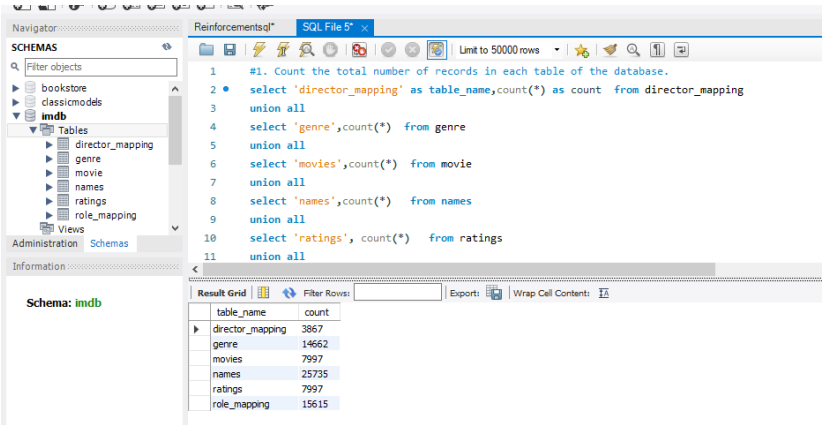
1. **Count the total number of records in each table of the database.**

   QUERY

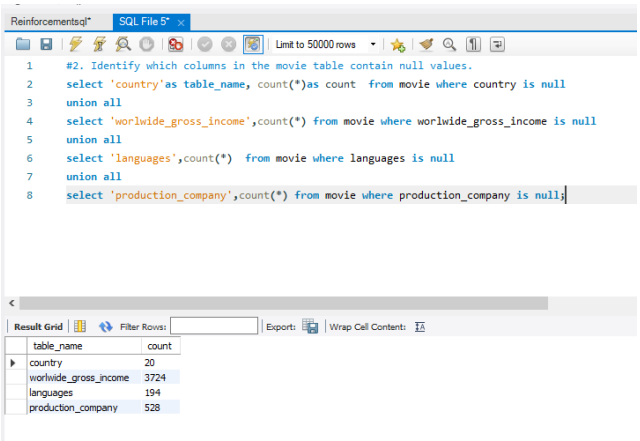
   select 'director_mapping' as table_name,count(*) as count from director_mapping
   union all
   select 'genre',count(*) from genre
   union all
   select 'movies',count(*) from movie
   union all
   select 'names',count(*) from names
   union all
   select 'ratings', count(*) from ratings
   union all
   select 'role_mapping',count(*) from role_mapping;



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

   QUERY

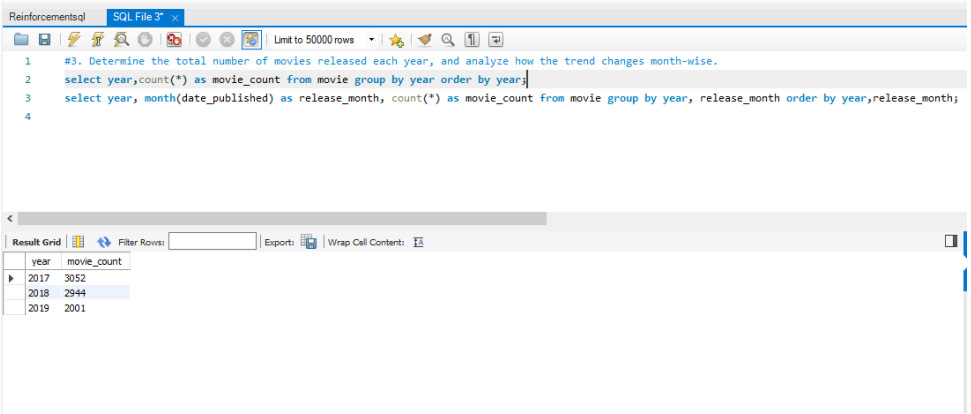
   select 'country'as table_name, count(*)as count from movie where country is null
   union all
   select 'worlwide_gross_income',count(*) from movie where worlwide_gross_income is null
   union all
   select 'languages',count(*) from movie where languages is null
   union all
   select 'production_company',count(*) from movie where production_company is null;



