

DATABASE MANAGEMENT AND DATABASE DESIGN

FINAL PROJECT REPORT

Tv-show/Movie Recommendation System

TEAM MEMBERS -

Shreyas Rai (002769838)
Smiti Agrawal (002781419)
Sarthak Srivastava (002756847)

GITHUB LINK -

https://github.com/shreyashusky/tv_Shows_Recommendation_System.git

<https://github.com/smitihusky/tvShowRecommendationSystem.git>

<https://github.com/sarthakhusky/TV-Shows-Recommendation-System.git>

ABOUT

These days, the small screen has some very big things to offer. From sitcoms to dramas to travel and talk shows, all the small screen shows are the best in showing diversity.

TV shows are definitely one of the biggest business markets out there. Also, it offers an opportunity to upcoming artists. Over the years, the number of TV shows has increased exponentially and so has their customer base.

A survey conducted by www.deadline.com says that the top TV Shows can have 15-20 million viewership. With the advancement in technology and availability of cheap internet services, the reach of TV Shows is going to increase in the near future.

Our domain is Top TV Shows aired till date. We will use social networking APIs to get the data for entities which represent Companies, Genre, Subscription, Ratings, Producers and Consumers for our domain.

CONTENTS-

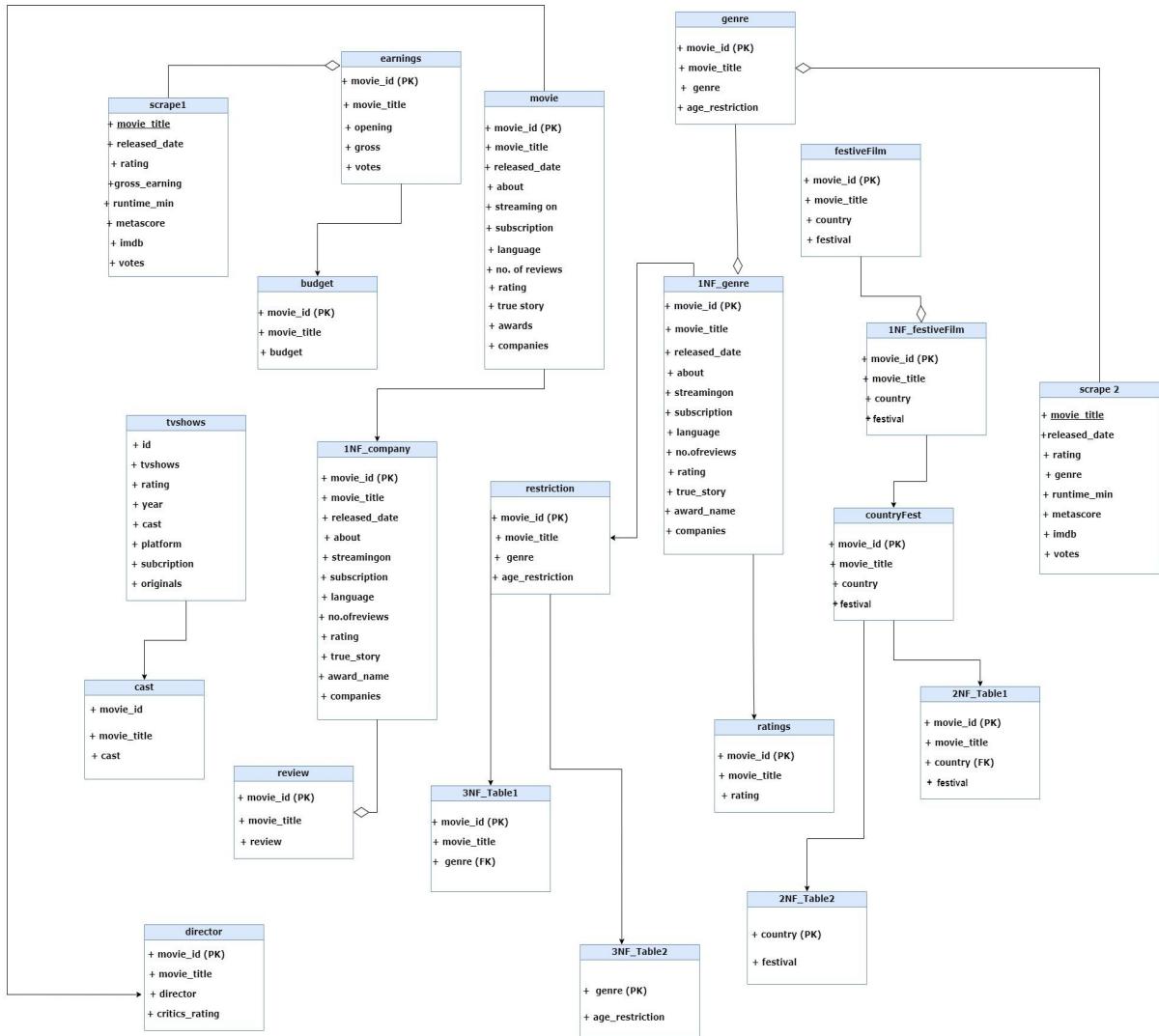
1.	Project Description	1
2.	Class Diagram	3
3.	Use-case Diagram	4
4.	SQL Create Statements	5
5.	Database Snippet	13
6.	Scraping Data & Cleaning Data	14
7.	UseCases SQL Query	17
8.	Data Visulization	32
9.	Data Normalization	34
10.	Views	42
11.	Procedures	52
12.	Conclusion	57

