

# Netflix Content Analysis Using SQL

This presentation analyzes Netflix's content dataset using SQL to uncover trends in content production, genre dominance, country contribution, rating distribution, and year-over-year growth patterns.



# Project Objective

The objective of this project is to analyze Netflix content data using SQL to uncover trends in content production, genre distribution, country contribution, ratings patterns, and growth over time. The goal is to derive meaningful business insights that can support strategic decision-making and content planning.





# Problem Statement:

With the rapid expansion of streaming platforms, understanding content distribution, genre trends, and production patterns has become crucial for strategic decision-making. However, raw content data alone does not provide meaningful insights. This project aims to analyze Netflix's content dataset using SQL to identify trends in content type distribution, genre dominance, country-wise production, rating patterns, and year-over-year growth in order to uncover actionable insights about Netflix's content strategy.

# SQL Queries Overview

## Key insights from streaming data

This section highlights the SQL queries used to analyze Netflix's content catalog, uncovering trends in genre distribution, country-wise production, rating patterns, content growth, and release trends over time.

### Most Dominant Genres

Identifies the most frequently occurring genres overall and year-wise to understand content preference trends and genre dominance patterns

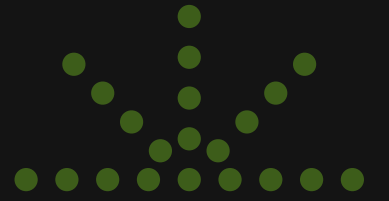
### Content Growth Trends

Analyzes year-over-year growth in content addition to detect expansion phases and potential content gaps.

### Country & Director Analysis

Examines country-wise production volume and directors contributing significantly to Netflix's content catalog.

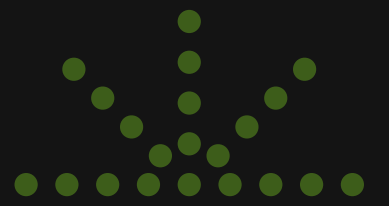
# 1. Retrieve all records where the content type is Movie.



```
SELECT
    *
FROM
    netflix_project
WHERE
    type = 'Movie';
```

| show_id | type  | title                            | director                       | cast   | country                 |
|---------|-------|----------------------------------|--------------------------------|--|-------------------------|
| s1      | Movie | Dick Johnson Is Dead             | Kirsten Johnson                |  | United States           |
| s7      | Movie | My Little Pony: A New Generation | Robert Cullen, JosÃ© Luis Ucha | Vanessa Hudgens, Kimiko Glenn, James Marsde...     |                         |
| s8      | Movie | Sankofa                          | Haile Gerima                   | Kofi Ghanaba, Oyafunmike Ogunlano, Alexandr...     | United States, Ghana, B |
| s10     | Movie | The Starling                     | Theodore Melfi                 | Melissa McCarthy, Chris O'Dowd, Kevin Kline, Ti... | United States           |

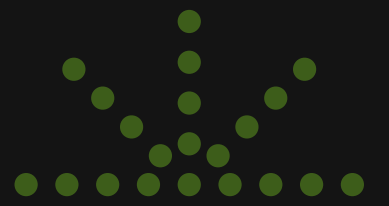
## 2. Count the total number of Movies and TV Shows available.



```
SELECT
    type,
    COUNT(*) AS total_count
FROM
    netflix_project
WHERE
    type IN ('Movie', 'TV Show')
GROUP BY
    type;
```

|   | type    | total_count |
|---|---------|-------------|
| ▶ | Movie   | 55          |
|   | TV Show | 45          |