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

QUERY

select year,count(*) as movie_count from movie group by year order by year;
select year, month(date_published) as release_month, count(*) as movie_count from movie group by year, release_month order by year,release_month;



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

QUERY

select count(*) as movie_count from movie where (country LIKE '%USA%' OR country LIKE '%India%') and year = '2019';

```
#4. How many movies were produced in either the USA or India in the year 2019?
select count(*) as movie_count from movie where (country LIKE '%USA%' OR country LIKE '%India%') and year = '2019';
```

movie_count
1059

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

QUERY

select distinct genre from genre;
select count(*) as exclusive_count from (select movie_id from genre group by movie_id having count(*) = 1) as exclusive_genre ;

```
#5. List the unique genres in the dataset, and count how many movies belong exclusively to one genre.
select distinct genre from genre;
select count(*) as exclusive_count from (select movie_id from genre group by movie_id having count(*) = 1) as exclusive_genre ;
```

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

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

QUERY

select genre, count(*)as movie_count from genre group by genre order by movie_count desc limit 1;

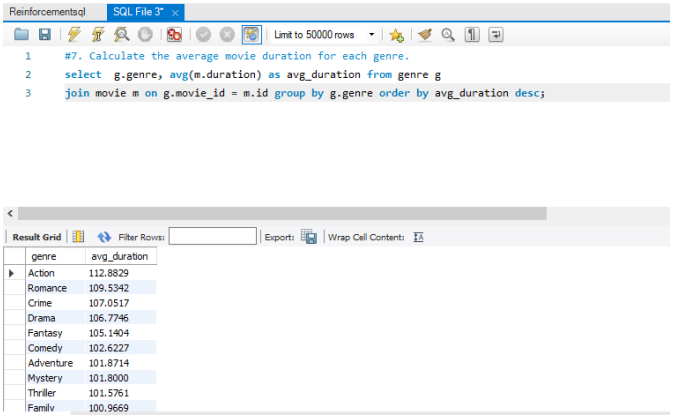
```
#6. Which genre has the highest total number of movies produced?
select genre, count(*)as movie_count from genre group by genre order by movie_count desc limit 1;
```

genre	movie_count
Drama	4285

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

QUERY

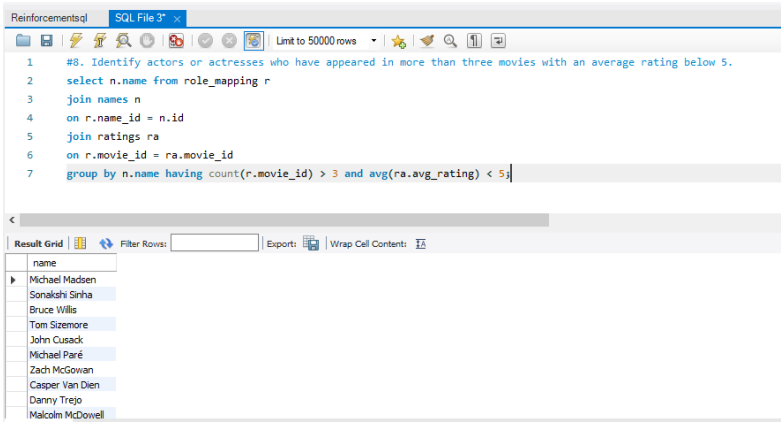
select g.genre, avg(m.duration) as avg_duration from genre g
join movie m on g.movie_id = m.id group by g.genre order by avg_duration desc;



8. **Identify actors or actresses who have appeared in more than three movies with an average rating below 5.**

