SQL Solutions for Analyzing Netflix Originals Data

A Data Analysis Project using MySQL

By : Tejaswini Srikireddy

Contact details : Linkedin , srikireddytejuteju@gmail.com

CSV files : netflix_originals , Genre_details

Objective

The objective of this project is to analyze the Netflix Originals dataset using SQL queries to extract meaningful insights and trends. The analysis leverages various SQL operations including GROUP BY, HAVING, ORDER BY, LIMIT, JOINS, WINDOW FUNCTIONS, and SUBQUERIES to develop insights into content production trends, genre performance, language diversity, and rating patterns.

Dataset Description

The project utilizes two datasets from the 'netflix_analysis' database:

1. Netflix_Originals: Contains information on each Netflix Original title, including Title, GenreID, Runtime, IMDb Score, Language, and Premiere Date.

2. Genre_Details: Contains GenreID and corresponding Genre name.

1. Average runtime of Netflix Originals for each genre

SQL Query:

```
SELECT gd.Genre, AVG(no.Runtime) AS avg_runtime
FROM netflix_analysis.Netflix_Originals no
JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
GROUP BY gd.Genre;
```

Explanation: This query joins Netflix_Originals and Genre_Details to group data by genre and calculates the average runtime per genre.

Approach: Understanding average runtime helps identify content length preferences in each genre.

Conclusion: The analysis reveals runtime expectations across genres, aiding in content planning.

Resultant output : 1stoutput

2. Languages with more than 10 Netflix Originals and their average IMDb score

SQL Query:

```
SELECT Language, COUNT(*) AS total_titles, AVG(IMDBScore) AS
avg_imdb_score
FROM netflix_analysis.Netflix_Originals
GROUP BY Language
HAVING COUNT(*) > 10;
```

Explanation: We group data by language and filter those with more than 10 entries to get meaningful averages.

Approach: This helps identify languages with a substantial number of releases and their viewer reception.

Conclusion: The result highlights which non-English languages have stronger representation and quality.

Resultant output : 2ndoutput

3. Top 5 genres with the most Netflix Originals

SQL Query:

```
SELECT gd.Genre, COUNT(*) AS total_titles
FROM netflix_analysis.Netflix_Originals no
JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
GROUP BY gd.Genre
ORDER BY total_titles DESC
LIMIT 5;
```

Explanation: Genre data is grouped and counted, then sorted in descending order to identify the top 5.

Approach: This highlights Netflix's focus genres.

Conclusion: The outcome shows the most favored content types by production volume.

Resultant output: 3rdoutput

4. Titles released after 2020 with IMDb score greater than 8.0

SQL Query:

```
SELECT Title, Premiere_Date, IMDBScore
FROM netflix_analysis.Netflix_Originals
WHERE YEAR(Premiere_Date) > 2020 AND IMDBScore > 8.0;
```

Explanation: This query filters recent, high-quality titles based on year and score.

Approach: It helps identify standout content in recent years.

Conclusion: These insights can help recommend quality, recent content to users.

Resultant output: 4thoutput

5. Genre with the most titles in languages other than English

SQL Query:

```
SELECT gd.Genre, COUNT(*) AS total_titles
FROM netflix_analysis.Netflix_Originals no
JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
WHERE Language != 'English'
GROUP BY gd.Genre
ORDER BY total_titles DESC
LIMIT 1;
```

Explanation: This filters non-English titles and identifies the genre with the highest count.

Approach: It highlights genre diversity in international content.

Conclusion: Results showcase Netflix's strongest genre in non-English markets.

Resultant output: 5thoutput

6. Latest title per genre (ranked by premiere date within each genre)

SQL Query:

```
WITH Ranked AS (
  SELECT no.Title, gd.Genre, no.Premiere_Date,
         RANK() OVER (PARTITION BY gd.Genre ORDER BY no.Premiere_Date
DESC) AS rk
  FROM netflix_analysis.Netflix_Originals no
  JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
)
SELECT Title, Genre, Premiere_Date
FROM Ranked
WHERE rk = 1;
```

Explanation: Window functions assign ranks by date within each genre, and the latest is selected.

Approach: This approach identifies the most recent release in each genre.

Conclusion: It gives insight into the freshest content Netflix has released across genres.

Resultant output: 6thoutput

7. Titles longer than the average runtime of their genre

SQL Query:

```
WITH GenreAvg AS (
  SELECT GenreID, AVG(Runtime) AS avg_runtime
  FROM netflix_analysis.Netflix_Originals
  GROUP BY GenreID
)
SELECT no.Title, gd.Genre, no.Runtime, ga.avg_runtime
FROM netflix_analysis.Netflix_Originals no
JOIN GenreAvg ga ON no.GenreID = ga.GenreID
JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
WHERE no.Runtime > ga.avg_runtime;
```

Explanation: We calculate average runtime per genre and filter titles exceeding that value.

Approach: This identifies outliers or detailed productions in each genre.

Conclusion: It can suggest deeper or more elaborate stories that stand out from the average.

Resultant output: 7thoutput

8. Number of Netflix Originals per year and average IMDb score

SQL Query:

```
SELECT YEAR(Premiere_Date) AS release_year,
       COUNT(*) AS total_titles,
       AVG(IMDBScore) AS avg_imdb_score
FROM netflix_analysis.Netflix_Originals
GROUP BY YEAR(Premiere_Date)
ORDER BY release_year;
```

Explanation: We extract year from premiere dates, group by year, and compute count and average score.

Approach: This shows trends in production volume and viewer reception over time.

Conclusion: We see Netflix's growth over years and changes in content quality.

Resultant output: 8thoutput

9. Genres with titles in the top 10% IMDb scores

SQL Query:

```
WITH RankedScores AS (
  SELECT
    no.IMDBScore,
    gd.Genre,
    PERCENT_RANK() OVER (ORDER BY no.IMDBScore) AS pr
  FROM netflix_analysis.Netflix_Originals no
  JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
)
```

```
SELECT DISTINCT Genre
FROM RankedScores
WHERE pr >= 0.9;
```

Explanation: A window function assigns percentile ranks, and genres with top 10% scores are selected.

Approach: This reveals genres that contribute to critically acclaimed content.

Conclusion: Genres identified here show high-quality, top-performing Netflix Originals.

Resultant output: 9thoutput

10. Genres where the minimum IMDb score is greater than 6.5

SQL Query:

```
SELECT gd.Genre, MIN(no.IMDBScore) AS min_imdb_score
FROM netflix_analysis.Netflix_Originals no
JOIN netflix_analysis.Genre_Details gd ON no.GenreID = gd.GenreID
GROUP BY gd.Genre
HAVING MIN(no.IMDBScore) > 6.5;
```

Explanation: Genres are grouped and the minimum IMDb score is calculated, filtering those above 6.5.

Approach: It ensures a baseline of quality across all titles within a genre.

Conclusion: These genres consistently perform well in viewer ratings.

Resultant output: 10thoutput