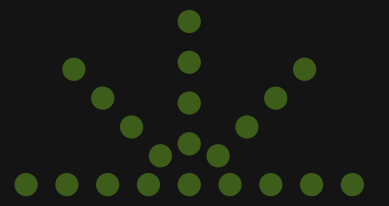
### 3. List all unique countries where Netflix content is produced.



```
SELECT DISTINCT
  title,
  country
FROM
  netflix_project
WHERE
  country IS NOT NULL
ORDER BY
  title;
```

|   | title                       | country |
|---|-----------------------------|---------|
| ▶ | A StoryBots Space Adventure |         |
|   | Angry Birds                 | Finland |
|   | Ankahi Kahaniya             |         |
|   | Avvai Shanmughi             |         |
|   | Bangkok Breaking            |         |

# 4. Find the oldest movie available on Netflix.



```
SELECT
    title, release_year
FROM
    netflix_project
WHERE
    type = 'Movie'
ORDER BY release_year ASC
LIMIT 1;
```

|   | title | release_year |
|---|-------|--------------|
| ▶ | Jaws  | 1975         |

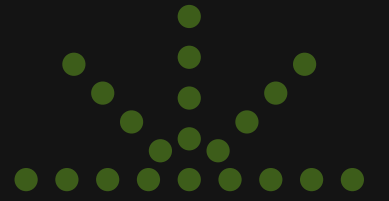
# 5. Display all titles released after 2018.



```
SELECT
    title, release_year
FROM
    netflix_project
WHERE
    release_year > 2018
ORDER BY release_year;
```

|   | title                   | release_year |
|---|-------------------------|--------------|
| ▶ | Paradise Hills          | 2019         |
|   | Titipo Titipo           | 2019         |
|   | Tobot Galaxy Detectives | 2019         |
|   | Dick Johnson Is Dead    | 2020         |
|   | Omo Ghetto: the Saga    | 2020         |

# 6. Identify content records where director information is missing.



```
SELECT
    *
FROM
    netflix_project
WHERE
    director IS NULL OR director = '';
```

|   | show_id | type    | title                               | director | cast   | country      | date_added         | release_year | rating | duration |
|---|---------|---------|-------------------------------------|----------|--|--------------|--------------------|--------------|--------|----------|
| ▶ | s2      | TV Show | Blood & Water                       |          | Ama Qamata, Khosi Ngema, Gail Mablane, Tha...    | South Africa | September 24, 2021 | 2021         | TV-MA  | 2 Season |
|   | s4      | TV Show | Jailbirds New Orleans               |          |  |              | September 24, 2021 | 2021         | TV-MA  | 1 Season |
|   | s5      | TV Show | Kota Factory                        |          | Mayur More, Jitendra Kumar, Ranjan Raj, Alam ... | India        | September 24, 2021 | 2021         | TV-MA  | 2 Season |
|   | s11     | TV Show | Vendetta: Truth, Lies and The Mafia |          |  |              | September 24, 2021 | 2021         | TV-MA  | 1 Season |

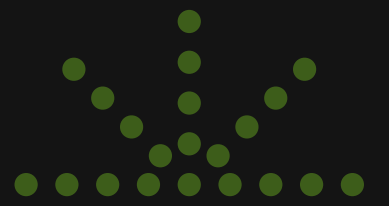
# 7. Count the number of titles added to Netflix each year.



```
SELECT
    RIGHT(date_added, 4) AS year_added,
    COUNT(*) AS total_titles_added
FROM
    netflix_project
WHERE
    date_added IS NOT NULL
GROUP BY
    RIGHT(date_added, 4)
ORDER BY
    year_added ASC;
```

|   | year_added | total_titles_added |
|---|------------|--------------------|
| ▶ | 2021       | 100                |

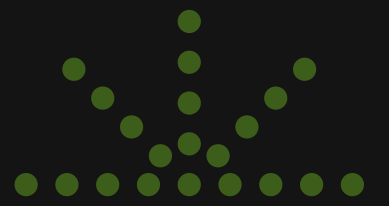
## 8. List titles with a rating of 'TV-MA'.



```
SELECT
    title, rating
FROM
    netflix_project
WHERE
    rating = 'TV-MA';
```

|   | title                 | rating |
|---|-----------------------|--------|
| ▶ | Blood & Water         | TV-MA  |
|   | Ganglands             | TV-MA  |
|   | Jailbirds New Orleans | TV-MA  |
|   | Kota Factory          | TV-MA  |
|   | Midnight Mass         | TV-MA  |

