
genre	14662
movie	7997
director_mapping	3867
role_mapping	15615
names	25735
ratings	7997

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

```
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;
```

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

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

```
/* Output format for the first part:

+---------------+-------------------+
| Year          |   number_of_movies|
+-------------------+---------------
|   2017        |   2134            |
|   2018        |       .           |
|   2019        |       .           |
+---------------+-------------------+


Output format for the second part of the question:
+---------------+-------------------+
|   month_num   |   number_of_movies|
+---------------+---------------
|   1           |    134            |
|   2           |    231            |
|   .           |       .           |
+---------------+-------------------+ */
```

```
SELECT
        YEAR(date_published) AS year,
        COUNT(*) AS number_of_movies
FROM movie
GROUP BY YEAR(date_published)
ORDER BY year;
```

year	number_of_movies
2017	3052
2018	2944
2019	2001

```
SELECT
        MONTH(date_published) AS month_num,
        COUNT(*) AS number_of_movies
```

```
FROM movie
GROUP BY MONTH(date_published)
ORDER BY month_num;
```

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??

```
SELECT COUNT(*) AS movie_count
FROM movie
WHERE country IN ('USA', 'India')
  AND YEAR(date_published) = 2019;
```

movie_count
887

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

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

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

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

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

	genre	movie_count
▶	Drama	4285

-- Q7. How many movies belong to only one genre?

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

	single_genre_movies
▶	3289

-- 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 genre g
JOIN movie m ON g.movie_id = m.id
GROUP BY g.genre;
```

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

-- 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         |
+---------------+-------------------+---------------------+*/
```

```
SELECT genre, COUNT(DISTINCT movie_id) AS movie_count,
        RANK() OVER (ORDER BY COUNT(DISTINCT movie_id) DESC) AS genre_rank
```

FROM genre
GROUP BY genre;

	genre	movie_count	genre_rank
▶	Drama	4285	1
	Comedy	2412	2
	Thriller	1484	3
	Action	1289	4
	Horror	1208	5
	Romance	906	6
	Crime	813	7
	Adventure	591	8
	Mystery	555	9
	Sci-Fi	375	10
	Fantasy	342	11
	Family	302	12
	Others	100	13

-- 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
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;

	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?

```
/* Output format:
+---------------+-------------------+---------------------+
| title         |       avg_rating  |       movie_rank    |
+---------------+-------------------+---------------------+
| Fan           |       9.6         |           5         |
|   .           |       .           |           .         |
|   .           |       .           |           .         |
|   .           |       .           |           .         |
+---------------+-------------------+---------------------+*/
```

```
-- 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;
```

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.

```
/* Output format:

+---------------+-------------------+
| median_rating |   movie_count     |
+-------------------+----------------
|   1           |       105         |
|   .           |       .           |
|   .           |       .           |
+---------------+-------------------+ */
```

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

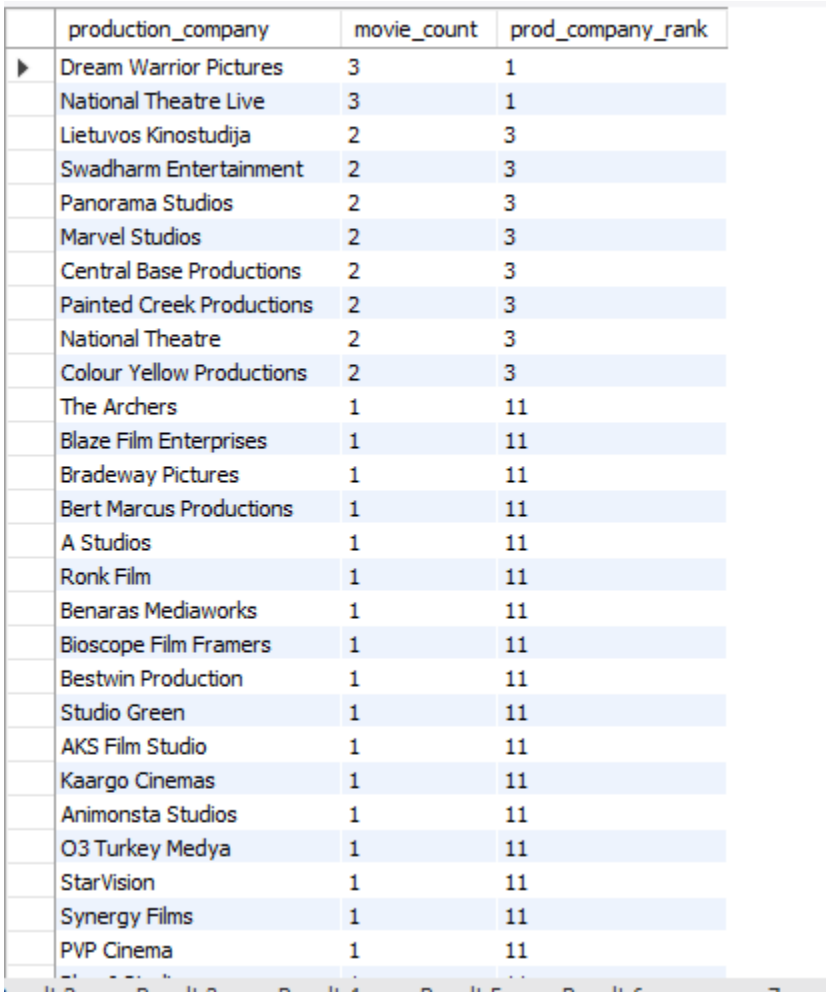
median_rating	movie_count
1	94
2	119
3	283
4	479
5	985
6	1975
7	2257
8	1030
9	429
10	346