ENTITY RELATIONSHIP DIAGRAM -

Explanation on some of the design decisions :

- We have collected data from web scraping(imdb) and kaggle. Where movie_id is a primary key.
- Each table has specific data which helps the user to find a particular move of his/her choice.
- In a movie table the movie_id is the primary key which helps to find the movies and avoid duplication. Where as movie_title is foreign key which is given as the name of the movie.
- A user can search a movie based on the directors, companies, age, festival, cast and county. Each movie will be distinguished and have a movie_id.
- Table connection helps the user to find the movies of different taste and genre and even cast and directors.
- Each data collected has uniformity and accuracy which help the user to find right movies
- Here we have a genre table where a movie can have multiple genres so we have applied 1NF on the Genre table which separated the multiple genres into columns under the same movie.
- After getting data we have selected the first row and stored the data in restrictonTable.
- In the restriction table we used 3NF to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management and we get 3NF_tables.
- For 2NF we have taken the table festiveFilm, where we use 1NF to get countryfest and then we applied 2NF rules and made 2NF_Table1 and 2NF_Table2.

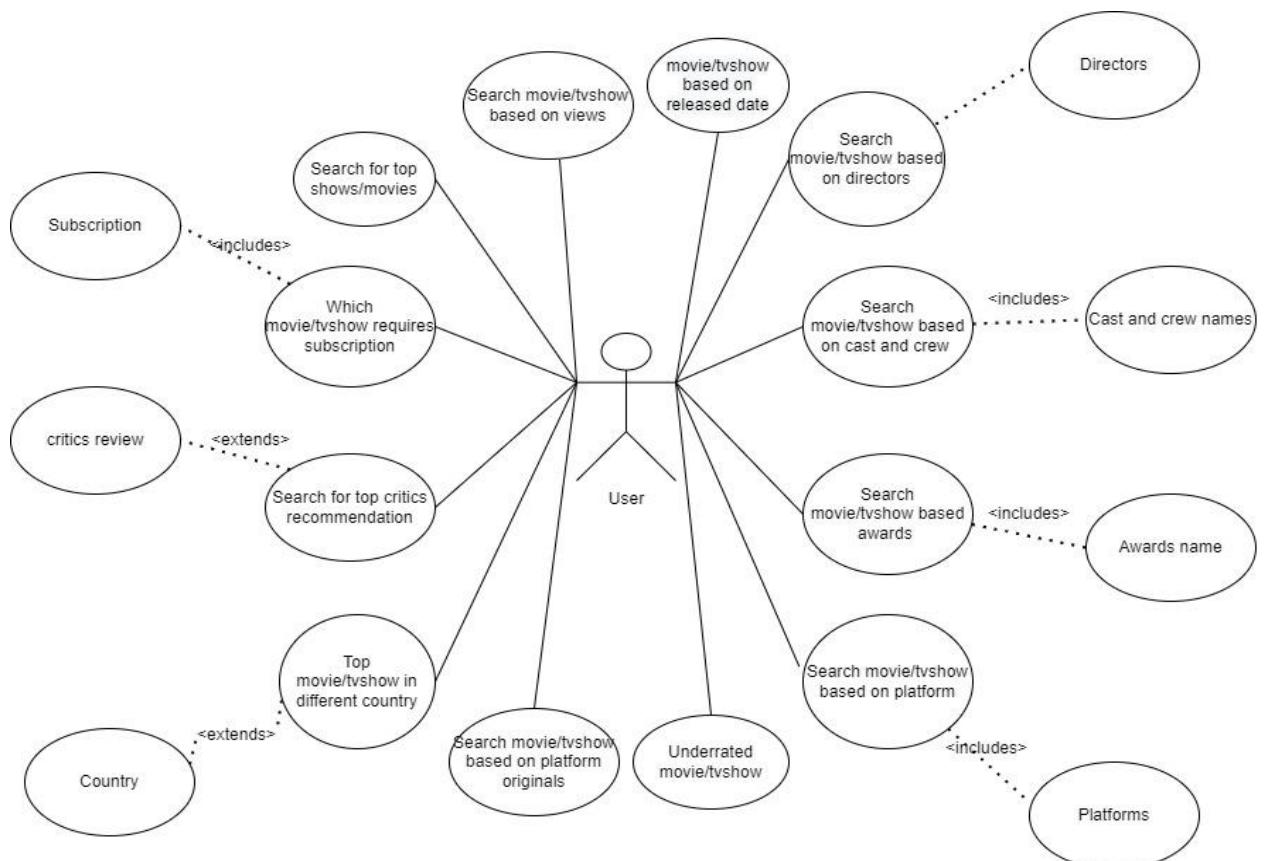
ER DIAGRAM



USE CASE DIAGRAM

Explanation on some of the design decisions :

When a user uses the tv-show/movie recommendations system he can browse for top movies/shows, in fact users can also search movies based on different platforms and which movie/tv-show is available on which platform and it will include the subscription details. Top Critic's recommendation is available which includes critics' reviews. The search can extend to the country and the top movie/tv-shows of different countries. A user can also search with directors, cast, or awards. This will include the recommendations based on cast and crew name and awards name. Users can also search for platform originals available on different platforms. This will help the users to find top recommendations based on his/her choice.