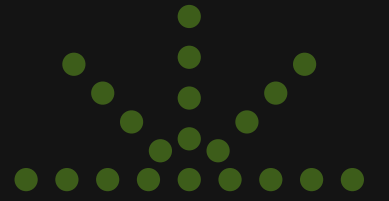
# 9. Find the top 5 countries with the highest number of Netflix titles.



```
SELECT
    country,
    COUNT(*) AS total_titles
FROM
    netflix_project
WHERE
    country IS NOT NULL
GROUP BY
    country
ORDER BY
    total_titles DESC
LIMIT 5;
```

|   | country        | total_titles |
|---|----------------|--------------|
| ▶ |                | 41           |
|   | United States  | 17           |
|   | Japan          | 13           |
|   | India          | 6            |
|   | United Kingdom | 5            |

# 10. Determine the most common rating for Movies and TV Shows separately



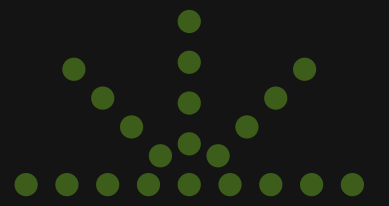
```
SELECT
    rating, COUNT(*) AS total_count
FROM
    netflix_project
WHERE
    type = 'Movie' AND rating IS NOT NULL
GROUP BY rating
ORDER BY total_count DESC
LIMIT 1;
```

```
SELECT
    rating, COUNT(*) AS total_count
FROM
    netflix_project
WHERE
    type = 'TV Show' AND rating IS NOT NULL
GROUP BY rating
ORDER BY total_count DESC
LIMIT 1;
```

|   | rating | total_count |
|---|--------|-------------|
| ▶ | TV-PG  | 12          |

|   | rating | total_count |
|---|--------|-------------|
| ▶ | TV-MA  | 22          |

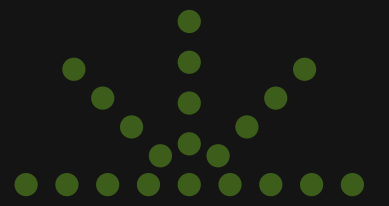
# 11. Calculate the average duration of movies (in minutes).



```
SELECT
    ROUND(AVG(SUBSTRING_INDEX(duration, ' ', 1)), 2) AS average_movie_duration
FROM
    netflix_project
WHERE
    type = 'Movie';
```

|   | average_movie_duration |
|---|------------------------|
| ▶ | 100.73                 |

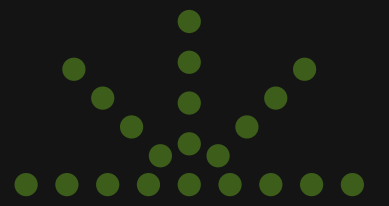
# 12. Find the most popular genre on Netflix.



```
SELECT
    listed_in,
    COUNT(*) AS total_titles
FROM
    netflix_project
WHERE
    listed_in IS NOT NULL
GROUP BY
    listed_in
ORDER BY
    total_titles DESC
LIMIT 1;
```

|   | listed_in  | total_titles |
|---|--|--------------|
| ▶ | Action & Adventure, Anime Features, Internati... | 12           |

