

NETFLIX ELT PROJECT

DATA ANALYSIS

BY
Nandha Kumar B

Tools Used



ABOUT DATA

The Netflix dataset comprises 12 fields and 8,807 records, including both movies and TV shows.

It captures the date added from 2013 to 2021 and the release years of content ranging from 1925 to 2021. Let's dive into the data analysis.



IMPORTING DATA IN PYTHON

```
In [1]: import pandas as pd  
df = pd.read_csv('C:\\\\Users\\\\nandh\\\\Downloads\\\\netflix_raw_dataset.csv')
```

```
In [2]: df.head(5)
```

Out[2]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	2021-09-25	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmmaker... Kirsten Johnson
1	s2	TV Show	Blood & Water	NaN	Ama Qamata, Khosi Ngema, Gail Mabalane, Thaban...	South Africa	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, TV Dramas, TV Mysteries	After crossing paths at a party, a Cape Town t...
2	s3	TV Show	Ganglands	Julien Leclercq	Sami Bouajila, Tracy Gotoas, Samuel Jouy, Nabi...	NaN	2021-09-24	2021	TV-MA	1 Season	Crime TV Shows, International TV Shows, TV Act...	To protect his family from a powerful drug lord...
3	s4	TV Show	Jailbirds New Orleans	NaN	NaN	NaN	2021-09-24	2021	TV-MA	1 Season	Docuseries, Reality TV	Feuds, flirtations and toilet talk go down amo...
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	2021-09-24	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...

CONNECTING DATA TO SQL

```
In [3]: import sqlalchemy as sal
```

```
# Modify the connection string to use pymysql
engine = sal.create_engine('mysql+pymysql://root:123456@127.0.0.1/Netflix_Dataset')
conn = engine.connect()
```

```
In [4]: df.to_sql('netflix', con=conn , index=False, if_exists = 'append')
conn.close()
```

```
In [5]: df[df.show_id=='s5023']
```

Out[5]:

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description
	5022	s5023	Movie	반드시 잡는다	Hong-seon Kim	Baek Yoon-sik	South Korea	2018-02-28	2017	TV- MA	110 min	Dramas, International Movies, Thrillers After people in his town start turning up dead...

LOADING DATA IN TO SQL USING SQLALCHEMY AND CHECKING IF OTHER LANGUAGE DATA ARE IMPORTED SUCCESSFULLY

NETFLIX DATA ANALYSIS

```
SELECT COUNT(*) FROM NETFLIX;
```

	COUNT(*)
▶	8807

**CHECKING IF ANY
DUPLICATES IS
AVAILABLE WITH SAME
TITLE, 7 DUPLICATES
AVAILABLE**

```
SELECT * FROM NETFLIX WHERE UPPER(TITLE) IN
(SELECT
    UPPER(TITLE)
FROM
    NETFLIX
GROUP BY UPPER(TITLE), TYPE
HAVING COUNT(*) > 1
)
ORDER BY TITLE;
```

	SHOW_ID	TYPE	TITLE	DIRECTOR	CAST	COUNTRY	DATE_ADDED
▶	s3963	Movie	15-Aug	Swapnaneel Jayakar	Rahul Pethe, Mrunmayee Deshpande, Adinath ...	India	2019-03-29
	s5967	Movie	15-Aug	Swapnaneel Jayakar	Rahul Pethe, Mrunmayee Deshpande, Adinath ...	India	2019-03-29
	s4523	Movie	22-Jul	Paul Greengrass	Anders Danielsen Lie, Jon Øigarden, Jonas Stra...	Norway, Iceland, United States	2018-10-10
	s5966	Movie	22-Jul	Paul Greengrass	Anders Danielsen Lie, Jon Øigarden, Jonas Stra...	Norway, Iceland, United States	2018-10-10
	s3997	TV Show	9-Feb	HULL	Shahd El Yaseen, Shaila Sabt, Hala, Hanadi Al-K...	NULL	2019-03-20
	s5965	TV Show	9-Feb	HULL	Shahd El Yaseen, Shaila Sabt, Hala, Hanadi Al-K...	NULL	2019-03-20
	s304	Movie	Esperando la carroza	Alejandro Doria	Luis Brandoni, China Zorrilla, Antonio Gasalla, J...	Argentina	2021-08-05