QUERY

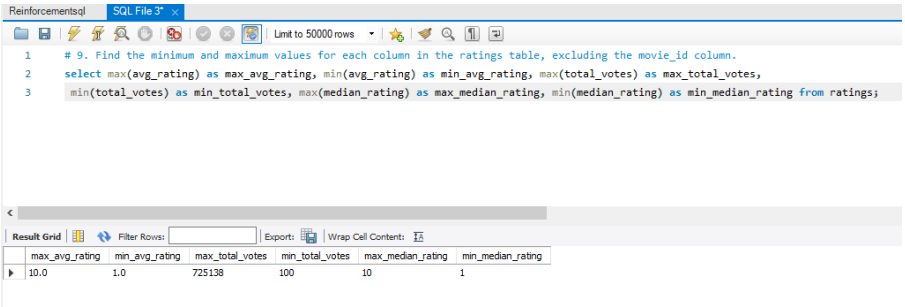
select n.name from role_mapping r
join names n
on r.name_id = n.id
join ratings ra
on r.movie_id = ra.movie_id
group by n.name having count(r.movie_id) > 3 and avg(ra.avg_rating) < 5;



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

QUERY

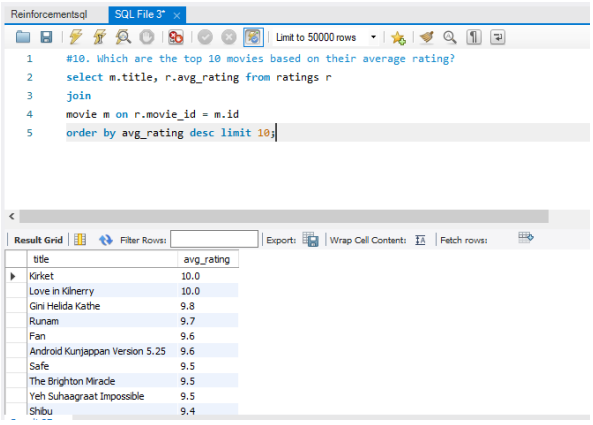
select max(avg_rating) as max_avg_rating, min(avg_rating) as min_avg_rating, max(total_votes) as max_total_votes,
min(total_votes) as min_total_votes, max(median_rating) as max_median_rating, min(median_rating) as min_median_rating from ratings;



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

QUERY

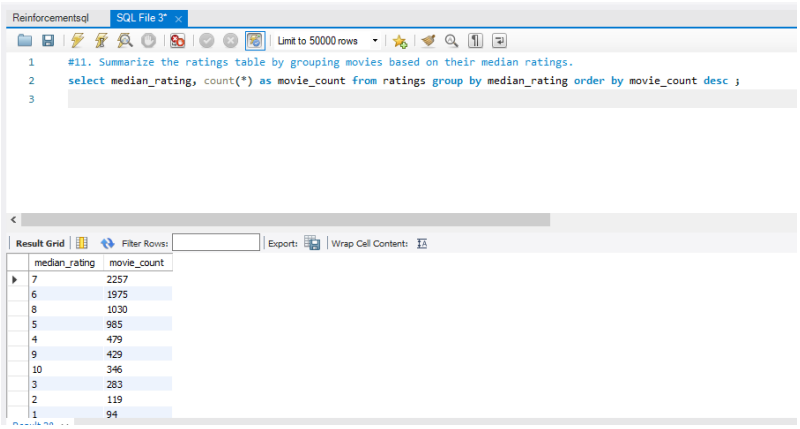
select m.title, r.avg_rating from ratings r
join
movie m on r.movie_id = m.id
order by avg_rating desc limit 10;