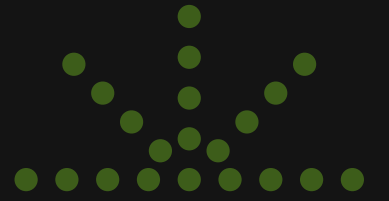
# 12. Find the most popular genre on Netflix.



```
SELECT
    listed_in,
    COUNT(*) AS total_titles
FROM
    netflix_project
WHERE
    listed_in IS NOT NULL
GROUP BY
    listed_in
ORDER BY
    total_titles DESC
LIMIT 1;
```

|   | listed_in  | total_titles |
|---|--|--------------|
| ▶ | Action & Adventure, Anime Features, Internati... | 12           |

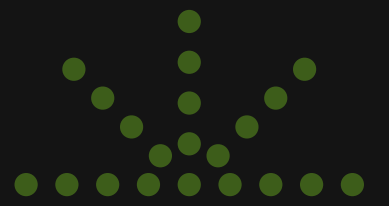
# 13. Identify which year had the highest number of releases.



```
SELECT
    release_year, COUNT(*) AS total_release
FROM
    netflix_project
GROUP BY release_year
ORDER BY release_year DESC
LIMIT 1;
```

|   | release_year | total_release |
|---|--------------|---------------|
| ▶ | 2021         | 53            |

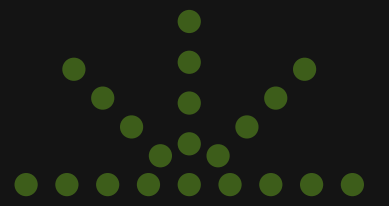
# 14. List directors who have directed more than 5 titles.



```
SELECT
    director, COUNT(*) AS total_titles
FROM
    netflix_project
WHERE
    director IS NOT NULL
GROUP BY director
HAVING COUNT(*) > 5
ORDER BY total_titles DESC;
```

|   | director | total_titles |
|---|----------|--------------|
| ▶ |          | 39           |

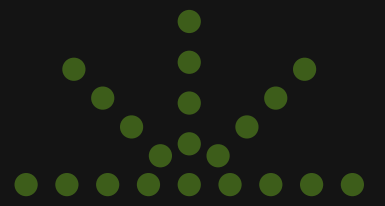
# 15. Find content that belongs to multiple genres.



```
SELECT
    title,
    listed_in
FROM
    netflix_project
WHERE
    listed_in LIKE '%,%';
```

|   | title                 | listed_in  |
|---|-----------------------|--|
| ▶ | Blood & Water         | International TV Shows, TV Dramas, TV Mysteries  |
|   | Ganglands             | Crime TV Shows, International TV Shows, TV Ac... |
|   | Jailbirds New Orleans | Docuseries, Reality TV                           |
|   | Kota Factory          | International TV Shows, Romantic TV Shows, TV... |
|   | Midnight Mass         | TV Dramas, TV Horror, TV Mysteries               |

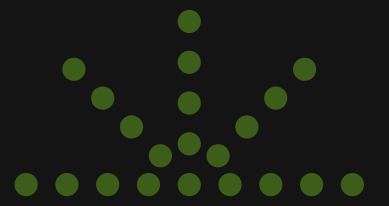
# 16. Compare the number of Movies vs TV Shows added year-wise.



```
SELECT
    RIGHT(date_added, 4) AS year_added,
    type,
    COUNT(*) AS total_added
FROM
    netflix_project
WHERE
    date_added IS NOT NULL
GROUP BY
    RIGHT(date_added, 4),
    type
ORDER BY
    year_added ASC, type;
```

|   | year_added | type    | total_added |
|---|------------|---------|-------------|
| ▶ | 2021       | Movie   | 55          |
|   | 2021       | TV Show | 45          |