NETFLIX DATA ANALYSIS

```
DELETE FROM NETFLIX
WHERE SHOW_ID IN (
  SELECT SHOW_ID
  FROM (
    SELECT *,  

           ROW_NUMBER() OVER(PARTITION BY TITLE, TYPE ORDER BY SHOW_ID) AS S_NO
    FROM NETFLIX
  ) AS CTE
  WHERE S_NO > 1
);
```

-- RESULT DATASET AFTER DELETING 7 DUPLICATES

```
SELECT COUNT(*) FROM NETFLIX;
```

	COUNT(*)
▶	8800

FOUND 7 DUPLICATE RECORDS WITH SAME TITLE, DATE ADDED, TYPE, DIRECTOR SO REMOVED THOSE DUPLICATES

NETFLIX DATA ANALYSIS

```
-- CREATE SEPARATE TABLE FOR SHOWS TO DO DATA ANALYSIS EFFECTIVELY
```

```
DROP TABLE IF EXISTS NETFLIX_SHOW;
```

```
CREATE TABLE NETFLIX_SHOW AS
SELECT
    SHOW_ID, TYPE, TITLE, DATE_ADDED, RELEASE_YEAR, RATING,
    (CASE
        WHEN DURATION IS NULL THEN RATING
        ELSE DURATION
    END) AS DURATION,
    DESCRIPTION
FROM NETFLIX;
```

**CREATING SEPARATE
NETFLIX_SHOW
TABLE TO DO DATA
ANALYSIS BY USING
JOINS AND CASE
EFFECTIVELY**

	SHOW_ID	TYPE	TITLE	DATE_ADDED	RELEASE_YEAR	RATING	DURATION	DESCRIPTION
▶	s1	Movie	Dick Johnson Is Dead	2021-09-25	2020	PG-13	90 min	As her father nears the end of his life, filmmaker...
	s10	Movie	The Starling	2021-09-24	2021	PG-13	104 min	A woman adjusting to life after a loss contends ...
	s100	TV Show	On the Verge	2021-09-07	2021	TV-MA	1 Season	Four women — a chef, a single mom, an heiress...
	s1000	Movie	Stowaway	2021-04-22	2021	TV-MA	116 min	A three-person crew on a mission to Mars faces...
	s1001	Movie	Wild Dog	2021-04-22	2020	TV-MA	126 min	A brash but brilliant Indian intelligence agent lea...
	s1002	Movie	Olobiri	2021-04-21	2015	TV-14	86 min	After drilling depletes a small village, a corporati...
	s1003	Movie	Tell Me When	2021-04-21	2021	TV-MA	97 min	Workaholic Will puts his humdrum life in LA on h...
	s1004	TV Show	Zero	2021-04-21	2021	TV-MA	1 Season	A shy teen with the power to turn invisible must...

NETFLIX DATA ANALYSIS

```
-- CHECKING DATA TRANSFORM FROM LISTED_IN INTO GENRE AS SEPARATE COLUMNS'S  
  
SELECT  
    SHOW_ID,  
    LISTED_IN,  
    TRIM(SUBSTRING_INDEX(LISTED_IN, ',', 1)) AS GENRE,  
    TRIM(SUBSTRING(LISTED_IN, LOCATE(',', LISTED_IN) + 1)) AS REMAINING_GENRES  
FROM  
    NETFLIX  
LIMIT 10;
```

