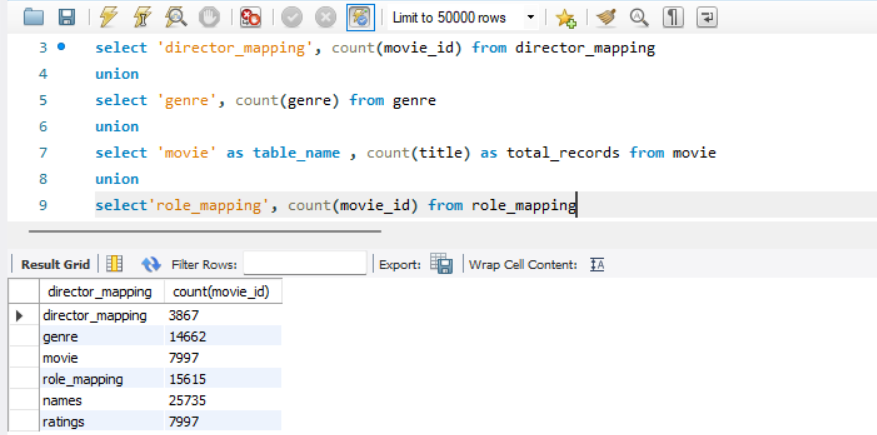
**Advanced SQL – Reinforcement Project – IMDB Dataset**

The dataset appears to represent an IMDb-style movie database, containing multiple tables that store information about movies, directors, genres, ratings, and roles played by actors. Below are the primary tables and their purpose:

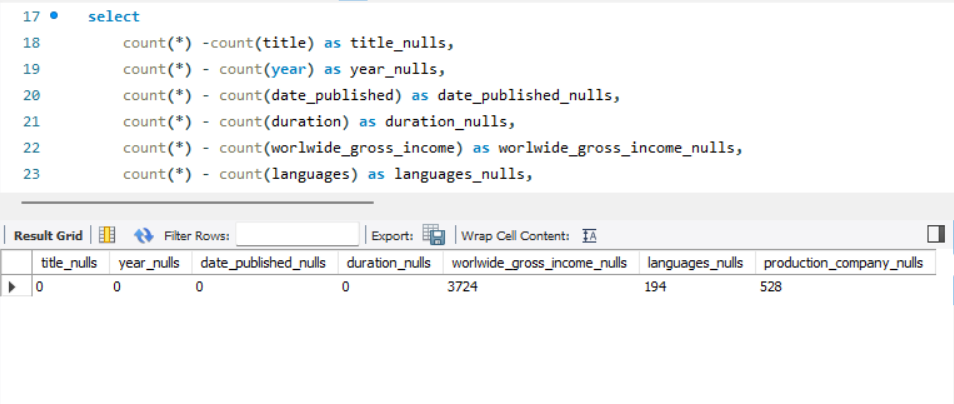
* **movie**: Contains general movie details such as title, release year, duration, country, and production company.
* **ratings**: Stores average ratings, total votes, and median ratings for movies.
* **names**: Holds information about actors, directors, and other contributors.
* **role\_mapping**: Maps actors to movies with their roles.
* **director\_mapping**: Links directors to their respective movies.
* **genre**: Associates movies with their respective genres.

**1. Count the total number of records in each table of the database.**

****

* Each SELECT statement counts rows in a table.
* AS table\_name and AS total\_records give column aliases for clarity.
* UNION ALL ensures all results are combined properly.

