**IMDB DATABASE – ADVANCED SQL PROJECT**

* **Name : Ganeshan S**
* **Date : 28/03/2025**
* **Course : Data Science and Data Analysis**
* **Batch & Year : Feb 2025**

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

delimiter $$

create procedure total\_count()

begin

select count(\*) from director\_mapping;

select count(\*) from genre;

select count(\*) from movie;

select count(\*) from names;

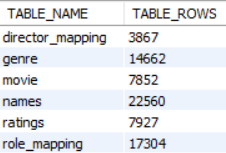
select count(\*) from ratings;

select count(\*) from role\_mapping;

end $$

delimiter ;

call total\_count;



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

select \* from movie;

select id from movie where id is null;

select title from movie where title is null;

select year from movie where year is null;

select date\_published from movie where date\_published is null;

select duration from movie where duration is null;

select country from movie where country is null;

select worlwide\_gross\_income from movie where worlwide\_gross\_income is null;

select languages from movie where languages is null;

select production\_company from movie where production\_company is null;



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

select

extract(month from date\_published) as release\_month,

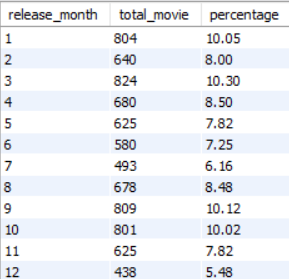
count(\*) as total\_movie,

round(count(\*) \* 100.0 / (select count(\*) from movie), 2) as percentage

from movie

group by extract(month from date\_published)

order by release\_month;



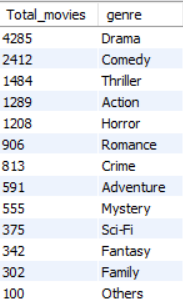
**-- 4. How many movies were produced in either the USA or India in the year 2019?**

select count(title), country, year from movie where year = 2019 group by country having country ="usa" or country ="india";



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

select count(movie\_id) as total\_movies, genre from genre group by genre order by count(movie\_id) desc;



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

select count(movie\_id), genre from genre group by genre order by count(movie\_id) desc limit 1;



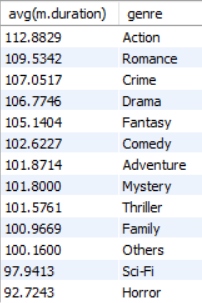
**-- 7. Calculate the average movie duration for each genre.**

select avg(m.duration), g.genre

from movie m left join genre g

on m.id = g.movie\_id

group by g.genre order by avg(m.duration) desc;



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

select m.category, count(m.name\_id) as movies, r.avg\_rating

from role\_mapping m inner join ratings r

on m.movie\_id = r.movie\_id

where r.avg\_rating < 5

group by m.name\_id, m.category, r.avg\_rating having count(m.name\_id) > 3 order by movies;



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

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;



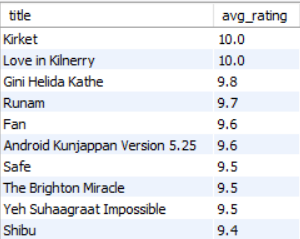
**-- 10. Which are the top 10 movies based on their average rating?**

select m.title, r.avg\_rating

from movie m inner join ratings r

on m.id = r.movie\_id

order by avg\_rating desc limit 10;



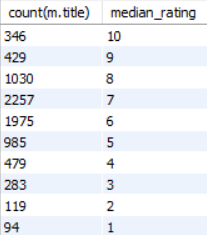
**-- 11. Summarize the ratings table by grouping movies based on their median ratings.**

select count(m.title),r.median\_rating

from movie m inner join ratings r

on m.id = r.movie\_id

group by r.median\_rating order by r.median\_rating desc;



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

select m.title, m.year, m.country, group\_concat(g.genre), r.total\_votes

from movie m inner join genre g

on m.id = g.movie\_id

inner join ratings r

on m.id = r.movie\_id

where year = 2017

and r.total\_votes > 1000 and m.country like '%usa%'

group by m.title, m.year, m.country, r.total\_votes

order by r.total\_votes desc;



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

select m.title, group\_concat(g.genre), r.avg\_rating

from movie m inner join genre g

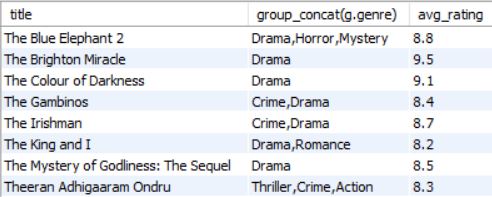
on m.id = g.movie\_id

inner join ratings r

on m.id = r.movie\_id

where r.avg\_rating > 8

group by m.title, r.avg\_rating having m.title like 'the%';