	SHOW_ID	LISTED_IN	GENRE	REMAINING_GENRES
▶	s1	Documentaries	Documentaries	Documentaries
	s10	Comedies, Dramas	Comedies	Dramas
	s100	TV Comedies, TV Dramas	TV Comedies	TV Dramas
	s1000	Dramas, International Movies, Thrillers	Dramas	International Movies, Thrillers
	s1001	Action & Adventure, International Movies	Action & Adventure	International Movies
	s1002	Dramas, International Movies, Thrillers	Dramas	International Movies, Thrillers
	s1003	Comedies, International Movies	Comedies	International Movies
	s1004	International TV Shows, TV Comedies, TV Dramas	International TV Shows	TV Comedies, TV Dramas

BEFORE CREATING
SEPARATE
NETFLIX_GENRE
TABLE , NEED TO
SEPARATE COMMAS(,)
MULTI DATA IN TO
DIFFERENT ROWS TO
DO GROUPING
& ANALYSIS
EFFECTING

NETFLIX DATA ANALYSIS

```
-- CREATING FUNCTION TO SPLIT STRING

DROP FUNCTION IF EXISTS SPLIT_STRING;

DELIMITER $$

CREATE FUNCTION SPLIT_STRING(str TEXT, delim VARCHAR(12), pos INT) RETURNS VARCHAR(255)
DETERMINISTIC
BEGIN
    DECLARE output TEXT;
    SET output = REPLACE(SUBSTRING(SUBSTRING_INDEX(str, delim, pos),
                                    LENGTH(SUBSTRING_INDEX(str, delim, pos -1)) + 1),
                          delim, '');
    RETURN output;
END$$

DELIMITER ;
```

**CREATING FUNCTION
TO SEPARATE STRING
BASED ON COMMA(,)
IN TO DIFFERENT
ROW TO DO GROUPING
& ANALYSIS
EFFECTIVELY**

NETFLIX DATA ANALYSIS

```
-- Insert split GENRE into the NETFLIX_GENRE table

INSERT INTO NETFLIX_GENRE (SHOW_ID, GENRE)
SELECT
    TRIM(SHOW_ID),
    TRIM(SPLIT_STRING(LISTED_IN, ',', numbers.n)) AS GENRE
FROM
    NETFLIX,
    (SELECT 1 n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL
     SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL
     SELECT 9 UNION ALL SELECT 10 UNION ALL SELECT 11 UNION ALL SELECT 12 UNION ALL
     SELECT 13 UNION ALL SELECT 14 UNION ALL SELECT 15 UNION ALL SELECT 16 UNION ALL
     SELECT 17 UNION ALL SELECT 18 UNION ALL SELECT 19 UNION ALL SELECT 20) numbers
WHERE
    TRIM(SPLIT_STRING(LISTED_IN, ',', numbers.n)) IS NOT NULL AND SPLIT_STRING(LISTED_IN, ',', numbers.n) != '';
```

-- View Netflix GENRE table

```
SELECT * FROM NETFLIX_GENRE;
```

```
-- create Netflix GENRE table from main Netflix table

DROP TABLE IF EXISTS NETFLIX_GENRE;

CREATE TABLE NETFLIX_GENRE (
    SHOW_ID VARCHAR(10),
    GENRE TEXT
);
```

	SHOW_ID	GENRE
▶	s1	Documentaries
	s10	Dramas
	s10	Comedies
	s100	TV Dramas
	s100	TV Comedies
	s1000	Thrillers
	s1000	International Movies

**USING FUNCTION
SPLIT_STRING TO
SEPARATE STRING
BASED ON COMMA(,)
IN TO DIFFERENT
ROW AND CREATE
SEPARATE
NETFLIX_GENRE
TABLE**

