--creating a database

create database Moviesdb;

--using the database

use Moviesdb;

--table has been imported via import wizard table name-> netflixAmazon

-- I will create a duplicate of the original table to make sure i have the original data in case if i perform any mistakes

select \* into netflixAmazon2 from netflixAmazon;

--i will use and perform operations on table netflixAmazon2

--exploring the table

select top 10 \* from netflixAmazon2;

--i dont require the column column1

alter table netflixAmazon2 drop column column1;

--datatypes of columns

EXEC sp\_help 'dbo.netflixAmazon2';

-- i want to delete data where the movies are not streamed on either platforms

select \* from netflixAmazon2 where Netflix=0 and Amazon\_Prime\_Video=0;

/\*hopefully there are no rows satisfying this condition

if there were any rows i would delete them \*/

--exploring the IMDB column

select distinct IMDB from netflixAmazon2 order by 1;

--i have values like d;} nan

--exploring Rottenm\_Tomatoes column

select distinct Rotten\_Tomatoes from netflixAmazon2 order by 1;

-- i found value like na

--i want to delete data where i dont have any scores from both

select \* from netflixAmazon2

where (IMDB='d;}' or IMDB='nan') and Rotten\_Tomatoes='na';

--delete

delete from netflixAmazon2 where (IMDB='d;}' or IMDB='nan') and Rotten\_Tomatoes='na';

--changing the values d;} nan and na to -1

update netflixAmazon2

set IMDB='-1' where IMDB='d;}' or IMDB='nan';

update netflixAmazon2

set Rotten\_Tomatoes='-1' where Rotten\_Tomatoes='na';

--changing the datatypes

alter table netflixAmazon2 alter column Rotten\_Tomatoes int;

--searching for duplicates

select Title, COUNT(\*) as Count

from netflixAmazon2

group by Title

having COUNT(\*) > 1

order by Count desc;

--

select \* from netflixAmazon2 order by title,genre;

select Title,genre,COUNT(\*) as DuplicateCount

from netflixAmazon2

group by Title, genre

having COUNT(\*) > 1;

alter table netflixAmazon2 add MinAge int;

update netflixAmazon2

set MinAge = TRY\_CasT(REPLACE(Rating, '+', '') as int);

alter table netflixAmazon2 drop column Rating;

--exploring title column

select distinct title from netflixAmazon2;

--removing

update netflixAmazon2

set Title = LTRIM(RTRIM(Title));

--creating a view where genres are merged

create view MergedGenresview as

select Title, Year,MinAge,IMDb,Rotten\_Tomatoes,

STRING\_AGG(Genre, ', ') as CombinedGenres,

MAX(Netflix) as Netflix,

MAX(Amazon\_Prime\_Video) as Amazon\_Prime\_Video

from netflixAmazon2

group by Title, Year, MinAge, IMDb, Rotten\_Tomatoes,Netflix,Amazon\_Prime\_Video;

select \* from MergedGenresView

drop view if exists MergedGenresView;

create view MergedGenresview as

select Title,Year,MinAge,IMDb,Rotten\_Tomatoes,

REPLACE(STRING\_AGG(Genre, ', '), '&', ',') as CombinedGenres,

MAX(Netflix) as Netflix,

MAX(Amazon\_Prime\_Video) as Amazon\_Prime\_Video

from netflixAmazon2

group by Title, Year, MinAge, IMDb, Rotten\_Tomatoes;

select \* from MergedGenresView where IMDB<0 or Rotten\_Tomatoes<0;

delete from netflixAmazon2 where IMDb<0 or Rotten\_Tomatoes<0;

--Data analysis

--1. counting the total number of movies

select COUNT(\*) as total\_movies from MergedGenresView;

--2. counting number of movies in each platform and their percentage

select SUM(Netflix) as NetflixCount, SUM(Amazon\_Prime\_Video) as AmazonPrimeCount,