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

QUERY

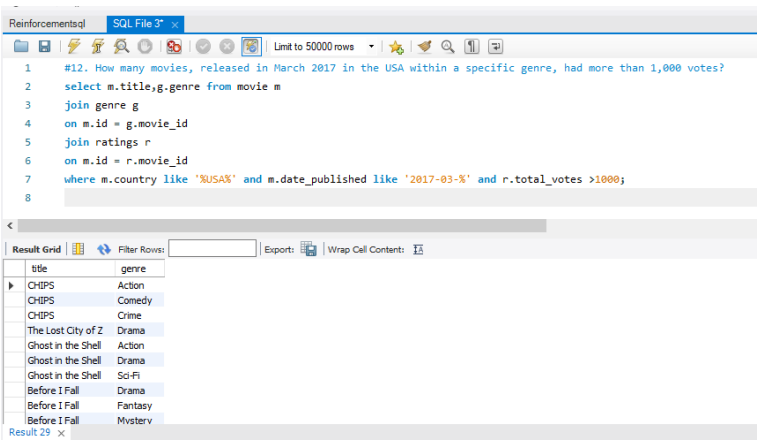
select median_rating, count(*) as movie_count from ratings group by median_rating order by movie_count desc ;



12. How many movies, released in March 2017 in the USA within a specific genre, had more than 1,000 votes?

QUERY

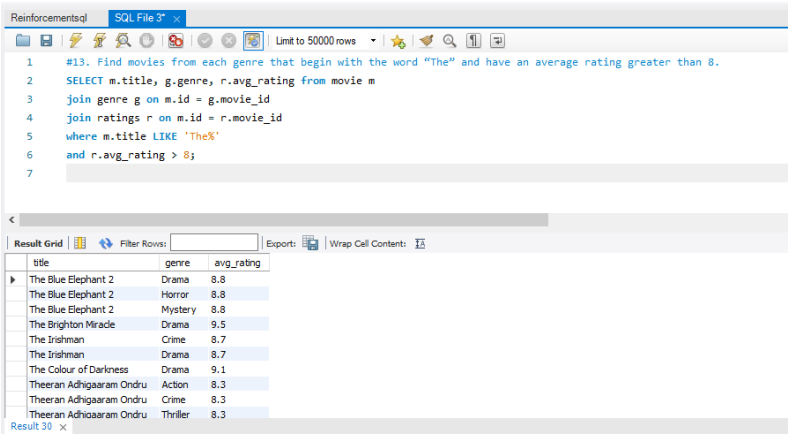
select m.title,g.genre from movie m
join genre g
on m.id = g.movie_id
join ratings r
on m.id = r.movie_id
where m.country like '%USA%' and m.date_published like '2017-03-%' and r.total_votes >1000;



13. Find movies from each genre that begin with the word "The" and have an average rating greater than 8.

QUERY

SELECT m.title, g.genre, r.avg_rating from movie m
join genre g on m.id = g.movie_id
join ratings r on m.id = r.movie_id
where m.title LIKE 'The%'
and r.avg_rating > 8;



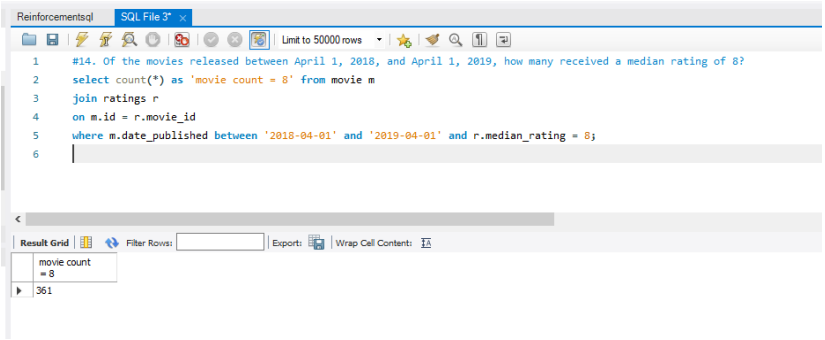
```sql
#13. Find movies from each genre that begin with the word "The" and have an average rating greater than 8.
SELECT m.title, g.genre, r.avg_rating from movie m
join genre g on m.id = g.movie_id
join ratings r on m.id = r.movie_id
where m.title LIKE 'The%'
and r.avg_rating > 8;
```