SQL CREATE STATEMENTS-

BUDGET

```
CREATE TABLE "budget" (
    "movie_id"      INTEGER,
    "movie_title"   TEXT,
    "budget"        INTEGER
    PRIMARY KEY("movie_id")
)
```

CAST

```
CREATE TABLE "cast" (
    "movie_id"      INTEGER,
    "movie_title"   TEXT,
    "cast"          TEXT
    PRIMARY KEY("movie_id")
)
```

DIRECTOR

```
CREATE TABLE "director" (
    "movie_id"      INTEGER,
    "movie_title"   TEXT,
    "director"      TEXT,
    "critics_rating" INTEGER
    PRIMARY KEY("movie_id")
)
```

EARNINGS

```
CREATE TABLE "earnings" (
    "movie_id"      INTEGER,
    "movie_title"   TEXT,
```

```
    "opening"      TEXT,  
    "gross" INTEGER,  
    "votes" INTEGER  
PRIMARY KEY("movie_id")  
)
```

FESTIVEFILM

```
CREATE TABLE "festiveFilm" (  
    "movie_id"      INTEGER,  
    "movie_title"   TEXT,  
    "country"       TEXT,  
    "festival"      TEXT  
PRIMARY KEY("movie_id")  
)
```

GENRE

```
CREATE TABLE "genre" (
```

```
"movie_id"    INTEGER,  
"movie_title" TEXT,  
"genre"TEXT,  
"certification" TEXT  
PRIMARY KEY("movie_id")  
)
```

MOVIE

```
CREATE TABLE "movie" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "released_date"    TEXT,  
    "about"TEXT,  
    "streamingon" TEXT,  
    "subscription" TEXT,  
    "language"    TEXT,  
    "no.ofreviews" INTEGER,  
    "ratings"    INTEGER,  
    "award_name"TEXT,  
    "true_story"    TEXT,  
    "companies"    TEXT  
PRIMARY KEY("movie_id")  
)
```

RATINGS

```
CREATE TABLE "ratings" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "rating"INTEGER  
PRIMARY KEY("movie_id")
```

)

REVIEW

```
CREATE TABLE "review" (
    "movie_id"      INTEGER,
    "movie_title"   TEXT,
    "review"        TEXT
PRIMARY KEY("movie_id")
)
```

SCRAPE1

```
CREATE TABLE "scrape1" (
    "field1"      INTEGER,
    "movie"        TEXT,
    "year"         INTEGER,
    "time_minute" INTEGER,
    "imdb_rating" REAL,
    "metascore"    TEXT,
```

```
"vote" INTEGER,  
"gross_earning"      TEXT  
PRIMARY KEY("movie_id")  
)
```

SCRAPE2

```
CREATE TABLE "scrape2" (  
    "field1" INTEGER,  
    "movie"      TEXT,  
    "year"      INTEGER,  
    "rating"    TEXT,  
    "genre"     TEXT,  
    "runtime_min" INTEGER,  
    "imdb"      REAL,  
    "metascore"  INTEGER,  
    "votes"      INTEGER,  
    "n_imdb"    REAL  
PRIMARY KEY("movie_id")  
)
```

TVShows

```
CREATE TABLE "tvshow" (  
    "id"      INTEGER,  
    "tvshows"   TEXT,  
    "rating"    REAL,  
    "year"      INTEGER,  
    "star_cast"  TEXT,  
    "field6"    TEXT,
```

```
"platform"      TEXT,  
"subscription" TEXT,  
"originals"     TEXT  
PRIMARY KEY("id")  
)
```

```
CREATE TABLE "restriction" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "genre"       TEXT,  
    "age_restriction" TEXT  
)
```

```
CREATE TABLE "countryFest" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "country"     TEXT,  
    "festival"    TEXT  
)
```

```
CREATE TABLE "1NF_festiveFilm" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "country"     TEXT,  
    "festival"    TEXT  
)
```

```
CREATE TABLE "1NF_genre" (  
    "movie_id"    INTEGER,  
    "movie_title" TEXT,  
    "released_date" TEXT,
```

```
"about" TEXT,  
"streamingon" TEXT,  
"subscription" TEXT,  
"language" TEXT,  
"no.ofreviews" INTEGER,  
"ratings" REAL,  
"award_name" TEXT,  
"true_story" TEXT,  
"companies" TEXT  
)
```

```
CREATE TABLE "1NFcompany" (  
    "movie_id" INTEGER,  
    "movie_title" TEXT,  
    "released_date" TEXT,  
    "about" TEXT,  
    "streamingon" TEXT,  
    "subscription" TEXT,  
    "language" TEXT,  
    "no.ofreviews" INTEGER,  
    "ratings" INTEGER,  
    "award_name" TEXT,  
    "true_story" TEXT,  
    "companies" TEXT  
)
```

```
CREATE TABLE "2NF_Table1" (  
    "movie_id" INTEGER,  
    "movie_title" TEXT,  
    "country" TEXT,  
    "festival" TEXT
```

```

)
CREATE TABLE "2NF_Table2" (
    "country"      TEXT,
    "festival"     TEXT
)

CREATE TABLE "3NF_Table1" (
    "movie_id"     INTEGER,
    "movie_title"  TEXT,
    "genre"        TEXT
)

CREATE TABLE "3NF_Table2" (
    "genre"        TEXT,
    "age_restriction" TEXT
)

```

Inserting data in Table

```

for index, row in df_movie.iterrows():

    cursor.execute("INSERT INTO Movies values (%s,%s,%s,%s)",
    (int(row['movie_id']),row['movie_title'],row['about'],row['ratings']))
    conn.commit()

```

DATABASE SNIPPET-

Screenshot of a database management tool interface showing the schema and code for 21 tables and 15 views.