COUNT(\*) as TotalMovies,

ROUND(100.0 \* SUM(Netflix) / COUNT(\*), 2) as NetflixPercentage,

ROUND(100.0 \* SUM(Amazon\_Prime\_Video) / COUNT(\*), 2) as AmazonPrimePercentage

from MergedGenresView;

--3. Average Rating per platform

select 'Netflix' as Platform, AVG(IMDb) as Avg\_IMDb, AVG(Rotten\_tomatoes) as AVG\_RottenTomatoes

from MergedGenresView where Netflix = 1

union all

select 'Amazon Prime Video' as Platform, AVG(IMDb) as Avg\_IMDb,AVG(Rotten\_tomatoes) as AVG\_RottenTomatoes

from MergedGenresView where Amazon\_Prime\_Video = 1;

--4. Movies per year

select [Year] as year ,count(\*) as num\_movies from MergedGenresView group by [Year] order by 1;

--5 top 5 highest rated titles by IMDB

select top 10 title,IMDB

from MergedGenresView order by 2 desc;

--top 5 highest rated titles by rotten tomato

select top 10 title,Rotten\_Tomatoes

from MergedGenresView order by 2 desc;

--lowest rated movies of all time

select top 5 title,IMDB, Rotten\_Tomatoes

from MergedGenresView order by IMDB,Rotten\_Tomatoes;

--6 top genres by average IMDB rating

select TRIM(value) as Genre, COUNT(\*) as num\_titles, AVG(IMDb) as Avg\_IMDb

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by TRIM(value) having COUNT(\*) > 5

order by Avg\_IMDb desc;

-- 7 Title Count by Platform and Age Category

select Platform, MinAge, COUNT(\*) as NumTitles

from MergedGenresView

cross apply (

VALUES

('Netflix', Netflix),

('Amazon Prime Video', Amazon\_Prime\_Video)

) as Platforms(Platform, IsAvailable)

where IsAvailable = 1 and MinAge is not null

group by Platform, MinAge

order by Platform, MinAge;

-- 8 Average Ratings Per Genre and Platform

select TRIM(value) as Genre,Platform,COUNT(\*) as NumTitles,AVG(IMDb) as Avg\_IMDb,

AVG(Rotten\_Tomatoes) as Avg\_RT

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

cross apply (

VALUES

('Netflix', Netflix),

('Amazon Prime Video', Amazon\_Prime\_Video)

) as Platforms(Platform, IsAvailable)

where IsAvailable = 1

group by TRIM(value), Platform

order by Genre, Platform;

-- 9 Year-on-Year Trends in Average IMDb Rating

select Year,COUNT(\*) as NumMovies,AVG(IMDb) as Avg\_IMDb

from MergedGenresView

group by Year

order by Year;

-- 10 Under-Rated movies (High IMDb, Low Rotten Tomatoes)

select Title,Year,IMDb,Rotten\_Tomatoes,CombinedGenres

from MergedGenresView

where IMDb >= 8 AND Rotten\_Tomatoes < 35

order by IMDb desc;

-- 11 Overhyped Movies(High RT, Low IMDb)

select Title,Year,IMDb,Rotten\_Tomatoes,CombinedGenres

from MergedGenresView

where Rotten\_Tomatoes > 70 AND IMDb < 6

order by Rotten\_Tomatoes desc;

-- 12 Highest Rated Title Per Genre

WITH GenreRanks as (

select TRIM(value) as Genre,Title,Year,IMDb,

ROW\_NUMBER() OVER (PARTITION BY TRIM(value) order by IMDb desc) as rn

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

)

select \* from GenreRanks

where rn = 1

order by Genre;

-- 13 IMDb Rating Distribution

select FLOOR(IMDb) as RatingBucket,COUNT(\*) as NumTitles

from MergedGenresView

group by FLOOR(IMDb)

order by RatingBucket;

-- 14 Find genres with high ratings but few titles (untapped opportunity)