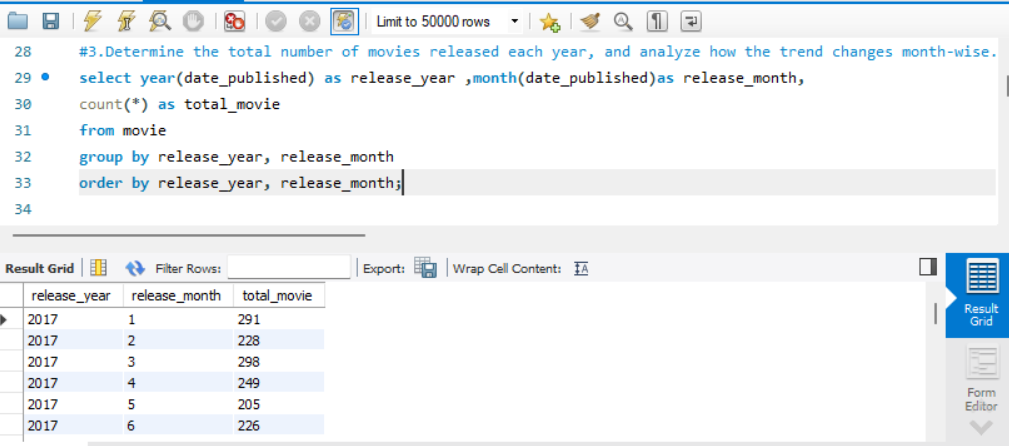
**2. Identify which columns in the movie table contain null values.**

****

* 3724 movies have a NULL worlwide\_gross\_income recorded.
* 194 movies have a missing languages.
* 528 movies have a null production\_company.

**3. Determine the total number of movies released each year, and analyze how the trend changes**

**month-wise.**

****

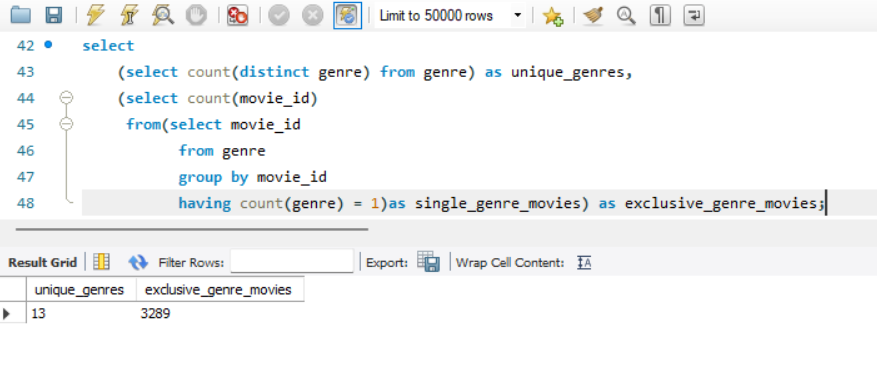
**4. How many movies were produced in either the USA or India in the year 2019?**

****

* **Filters only movies from** 🇺🇸 **USA** and 🇮🇳 **India**.
* **Only includes movies released in** **2019**.
* **295 movies** were released in **INDIA** in **2019**.
* **592 movies** were released in **USA** in **2019**.

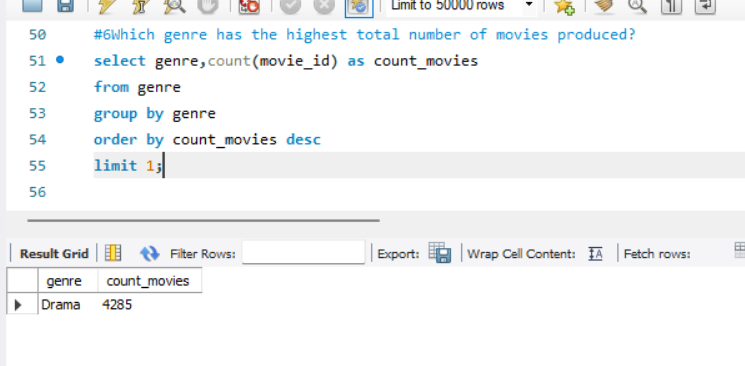
**5. List the unique genres in the dataset, and count how many movies belong exclusively to one**

**genre.**

****

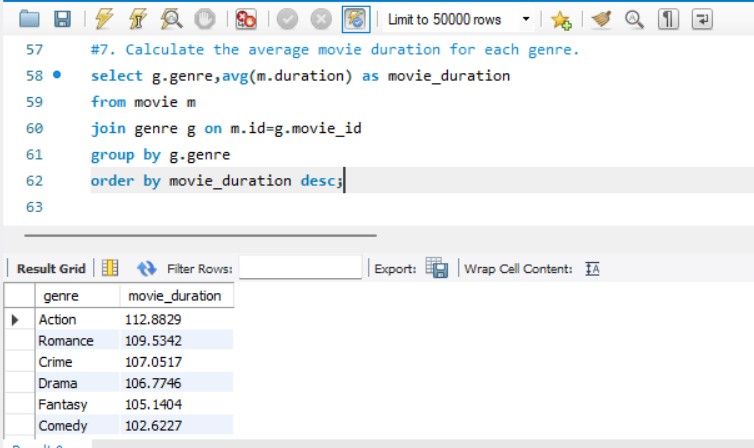
* The COUNT(DISTINCT genre) function counts the number of unique genres in the genre table.
* Unique\_genres → The number of different genres.
* Exclusive\_genre\_movies → The number of movies assigned to exactly one genre.