NETFLIX DATA ANALYSIS

```
-- Insert split countries into the NETFLIX_COUNTRY table

INSERT INTO NETFLIX_COUNTRY (SHOW_ID, COUNTRY)
SELECT
    TRIM(SHOW_ID),
    TRIM(SPLIT_STRING(COUNTRY, ',', numbers.n)) AS COUNTRY
FROM
    NETFLIX,
    (SELECT 1 n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL
     SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL
     SELECT 9 UNION ALL SELECT 10) numbers
WHERE
    TRIM(SPLIT_STRING(COUNTRY, ',', numbers.n)) IS NOT NULL AND SPLIT_STRING(COUNTRY, ',', numbers.n) != '';
```

```
DROP TABLE IF EXISTS NETFLIX_COUNTRY;
```

```
CREATE TABLE NETFLIX_COUNTRY (
    SHOW_ID VARCHAR(10),
    COUNTRY TEXT
);
```

	SHOW_ID	COUNTRY
▶	s1	United States
	s10	United States
	s100	United States
	s100	France
	s1000	United States
	s1000	Germany
	s1002	United States
	s1002	Nigeria
	s1002	Canada
	s1003	Mexico

**USING FUNCTION
SPLIT_STRING TO
SEPARATE STRING
BASED ON COMMA(),
IN TO DIFFERENT
ROW AND CREATE
SEPARATE
NETFLIX_COUNTRY
TABLE AND SEE ID
\$1000 WAS
SEPARATED IN TWO
ROWS AS UNITED
STATES & GERMANY**

NETFLIX DATA ANALYSIS

```
-- Insert split Director into the NETFLIX_Director table

INSERT INTO NETFLIX_DIRECTOR (SHOW_ID, DIRECTOR)
SELECT
    SHOW_ID,
    SPLIT_STRING(DIRECTOR, ',', numbers.n) AS DIRECTOR
FROM
    NETFLIX,
    (SELECT 1 n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL
     SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL
     SELECT 9 UNION ALL SELECT 10) numbers
WHERE
    SPLIT_STRING(DIRECTOR, ',', numbers.n) IS NOT NULL AND SPLIT_STRING(DIRECTOR, ',', numbers.n) != '';

-- View Netflix Director table

SELECT * FROM NETFLIX_DIRECTOR;
```

```
-- create Netflix Director table from main Netflix table

DROP TABLE IF EXISTS NETFLIX_DIRECTOR;

CREATE TABLE NETFLIX_DIRECTOR (
    SHOW_ID VARCHAR(10),
    DIRECTOR TEXT
);
```

	SHOW_ID	DIRECTOR
▶	s1	Kirsten Johnson
	s10	Theodore Melfi
	s1000	Joe Penna
	s1001	Ahishor Solomon
	s1002	Curtis Graham
	s1003	Gerardo Gatica
	s1007	Suhas Kadav
	s1008	Suhas Kadav
	s1009	Suhas Kadav
	s1010	Sumit Das
	s1010	Akshay Sanjee

**USING FUNCTION
SPLIT_STRING TO
SEPARATE STRING
BASED ON COMMA(,)
IN TO DIFFERENT
ROW AND CREATE
SEPARATE
NETFLIX_DIRECTOR
TABLE AND SEE ID
S1010 WAS
SEPARATED IN TWO
ROWS AS SUMIT DAS
& AKSHAY SANJEE**

NETFLIX DATA ANALYSIS

```
INSERT INTO NETFLIX_CAST (SHOW_ID, CAST)
SELECT
    SHOW_ID,
    SPLIT_STRING(CAST, ',', numbers.n) AS CAST
FROM
    NETFLIX,
    (SELECT 1 n UNION ALL SELECT 2 UNION ALL SELECT 3 UNION ALL SELECT 4 UNION ALL
     SELECT 5 UNION ALL SELECT 6 UNION ALL SELECT 7 UNION ALL SELECT 8 UNION ALL
     SELECT 9 UNION ALL SELECT 10 UNION ALL SELECT 11 UNION ALL SELECT 12 UNION ALL
     SELECT 13 UNION ALL SELECT 14 UNION ALL SELECT 15 UNION ALL SELECT 16 UNION ALL
     SELECT 17 UNION ALL SELECT 18 UNION ALL SELECT 19 UNION ALL SELECT 20) numbers
WHERE
    SPLIT_STRING(CAST, ',', numbers.n) IS NOT NULL AND SPLIT_STRING(CAST, ',', numbers.n) != '';
-- View Netflix CAST table
```

```
-- create Netflix CAST table from main Netflix table

DROP TABLE IF EXISTS NETFLIX_CAST;

CREATE TABLE NETFLIX_CAST (
    SHOW_ID VARCHAR(10),
    CAST TEXT
);
```