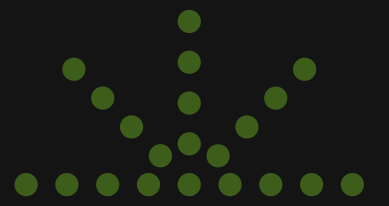
# 17. Identify countries producing only Movies and not TV Shows.



```
SELECT
    country
FROM
    netflix_project
WHERE
    country IS NOT NULL
GROUP BY
    country
HAVING
    SUM(CASE WHEN type = 'Movie' THEN 1 ELSE 0 END) > 0
    AND
    SUM(CASE WHEN type = 'TV Show' THEN 1 ELSE 0 END) = 0;
```

|   | country   |
|---|---|
| ▶ | United States, Ghana, Burkina Faso, United Kin... |
|   | Germany, Czech Republic                           |
|   | United States, India, France                      |
|   | China, Canada, United States                      |
|   | South Africa, United States, Japan                |

# 18. Rank countries based on the number of titles produced.

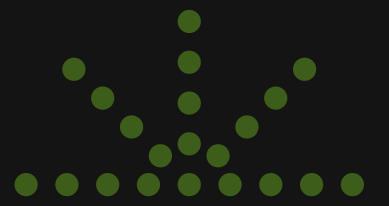


```
SELECT
    country,
    COUNT(*) AS total_titles,
    RANK() OVER (ORDER BY COUNT(*) DESC) AS country_rank
FROM
    netflix_project
WHERE
    country IS NOT NULL
GROUP BY
    country
ORDER BY
    country_rank;
```

|   | country        | total_titles | country_rank |
|---|----------------|--------------|--------------|
| ▶ |                | 41           | 1            |
|   | United States  | 17           | 2            |
|   | Japan          | 13           | 3            |
|   | India          | 6            | 4            |
|   | United Kingdom | 5            | 5            |

Result 55

# 19. Identify year-over-year growth in content addition.



```
WITH yearly_data AS (  
  SELECT  
    RIGHT(date_added, 4) AS year_added,  
    COUNT(*) AS total_added  
  FROM  
    netflix_project  
  WHERE  
    date_added IS NOT NULL  
  GROUP BY  
    RIGHT(date_added, 4)  
)
```

```
SELECT  
  year_added,  
  total_added,  
  total_added - LAG(total_added) OVER (ORDER BY year_added) AS yoy_growth  
FROM  
  yearly_data  
ORDER BY  
  year_added;
```

|   | year_added | total_added | yoy_growth |
|---|------------|-------------|------------|
| ▶ | 2021       | 100         | NULL       |

# 20. Find the top 3 genres per country.



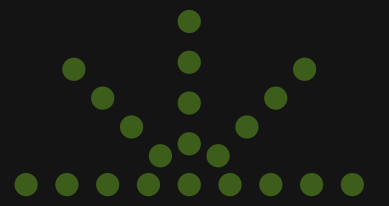
```
WITH genre_split AS (  
  SELECT  
    country,  
    TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(listed_in, ',', numbers.n), ',', -1)) AS genre  
  FROM  
    netflix_project  
  JOIN  
    (SELECT 1 n UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5) numbers  
  ON CHAR_LENGTH(listed_in) - CHAR_LENGTH(REPLACE(listed_in, ',', '')) >= numbers.n - 1  
  WHERE  
    country IS NOT NULL  
)
```

```
genre_count AS (  
  SELECT  
    country,  
    genre,  
    COUNT(*) AS total_titles  
  FROM  
    genre_split  
  GROUP BY  
    country, genre  
)
```

```
FROM (  
  SELECT  
    country,  
    genre,  
    total_titles,  
    RANK() OVER (PARTITION BY country ORDER BY total_titles DESC) AS rank_num  
  FROM  
    genre_count  
  ) ranked  
WHERE  
  rank_num <= 3  
ORDER BY  
  country, rank_num;
```

|   | country   | genre                    | total_titles |
|---|-----------|--------------------------|--------------|
| ▶ |           | International Movies     | 11           |
|   |           | International TV Shows   | 8            |
|   |           | Children & Family Movies | 8            |
|   | Australia | Reality TV               | 1            |
|   | Australia | International TV Shows   | 1            |