**6. Which genre has the highest total number of movies produced?**

****

* Sorts the results in descending order based on the count of movies in each genre (most popular genre first).
* Ensures that only the **top** genre (with the highest count) is returned.

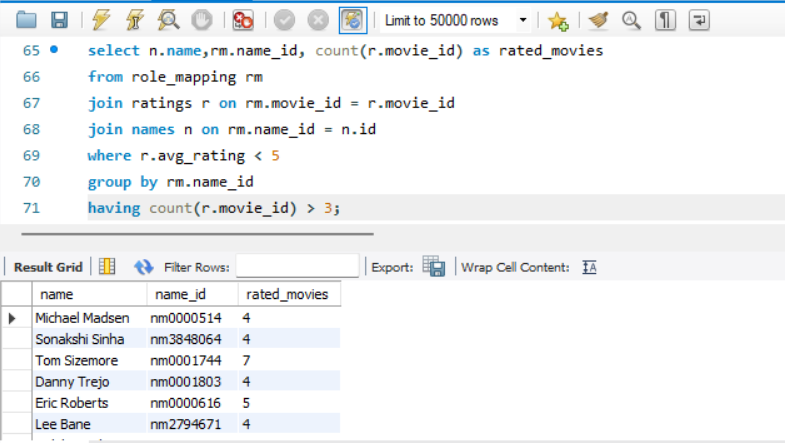
**7. Calculate the average movie duration for each genre.**

****

* Combines the movie table (m) and the genre table (g) using movie\_id as the common key.
* Sorts the genres from **longest to shortest** average movie duration.

**8. Identify actors or actresses who have appeared in more than three movies with an average**

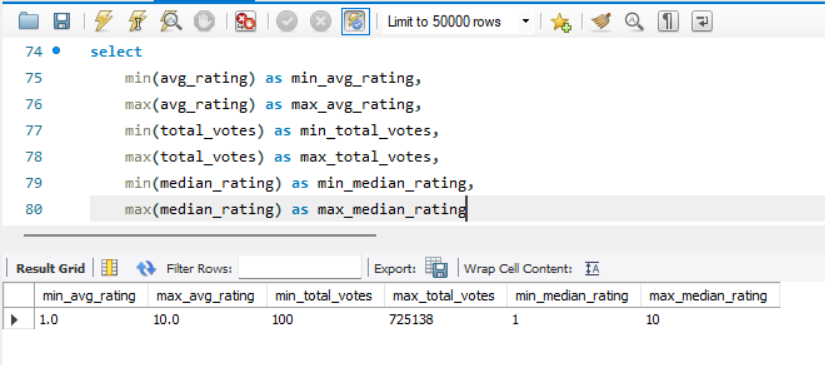
**rating below 5.**

****

* Connects the role\_mapping table to the ratings table using movie\_id and also role\_mapping table to the names table using name\_id.
* Groups results by name\_id, so each row represents a unique person.
* Filters only movies that have an **average rating below 5**.

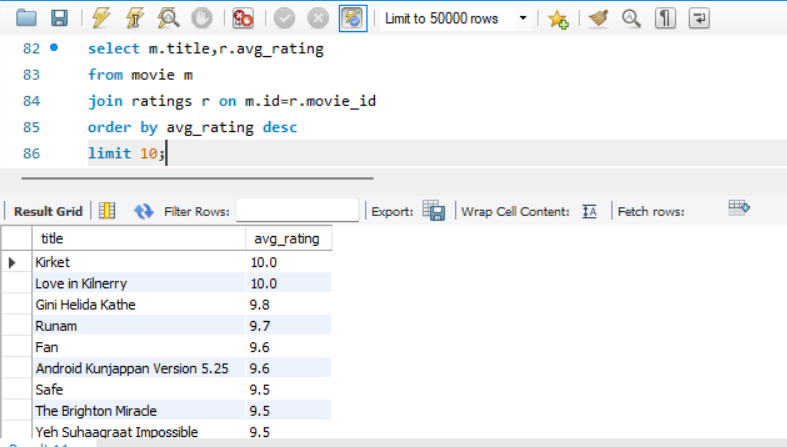
**9. Find the minimum and maximum values for each column in the ratings table, excluding the**