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

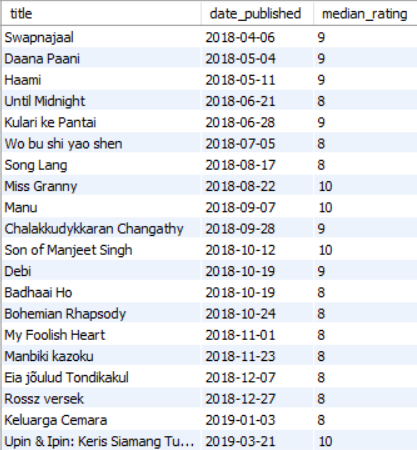
select m.title, m.date\_published, r.median\_rating

from movie m inner join ratings r

on r.movie\_id = m.id

where m.date\_published between '2018-04-01' and '2019-04-01'

and r.avg\_rating = 8 order by m.date\_published asc;



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

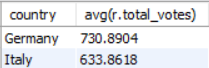
select m.country, avg(r.total\_votes)

from movie m inner join ratings r

on r.movie\_id = m.id

where m.country = "germany" or m.country = "italy"

group by m.country;



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

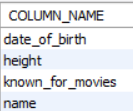
select id from names where id is null;

select name from names where name is null;

select height from names where height is null;

select date\_of\_birth from names where date\_of\_birth is null;

select known\_for\_movies 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?**

select rm.name\_id,n.name,rm.category,m.title as movie\_name,r.median\_rating

from role\_mapping rm

inner join movie m

on rm.movie\_id = m.id

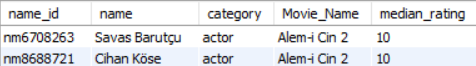
inner join ratings r

on rm.movie\_id = r.movie\_id

inner join names n

on rm.name\_id = n.id

where r.median\_rating>=8 order by r.median\_rating desc limit 2;



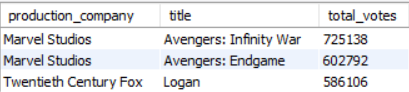
**-- 18. Which are the top three production companies based on the total number of votes their movies received?**

select m.production\_company, m.title, r.total\_votes

from movie m inner join ratings r

on r.movie\_id = m.id

order by total\_votes desc limit 3;



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

select count(dm.name\_id) as movie\_count, dm.name\_id,n.name

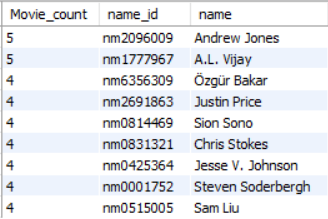
from director\_mapping dm inner join names n

on dm.name\_id = n.id

inner join movie m

on dm.movie\_id = m.id

group by dm.name\_id having movie\_count>3 order by movie\_count desc;



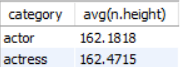
**-- 20. Calculate the average height of actors and actresses separately.**

select rm.category, avg(n.height)

from role\_mapping rm inner join names n

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.**

select dm.movie\_id, n.name, m.title, m.date\_published, m.country

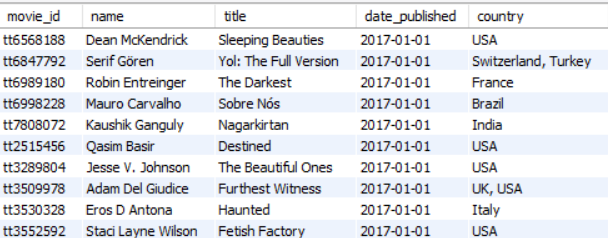
from director\_mapping dm inner join names n

on dm.name\_id = n.id

inner join movie m

on dm.movie\_id = m.id

order by m.date\_published asc limit 10;



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

select m.title, r.total\_votes, group\_concat(distinct g.genre) as genre

from movie m inner join genre g

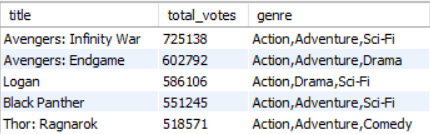
on m.id = g.movie\_id

inner join ratings r

on m.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.**

select m.title, m.duration, m.production\_company, group\_concat(distinct g.genre) as genre

from movie m inner join genre g

on m.id = movie\_id

group by m.title, m.duration, m.production\_company

order by m.duration desc limit 1;



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

select m.title, m.year, r.total\_votes

from movie m inner join ratings r

on m.id = r.movie\_id

having m.year = 2018

order by r.total\_votes desc limit 15;



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

select

languages,

count(\*) as movie\_count

from (

select

trim(substring\_index(substring\_index(languages, ',', n.n), ',', -1)) as languages

from movie

inner join (

select 1 as n union all select 2 union all select 3 union all select 4 union all select 5

union all select 6 union all select 7 union all select 8 union all select 9 union all select 10

) n on char\_length(languages) - char\_length(replace(languages, ',', '')) >= n.n - 1

) sub

group by languages

order by movie\_count desc;