Database Structure Tab:

- Tables (21):**
 - 1NF_festiveFilm
 - 1NF_genre
 - 1NFcompany
 - 2NF_Table1
 - 2NF_Table2
 - 3NF_Table1
 - 3NF_Table2
 - budget
 - cast
 - countryFest
 - director
 - earnings
 - festiveFilm
 - genre
 - movie
 - ratings
 - restriction
 - review
 - scrape1
 - scrape2
 - tvshow
- Indices (0):** No indices are present.
- Views (15):**
 - [actionpacked] AS select m.movie_title, g.genre, m.ratings from genre g, movie m where g.movie_id=m.movie_id and m.award LIKE "%TRUE%" and g.genre like "%Action%" order by m.ratings desc
 - [activator] AS select m.movie_title, c.cast, m.release_date from movie m, cast c where m.movie_id=c.movie_id and c.cast like "%Robert Downey Jr.%"
 - [bestdirector] AS select c.movie_title as Movie, d.director, c.cast from cast c, director d, ratings r where c.movie_id=d.movie_id and d.movie_id=r.movie_id and r.rating>8 order by r.rating desc
 - [dirberlin] AS select d.movie_title, d.director, f.festival from director d, festival f where d.movie_id=f.movie_id and festival like "%Berlin%"
 - [directorearning] AS select distinct d.director,sum(e.gross) as total_gross from director d, earnings e where d.movie_id=e.movie_id group by d.director order by e.gross desc limit 1
 - [directorrestriction] AS select d.movie_title,d.director from director d , genre g where d.movie_id=g.movie_id and g.certification = "PG"
 - [familyfest] AS select f.movie_title,g.genre,f.festival from restriction g,festiveFilm f where g.movie_id=f.movie_id and festival="Cannes" and genre like "%Family%"
 - [parentsguide] AS select g.movie_title, g.certification, r.rating from genre g, ratings r where g.movie_id=r.movie_id and g.certification = "PG13" order by rating desc limit 20
 - [profitablemovies] AS select e.movie_title,(e.gross-b.budget) as profit from earnings e, budget b where e.movie_id=b.movie_id order by profit desc limit 10
 - [realstory] AS select distinct(m.movie_title), e.votes from movie m,earnings e where e.movie_id=m.movie_id and m.biography="1" order by votes desc
 - [romantic] AS select c.cast from cast c,genre g where c.movie_id=g.movie_id and genre like "%Romance%"
 - [superhit] AS select r.movie_title as superhit_movies, f.country from ratings r, festiveFilm f where r.rating>7.1 group by r.movie_title limit 10
 - [tophulu] AS select m.movie_title from review r, movie m where r.movie_title=m.movie_title and m.streaming_on like "%shuul%" and r.review like "%Best%"

SCRAPING DATA

Scrapping#1

```
In [5]: # Storing each of the urls of 50 movies
for page in pages:
    # Getting the contents from the each url
    page = requests.get('https://www.imdb.com/search/title/?groups=top_1000&start=' + str(page) + '&ref_=adv_nxt', headers=headers)
    soup = BeautifulSoup(page.text, 'html.parser')

    # Aiming the part of the html we want to get the information from
    movie_div = soup.find_all('div', class_='lister-item mode-advanced')

    # Controlling the loop's rate by pausing the execution of the loop for a specified amount of time
    # Waiting time between requests for a number between 2-10 seconds
    sleep(randint(2,10))

    for container in movie_div:
        # Scraping the movie's name
        name = container.h3.a.text
        titles.append(name)

        # Scraping the movie's year
        year = container.h3.find('span', class_='lister-item-year').text
        years.append(year)

        # Scraping the movie's length
        runtime = container.find('span', class_='runtime').text if container.p.find('span', class_='runtime') else '-'
        time.append(runtime)

        # Scraping the rating
        imdb = float(container.strong.text)
        imbd_ratings.append(imdb)

        # Scraping the metascore
        m_score = container.find('span', class_='metascore').text if container.find('span', class_='metascore') else '-'
        metascores.append(m_score)

        # Scraping votes and gross earnings
        nv = container.find_all('span', attrs={'name': 'nv'})
```

Scrapping#2

Jupyter Assignment4 Last Checkpoint: 18 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted Python 3 (ipykernel) C

```
In [39]: # create a empty list for storing
# movie information
list = []

# Iterating over movies to extract
# each movie's details
for index in range(0, len(movies)):

    # Separating movie into: 'place',
    # 'title', 'year'
    movie_string = movies[index].get_text()
    movie = (' '.join(movie_string.split()).replace('.', ' '))
    movie_title = movie[len(str(index))+1:-7]
    year = re.search('\((.*?)\)', movie_string).group(1)
    place = movie[:len(str(index))-(len(movie))]
    data = {"place": place,
            "movie_title": movie_title,
            "rating": ratings[index],
            "year": year,
            "star_cast": crew[index],
            }
    list.append(data)

In [40]: for movie in list:
    print(movie['place'], '-', movie['movie_title'], '('+movie['year'] +
      ') -', 'Starring:', movie['star_cast'], movie['rating'])

1 - Planet Earth II (2016) - Starring: David Attenborough, Chadden Hunter 9.437618220563438
2 - Breaking Bad (2008) - Starring: Bryan Cranston, Aaron Paul 9.427990952668377
3 - Planet Earth (2006) - Starring: Sigourney Weaver, David Attenborough 9.41470735247401
4 - Band of Brothers (2001) - Starring: Scott Grimes, Damian Lewis 9.391704537225568
5 - Chernobyl (2019) - Starring: Jessie Buckley, Jared Harris 9.317638272372545
6 - The Wire (2002) - Starring: Dominic West, Lance Reddick 9.291195945250243
7 - Blue Planet II (2017) - Starring: David Attenborough, Peter Drost 9.23102995832745
8 - Avatar: The Last Airbender (2005) - Starring: Dee Bradley Baker, Zach Tyler Eisen 9.229296504039295
9 - Cosmos: A Spacetime Odyssey (2014) - Starring: Neil deGrasse Tyson, Christopher Emerson 9.202989130190462
1 - The Sopranos (1999) - Starring: James Gandolfini, Lorraine Bracco 9.200325581901142
```