**movie\_id column.**

****

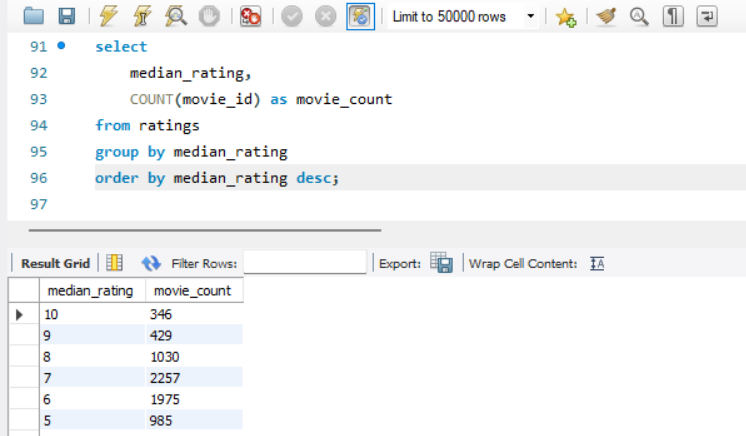
* Finds the lowest&highest average rating,number of votes,median rating, in the dataset.

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

****

* Combines the movie table (m) and the ratings table (r) using the movie\_id key.
* Sorts the movies in **descending order** based on their **average rating**.
* Restricts the output to only **the top 10 highest-rated movies**.

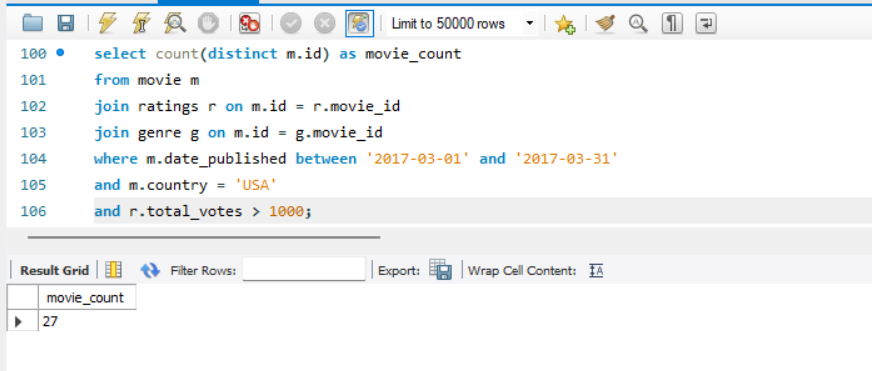
**11. Summarize the ratings table by grouping movies based on their median ratings.**

****

* Groups the data by median\_rating, so each row represents a unique median rating and the number of movies with that rating.