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

```
/* Output format:
+------------------+-------------------+---------------------+
|production_company|movie_count        |    prod_company_rank|
+------------------+-------------------+---------------------+
| The Archers      |        1          |            1        |
+------------------+-------------------+---------------------+*/
```

```
SELECT m.production_company, COUNT(*) AS movie_count,
        RANK() OVER (ORDER BY COUNT(*) DESC) AS prod_company_rank
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE r.avg_rating > 8 AND m.production_company IS NOT NULL
```

GROUP BY m.production_company;



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

-- 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         |
|   .           |       .           |
|   .           |       .           |
+---------------+-------------------+ */
```

SELECT g.genre, COUNT(*) AS movie_count
FROM movie m

```
JOIN ratings r ON m.id = r.movie_id
JOIN genre g ON m.id = g.movie_id
WHERE r.total_votes > 1000
  AND MONTH(m.date_published) = 3
  AND YEAR(m.date_published) = 2017
  AND m.country = 'USA'
GROUP BY g.genre;
```

genre	movie_count
Action	4
Comedy	8
Crime	5
Drama	16
Fantasy	2
Mystery	2
Romance	3
Sci-Fi	4
Thriller	4
Horror	5
Family	1

-- 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 movie m
JOIN ratings r ON m.id = r.movie_id
JOIN genre g ON m.id = g.movie_id
WHERE m.title LIKE 'The %' AND r.avg_rating > 8;
```

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

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

	movie_count
▶	361

-- 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.

```
SELECT country, SUM(r.total_votes) AS total_votes
FROM movie m
JOIN ratings r ON m.id = r.movie_id
WHERE country IN ('Germany', 'Italy')
GROUP BY country;
```

	country	total_votes
▶	Germany	106710
	Italy	77965

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

```
/*Hint: You can find null values for individual columns or follow below output format
+---------------+-------------------+---------------------+----------------------+
| name_nulls    |   height_nulls    |date_of_birth_nulls  |known_for_movies_nulls|
+---------------+-------------------+---------------------+----------------------+
|       0       |           123     |          1234       |    12345             |
+---------------+-------------------+---------------------+----------------------+*/
```

```
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;
```

name_nulls	height_nulls	date_of_birth_nulls	known_for_movies_nulls
0	17335	13431	15226

-- 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_genres AS (
	SELECT g.genre
	FROM genre g
	JOIN ratings r ON g.movie_id = r.movie_id
	WHERE r.avg_rating > 8
	GROUP BY g.genre
	ORDER BY COUNT(*) DESC
	LIMIT 3
)
```

```
SELECT n.name AS director_name, COUNT(*) AS movie_count
FROM director_mapping d
JOIN names n ON d.name_id = n.id
JOIN movie m ON d.movie_id = m.id
JOIN ratings r ON m.id = r.movie_id
JOIN genre g ON m.id = g.movie_id
WHERE r.avg_rating > 8 AND g.genre IN (SELECT genre FROM top_genres)
GROUP BY n.name
ORDER BY movie_count DESC
LIMIT 3;
```

director_name	movie_count
James Mangold	4
Joe Russo	3
Anthony Russo	3

```
-- Q20. Who are the top two actors whose movies have a median rating >= 8?
/* Output format:

+---------------+-------------------+
| actor_name    |   movie_count     |
+-------------------+----------------
|Christain Bale |       10          |
|   .           |       .           |
+---------------+-------------------+ */
SELECT n.name AS actor_name, COUNT(*) AS movie_count
FROM role_mapping rm
JOIN names n ON rm.name_id = n.id
JOIN ratings r ON rm.movie_id = r.movie_id
WHERE rm.category = 'actor' AND r.median_rating >= 8
GROUP BY n.name
ORDER BY movie_count DESC
LIMIT 2;
```

actor_name	movie_count
Mammootty	8
Mohanlal	5

-- Q22. Rank actors with movies released in India based on their average ratings. Which actor is at the top of the list?
-- Note: The actor should have acted in at least five Indian movies.
-- (Hint: You should use the weighted average based on votes. If the ratings clash, then the total number of votes should act as the tie breaker.)