title	genre	avg_rating
The Blue Elephant 2	Drama	8.8
The Blue Elephant 2	Horror	8.8
The Blue Elephant 2	Mystery	8.8
The Brighton Miracle	Drama	9.5
The Irishman	Crime	8.7
The Irishman	Drama	8.7
The Colour of Darkness	Drama	9.1
Theeran Adhigaaram Ondru	Action	8.3
Theeran Adhigaaram Ondru	Crime	8.3
Theeran Adhigaaram Ondru	Thriller	8.3

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

QUERY

select count(*) as 'movie count = 8' 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;



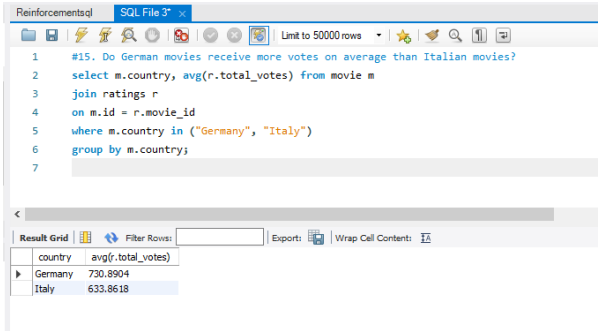
```sql
#14. Of the movies released between April 1, 2018, and April 1, 2019, how many received a median rating of 8?
select count(*) as 'movie count = 8' from movie m
join ratings r
on m.id = r.movie_id
where m.date_published between '2018-04-01' and '2019-04-01' and r.median_rating = 8;
```

movie count = 8
361

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

QUERY

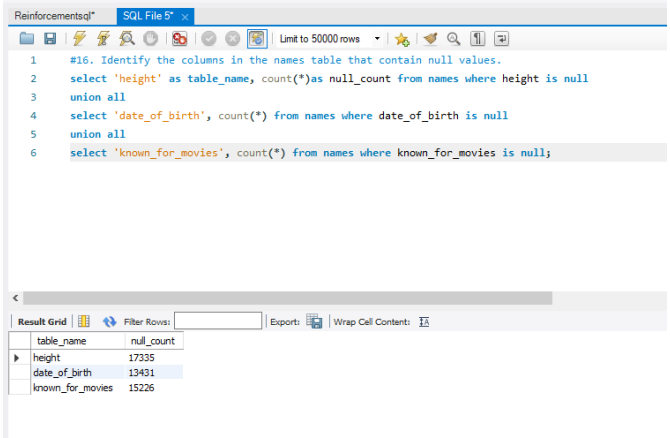
select m.country, avg(r.total_votes) from movie m
join ratings r
on m.id = r.movie_id
where m.country in ("Germany", "Italy")
group by m.country;



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

QUERY

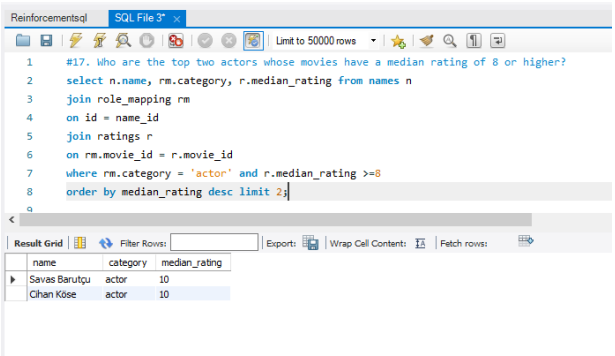
```
select 'height' as table_name, count(*)as null_count from names where height is null
union all
select 'date_of_birth', count(*) from names where date_of_birth is null
union all
select 'known_for_movies', count(*) from names where known_for_movies is null;
```