**12. How many movies, released in March 2017 in the USA within a specific genre, had more**

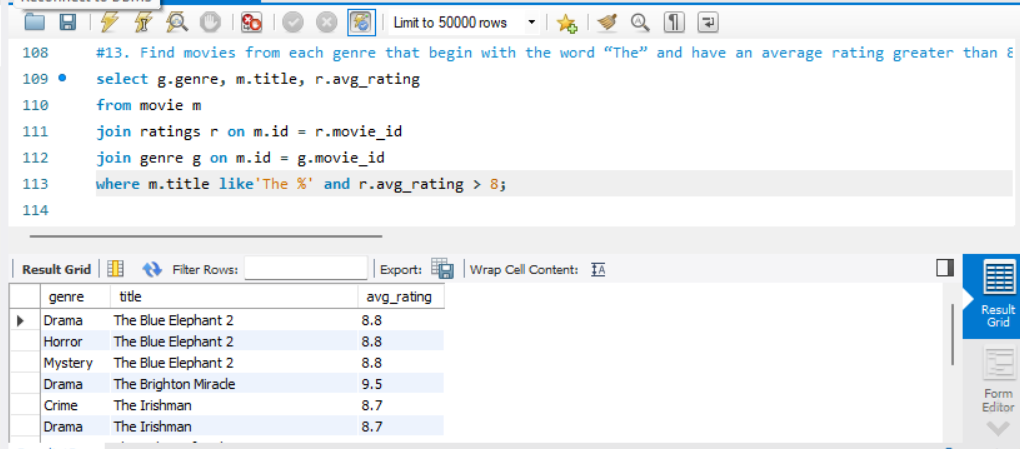
**than 1,000 votes?**

****

* Filters movies that were released in March 2017.
* Filters movies that have received more than 1000 votes to focus on well-rated or popular movies.

**13. Find movies from each genre that begin with the word “The” and have an average rating**

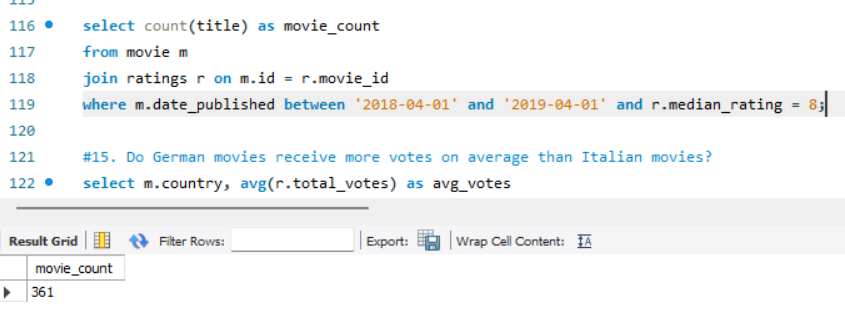
**greater than 8.**

****

* Filters movies whose title **starts with "The"** (e.g., *The Godfather*, *The Dark Knight*).
* The **LIKE 'The %'** pattern means:
  + 'The ' → Starts with "The " (case-sensitive in some databases).
  + '% ' → Anything can follow after "The ".

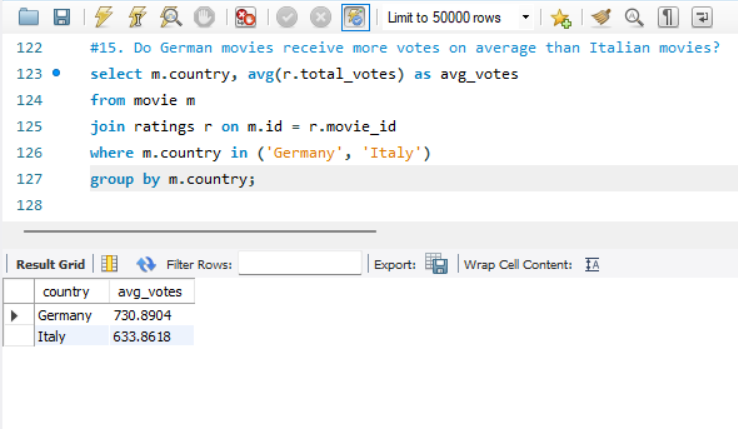
**14. Of the movies released between April 1, 2018, and April 1, 2019, how many received a**

**median rating of 8?**

****

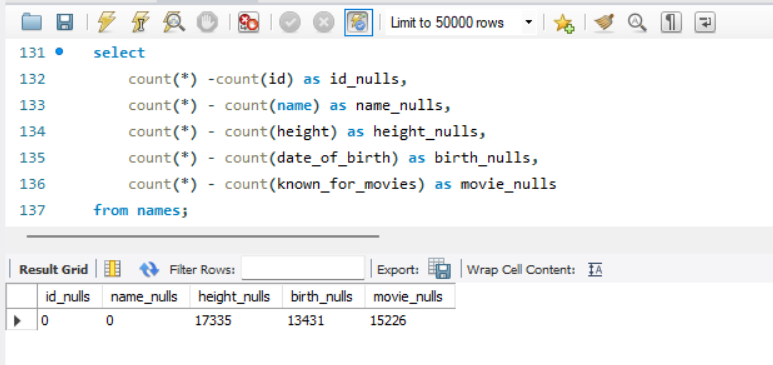
* Ensures that only movies **with ratings** are considered.
* Filters movies that have a **median rating of exactly 8**.
* Counts the number of movies that match all these conditions.

