**Segment 1: Database - Tables, Columns, Relationships**

What are the different tables in the database and how are they connected to each other in the database?

OUTPUT

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

OUTPUT

Identify which columns in the movie table have null values.

select max(case when title is null then 1 else 0 end) as title ,

max(case when id is null then 1 else 0 end) as id,

max(case when year is null then 1 else 0 end) as year,

max(case when date\_published is null then 1 else 0 end) as date\_published,

max(case when duration is null then 1 else 0 end) as duration,

max(case when country is null then 1 else 0 end) as country,

max(case when worlwide\_gross\_income is null then 1 else 0 end) as worlwide\_gross\_income,

max(case when languages is null then 1 else 0 end) as languages,

max(case when production\_company is null then 1 else 0 end) as production\_company

from movies;

OUTPUT

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Segment 2: Movie Release Trends**

Determine the total number of movies released each year and analyse the month-wise trend

Total number of movie released each year

SELECT

year,

COUNT(\*) AS total\_movies

FROM

movies

GROUP BY

year

ORDER BY

year;

OUTPUT

|  |  |
| --- | --- |
| 2017 | 3052 |
| 2018 | 2944 |
| 2019 | 2001 |

Month-wise trend of movie releases

SELECT

year,

MONTH(date\_published) AS release\_month,

COUNT(\*) AS total\_movies

FROM

.movies

GROUP BY

year,

MONTH(date\_published)

ORDER BY

year, release\_month;

OUTPUT

|  |  |  |
| --- | --- | --- |
| 2017 |  | 3052 |
| 2018 |  | 294 |
| 2019 |  | 2001 |

* Calculate the number of movies produced in the USA or India in the year 2019.

SELECT

country,

COUNT(\*) AS total\_movies

FROM

movies

WHERE

(country = 'USA' OR country = 'India') AND year = 2019

GROUP BY

country;

OUTPUT

|  |  |
| --- | --- |
| India | 295 |
| USA | 592 |

**Segment 3: Production Statistics and Genre Analysis**

Retrieve the unique list of genres present in the dataset.

SELECT DISTINCT

genre

FROM

genre;

OUTPUT

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

Identify the genre with the highest number of movies produced overall.

SELECT

genre,

COUNT(\*) AS total\_movies

FROM

genre

GROUP BY

genre

ORDER BY

total\_movies DESC

LIMIT 1;

OUTPUT

|  |  |
| --- | --- |
| Drama | 4285 |

Determine the count of movies that belong to only one genre.

SELECT COUNT(\*) AS single\_genre\_movies

FROM (

SELECT movie\_id

FROM genre

GROUP BY movie\_id

HAVING COUNT(DISTINCT genre) = 1

) AS single\_genre\_movies\_list;

OUTPUT

|  |
| --- |
| 3289 |

Calculate the average duration of movies in each genre.

SELECT

g.genre,

AVG(m.duration) AS average\_duration

FROM

genre g

JOIN

movies m ON g.movie\_id = m.id

GROUP BY

g.genre;

OUTPUT

|  |  |
| --- | --- |
| Drama | 106.7746 |
| Fantasy | 105.1404 |
| Thriller | 101.5761 |
| Comedy | 102.6227 |
| Horror | 92.7243 |
| Family | 100.9669 |
| Romance | 109.5342 |
| Adventure | 101.8714 |
| Mystery | 101.8000 |
| Action | 112.8829 |
| Crime | 107.0517 |
| Sci-Fi | 97.9413 |
| Others | 100.1600 |

Find the rank of the 'thriller' genre among all genres in terms of the number of movies produced.

SELECT

genre,

RANK() OVER (ORDER BY COUNT(movie\_id) DESC) AS genre\_rank

FROM

genre

WHERE

genre = 'Thriller'

GROUP BY

genre;

OUTPUT

|  |  |
| --- | --- |
| Thriller | 1 |

**Segment 4: Ratings Analysis and Crew Members**

Retrieve the minimum and maximum values in each column of the ratings table (except movie\_id).

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;

OUTPUT

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 1 | 10 | 100 | 725138 | 1 | 10 |

Identify the top 10 movies based on average rating

SELECT

m.id,

m.title,

r.avg\_rating

FROM

movies m

JOIN

ratings r ON m.id = r.movie\_id

ORDER BY

r.avg\_rating DESC

LIMIT 10;

OUTPUT

|  |  |  |
| --- | --- | --- |
| tt6735740 | Love in Kilnerry | 10 |
| tt10914342 | Kirket | 10 |
| tt9537008 | Gini Helida Kathe | 9.8 |
| tt10370434 | Runam | 9.7 |
| tt10867504 | Fan | 9.6 |
| tt9526826 | Android Kunjappan Version 5.25 | 9.6 |
| tt9680166 | Yeh Suhaagraat Impossible | 9.5 |
| tt10901588 | The Brighton Miracle | 9.5 |
| tt10869474 | Safe | 9.5 |
| tt10405902 | Shibu | 9.4 |

Summarise the ratings table based on movie counts by median ratings

SELECT

r.median\_rating,

COUNT(r.movie\_id) AS movie\_count

FROM

ratings r

GROUP BY

r.median\_rating

ORDER BY

r.median\_rating;

OUTPUT

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

Identify the production house that has produced the most number of hit movies (average rating > 8).

SELECT

m.production\_company,

COUNT(r.movie\_id) AS hit\_movie\_count

FROM

movies m

JOIN

ratings r ON m.id = r.movie\_id

WHERE

r.avg\_rating > 8

GROUP BY

m.production\_company

ORDER BY

hit\_movie\_count DESC

LIMIT 1;

OUTPUT

|  |  |
| --- | --- |
|  | 21 |

Determine the number of movies released in each genre during March 2017 in the USA with more than 1,000 votes.

select count(\*)

from movies m

join genre g on g.movie\_id = m.id

join ratings r on r.movie\_id = m.id

where r.total\_votes > 1000 and lower(country) = "usa" and

year( str\_to\_date (m.date\_published,'%m/%d/%Y')) = 2017 and

month( str\_to\_date (m.date\_published,'%m/%d/%Y')) = 3;

OUTPUT

|  |
| --- |
| 54 |

Retrieve movies of each genre starting with the word 'The' and having an average rating > 8.

SELECT

m.id,

m.title,

g.genre,

r.avg\_rating

FROM

movies m

JOIN

genre g ON m.id = g.movie\_id

JOIN

ratings r ON m.id = r.movie\_id

WHERE

g.genre LIKE 'The%'

AND r.avg\_rating > 8;

OUTPUT

NULL

**Segment 5: Crew Analysis**

Identify the columns in the names table that have null values.

select max(case when known\_for\_movies is null then 1 else 0 end) as known\_for\_movies ,

max(case when id is null then 1 else 0 end) as id,

max(case when date\_of\_birth is null then 1 else 0 end) as date\_of\_birth,

max(case when name is null then 1 else 0 end) as name,

max(case when height is null then 1 else 0 end) as height

from names;

OUTPUT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0 | 0 | 0 | 0 | 0 |

Determine the top three directors in the top three genres with movies having an average rating > 8

select M.title , g.genre, r.avg\_rating

from movies as m

inner join genre as g on m.id = g.movie\_id

inner join ratings as r on m.id = r.movie\_id

where r.avg\_rating >8

order by r.avg\_rating desc

limit 3;

OUTPUT

|  |  |  |
| --- | --- | --- |
| Kirket | Drama | 10 |
| Love in Kilnerry | Comedy | 10 |
| Gini Helida Kathe | Drama | 9.8 |

Find the top two actors whose movies have a median rating >= 8.

select n.name , r.median\_rating

from movies as m

inner join names as n on m.id = n.known\_for\_movies

inner join ratings as r on m.id = r.movie\_id

where r.median\_rating >=8

order by r.median\_rating

limit 3 ;

OUTPUT

|  |  |
| --- | --- |
| Casey Affleck | 8 |
| Annette Bening | 8 |
| David Thewlis | 8 |

Identify the top three production houses based on the number of votes received by their movies.

SELECT production\_company,

SUM(total\_votes) AS total\_votes\_received

FROM movies

JOIN ratings ON movies.id = ratings.movie\_id

GROUP BY production\_company

ORDER BY total\_votes\_received DESC

LIMIT 3;

OUTPUT

|  |  |
| --- | --- |
| Marvel Studios | 2656967 |
| Twentieth Century Fox | 2411163 |
| Warner Bros. | 2396057 |

Rank actors based on their average ratings in Indian movies released in India.

OUTPUT

Identify the top five actresses in Hindi movies released in India based on their average ratings.

OUTPUT

**Segment 6: Broader Understanding of Data**

Classify thriller movies based on average ratings into different categories.

SELECT

tm.movie\_id,

tm.title,

tm.avg\_rating,

CASE

WHEN tm.avg\_rating >= 9.0 THEN 'Masterpiece'

WHEN tm.avg\_rating >= 8.0 AND tm.avg\_rating < 9.0 THEN 'Highly Acclaimed'

WHEN tm.avg\_rating >= 7.0 AND tm.avg\_rating < 8.0 THEN 'Solid Performers'

WHEN tm.avg\_rating >= 6.0 AND tm.avg\_rating < 7.0 THEN 'Above Average'

ELSE 'Below Average'

END AS rating\_category

FROM

ThrillerMovies tm;

OUTPUT

|  |  |  |  |
| --- | --- | --- | --- |
| tt0012494 | Der müde Tod | 7.7 | Solid Performers |
| tt0360556 | Fahrenheit 451 | 4.9 | Below Average |
| tt0837563 | Pet Sematary | 5.8 | Below Average |
| tt0862930 | Dukun | 6.9 | Above Average |
| tt0972544 | Back Roads | 7 | Solid Performers |
| tt10039344 | Countdown | 5.4 | Below Average |
| tt10048556 | Staged Killer | 3.3 | Below Average |
| tt10055770 | Vellaipookal | 7.3 | Solid Performers |
| tt10121762 | Uriyadi 2 | 7.3 | Solid Performers |
| tt10122392 | Incitement | 7.5 | Solid Performers |
| tt10133300 | Rakshasudu | 8.4 | Highly Acclaimed |

analyse the genre-wise running total and moving average of the average movie duration.

OUTPUT

Identify the five highest-grossing movies of each year that belong to the top three genres.

WITH RankedMovies AS (

SELECT

m.id AS movie\_id,

m.title,

m.year,

m.worlwide\_gross\_income,

g.genre,

ROW\_NUMBER() OVER (PARTITION BY m.year, g.genre ORDER BY m.worlwide\_gross\_income DESC) AS genre\_rank

FROM

movies m

JOIN genre g ON m.id = g.movie\_id

)

SELECT

movie\_id,

title,

year,

worlwide\_gross\_income,

genre

FROM

RankedMovies

WHERE

genre\_rank <= 5;

OUTPUT

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| tt1724970 | Beyond Skyline | 2017 | $ 992181 | Action |
| tt4600952 | Zashchitniki | 2017 | $ 9765483 | Action |
| tt6413410 | V.I.P. | 2017 | $ 9710283 | Action |
| tt1935194 | Overdrive | 2017 | $ 9650552 | Action |
| tt2283362 | Jumanji: Welcome to the Jungle | 2017 | $ 962102237 | Action |
| tt1724970 | Beyond Skyline | 2017 | $ 992181 | Adventure |
| tt6673840 | Vremya pervykh | 2017 | $ 9677311 | Adventure |

Determine the top two production houses that have produced the highest number of hits among multilingual movies.

WITH HitMovies AS (

SELECT

production\_company,

COUNT(\*) AS hit\_count

FROM

movies

WHERE

worlwide\_gross\_income IS NOT NULL

AND worlwide\_gross\_income != '$0'

GROUP BY

production\_company

HAVING

COUNT(\*) > 1

)

SELECT

production\_company,

hit\_count

FROM

HitMovies

ORDER BY

hit\_count DESC

LIMIT 2;

OUTPUT

|  |  |
| --- | --- |
|  | 528 |
| Netflix | 20 |