# 21. Calculate the percentage contribution of each content type (Movie vs TV Show).



```
SELECT
  type,
  COUNT(*) AS total_count,
  ROUND(
    COUNT(*) * 100.0 / (SELECT COUNT(*) FROM netflix_project),
    2
  ) AS percentage_contribution
FROM
  netflix_project
GROUP BY
  type;
```

|   | type    | total_count | percentage_contribution |
|---|---------|-------------|-------------------------|
| ▶ | Movie   | 55          | 55.00                   |
|   | TV Show | 45          | 45.00                   |

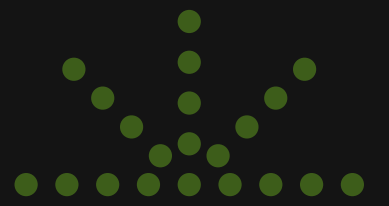
## 22. Use window functions to rank years by content release count.



```
SELECT
  release_year,
  COUNT(*) AS total_releases,
  RANK() OVER (ORDER BY COUNT(*) DESC) AS year_rank
FROM
  netflix_project
GROUP BY
  release_year
ORDER BY
  year_rank;
```

|   | release_year | total_releases | year_rank |
|---|--------------|----------------|-----------|
| ▶ | 2021         | 53             | 1         |
|   | 2020         | 8              | 2         |
|   | 2018         | 4              | 3         |
|   | 2013         | 3              | 4         |
|   | 2019         | 3              | 4         |

## 23. Identify directors whose content consistently receives higher maturity ratings.



```
SELECT
  director,
  COUNT(*) AS total_titles
FROM
  netflix_project
WHERE
  director IS NOT NULL
GROUP BY
  director
```

```
HAVING
  SUM(CASE
    WHEN rating IN ('TV-MA', 'R', 'NC-17')
    THEN 1
    ELSE 0
  END) = COUNT(*);
```

|   | director            | total_titles |
|---|---------------------|--------------|
| ► | Julien Ledercq      | 1            |
|   | Mike Flanagan       | 1            |
|   | Haile Gerima        | 1            |
|   | Kongkiat Komesiri   | 1            |
|   | Christian Schwochow | 1            |

## 24. Detect content gaps (years with unusually low releases).

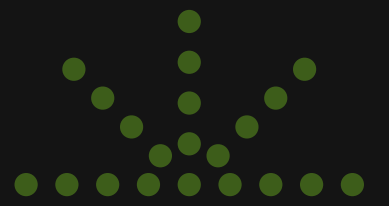


```
WITH yearly_counts AS (  
  SELECT  
    release_year,  
    COUNT(*) AS total_releases  
  FROM  
    netflix_project  
  GROUP BY  
    release_year  
)
```

```
SELECT  
  release_year,  
  total_releases  
FROM  
  yearly_counts  
WHERE  
  total_releases < (  
    SELECT AVG(total_releases)  
    FROM yearly_counts  
  )  
ORDER BY  
  total_releases ASC;
```

|   | release_year | total_releases |
|---|--------------|----------------|
| ▶ | 1993         | 1              |
|   | 1996         | 1              |
|   | 1998         | 1              |
|   | 1997         | 1              |
|   | 1975         | 1              |

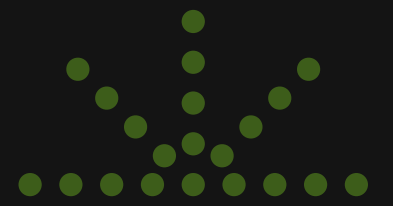
# 25. Analyze the shift in content type trend over the last decade.



```
SELECT
    release_year,
    SUM(CASE WHEN type = 'Movie' THEN 1 ELSE 0 END) AS total_movies,
    SUM(CASE WHEN type = 'TV Show' THEN 1 ELSE 0 END) AS total_tv_shows
FROM
    netflix_project
WHERE
    release_year >= YEAR(CURDATE()) - 10
GROUP BY
    release_year
ORDER BY
    release_year ASC;
```

|   | release_year | total_movies | total_tv_shows |
|---|--------------|--------------|----------------|
| ▶ | 2017         | 1            | 1              |
|   | 2018         | 2            | 2              |
|   | 2019         | 1            | 2              |
|   | 2020         | 4            | 4              |
|   | 2021         | 22           | 31             |