	SHOW_ID	CAST
▶	s10	Ravi Kapoor
	s10	Loretta Devine
	s10	Kimberly Quinn
	s10	Rosalind Chao
	s10	Laura Harrier
	s10	Skyler Gisondo
	s10	Daveed Diggs
	s10	Timothy Olyphant
	s10	Kevin Kline

**USING FUNCTION
SPLIT_STRING TO
SEPARATE STRING
BASED ON COMMA(,)
IN TO DIFFERENT
ROW AND CREATE
SEPARATE
NETFLIX_CAST
TABLE**

NETFLIX DATA ANALYSIS

```
-- FILLING COUNTRY TABLE NULL VALUE WITH PREVIOUS DIRECTOR COUNTRY
```

```
INSERT INTO NETFLIX_COUNTRY
SELECT SHOW_ID, M.COUNTRY
FROM NETFLIX AS N
INNER JOIN (
    SELECT DIRECTOR, COUNTRY
    FROM NETFLIX_COUNTRY AS NC
    INNER JOIN NETFLIX_DIRECTOR AS ND ON NC.SHOW_ID = ND.SHOW_ID
    GROUP BY DIRECTOR, COUNTRY
) M ON N.DIRECTOR = M.DIRECTOR
WHERE N.COUNTRY IS NULL;
```

```
INSERT INTO NETFLIX_COUNTRY SELECT SHOW_ID, M.COUNTRY FROM NETFLIX AS N INNER JOIN (    SELECT DIRECTOR, CO... 184 row(s) affected Records:
```

HANDLING NULL VALUES : FILLING POSSIBLE NULL VALUES IN COUNTRY WITH PREVIOUS DIRECTOR'S FILM COUNTRY NAME

ELT / DATA CLEANING / DATA MANIPULATION / HANDLING NULL VALUES

DELETED 7
DUPLICATE DATA

CREATED
FUNCTION TO
SEPARATE
COMMA(,) VALUES

USING FUNCTION
CREATED
SEPARATE TABLE
WITH VALUES

NEW TABLE CREATED
AS NETFLIX SHOW,
CAST, DIRECTOR,
GENRE, COUNTRY

FOUND MORE NULL
VALUES IN COUNTRY
TABLE

FILLED NULL VALUES
WITH DIRECTOR
PREVIOUS COUNTRY
MATCH

NETFLIX DATA ANALYSIS

1. FOR EACH DIRECTOR, COUNT THE NUMBER OF MOVIES AND TV SHOWS CREATED IN SEPARATE COLUMNS, FOCUSING ON DIRECTORS WHO HAVE CREATED BOTH.

```
SELECT ND.DIRECTOR,
       COUNT(CASE WHEN NS.TYPE='Movie' THEN NS.SHOW_ID END) AS NO_OF_MOVIES,
       COUNT(CASE WHEN NS.TYPE='Tv Show' THEN NS.SHOW_ID END) AS NO_OF_TV_SHOWS
  FROM NETFLIX_SHOW AS NS INNER JOIN NETFLIX_DIRECTOR AS ND ON
    NS.SHOW_ID = ND.SHOW_ID
 GROUP BY DIRECTOR
 having COUNT(DISTINCT NS.TYPE)>1 ;
```