WITH GenreStats as (

select TRIM(value) as Genre,COUNT(\*) as NumTitles,AVG(IMDb) as Avg\_IMDb

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by TRIM(value)

)

select \* from GenreStats

where NumTitles < 500 AND Avg\_IMDb > 6

order by Avg\_IMDb desc;

-- 15 Calculate IMDb rating volatility within genres

select

TRIM(value) as Genre,

COUNT(\*) as NumTitles,

AVG(IMDb) as Avg\_IMDb,

STDEV(IMDb) as StdDev\_IMDb

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by TRIM(value)

having COUNT(\*) > 5

order by StdDev\_IMDb desc;

-- 16 Analyze genre popularity over time

select Year,TRIM(value) as Genre,COUNT(\*) as NumTitles

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by Year, TRIM(value)

order by Year, NumTitles desc;

-- 17 Most Increasing Genre (YOY Growth)

WITH GenreYearCounts as (

select Year,TRIM(value) as Genre,COUNT(\*) as NumTitles

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by Year, TRIM(value)

),

GenreWithLag as (

select \*,

LAG(NumTitles) OVER (PARTITION BY Genre order by Year) as PrevYearCount,

NumTitles - LAG(NumTitles) OVER (PARTITION BY Genre order by Year) as Growth

from GenreYearCounts

)

select top 5 \* from GenreWithLag

where Growth is not null

order by Growth desc;

-- 18 Find hidden gems with high ratings in less popular genres

WITH GenrePopularity as (

select TRIM(value) as Genre,count(\*) as GenreCount

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by TRIM(value)

)

select m.Title, m.Year, m.IMDb, m.Rotten\_Tomatoes, TRIM(g.value) as Genre,gp.GenreCount

from MergedGenresView m

cross apply STRING\_SPLIT(m.CombinedGenres, ',') g

join GenrePopularity gp ON TRIM(g.value) = gp.Genre

where m.IMDb > 7.5 AND gp.GenreCount < 100

order by m.IMDb desc;

-- 19 Detect highly polarizing titles

-- Big differences between IMDb and Rotten Tomatoes, regardless of which is higher

select Title,Year,IMDb,Rotten\_Tomatoes,ABS((IMDb \* 10) - Rotten\_Tomatoes) as RatingGap,CombinedGenres

from MergedGenresView

where ABS((IMDb \* 10) - Rotten\_Tomatoes) >= 50

order by RatingGap desc;

-- 20 Platform similarity score by genre

-- How similar are Netflix and Prime’s genre libraries?

WITH PlatformTotals as (

select

SUM(case when Netflix = 1 then 1 else 0 end) as TotalNetflix,

SUM(case when Amazon\_Prime\_Video = 1 then 1 else 0 end) as TotalPrime

from MergedGenresView

),

GenrePlatformCounts as (

select LTRIM(RTRIM(value)) as Genre,

SUM(case when Netflix = 1 then 1 else 0 end) as Netflix\_Count,

SUM(case when Amazon\_Prime\_Video = 1 then 1 else 0 end) as Prime\_Count

from MergedGenresView

cross apply STRING\_SPLIT(CombinedGenres, ',')

group by LTRIM(RTRIM(value))

),

GenreSimilarity as (

select g.Genre,g.Netflix\_Count,g.Prime\_Count,

-- compute shares safely, avoiding division by zero

case

when pt.TotalNetflix = 0 then 0

else CasT(g.Netflix\_Count as float) / pt.TotalNetflix

end as Netflix\_Share,

case

when pt.TotalPrime = 0 then 0

else CasT(g.Prime\_Count as float) / pt.TotalPrime

end as Prime\_Share

from GenrePlatformCounts g

cross join PlatformTotals pt

)

select Genre, Netflix\_Count, Prime\_Count, Netflix\_Share, Prime\_Share,

1.0 - ABS(Netflix\_Share - Prime\_Share) as SimilarityScore

from GenreSimilarity

order by SimilarityScore desc;