**15. Do German movies receive more votes on average than Italian movies?**

****

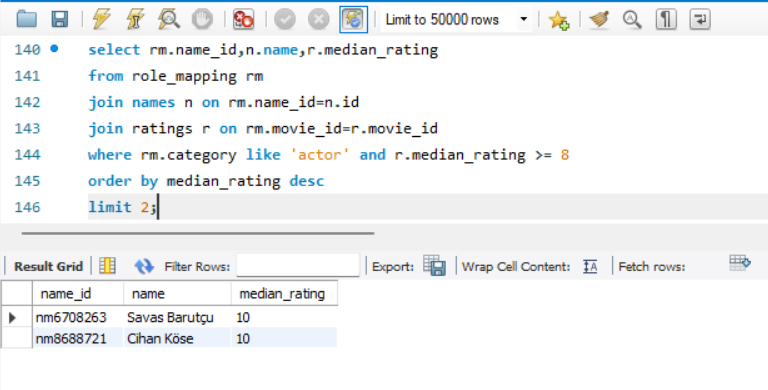
* Links the movie table with the ratings table using the movie\_id key.
* Filters the results to **only include movies from Germany and Italy**.
* Groups the results by country so that the **average votes** are calculated separately for each country.

**16. Identify the columns in the names table that contain null values.**

****

* Subtracting the non-NULL count from the total row count gives the **number of NULL values** in that column.
* **17335 NULL values** in the height column.
* **13431 NULL values** in the birth column.
* **15226 NULL values** in the height column

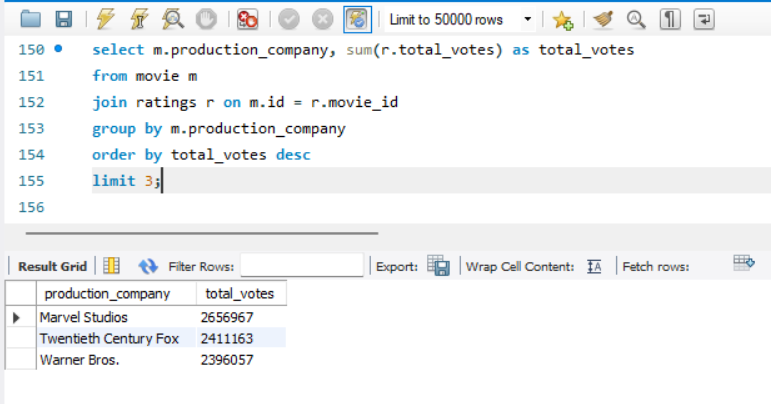
**17. Who are the top two actors whose movies have a median rating of 8 or higher?**

****

* Filters only rows where the person is categorized as an **actor**.
* Filters only movies with a **median rating of 8 or higher**.

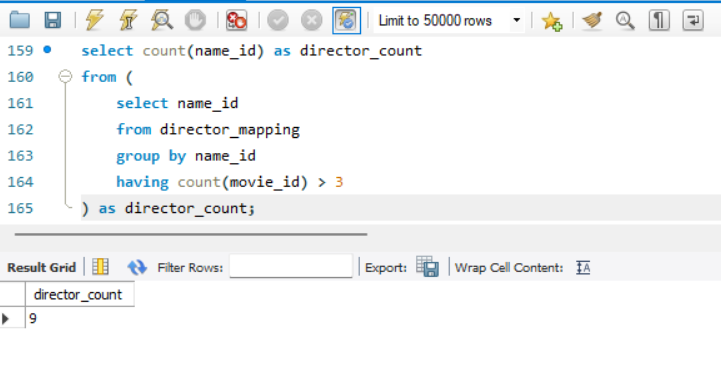
**18. Which are the top three production companies based on the total number of votes their**