	DIRECTOR	NO_OF_MOVIES	NO_OF_TV_SHOWS
▶	Abhishek Chaubey	1	1
	Alban Teurlai	1	1
	Anurag Kashyap	1	1
	Hiroyuki Seshita	2	1
	John Paul Tremblay	4	1
	Julia Reichert	1	1
	Mike Smith	1	1
	Robb Wells	4	1
	Sam Rockwell	1	1

NETFLIX DATA ANALYSIS

2. WHICH COUNTRY HAS HIGHEST NUMBER OF COMEDY MOVIES

```
SELECT NC.COUNTRY, COUNT(*) AS NO_OF_COMEDY_MOVIES FROM NETFLIX_COUNTRY AS NC  
JOIN NETFLIX_GENRE AS NG ON NC.SHOW_ID = NG.SHOW_ID  
JOIN NETFLIX_SHOW AS NS ON NS.SHOW_ID = NG.SHOW_ID  
WHERE GENRE = 'Comedies' AND NS.TYPE='Movie'  
GROUP BY NC.COUNTRY  
ORDER BY NO_OF_COMEDY_MOVIES DESC;
```

	COUNTRY	NO_OF_COMEDY_MOVIES
▶	United States	685
	India	335
	Canada	95
	United Kingdom	92
	Egypt	63
	Turkey	58
	France	51
	Spain	49
	Nigeria	43
	Germany	42
	Hong Kong	32
	China	31
	Philippines	29
	Mexico	27
	Brazil	24

NETFLIX DATA ANALYSIS

3. FOR EACH YEAR WHICH DIRECTOR HAS MAXIMUM NUMBER OF MOVIES RELEASED

- WITH CTE AS

```
( SELECT YEAR(DATE_ADDED) AS DATE_YEAR, DIRECTOR, COUNT(NS.SHOW_ID) AS NO_OF_MOVIES FROM NETFLIX_SHOW AS NS  
INNER JOIN NETFLIX_DIRECTOR AS ND ON  
ND.SHOW_ID = NS.SHOW_ID  
WHERE NS.TYPE='Movie'  
GROUP BY ND.DIRECTOR, DATE_YEAR  
)
```

CTE2 AS

```
(SELECT *,  
ROW_NUMBER() OVER(PARTITION BY DATE_YEAR ORDER BY NO_OF_MOVIES DESC) AS RN  
FROM CTE  
ORDER BY DATE_YEAR, NO_OF_MOVIES DESC  
)  
SELECT * FROM CTE2 WHERE RN =1;
```

	DATE_YEAR	DIRECTOR	NO_OF_MOVIES	RN
▶	2008	Sorin Dan Mihalcescu	1	1
	2009	Joe Dante	1	1
	2010	Jim Monaco	1	1
	2011	Larry Elkan	1	1
	2012	Constance Marks	1	1
	2013	Jason Moore	1	1
	2014	Leo Riley	2	1
	2015	Jay Karas	2	1
	2016	Jay Karas	4	1
	2017	Jay Chapman	7	1
	2018	Jan Suter	12	1
	2019	Martin Scorsese	7	1
	2020	Youssef Chahine	11	1

NETFLIX DATA ANALYSIS

4. WHAT IS AVERAGE DURATION OF MOVIES IN EACH GENRE

```
SELECT
    NS.SHOW_ID, GENRE, ROUND(AVG(REPLACE(DURATION, ' MIN', '')), 0) AS AVG_DURATION
FROM
    NETFLIX_SHOW AS NS
    JOIN
        NETFLIX_GENRE AS NG ON NG.SHOW_ID = NS.SHOW_ID
WHERE
    NS.TYPE = 'Movie'
GROUP BY GENRE;
```

