In [1]:

```
import pandas as pd
import pymysql as p
import numpy as np

import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
# DATA CLEANING:
```

In [3]:

```
df = pd.read_csv("netflix_titles.csv")


df["cast"]=df["cast"].str.split(',')
df=df.explode("cast").reset_index(drop=True)


df["country"]=df["country"].str.split(',')
df=df.explode("country").reset_index(drop=True)


df["listed_in"]=df["listed_in"].str.split(',')
df=df.explode("listed_in").reset_index(drop=True)

df["director"]=df["director"].str.split(',')
df=df.explode("director").reset_index(drop=True)


df.dropna(inplace=True)


df['date_added'] = df['date_added'].str.strip()  # Remove leading/trailing spaces
df['date_added'] = pd.to_datetime(df['date_added'])
df['day_added'] = df['date_added'].dt.day
df['year_added'] = df['date_added'].dt.year
df['month_added'] = df['date_added'].dt.month
df['year_added'] = df['year_added'].astype(int)
df['day_added'] = df['day_added'].astype(int)


df = df.drop_duplicates(subset=["show_id"])
```

In [4]:

```
pip install sqlalchemy pymysql
```

```
Requirement already satisfied: sqlalchemy in c:\users\lenovo\anaconda3\lib\site-packages
(2.0.39)
Requirement already satisfied: pymysql in c:\users\lenovo\anaconda3\lib\site-packages (1.
1.1)
Requirement already satisfied: greenlet!=0.4.17 in c:\users\lenovo\anaconda3\lib\site-pac
kages (from sqlalchemy) (3.1.1)
Requirement already satisfied: typing-extensions>=4.6.0 in c:\users\lenovo\anaconda3\lib\
site-packages (from sqlalchemy) (4.12.2)
Note: you may need to restart the kernel to use updated packages.
```

In [31]:

```
from sqlalchemy import create_engine


# MySQL connection details
username = "root"
```

```
password = ""
host = "localhost"
database = "netflix_titles"

# Create SQLAlchemy engine (using pymysql as driver)
engine = create_engine(f"mysql+pymysql://{username}:{password}@{host}/{database}")

# Export DataFrame to MySQL
df.to_sql(
    name="netflix_titles_details2",     # table name
    con=engine,
    if_exists="replace", # options: 'fail', 'replace', 'append'
    index=False          # don't write DataFrame index as a column
)

print("DataFrame exported successfully to MySQL!")
```

```
DataFrame exported successfully to MySQL!
```

In [ ]:

```

```

In [5]:

```
mydb = p.connect(
    host="localhost",
    user="root",
    password=""
)

cursor = mydb.cursor()

cursor.execute("USE netflix_titles")
```

Out[5]:

```
0
```

In [25]:

```
# Q1:

cursor.execute("SELECT DISTINCT(type) from netflix_titles_details2")


results = cursor.fetchall()

data = pd.DataFrame(results, columns=['type'])

data
```

Out[25]:

	type
0	Movie
1	TV Show

In [26]:

```
# Q2:
cursor.execute("SELECT country, COUNT(*) as title_count FROM netflix_titles_details2 GROU
P BY country")

results = cursor.fetchall()

data = pd.DataFrame(results, columns=['country', 'title_count'])
data
```

Out[26]:

	country	title_count
0	Argentina	39
1	Australia	43
2	Austria	4
3	Bangladesh	2
4	Belgium	8
...	...	...
63	United Kingdom	253
64	United States	1505
65	Uruguay	2
66	Venezuela	1
67	Vietnam	4

68 rows × 2 columns

In [27]:

```
# Q3:

cursor.execute("SELECT rating , Count(*) from netflix_titles_details2 GROUP BY rating")


results = cursor.fetchall()

data = pd.DataFrame(results, columns=['rating','count(*)'])

data
```

Out[27]:

	rating	count(*)
0	G	35
1	NC-17	1
2	NR	175
3	PG	176
4	PG-13	278
5	R	501
6	TV-14	917
7	TV-G	54
8	TV-MA	1189
9	TV-PG	358
10	TV-Y	24
11	TV-Y7	48
12	TV-Y7-FV	11
13	UR	7

In [28]:

```
# Q4:

cursor.execute("SELECT listed_in as genres, Count(*) as total from netflix_titles_details
2 GROUP BY rating")
```

```
results = cursor.fetchall()

data = pd.DataFrame(results, columns=['listed_in','total'])

data
```

Out[28]:

	listed_in	total
0	Documentaries	35
1	Dramas	1
2	Cult Movies	175
3	Dramas	176
4	Horror Movies	278
5	International Movies	501
6	Comedies	917
7	Children & Family Movies	54
8	Stand-Up Comedy	1189
9	Children & Family Movies	358
10	Children & Family Movies	24
11	Children & Family Movies	48
12	Children & Family Movies	11
13	Action & Adventure	7

In [29]:

```
# Q5:

cursor.execute("SELECT director as Name_of_Director, Count(*) as total from netflix_titles_details2 GROUP BY director")


results = cursor.fetchall()

data = pd.DataFrame(results, columns=['Name_of_Director','frequency'])

data
```

Out[29]:

	Name_of_Director	frequency
0	A. L. Vijay	2
1	A. Salaam	1
2	A.R. Murugadoss	3
3	Aadish Keluskar	1
4	Aamir Bashir	1
...	...	...
2845	Zhang Yimou	1
2846	Ziga Virc	1
2847	Zoe Berriatúa	2
2848	Zoe Lister-Jones	1

	Name_of_Director	frequency
2849	Zoya Akhtar	5

2850 rows × 2 columns

In [30]:

```
# Q6:

cursor.execute("SELECT year_added as ReleaseYear, count(*) as totalReleases from netflix_
titles_details2 GROUP BY year_added")
results = cursor.fetchall()


data=pd.DataFrame(results,columns=["Release Year","No. of Releases in this year"])
print(data)

print(" ")
print(" ")
print(" ")



# Sort by number of releases in descending order and get top 5
top_5 = data.sort_values(by="No. of Releases in this year", ascending=False).head(5)
top_5
```

```
    Release Year  No. of Releases in this year
0           2008                             1
1           2009                             2
2           2010                             1
3           2011                            13
4           2012                             4
5           2013                             7
6           2014                            14
7           2015                            50
8           2016                           211
9           2017                           805
10          2018                          1140
11          2019                          1386
12          2020                           140
```

Out[30]:

	Release Year	No. of Releases in this year
11	**2019**	**1386**
10	**2018**	**1140**
9	**2017**	**805**
8	**2016**	**211**
12	**2020**	**140**

In [ ]: