SQL Scripts

1. Average movie runtime change over the years:

SELECT

D.year,

AVG(F.runtimeMinutes) AS average\_runtime

FROM

Fact\_movie F

JOIN

Dim\_date D ON F.startYear = D.year

GROUP BY

D.year

ORDER BY

D.year;

1. Is there a correlation between the average movie rating and the year of release?

SELECT

D.year,

AVG(F.averageRating) AS average\_rating

FROM

Fact\_movie F

JOIN

Dim\_date D ON F.startYear = D.year

GROUP BY

D.year

ORDER BY

D.year;

1. distribution of movie releases per year

SELECT

D.year,

COUNT(F.tconst) AS movie\_count

FROM

Fact\_movie F

JOIN

Dim\_date D ON F.startYear = D.year

GROUP BY

D.year

ORDER BY

D.year;

1. Which genres have seen the most significant ratings in popularity over the past decade?

SELECT

G.genre,

AVG(F.averageRating) AS average\_rating

FROM

Fact\_movie F

JOIN

Dim\_movie D ON F.tconst = D.tconst

JOIN

Bridge\_MovieGenre BG ON F.tconst = BG.tconst

JOIN

Dim\_genre G ON BG.genre\_id = G.genre\_id

JOIN

Dim\_date DD ON F.startYear = DD.year

WHERE

DD.year BETWEEN YEAR(CURDATE()) - 10 AND YEAR(CURDATE()) - 1

GROUP BY

G.genre

ORDER BY

average\_rating DESC;

1. Show the top 5 genres as compared to gross earnings for the 9 box office movies.

SELECT

G.genre,

SUM(E.gross) AS total\_earnings

FROM

Fact\_movieEarning E

JOIN

Dim\_movie D ON E.tconst = D.tconst

JOIN

Bridge\_MovieGenre BG ON E.tconst = BG.tconst

JOIN

Dim\_genre G ON BG.genre\_id = G.genre\_id

WHERE

E.movie\_id IN (SELECT TOP 9 movie\_id FROM Fact\_movie ORDER BY numVotes DESC)

GROUP BY

G.genre

ORDER BY

total\_earnings DESC

LIMIT 5;

1. Correlation between the movie's runtime and its average rating

SELECT

AVG(F.runtimeMinutes) AS avg\_runtime,

AVG(F.averageRating) AS avg\_rating,

CORR(F.runtimeMinutes, F.averageRating) AS correlation

FROM

Fact\_movie F

WHERE

F.runtimeMinutes IS NOT NULL

AND F.averageRating IS NOT NULL;

1. Correlation between the movie's runtime and its average gross

SELECT

AVG(F.runtimeMinutes) AS avg\_runtime,

AVG(E.gross) AS avg\_gross,

CORR(F.runtimeMinutes, E.gross) AS correlation

FROM

Fact\_movie F

JOIN

Fact\_movieEarning E ON F.tconst = E.tconst

WHERE

F.runtimeMinutes IS NOT NULL

AND E.gross IS NOT NULL;

1. Relationship between the number of votes and the average rating of movies

SELECT

AVG(F.numVotes) AS avg\_num\_votes,

AVG(F.averageRating) AS avg\_rating,

CORR(F.numVotes, F.averageRating) AS correlation

FROM

Fact\_movie F

WHERE

F.numVotes IS NOT NULL

AND F.averageRating IS NOT NULL;

1. Identify directors with the most films rated above 7. Sort them in the descending order

SELECT

P.primaryName AS director\_name,

COUNT(F.tconst) AS num\_films\_above\_7

FROM

Bridge\_MoviePerson BP

JOIN

Dim\_person P ON BP.person\_id = P.person\_id

JOIN

Fact\_movie F ON BP.tconst = F.tconst

WHERE

BP.role = 'director'

AND F.averageRating > 7

GROUP BY

P.primaryName

ORDER BY

num\_films\_above\_7 DESC;

1. Determine the directors who have directed the highest number of films overall and their respective gross earnings trends

WITH DirectorFilmCounts AS (

SELECT

BP.person\_id,

P.primaryName AS director\_name,

COUNT(DISTINCT F.tconst) AS num\_films

FROM

Bridge\_MoviePerson BP

JOIN

Dim\_person P ON BP.person\_id = P.person\_id

JOIN

Fact\_movie F ON BP.tconst = F.tconst

WHERE

BP.role = 'director'

GROUP BY

BP.person\_id, P.primaryName

)

SELECT

DFC.director\_name,

DFC.num\_films,

COALESCE(SUM(E.gross), 0) AS total\_earnings

FROM

DirectorFilmCounts DFC

LEFT JOIN

Bridge\_MoviePerson BP ON DFC.person\_id = BP.person\_id

LEFT JOIN

Fact\_movieEarning E ON BP.tconst = E.tconst

GROUP BY

DFC.director\_name, DFC.num\_films

ORDER BY

DFC.num\_films DESC;