SHOW_ID	GENRE	AVG_DURATION
s1	Documentaries	82
s10	Comedies	104
s10	Dramas	113
s1000	International Movies	110
s1000	Thrillers	107
s1001	Action & Adventure	114
s1006	Children & Family Movies	80
s1019	Music & Musicals	106
s102	Sports Movies	95
s1022	Independent Movies	101
s1028	Horror Movies	98
s1028	Romantic Movies	111
s1028	Sci-Fi & Fantasy	106
s1042	Anime Features	93

NETFLIX DATA ANALYSIS

5. FIND THE LIST OF DIRECTORS WHO HAVE CREATED HORROR AND COMEDY MOVIES BOTH. DISPLAY DIRECTOR NAMES ALONG WITH NUMBER OF COMEDY AND HORROR MOVIES DIRECTED BY THEM

```
SELECT ND.DIRECTOR,
       COUNT(DISTINCT CASE WHEN NG.GENRE = 'Comedies' THEN NS.SHOW_ID END) AS NO_OF_COMEDY_MOVIES,
       COUNT(DISTINCT CASE WHEN NG.GENRE = 'Horror Movies' THEN NS.SHOW_ID END) AS NO_OF_HORROR_MOVIES
  FROM NETFLIX_SHOW AS NS
 INNER JOIN NETFLIX_GENRE AS NG ON NS.SHOW_ID = NG.SHOW_ID
 INNER JOIN NETFLIX_DIRECTOR AS ND ON NS.SHOW_ID = ND.SHOW_ID
 WHERE TYPE = 'Movie'
   AND NG.GENRE IN ('Comedies', 'Horror Movies')
 GROUP BY ND.DIRECTOR
 HAVING COUNT(DISTINCT NG.GENRE) = 2;
```

DIRECTOR	NO_OF_COMEDY_MOVIES	NO_OF_HORROR_MOVIES
Adam Egypt Mortimer	1	1
Dennis Widmyer	1	1
Dibakar Banerjee	1	1
Gary Shore	1	1
Karan Johar	1	1
Kevin Kolsch	1	1
Kevin Smith	1	1
Nicholas McCarthy	1	1
Sarah Adina Smith	1	1
Scott Stewart	1	1
Ahmed Zein	2	2
Álex de la Iglesia	2	1
Amar Kaushik	1	1

NETFLIX DATA ANALYSIS

6. TOP 5 MOST POPULAR ACTORS BASED ON TV SHOWS

```
SELECT CAST, COUNT(DISTINCT NS.SHOW_ID) AS NO_OF_SHOWS
FROM NETFLIX_CAST AS NC JOIN NETFLIX_SHOW AS NS ON NS.SHOW_ID=NC.SHOW_ID
WHERE NS.TYPE='Tv Show'
GROUP BY CAST
ORDER BY NO_OF_SHOWS DESC
LIMIT 5;
```

CAST	NO_OF_SHOWS
Takahiro Sakurai	23
Yuki Kaji	17
Ai Kayano	16
Junichi Suwabe	16
David Attenborough	14

NETFLIX DATA ANALYSIS

7. MOST FREQUENT DIRECTOR - COUNTRY COLLABRATION

```
SELECT NC.COUNTRY, ND.DIRECTOR, COUNT(*) AS NO_OF_COLLABORATIONS
FROM NETFLIX_COUNTRY AS NC
JOIN NETFLIX_SHOW AS NS ON NC.SHOW_ID = NS.SHOW_ID
JOIN NETFLIX_DIRECTOR AS ND ON NS.SHOW_ID = ND.SHOW_ID
GROUP BY NC.COUNTRY, ND.DIRECTOR
ORDER BY NO_OF_COLLABORATIONS DESC
LIMIT 10;
```

COUNTRY	DIRECTOR	NO_OF_COLLABORATIONS
India	Rajiv Chilaka	21
United States	Marcus Raboy	16
India	Suhas Kadav	16
United States	Jay Karas	15
Philippines	Cathy Garcia-Molina	13
Egypt	Youssef Chahine	12
United States	Martin Scorsese	12