```
/* Output format:
+---------------+-------------------+---------------------+----------------------+-----------------+
| actor_name    |   total_votes     |   movie_count       | actor_avg_rating     |actor_rank       |
+---------------+-------------------+---------------------+----------------------+-----------------+
|   Yogi Babu   |           3455    |          11         |    8.42              |      1          |
|       .       |           .       |          .          |    .                 |      .          |
|       .       |           .       |          .          |    .                 |      .          |
|       .       |           .       |          .          |    .                 |      .          |
+---------------+-------------------+---------------------+----------------------+-----------------+*/
```

```
SELECT n.name AS actor_name,
        SUM(r.total_votes) AS total_votes,
        COUNT(*) AS movie_count,
        ROUND(SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes), 2) AS actor_avg_rating,
        RANK() OVER (ORDER BY SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes)
DESC, SUM(r.total_votes) DESC) AS actor_rank
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 rm.category = 'actor' AND m.country = 'India'
GROUP BY n.name
HAVING COUNT(*) >= 5;
```

actor_name	total_votes	movie_count	actor_avg_rating	actor_rank
Vijay Sethupathi	23114	5	8.42	1
Fahadh Faasil	13557	5	7.99	2
Yogi Babu	8500	11	7.83	3
Joju George	3926	5	7.58	4
Ammy Virk	2504	6	7.55	5
Dileesh Pothan	6235	5	7.52	6
Kunchacko Boban	5628	6	7.48	7
Pankaj Tripathi	40728	5	7.44	8
Rajkummar Rao	42560	6	7.37	9
Dulquer Salmaan	17666	5	7.30	10
Amit Sadh	13355	5	7.21	11
Tovino Thomas	11596	8	7.15	12
Mammootty	12613	8	7.04	13
Nassar	4016	5	7.03	14
Karamjit Anmol	1970	6	6.91	15
Hareesh Kanaran	3196	5	6.58	16
Naseeruddin Shah	12604	5	6.54	17
Anandraj	2750	6	6.54	18
Mohanlal	17244	6	6.51	19
Aju Varghese	2237	5	6.43	20
Siddique	5953	7	6.43	21
Prakash Raj	8548	6	6.37	22
Jimmy Sheirgill	3826	6	6.29	23
Mahesh Achanta	2716	6	6.21	24
Biju Menon	1916	5	6.21	25
Suraj Venjaramo...	4284	6	6.19	26
Abir Chatterjee	1413	5	5.80	27
Sunny Deol	4594	5	5.71	28
Radha Ravi	1483	5	5.70	29
Prabhu Deva	2044	5	5.68	30

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

-- Note: The actresses should have acted in at least three Indian movies.

-- (Hint: You should use the weighted average based on votes. If the ratings clash, then the total number of votes should act as the tie breaker.)

```
/* Output format:
+---------------+-------------------+---------------------+----------------------+-----------------+
| actress_name  |   total_votes     |   movie_count       | actress_avg_rating   |actress_rank     |
+---------------+-------------------+---------------------+----------------------+-----------------+
|   Tabu        |           3455    |          11         |    8.42              |      1          |
|       .       |           .       |          .          |    .                 |      .          |
|       .       |           .       |          .          |    .                 |      .          |
|       .       |           .       |          .          |    .                 |      .          |
+---------------+-------------------+---------------------+----------------------+-----------------+*/
```

```
SELECT n.name AS actress_name,
	SUM(r.total_votes) AS total_votes,
	COUNT(*) AS movie_count,
	ROUND(SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes), 2) AS
actress_avg_rating,
	RANK() OVER (ORDER BY SUM(r.avg_rating * r.total_votes) / SUM(r.total_votes)
DESC, SUM(r.total_votes) DESC) AS actress_rank
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 rm.category = 'actress' AND m.country = 'India' AND m.languages LIKE '%Hindi%'
GROUP BY n.name
HAVING COUNT(*) >= 3;
```

actress_name	total_votes	movie_count	actress_avg_rating	actress_rank
Taapsee Pannu	18061	3	7.74	1
Kriti Sanon	21967	3	7.05	2
Divya Dutta	8579	3	6.88	3
Shraddha Kapoor	26779	3	6.63	4
Kriti Kharbanda	2549	3	4.80	5
Sonakshi Sinha	4025	4	4.18	6