The background of the slide is a dark, high-contrast image of a person's silhouette writing on a glass wall. The person is on the left, facing right, with their arms raised. The glass wall has a grid pattern. Several bright red circles of varying sizes are scattered across the image: a large one in the top left, a medium one in the top right, a small one in the bottom left, and a large one in the bottom right.


[Home](#)

[About](#)

[Contact](#)

NETFLIX

SQL PROJECT

[Home](#)[About](#)[Contact](#)


OBJECTIVE

NETFLIX



● Understanding the Basics

The objective of this project is to analyze and categorize Netflix content using Microsoft SQL Server by leveraging structured querying techniques. The project aims to extract meaningful insights from the content dataset, such as identifying trends in genre, country-wise production, release year distribution, and categorizing content based on the presence of sensitive keywords (e.g., "kill" and "violence") in the description. A key goal is to implement data filtering, transformation, and aggregation techniques in T-SQL to support content moderation, recommendation strategies, and data-driven decision-making for digital streaming platforms.



[Home](#)[About](#)[Contact](#)

TOOLS AND TECHNIQUES USED

[Read More](#)

● MSSQL Server

Used Common Table Expressions (CTEs) to simplify and structure complex queries.

Applied aggregation functions like COUNT, AVG, MAX, and MIN to summarize content trends.

Implemented window functions such as ROW_NUMBER and RANK to rank content within categories.

Used STRING_SPLIT to separate and analyze multi-genre values from the dataset.

Utilized DATEADD and interval logic to filter content released in the last five years.

Categorized content as 'Good' or 'Bad' using CASE statements based on keyword presence in descriptions.

Grouped and summarized data using GROUP BY to analyze content distribution by year, country, and type.






[Home](#)

[About](#)

[Contact](#)



THE DATA ANALYSIS PROCESS

● **Data Collection**

Used Kaggle Netflix dataset to analyse the content trends in MSSQL.

● **Creation of table**


Created a database and table in MSSQL and then imported data using the Import Wizard of SQL.

● **Data Analysis**

Explored the dataset and then identified the business requirements that can be identified using SQL.

● **Interpretation, and Reporting**

Analysed and interpreted data using SQL queries.



SQL TABLE AND CONTENTS


[Home](#)[About](#)[Contact](#)

This table contains metadata about TV Shows and Movies available on Netflix. It includes attributes such as content type, title, director, cast, release information, and description. The structure is designed to support content filtering, classification, and trend analysis.

Key Columns:

- **show_id**: Unique identifier for each content entry
- **type**: Indicates whether the content is a Movie or TV Show
- **title**: Name of the show or movie
- **director**: Director's name (nullable)
- **cast**: List of main cast members (nullable)
- **country**: Country of origin (nullable)
- **date_added**: Date when the content was added to Netflix
- **release_year**: Year the content was originally released
- **rating**: Age rating classification
- **duration**: Length of the content (e.g., "90 min", "2 Seasons")
- **listed_in**: Genre(s) or category tags
- **description**: Brief synopsis or summary of the content

Column Name	Data Type	Allow Nulls
show_id	varchar(6)	<input checked="" type="checkbox"/>
type	varchar(10)	<input checked="" type="checkbox"/>
title	varchar(150)	<input checked="" type="checkbox"/>
director	varchar(210)	<input checked="" type="checkbox"/>
cast	varchar(1000)	<input checked="" type="checkbox"/>
country	varchar(150)	<input checked="" type="checkbox"/>
date_added	varchar(50)	<input checked="" type="checkbox"/>
release_year	int	<input checked="" type="checkbox"/>
rating	varchar(50)	<input checked="" type="checkbox"/>
duration	varchar(55)	<input checked="" type="checkbox"/>
listed_in	varchar(130)	<input checked="" type="checkbox"/>
description	varchar(250)	<input checked="" type="checkbox"/>

[Home](#)[About](#)[Contact](#)

BUSINESS PROBLEMS


```
# Netflix-Project - MSSQL
```

```
-- Business Problems solved in MSSQL
```