Scraping#3

The screenshot shows a Jupyter Notebook interface with the title "Assignment4" and a status bar indicating "Last Checkpoint: 18 hours ago (unsaved changes)". The notebook has tabs for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a toolbar with various icons. The Python version is listed as "Python 3 (ipykernel)". The code in the cells includes:

```
In [28]: final_df.to_csv('scrape2.csv')

In [57]: #scraping data for tv shows from imdb
from bs4 import BeautifulSoup
import requests
import re
import pandas as pd

In [34]: url = 'https://www.imdb.com/chart/toptv/'
response = requests.get(url)
soup = BeautifulSoup(response.text, "html.parser")

In [38]: movies = soup.select('td.titleColumn')
crew = [a.attrs.get('title') for a in soup.select('td.titleColumn a')]
ratings = [b.attrs.get('data-value')
           for b in soup.select('td.posterColumn span[name=ir]')]
```

CLEANING DATA

The screenshot shows a Jupyter Notebook interface with the title "Assignment4" and a status bar indicating "Last Checkpoint: 18 hours ago (unsaved changes)". The notebook has tabs for File, Edit, View, Insert, Cell, Kernel, Widgets, Help, and a toolbar with various icons. The Python version is listed as "Python 3 (ipykernel)". The code in the cells includes:

```
In [8]: # Cleaning 'year' column
movies['year'] = movies['year'].str.extract('(\d+)').astype(int)
movies.head(3)

Out[8]:
      movie   year time_minute  imbd_rating  metascore    vote gross_earning
0  A Christmas Story  1983        93 min       7.9       77  151,583     $20.61M
1  Triangle of Sadness  2022       147 min       7.7       63   27,718          -
2  All Quiet on the Western Front  2022       148 min       7.8       76   74,549          -
```

```
In [9]: # Cleaning 'time_minute' column
movies['time_minute'] = movies['time_minute'].str.extract('(\d+)').astype(int)
movies.head(3)

Out[9]:
      movie   year time_minute  imbd_rating  metascore    vote gross_earning
0  A Christmas Story  1983         93       7.9       77  151,583     $20.61M
1  Triangle of Sadness  2022        147       7.7       63   27,718          -
2  All Quiet on the Western Front  2022        148       7.8       76   74,549          -
```

```
In [10]: # Cleaning 'metascore' column
movies['metascore'] = movies['metascore'].str.extract('(\d+)')
# convert it to float and if there are dashes turn it into NaN
movies['metascore'] = pd.to_numeric(movies['metascore'], errors='coerce')

In [11]: # Cleaning 'vote' column
movies['vote'] = movies['vote'].str.replace(',', '').astype(int)
movies.head(3)

Out[11]:
      movie   year time_minute  imbd_rating  metascore    vote gross_earning
0  A Christmas Story  1983         93       7.9       77.0  151583     $20.61M
```

```

In [12]: # Cleaning 'gross_earning' column
          # left strip $ and right strip M
movies['gross_earning'] = movies['gross_earning'].map(lambda x: x.lstrip('$').rstrip('M'))
          # convert it to float and if there are dashes turn it into NaN
movies['gross_earning'] = pd.to_numeric(movies['gross_earning'], errors='coerce')
movies.head(3)

```

```

Out[12]:
      movie   year  time_minute  imdb_rating  metascore    vote  gross_earning
0   A Christmas Story  1983           93       7.9     77.0  151583     20.61
1  Triangle of Sadness  2022          147       7.7     63.0  27718      NaN
2  All Quiet on the Western Front  2022          148       7.8     76.0  74549      NaN

```

Output:

	movie	year	time_minute	imdb_rating	metascore	vote	gross_earning
0	A Christmas Story	1983	93	7.9	77.0	151583	20.61
1	Triangle of Sadness	2022	147	7.7	63.0	27718	NaN
2	All Quiet on the Western Front	2022	148	7.8	76.0	74549	NaN
3	Top Gun: Maverick	2022	130	8.4	78.0	438268	NaN
4	Everything Everywhere All at Once	2022	139	8.1	81.0	244709	NaN
...
995	Airlift	2016	130	7.9	NaN	57326	NaN
996	Paan Singh Tomar	2012	135	8.2	NaN	36590	0.04
997	Anand	1971	122	8.1	NaN	34030	NaN
998	Sarfarosh	1999	174	8.1	NaN	25937	NaN
999	The Breath	2009	128	8.0	NaN	33972	NaN

1000 rows × 7 columns

```
In [15]: movies.to_csv('c_movies.csv')
```

```
In [18]: movies["gross_earning"] = movies["gross_earning"].fillna("Null")
```

	movie	year	time_minute	imdb_rating	metascore	vote	gross_earning
0	A Christmas Story	1983	93	7.9	77.0	151583	20.61
1	Triangle of Sadness	2022	147	7.7	63.0	27718	NaN
2	All Quiet on the Western Front	2022	148	7.8	76.0	74549	NaN
3	Top Gun: Maverick	2022	130	8.4	78.0	438268	NaN
4	Everything Everywhere All at Once	2022	139	8.1	81.0	244709	NaN
...
995	Airlift	2016	130	7.9	NaN	57326	NaN
996	Paan Singh Tomar	2012	135	8.2	NaN	36590	0.04
997	Anand	1971	122	8.1	NaN	34030	NaN
998	Sarfarosh	1999	174	8.1	NaN	25937	NaN
999	The Breath	2009	128	8.0	NaN	33972	NaN