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

QUERY

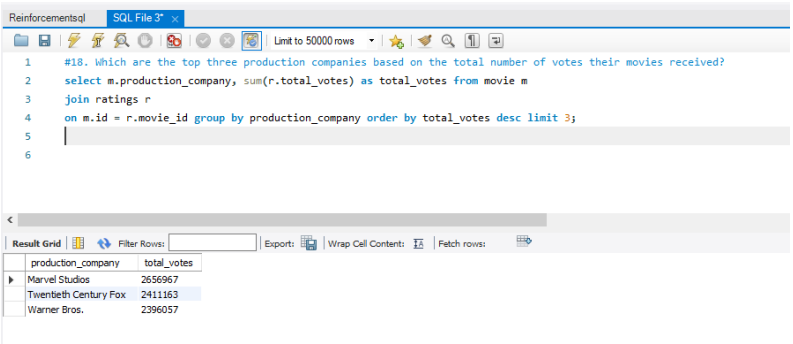
```
select n.name, rm.category, r.median_rating from names n
join role_mapping rm
on id = name_id
join ratings r
on rm.movie_id = r.movie_id
where rm.category = 'actor' and r.median_rating >=8
order by median_rating desc limit 2;
```



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

QUERY

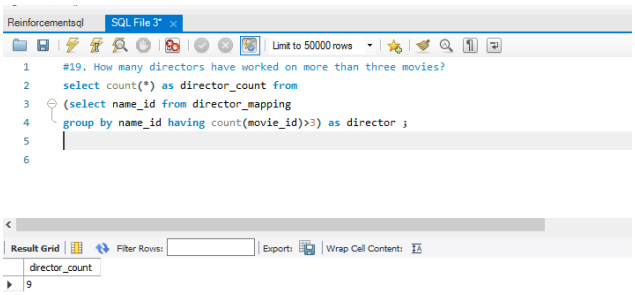
select m.production_company, sum(r.total_votes) as total_votes from movie m
join ratings r
on m.id = r.movie_id group by production_company order by total_votes desc limit 3;



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

QUERY

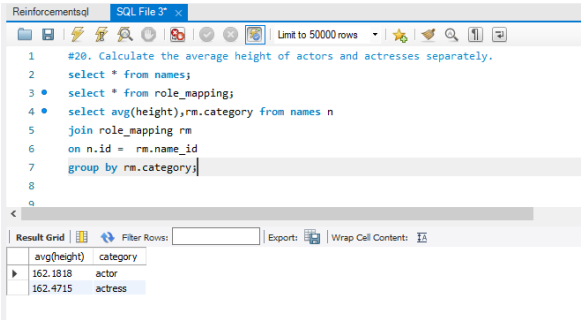
select count(*) as director_count from
(select name_id from director_mapping
group by name_id having count(movie_id)>3) as director ;



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

QUERY

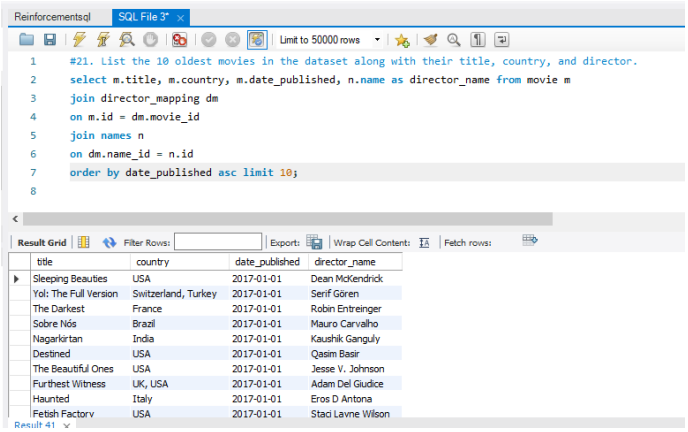
```
select * from names;
select * from role_mapping;
select avg(height),rm.category from names n
join role_mapping rm
on n.id =  rm.name_id
group by rm.category;
```