**movies received?**

****

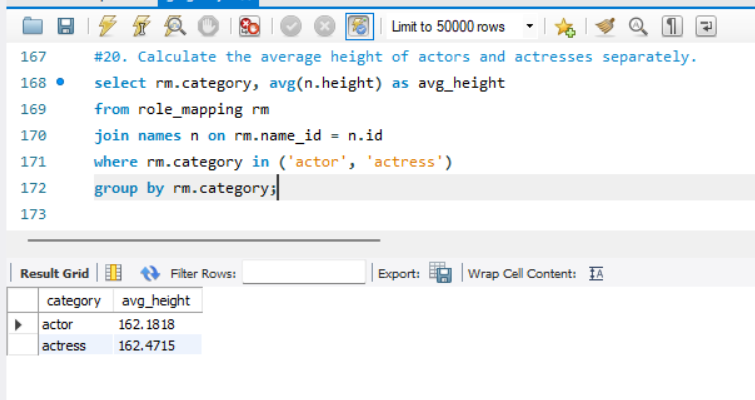
* Calculates the **total number of votes** each production company’s movies have received.
* Returns **only the top 3 production companies**.

**19. How many directors have worked on more than three movies?**

****

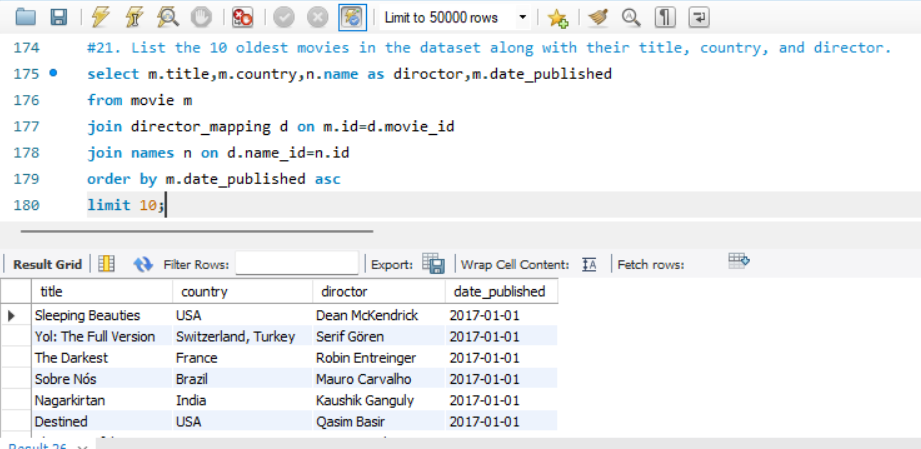
* The outer query **counts the number of rows** returned by the inner query.
* Each row in the inner query represents **one director**, so counting these rows gives the total number of directors who meet the condition.

**20. Calculate the average height of actors and actresses separately.**

****

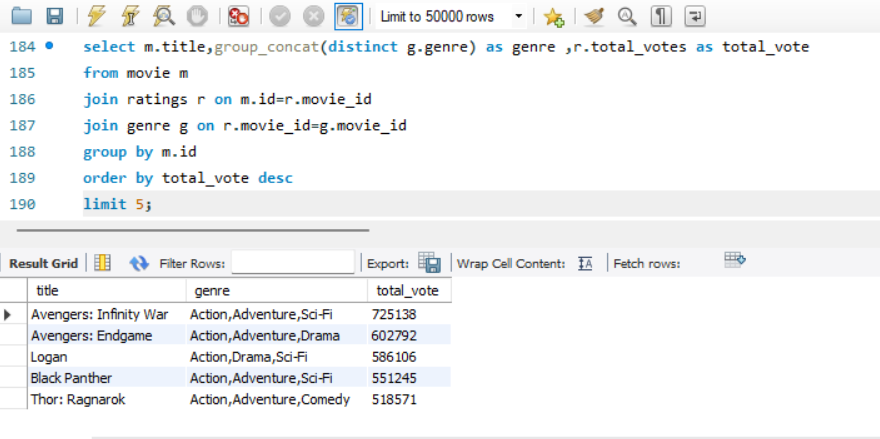
* Filters the results to include **only actors and actresses** (excluding other roles like directors, producers, etc.).
* Groups the data into two categories: actor and actress.