1000 rows × 7 columns

```
In [15]: movies.to_csv('c_movies.csv')
```

```
In [18]: movies["gross_earning"] = movies["gross_earning"].fillna("Null")
```

USECASES ON PROJECT

5 USE-CASES (SMITI AGRAWAL)

**every use case includes joins of 2 tables

1. What are the family movies that are released during the Cannes film festival

?

```
In [104]: #UseCase
#What are the family movies that are released during the Cannes film festival ?
run_query('select g.movie_title,g.genre,f.festival
          from genre g,festiveFilm f
          where g.movie_id=f.movie_id and festival="Cannes" and genre like "%family%"')
```

	movie_title	genre	festival
0	Tangled	Animation Family	Cannes
1	Alice in Wonderland	Family Fantasy Adventure	Cannes
2	Monsters University	Animation Family	Cannes
3	Cars 2	Animation Family Adventure Comedy	Cannes
4	Toy Story 3	Animation Family Comedy	Cannes
...
66	Fantasia 2000	Animation Family Music	Cannes
67	Looney Tunes: Back in Action	Animation Comedy Family	Cannes
68	Oceans	Documentary Family	Cannes
69	The Lion King	Family Animation Drama	Cannes
70	Cloudy with a Chance of Meatballs 2	Animation Family Comedy	Cannes

71 rows x 3 columns

2. What are the top 10 movies having higher profits?

```
In [107]: #What are the top 10 movies having higher profits?
run_query('select e.movie_title,(e.gross-b.budget) as profit
          from earnings e, budget b
          where e.movie_id=b.movie_id
          order by profit desc limit 10 ')
```

	movie_title	profit
0	Avatar	2550965087
1	Titanic	1645034188
2	Jurassic World	1363528810
3	Furious 7	1316249360
4	The Avengers	1299557910
5	Avengers: Age of Ultron	1125403694
6	Frozen	1124219009
7	The Lord of the Rings: The Return of the King	1024888979
8	Iron Man 3	1015439994
9	Transformers: Dark of the Moon	928746996

3. Which director's movie has earned maximum money ?

```
In [116]: #Which director's movie has earned the maximum money ?
run_query('select distinct d.director,sum(e.gross) as total_gross
          from director d, earnings e
          where d.movie_id=e.movie_id
          group by d.director
          order by e.gross desc limit 1')
```

director	total_gross
0 James cameron	2787965087

4. Highest rated movies released in the year 2000, USA?

```
In [133]: #Highest rated movies released in year 2000, USA?
run_query('select m.movie_title,m.released_date, f.country, m.ratings
          from movie m,festiveFilm f
          where m.movie_id=f.movie_id and ratings>7 and m.released_date like "%2000%" and f.country like "%United States"
          order by ratings desc')

Out[133]:
      movie_title   released_date           country  ratings
0        Gladiator  2000-05-01  United Kingdom  United States of America    7.9
1       Cast Away  2000-12-22  United States of America    7.5
2  The Emperor's New Groove  2000-12-09  United States of America    7.2
```

5. What are real story based movies that have earned the highest votes?

```
In [142]: #What are real story based movies that has earned highest votes?
run_query('select distinct(m.movie_title), e.votes
          from movie m,earnings e where e.movie_id=m.movie_id and m.true_story="1"
          order by votes desc')

Out[142]:
      movie_title  votes
0            2012  4903
1            Zodiac  2023
2            Robin Hood  1398
3            Pompeii  1267
4  The Legend of Zorro  893
5  Gulliver's Travels  621
6             Ali  447
7            金陵十三釵  187
8        Town & Country  16
```

5 USE-CASES (SHREYAS RAI)

**every use case includes joins of 2 tables

1. What are top rated movies for children above age 13?

```
In [148]: #Top rated movies for children above age 13
run_query('select g.movie_title, g.certification, r.rating
          from genre g, ratings r
          where g.movie_id=r.movie_id and g.certification = "PG13"
          order by rating desc limit 20')

Out[148]:
      movie_title  certification  rating
0        The Dark Knight  PG13  8.2
1        Interstellar  PG13  8.1
2        Inception  PG13  8.1
3  The Lord of the Rings: The Return of the King  PG13  8.1
4  The Lord of the Rings: The Fellowship of the Ring  PG13  8.0
5  The Lord of the Rings: The Two Towers  PG13  8.0
6  Guardians of the Galaxy  PG13  7.9
7        Gladiator  PG13  7.9
8  The Wolf of Wall Street  PG13  7.9
9        The Departed  PG13  7.9
10       Shutter Island  PG13  7.8
11  Terminator 2: Judgment Day  PG13  7.7
12  The Dark Knight Rises  PG13  7.6
13  The Hobbit: The Desolation of Smaug  PG13  7.6
14  Edge of Tomorrow  PG13  7.6
15  Captain America: The Winter Soldier  PG13  7.6
16        The Martian  PG13  7.6
17       Finding Nemo  PG13  7.6
```

2. List award winning movies for Action packed

```
In [181]: #List award winning movies for Action packed
run_query('select m.movie_title, g.genre, m.ratings
          from genre g, movie m
          where g.movie_id=m.movie_id
          and m.award_name = "1"
          and g.genre like "%Action%"
          order by m.ratings desc')
```