1. Count the number of Movies vs TV Shows
2. Find the most common rating for movies and TV shows
3. List all movies released in a specific year (e.g., 2020)
4. Identify the longest movie
5. Find content added in the last 5 years
6. Find all the movies/TV shows by director 'Rajiv Chilaka'!
7. List all TV shows with more than 5 seasons
8. Count the number of content items in each genre
9. Find each year and the average numbers of content release in India on netflix.
return top 5 year with highest avg content release!
10. List all movies that are documentaries
11. Find all content without a director
12. Find how many movies actor 'Salman Khan' appeared in last 10 years
- 13.

```
Categorize the content based on the presence of the keywords 'kill' and 'violence' in  
the description field. Label content containing these keywords as 'Bad' and all other  
content as 'Good'. Count how many items fall into each category.
```

3This project aims to solve 13 real-world business questions using the Netflix content dataset. By leveraging advanced SQL features like CTEs, window functions, aggregation, and string manipulation in MS SQL Server, we extracted insights related to content trends, user behavior, regional analysis, and content moderation. These queries help uncover patterns such as most frequent actors, top content-producing countries, keyword-based content categorization, and time-based content trends on Netflix.



[Home](#)[About](#)[Contact](#)

```
-- Q1. Count the number of movies vs tv shows
```


```
select distinct(count(show_id)) as total_count, type
from netflix
group by type;
```

To understand the overall content mix on Netflix, we counted the number of entries categorized as TV Shows and Movies. This helps identify which type of content dominates the platform. By using a simple **GROUP BY** query on the type column, we were able to get a clear breakdown of how many Movies and TV Shows are present in the dataset.

[Read More](#)

```
-- Q2. Find the most common rating for movies and TV shows
WITH counted AS (
  SELECT
    type,
    rating,
    COUNT(*) AS total_count
  FROM netflix
  GROUP BY type, rating
)
SELECT
  type,
  rating
FROM (
  SELECT
    type,
    rating,
    total_count,
    RANK() OVER (PARTITION BY type ORDER BY total_count DESC) AS ranking
  FROM counted
) AS t1
WHERE ranking = 1;
```

This query identifies the most frequently used rating (like TV-MA, PG, etc.) for both Movies and TV Shows. A Common Table Expression (CTE) is used to count the number of times each rating appears for each content type. Then, a window function RANK() is applied to rank the ratings within each type based on their frequency. Finally, the query filters out the top-ranked rating per type, giving us the most common rating separately for Movies and TV Shows.

[Home](#)[About](#)[Contact](#)

```
-- Q3. List all the Movies which were released in 2020
```

```
select *  
from netflix  
where release_year = 2020 and type = 'Movie';
```


Filtered the dataset by release_year and type to retrieve all movie titles released in the selected year.

```
-- Q4. Identify the longest movie
```

```
select *  
from netflix  
where type = 'Movie'  
AND duration = (SELECT MAX(duration) from netflix)
```

Filtered by type = 'Movie' and used ordering on the duration field to find the longest movie.

[Read More](#)

[Home](#)[About](#)[Contact](#)

```
-- Q5. Find content added in the last 5 years

SELECT *
FROM netflix
WHERE date_added >= DATEADD(YEAR, -5, GETDATE());
```

Used the date_added field along with DATEADD() and GETDATE() to filter content added in the most recent 5-year span.

```
--Q6. Find all the movies/TV shows by director 'Rajiv Chilaka'!

SELECT DISTINCT(title) as name, *
FROM netflix
where director = 'Rajiv Chilaka';
```

Queried content where the director column contains 'Rajiv Chilaka', allowing analysis of this director's contribution to Netflix.

[Read More](#)

[Home](#)[About](#)[Contact](#)

```
--Q7.List all TV shows with more than 5 seasons

SELECT DISTINCT(title) as tv_show_name , duration
from netflix
where type = 'TV Show'
      AND duration > '5 seasons' ;|
```

Extracted the number of seasons from the duration field for TV Shows and filtered out entries where the number exceeds 5.

```
-- Q8.Count the number of content items in each genre


WITH SplitGenres AS (
  SELECT
    value AS genre
  FROM netflix
  CROSS APPLY STRING_SPLIT(listed_in, ',')
),
TrimmedGenres AS (
  SELECT LTRIM(RTRIM(genre)) AS genre
  FROM SplitGenres
)
SELECT
  genre,
  COUNT(*) AS content_count
FROM TrimmedGenres
GROUP BY genre
ORDER BY content_count DESC;
```

To determine how frequently each genre appears on Netflix, we used the listed_in column, which often contains multiple genres per title (e.g., "Dramas, International Movies"). A Common Table Expression (CTE) was used in combination with the STRING_SPLIT function to break these multi-genre values into individual entries. After splitting, we grouped by genre and used aggregation (COUNT) to compute the total number of content items per genre. This allowed for an accurate representation of genre distribution across the platform.

[Home](#)[About](#)[Contact](#)

```
--Q9.Find each year and the average numbers of content release in India on netflix. Return top 5 year with highest avg content release!  
  
WITH IndianContent AS (  
    SELECT  
        release_year  
    FROM netflix  
    WHERE country LIKE '%India%' -- filters rows where India is mentioned  
) ,  
ContentPerYear AS (  
    SELECT  
        release_year,  
        COUNT(*) AS content_count  
    FROM IndianContent  
    GROUP BY release_year  
) ,  
AvgContentPerYear AS (  
    SELECT  
        release_year,  
        ROUND(AVG(content_count) OVER (), 2) AS avg_content,  
        content_count  
    FROM ContentPerYear  
)  
SELECT TOP 5  
    release_year,  
    avg_content,  
    content_count  
FROM AvgContentPerYear  
ORDER BY content_count DESC;
```

Using CTEs first filtered content by country = 'India', grouped by release_year, and calculated the yearly average count using aggregation.

[Home](#)[About](#)[Contact](#)


```
-- Q10. List all movies that are documentaries
```

```
SELECT DISTINCT(title)
from netflix
where type = 'Movie'
      AND listed_in like '%Documentaries%' ;
```

Filtered by type = 'Movie' and where listed_in contains the word 'Documentary' to identify non-fictional content.

```
-- Q11. Find all content without a director
```

```
SELECT *
from netflix
where director is null;
```

Retrieved entries where the director column is NULL, indicating missing metadata or collaborative productions.


[Home](#)[About](#)[Contact](#)

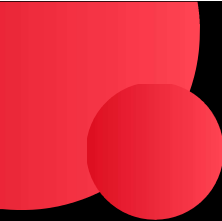
```
-- Q12. Find how many movies actor 'Salman Khan' appeared in last 10 years!  
  
select show_id, title, cast, date_added  
from netflix  
where type = 'Movie'  
      AND cast like '% Salman Khan%'  
      AND date_added >= DATEADD(YEAR, -10, GETDATE()) ;
```

Filtered movies from the last 10 years and used pattern matching in the cast column to check for actor presence.

```
-- Q13. Categorize the content based on the presence of the keywords 'kill' and 'violence' in the description field. Label content containing these keywords as 'Bad' and all other  
--content as 'Good'. Count how many items fall into each category.  
  
SELECT  
  CASE  
    WHEN LOWER(description) LIKE '%kill%' OR LOWER(description) LIKE '%violence%' THEN 'Bad'  
    ELSE 'Good'  
  END AS category,  
  COUNT(*) AS total_items  
FROM netflix  
GROUP BY  
  CASE  
    WHEN LOWER(description) LIKE '%kill%' OR LOWER(description) LIKE '%violence%' THEN 'Bad'  
    ELSE 'Good'  
  END;
```


Used CASE, LOWER(), and LIKE to label content containing keywords like 'kill' or 'violence' as 'Bad', and others as 'Good'.
Counted total items in each category.

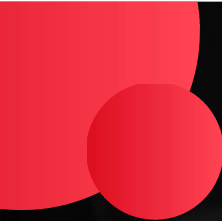


[Home](#)[About](#)[Contact](#)

CONCLUSION

This project provided valuable insights into the structure and trends of Netflix content using Microsoft SQL Server. By applying advanced SQL techniques such as Common Table Expressions (CTEs), aggregation functions, window functions, string manipulation, and date filtering, we successfully addressed multiple real-world business questions. The analysis helped uncover patterns in content types, genres, country-wise production, and release trends, while also demonstrating methods for content classification based on description data. These insights can support strategic decisions in content recommendation, regional targeting, and platform optimization. Overall, the project showcases the power of SQL in transforming raw data into meaningful, actionable insights.





[Home](#)

[About](#)

[Contact](#)

[See You Next](#)

THANK YOU