-- 21 IMDb volatility over time

-- Do certain years produce wildly inconsistent ratings?

select Year, COUNT(\*) as NumTitles, AVG(IMDb) as Avg\_IMDb, STDEV(IMDb) as StdDev\_IMDb

from MergedGenresView

group by Year

order by StdDev\_IMDb desc;

-- 22 Genre affinity mapping

-- Identify genres that co-occur frequently

WITH GenrePairs as (

select distinct a.Title, TRIM(x.value) as Genre1,TRIM(y.value) as Genre2

from MergedGenresView a

cross apply STRING\_SPLIT(CombinedGenres, ',') x

cross apply STRING\_SPLIT(CombinedGenres, ',') y

where TRIM(x.value) < TRIM(y.value)

)

select top 5 Genre1, Genre2, COUNT(\*) as PairCount

from GenrePairs

group by Genre1, Genre2

having COUNT(\*) > 5

order by PairCount desc;

-- 23 Cohort analysis: average IMDb by release decade

-- Helps see how ratings differ across eras

select (Year / 10) \* 10 as Decade,COUNT(\*) as NumTitles,AVG(IMDb) as Avg\_IMDb

from MergedGenresView

group by (Year / 10) \* 10

order by Decade;

--24 genres where Netflix has higher average IMDb than Amazon Prime Video

WITH GenrePlatformAvg as (

select LTRIM(RTRIM(g.value)) as Genre,m.Netflix,m.Amazon\_Prime\_Video,m.IMDb

from MergedGenresView m

cross apply STRING\_SPLIT(m.CombinedGenres, ',') g

where m.IMDb is not null

),

AvgIMDbByPlatform as (

select Genre,'Netflix' as Platform,AVG(IMDb) as AvgIMDb

from GenrePlatformAvg

where Netflix = 1

group by Genre

union all

select Genre, 'Amazon Prime Video' as Platform,AVG(IMDb) as AvgIMDb

from GenrePlatformAvg

where Amazon\_Prime\_Video = 1

group by Genre

),

AvgIMDbComparison as (

select n.Genre,n.AvgIMDb as NetflixAvgIMDb,p.AvgIMDb as PrimeAvgIMDb,n.AvgIMDb - p.AvgIMDb as IMDbDifference

from

(select Genre, AvgIMDb from AvgIMDbByPlatform where Platform = 'Netflix') n

inner join

(select Genre, AvgIMDb from AvgIMDbByPlatform where Platform = 'Amazon Prime Video') p

ON n.Genre = p.Genre

)

select \* from AvgIMDbComparison

where IMDbDifference > 0

order by IMDbDifference desc;

--25 corelation betweeen IMDB and Rotten tomatoes

WITH stats AS (

SELECT

COUNT(\*) AS n,

SUM(CAST(IMDb AS FLOAT)) AS sum\_x,

SUM(CAST(Rotten\_Tomatoes AS FLOAT)) AS sum\_y,

SUM(CAST(IMDb AS FLOAT) \* CAST(Rotten\_Tomatoes AS FLOAT)) AS sum\_xy,

SUM(CAST(IMDb AS FLOAT) \* CAST(IMDb AS FLOAT)) AS sum\_xx,

SUM(CAST(Rotten\_Tomatoes AS FLOAT) \* CAST(Rotten\_Tomatoes AS FLOAT)) AS sum\_yy

FROM MergedGenresview

WHERE IMDb IS NOT NULL AND Rotten\_Tomatoes IS NOT NULL

)

SELECT

CASE

WHEN (n \* sum\_xx - sum\_x \* sum\_x) = 0 OR (n \* sum\_yy - sum\_y \* sum\_y) = 0 THEN NULL

ELSE

(n \* sum\_xy - sum\_x \* sum\_y)

/

SQRT(

(n \* sum\_xx - sum\_x \* sum\_x) \*

(n \* sum\_yy - sum\_y \* sum\_y)

)

END AS Correlation

FROM stats;