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

QUERY

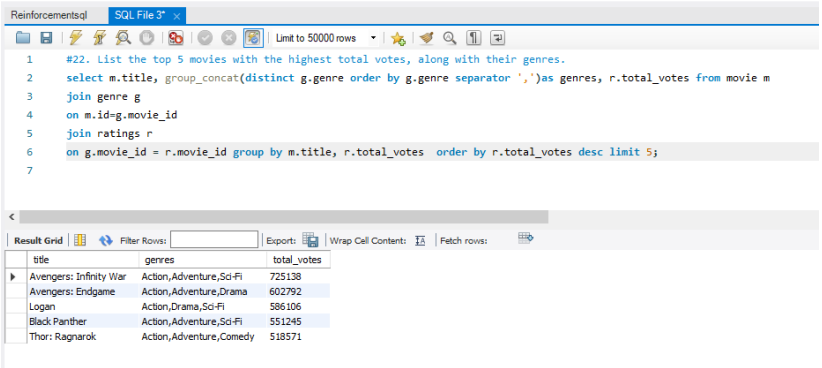
```
select m.title, m.country, m.date_published, n.name as director_name from movie m
join director_mapping dm
on m.id = dm.movie_id
join names n
on dm.name_id = n.id
order by date_published asc limit 10;
```



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

QUERY

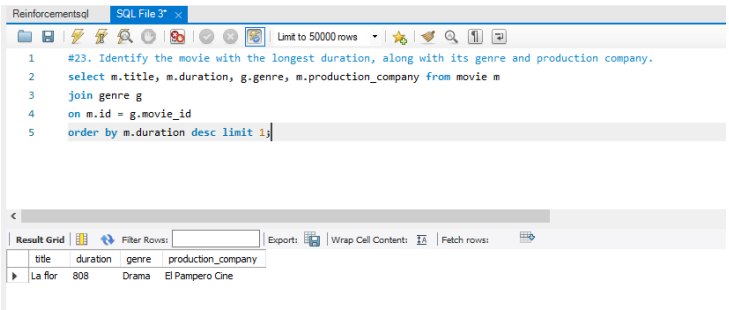
```
select m.title, group_concat(distinct g.genre order by g.genre separator ',')as genres,
r.total_votes from movie m
join genre g
on m.id=g.movie_id
join ratings r
on g.movie_id = r.movie_id group by m.title, r.total_votes  order by r.total_votes desc
limit 5;
```



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

QUERY

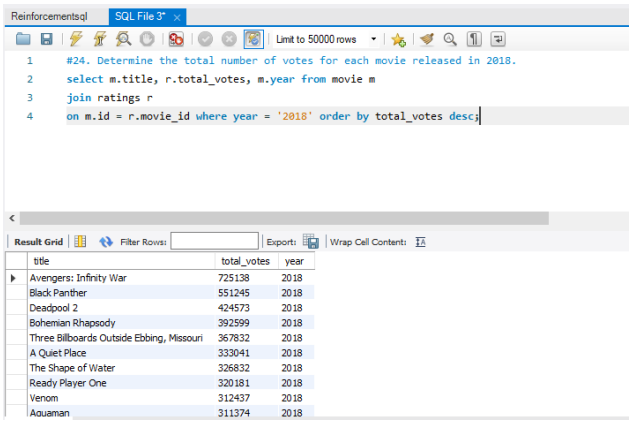
```
select m.title, m.duration, g.genre, m.production_company from movie m
join genre g
on m.id = g.movie_id
order by m.duration desc limit 1;
```



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

QUERY

select m.title, r.total_votes, m.year from movie m
join ratings r
on m.id = r.movie_id where year = '2018' order by total_votes desc;



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

QUERY

select languages,count(*) as movie_count from movie group by languages order by movie_count desc limit 1;