**21. List the 10 oldest movies in the dataset along with their title, country, and director.**

****

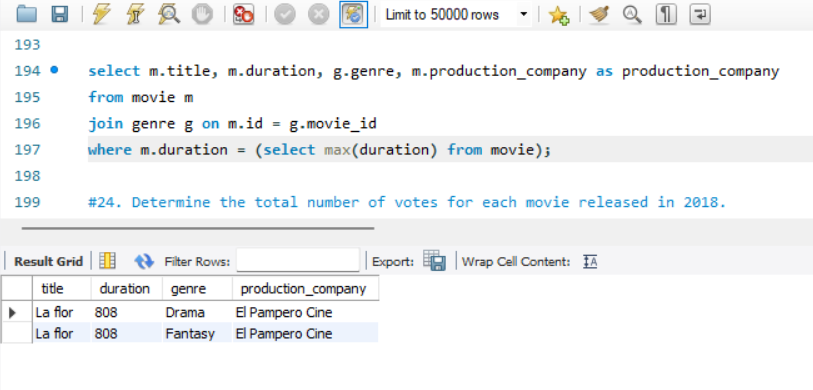
* Sorts the movies in **ascending order by release date** (oldest movies first).
* First oldest movie **Sleeping Beauties.**
* Retrieves only **the first 10 movies** based on release date.

**22. List the top 5 movies with the highest total votes, along with their genres.**

****

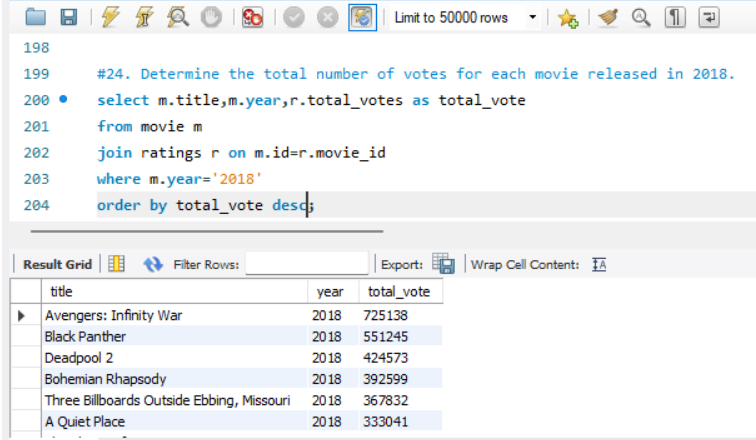
* Aggregates multiple genres for each movie into a single string, **separated by commas** (e.g., "Action, Adventure, Sci-Fi").
* DISTINCT ensures that duplicate genres are not repeated.

**23. Identify the movie with the longest duration, along with its genre and production company.**

****

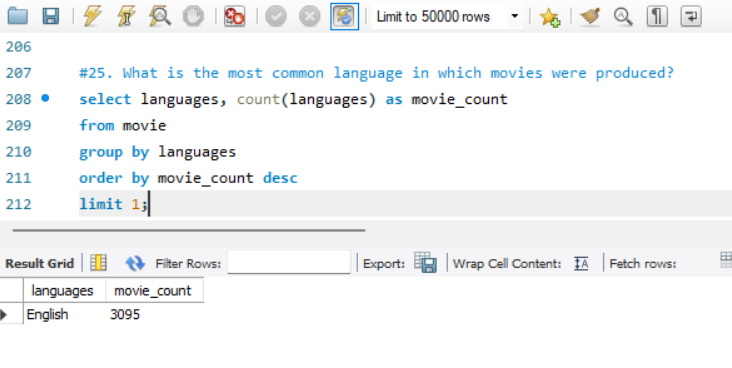
* Finds the **maximum movie duration** in the database.
* Returns a single value, which is used to filter movies in the main query.

**24. Determine the total number of votes for each movie released in 2018.**

****

* Links the movie table with the ratings table using movie\_id.
* Filters the results to **only include movies released in 2018**.
* **Avengers:Infinity War** getting most votes it’s appear in top of the table.

**25. What is the most common language in which movies were produced?**

****

* Counts how many movies exist for each language.
* Sorts the results **in descending order**, ensuring the language with **the most movies appears first**.
* Returns **English the top language** with the highest number of movies.