1. List the top 10 actors/actresses with the most films rated between 4 and 7.

WITH ActorFilmCounts AS (

SELECT

BP.person\_id,

P.primaryName AS actor\_name,

COUNT(DISTINCT F.tconst) AS num\_films

FROM

Bridge\_MoviePerson BP

JOIN

Dim\_person P ON BP.person\_id = P.person\_id

JOIN

Fact\_movie F ON BP.tconst = F.tconst

WHERE

BP.role IN ('actor', 'actress')

AND F.averageRating BETWEEN 4 AND 7

GROUP BY

BP.person\_id, P.primaryName

)

SELECT

AFC.actor\_name,

AFC.num\_films

FROM

ActorFilmCounts AFC

ORDER BY

AFC.num\_films DESC

LIMIT 10;

1. Compare the top 5 actors and actresses based on movie ratings.

WITH ActorRatingAverages AS (

SELECT

BP.person\_id,

P.primaryName AS actor\_name,

AVG(F.averageRating) AS avg\_rating

FROM

Bridge\_MoviePerson BP

JOIN

Dim\_person P ON BP.person\_id = P.person\_id

JOIN

Fact\_movie F ON BP.tconst = F.tconst

WHERE

BP.role IN ('actor', 'actress')

AND F.averageRating IS NOT NULL

GROUP BY

BP.person\_id, P.primaryName

)

SELECT

ARA.actor\_name,

ARA.avg\_rating

FROM

ActorRatingAverages ARA

ORDER BY

ARA.avg\_rating DESC

LIMIT 5;

1. How does movie performance vary across different seasons of the year? (based on gross earnings)

SELECT

DD.season,

AVG(E.gross) AS average\_earnings

FROM

Fact\_movieEarning E

JOIN

Dim\_date DD ON E.date\_id = DD.date\_id

WHERE

E.gross IS NOT NULL

GROUP BY

DD.season

ORDER BY

DD.season;

1. Top 3 movies based on the season (spring, summer, fall)

WITH SeasonalMovieEarnings AS (

SELECT

DD.season,

E.tconst,

D.primaryTitle,

E.gross

FROM

Fact\_movieEarning E

JOIN

Dim\_date DD ON E.date\_id = DD.date\_id

JOIN

Dim\_movie D ON E.tconst = D.tconst

WHERE

E.gross IS NOT NULL

)

SELECT

SME.season,

SME.primaryTitle AS movie\_title,

SME.gross

FROM

SeasonalMovieEarnings SME

WHERE

SME.season IN ('spring', 'summer', 'fall')

ORDER BY

SME.season, SME.gross DESC

LIMIT 3;

1. Identify movies that have had the widest release across multiple regions.

WITH MovieRegionCounts AS (

SELECT

M.tconst,

D.primaryTitle AS movie\_title,

COUNT(DISTINCT MR.region\_id) AS num\_regions

FROM

Bridge\_MovieRegion MR

JOIN

Dim\_movie M ON MR.tconst = M.tconst

JOIN

Dim\_region D ON MR.region\_id = D.region\_id

GROUP BY

M.tconst, D.primaryTitle

)

SELECT

MRC.movie\_title,

MRC.num\_regions

FROM

MovieRegionCounts MRC

ORDER BY

MRC.num\_regions DESC

LIMIT 5;

1. Create a dashboard for the 9 box office movies that shows movie-wise: 1. No. of directors 2. No. of actors/actresses 3. No. of writers 4. Total worldwide gross earnings 5. Average ratings 6. No. of regions, it was released 7. No. of genres

WITH MovieDashboard AS (

SELECT

M.tconst,

COUNT(DISTINCT CASE WHEN BP.role = 'director' THEN BP.person\_id END) AS num\_directors,

COUNT(DISTINCT CASE WHEN BP.role IN ('actor', 'actress') THEN BP.person\_id END) AS num\_actors\_actresses,

COUNT(DISTINCT CASE WHEN BP.role = 'writer' THEN BP.person\_id END) AS num\_writers,

SUM(E.gross) AS total\_earnings,

AVG(F.averageRating) AS avg\_rating,

COUNT(DISTINCT MR.region\_id) AS num\_regions,

COUNT(DISTINCT BG.genre\_id) AS num\_genres

FROM

Dim\_movie M

LEFT JOIN

Bridge\_MoviePerson BP ON M.tconst = BP.tconst

LEFT JOIN

Fact\_movieEarning E ON M.tconst = E.tconst

LEFT JOIN

Fact\_movie F ON M.tconst = F.tconst

LEFT JOIN

Bridge\_MovieRegion MR ON M.tconst = MR.tconst

LEFT JOIN

Bridge\_MovieGenre BG ON M.tconst = BG.tconst

WHERE

F.numVotes IS NOT NULL

GROUP BY

M.tconst

)

SELECT

MD.tconst,

MD.num\_directors,

MD.num\_actors\_actresses,

MD.num\_writers,

MD.total\_earnings,

MD.avg\_rating,

MD.num\_regions,

MD.num\_genres

FROM

MovieDashboard MD

ORDER BY

MD.total\_earnings DESC

LIMIT 9;