Out[181]:

	movie_title	genre	ratings
0	Inception	Action Thriller Science Fiction My...	8.1
1	The Lord of the Rings: The Return of the King	Adventure Fantasy Action	8.1
2	The Lord of the Rings: The Fellowship of the Ring	Adventure Fantasy Action	8.0
3	The Lord of the Rings: The Two Towers	Adventure Fantasy Action	8.0
4	Gladiator	Action Drama Adventure	7.9
5	Terminator 2: Judgment Day	Action Thriller Science Fiction	7.7
6	The Dark Knight Rises	Action Crime Drama Thriller	7.6
7	Edge of Tomorrow	Action Science Fiction	7.6
8	X-Men: Days of Future Past	Action Adventure Fantasy Science F...	7.5
9	Batman Begins	Action Crime Drama	7.5
10	The Avengers	Science Fiction Action Adventure	7.4
11	Star Trek Into Darkness	Action Adventure Science Fiction	7.4
12	The Hunger Games: Catching Fire	Adventure Action Science Fiction	7.4
13	Avatar	Action Adventure Fantasy Science F...	7.2

3. List the movies which has less ratings but earned more

```
In [160]: #Movies which has less ratings but earned more
run_query('select e.movie_title
          from earnings e, ratings r
          where e.movie_id=r.movie_id
          and r.rating<5
          order by e.gross desc')
```

Out[160]:

	movie_title
0	Independence Day: Resurgence
1	The Last Airbender
2	Eragon
3	Batman & Robin
4	Gulliver's Travels
5	Fantastic Four
6	Speed 2: Cruise Control
7	Jack and Jill
8	Nutty Professor II: The Klumps
9	Cats & Dogs 2 : The Revenge of Kitty Galore
10	Catwoman
11	Stealth
12	How Do You Know
13	Battlefield Earth
14	Town & Country
15	The Adventures of Pluto Nash
16	A Sound of Thunder

4. Movies which were superhit in more than one country

```
In [177]: #Movies which were superhit in more than one country
run_query('select r.movie_title as superhit_movies, f.country
           from ratings r, festiveFilm f
           where r.rating>7.1
           group by r.movie_title limit 10')
```

```
Out[177]:
superhit_movies          country
0      A Beautiful Mind  United States of America United Kingdom
1      All That Jazz     United States of America United Kingdom
2      American Gangster United States of America United Kingdom
3      Avatar            United States of America United Kingdom
4      Avengers: Age of Ultron United States of America United Kingdom
5      Batman Begins     United States of America United Kingdom
6      Big Hero 6        United States of America United Kingdom
7      Black Hawk Down   United States of America United Kingdom
8      Blood Diamond    United States of America United Kingdom
9      Captain America: The Winter Soldier United States of America United Kingdom
```

5. Cast and directors with maximum hit movies

```
In [170]: #Cast and directors with maximum hit movies
run_query('select c.movie_title as Movie, d.director, c.cast
           from cast c, director d, ratings r
           where c.movie_id=d.movie_id
           and d.movie_id=r.movie_id
           and r.rating>8
           order by r.rating desc')
```

```
Out[170]:
Movie          director          cast
0  The Dark Knight  Roberto Benigni  Bruce Wayne fecafdf Christian Bale ...
1  Interstellar   Makoto Shinkai  Joseph Cooper febbfceeb Matthew McCon...
2  Inception       Nitesh Tiwari  Dom Cobb fecaeae Leonardo DiCaprio ...
3 The Lord of the Rings: The Return of the King Darius Marder  Frodo Baggins febcff Elijah Wood ...
```

5 USE-CASES (SARTHAK SHRIVASTAV)

**every use case includes joins of 2 tables

1. What are the best movies available on Hulu?

```
In [197]: #what are the Best movies available on Hulu?
run_query('select m.movie_title
           from review r, movie m
           where r.movie_title=m.movie_title and m.`streaming on` like "%hulu%" and r.review like "%Best%"')
```

Out[197]:

	movie_title
0	Avatar
1	The Hobbit: The Desolation of Smaug
2	The Hobbit: The Desolation of Smaug
3	X-Men: Days of Future Past
4	Star Trek Into Darkness
5	The Great Gatsby
6	The Dark Knight
7	Edge of Tomorrow
8	Maleficent
9	Dawn of the Planet of the Apes
10	Captain America: The Winter Soldier
11	How to Train Your Dragon
12	Guardians of the Galaxy
13	The Curious Case of Benjamin Button
14	X-Men: First Class
15	Ratatouille
16	Batman Begins
17	Mad Max: Fury Road

2. Which directors has age restrictions as PG?

```
In [365]: #which directors has age restriction as PG
run_query('select d.movie_title,d.director
           from director d , genre g where d.movie_id=g.movie_id and g.certification = "PG" ')
```

Out[365]:

	movie_title	director
0	Tangled	Nathan Greno
1	Harry Potter and the Half-Blood Prince	David Yates
2	The Chronicles of Narnia: Prince Caspian	Andrew Adamson
3	Monsters University	Dan Scanlon
4	Oz: The Great and Powerful	Sam Raimi
...
111	Arthur et les Minimoys	Luc Besson
112	Oceans	Gary Ross
113	Don Gato: El inicio de la pandilla	Andrés Couturier
114	The Lion King	Rob Minkoff
115	Journey 2: The Mysterious Island	Brad Peyton