# 26. Determine the most dominant genre each year.



```
WITH genre_split AS (  
  SELECT  
    release_year,  
    TRIM(SUBSTRING_INDEX(SUBSTRING_INDEX(listed_in, ',', numbers.n), ',', -1)) AS genre  
  FROM  
    netflix_project  
  JOIN  
    (SELECT 1 n UNION SELECT 2 UNION SELECT 3 UNION SELECT 4 UNION SELECT 5) numbers  
  ON CHAR_LENGTH(listed_in) - CHAR_LENGTH(REPLACE(listed_in, ',', '')) >= numbers.n - 1  
)
```

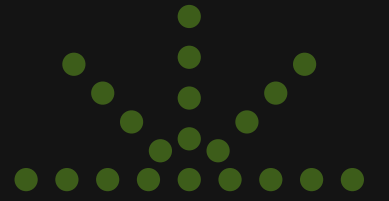
```
genre_count AS (  
  SELECT  
    release_year,  
    genre,  
    COUNT(*) AS total_titles  
  FROM  
    genre_split  
  GROUP BY  
    release_year, genre  
)
```

```
SELECT  
  
  release_year,  
  
  genre,  
  
  total_titles
```

```
FROM (  
  SELECT  
    release_year,  
    genre,  
    total_titles,  
    RANK() OVER (PARTITION BY release_year ORDER BY total_titles DESC) AS rank_num  
  FROM  
    genre_count  
) ranked  
WHERE  
  rank_num = 1  
ORDER BY  
  release_year;
```

|   | release_year | genre              | total_titles |
|---|--------------|--------------------|--------------|
| ▶ | 1975         | Dramas             | 1            |
|   | 1975         | Classic Movies     | 1            |
|   | 1975         | Action & Adventure | 1            |
|   | 1978         | Thrillers          | 1            |
|   | 1978         | Horror Movies      | 1            |

# 27. Find countries where Netflix content growth is declining.



```
WITH yearly_country_data AS (  
  SELECT  
    country,  
    RIGHT(date_added, 4) AS year_added,  
    COUNT(*) AS total_added  
  FROM  
    netflix_project  
  WHERE  
    country IS NOT NULL  
    AND date_added IS NOT NULL  
  GROUP BY  
    country, RIGHT(date_added, 4)  
)
```

```
growth_data AS (  
  SELECT  
    country,  
    year_added,  
    total_added,  
    total_added - LAG(total_added)  
      OVER (PARTITION BY country ORDER BY year_added) AS yoy_growth  
  FROM  
    yearly_country_data  
)
```

```
SELECT  
  country,  
  year_added,  
  total_added,  
  yoy_growth  
FROM  
  growth_data  
WHERE  
  yoy_growth < 0  
ORDER BY  
  country, year_added;
```

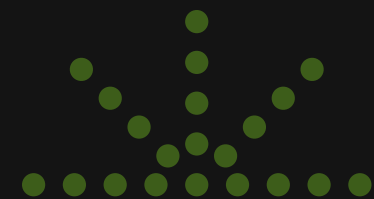
|  | country | year_added | total_added | yoy_growth |
|--|---------|------------|-------------|------------|
|  |         |            |             |            |



# Summary

This project analyzed Netflix's content dataset using SQL to uncover meaningful insights about content distribution, genre dominance, country-wise production, rating patterns, and year-over-year growth trends.

Through data cleaning, aggregation, conditional logic, and window functions, the analysis highlighted strategic shifts in content focus, dominant genres across years, and production trends across countries.



# Thank you