NETFLIX DATA ANALYSIS

8. FOR EACH YEAR, WHICH 3 GENRES HAD THE HIGHEST NUMBER OF SHOWS RELEASED?

```
WITH RankedGenres AS (
    SELECT RELEASE_YEAR, GENRE, COUNT(*) AS NO_OF_SHOWS,
           ROW_NUMBER() OVER(PARTITION BY RELEASE_YEAR ORDER BY COUNT(*) DESC) AS RN
    FROM NETFLIX_SHOW
    LEFT JOIN NETFLIX_GENRE ON NETFLIX_SHOW.SHOW_ID = NETFLIX_GENRE.SHOW_ID
    GROUP BY RELEASE_YEAR, GENRE
    ORDER BY RELEASE_YEAR DESC
)
SELECT * FROM RankedGenres
WHERE RN <= 3;
```

	RELEASE_YEAR	GENRE	NO_OF_SHOWS	RN
▶	2021	International TV Shows	149	1
	2021	International Movies	141	2
	2021	Dramas	92	3
	2020	International Movies	239	1
	2020	International TV Shows	214	2
	2020	Dramas	195	3
	2019	International Movies	282	1
	2019	Dramas	242	2

NETFLIX DATA ANALYSIS

9. HOW MANY ACTORS HAVE COLLABORATED WITH DIRECTOR RAJIV CHILAKA

```
SELECT COUNT(DISTINCT CAST) AS NO_OF_COLLABRATION  
FROM NETFLIX_SHOW AS NS  
JOIN NETFLIX_CAST AS NC ON NS.SHOW_ID = NC.SHOW_ID  
JOIN NETFLIX_DIRECTOR AS ND ON NS.SHOW_ID = ND.SHOW_ID  
WHERE ND.DIRECTOR = 'Rajiv Chilaka';
```

NO_OF_COLLABRATION
24

NETFLIX DATA ANALYSIS

10. WHICH GENRE REIGNS SUPREME IN EACH COUNTRY

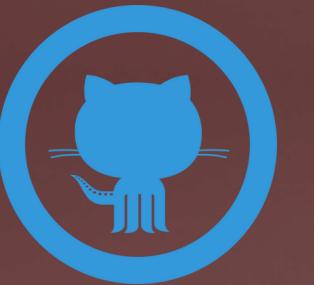
```
WITH RankedGenresPerCountry AS (
    SELECT NC.COUNTRY, GENRE, COUNT(*) AS NO_OF_SHOWS,
           ROW_NUMBER() OVER(PARTITION BY NC.COUNTRY ORDER BY COUNT(*) DESC) AS RN
    FROM NETFLIX_SHOW AS NS
    JOIN NETFLIX_COUNTRY AS NC ON NC.SHOW_ID = NS.SHOW_ID
    JOIN NETFLIX_GENRE AS NG ON NS.SHOW_ID = NG.SHOW_ID
    GROUP BY NC.COUNTRY, GENRE
    ORDER BY NC.COUNTRY ASC, NO_OF_SHOWS DESC
)
SELECT * FROM RankedGenresPerCountry
WHERE RN = 1;
```

COUNTRY	GENRE	NO_OF_SHOWS	RN
Afghanistan	Documentaries	1	1
Albania	Dramas	1	1
Algeria	International Movies	4	1
Angola	Action & Adventure	1	1
Argentina	International Movies	58	1
Armenia	International Movies	1	1
Australia	Dramas	38	1
Austria	International Movies	9	1
Azerbaijan	Romantic TV Shows	1	1
Bahamas	Action & Adventure	1	1
Bangladesh	Dramas	4	1
Belarus	British TV Shows	1	1
Belgium	International Movies	58	1



DATA ANALYSIS PROJECT

THANK YOU



/NANDHAKUMARB1



/NANDHAKUMARBHASKAR