3. What movies of directors are in the Berlin Festival?

```
In [370]: #What movies of directors are in Berlin Festival?  
run_query(' select d.movie_title, d.director, f.festival  
          from director d , festiveFilm f  
          where d.movie_id =f.movie_id and festival like "%Berlin%"')  
  
Out[370]:
```

	movie_title	director	festival
0	Spectre	Sam mendes	Berlin
1	Harry Potter and the Half-Blood Prince	David Yates	Berlin
2	Quantum of Solace	Marc Forster	Berlin
3	Robin Hood	Otto Bathurst	Berlin
4	King Kong	Peter Jackson	Berlin
5	Skyfall	Sam Mendes	Berlin
6	X-Men: Days of Future Past	Steven Spielberg	Berlin
7	Up	David Fincher	Berlin
8	Monsters vs Aliens	Jonathan Demme	Berlin
9	Iron Man	George Lucas	Berlin
10	G.I. Joe: The Rise of Cobra	Martin Scorsese	Berlin
11	The Jungle Book	Ridley Scott	Berlin
12	Terminator 3: Rise of the Machines	Can Ulkay	Berlin
13	Inception	Nitesh Tiwari	Berlin
14	Alice Through the Looking Glass	Andrew Stanton	Berlin
15	Alexander	Stanley Kubrick	Berlin
16	Harry Potter and the Order of the Phoenix	Francis Ford Coppola	Berlin
17	Harry Potter and the Goblet of Fire	Ridley Scott	Berlin

4. Which actor were active in which year?

```
In [378]: #Which actor were active in which year?  
run_query('select m.movie_title, c.cast, m.released_date  
          from movie m, cast c  
          where m.movie_id=c.movie_id and c.cast like "%Robert Downey Jr.%")  
  
Out[378]:
```

	movie_title	cast	released_date
0	Avengers: Age of Ultron	Tony Stark / Iron Man edcde Robert Do...	2015-04-22
1	The Avengers	Tony Stark / Iron Man fecaeb Robert D...	2012-04-25
2	Captain America: Civil War	Steve Rogers / Captain America ecbdead ...	2016-04-27
3	Iron Man 3	Tony Stark / Iron Man feacaedff Rober...	2013-04-18
4	Iron Man	Tony Stark / Iron Man fecafee Robert ...	2008-04-30
5	Iron Man 2	Tony Stark / Iron Man fec Robert Down...	2010-04-28
6	The Incredible Hulk	Bruce Banner / The Hulk fecafcf Edward...	2008-06-12
7	Sherlock Holmes: A Game of Shadows	Sherlock Holmes fecafba Robert Downey...	2011-11-22
8	Tropic Thunder	Tugg Speedman fedcaf Ben Stiller ...	2008-08-09
9	Sherlock Holmes	Sherlock Holmes fecef Robert Downey J...	2009-12-23
10	Zodiac	Robert Graysmith fecafddd Jake Gyllen...	2007-03-02

5. Which actor have done a Romantic Movie?

```
In [380]: #Which actor have done a Romantic Movie?  
run_query('select c.cast from cast c,genre g where c.movie_id=g.movie_id and genre like "%Romance"')  
  
Out[380]:  
cast  
0 Jay Gatsby feecaea Leonardo DiCaprio ...  
1 Prince Dastan feccafbe Jake Gyllenhaal...  
2 Maleficent feccafcb Angelina Jolie ...  
3 James Stewart / Jay Fennel feecae Jos...  
4 Daisy feecafb Cate Blanchett Ben...  
5 Alexander fecafed Colin Farrell ...  
6 Isabella 'Bella' Swan fececaff Kriste...  
7 Lisa Jorgenson fecafee Reese Withersp...  
8 Porter Stoddard fecae Warren Beatty ...  
9 Jack Byrnes feaca Robert De Niro ...  
10 Frank Tupelo fec Johnny Depp Eli...  
11 Carrie Bradshaw feecc Sarah Jessica P...  
12 Giselle feccaff Amy Adams Rober...  
13 Amanda Woods fecaf Cameron Diaz ...  
14 Jane Adler fecaedc Meryl Streep ...  
15 Sayuri fecafcf Zhang Ziyi Hatsum...  
16 Michael Newman feecafbf Adam Sandler ...  
17 Capt. Spurgeon 'Fish' Tanner febcrafdf ...  
18 Harry Sanborn febcffcc Jack Nicholson...  
19 Skeeter Bronson fecad Adam Sandler ...
```

PREVIOUS ASSIGNMENTS USECASES

(Modeling 10 questions 10*3=30 questions)

- What are the relevant shows for the given age group?

```
In [200]: #What are the relevant shows for the given age group?  
run_query('select movie_title, certification from genre where certification="PG13" OR certification="PG"')  
  
Out[200]:  
movie_title certification  
0 Avatar PG13  
1 Pirates of the Caribbean: At World's End PG13  
2 Spectre PG13  
3 The Dark Knight Rises PG13  
4 John Carter PG13  
... ...  
459 Journey 2: The Mysterious Island PG  
460 Cloudy with a Chance of Meatballs 2 PG13  
461 Red Dragon PG13  
462 Hidalgo PG13  
463 Jack and Jill PG13  
  
464 rows x 2 columns
```

- What are the underrated shows to watch

```
In [211]: #What are the underrated show to watch ?
run_query('select movie_title from movie where ratings<7 and `no.of reviews`>5000')
```

```
Out[211]:
```

	movie_title
0	Batman v Superman: Dawn of Justice
1	Man of Steel
2	The Amazing Spider-Man
3	Jurassic World
4	Skyfall
5	Iron Man 3
6	World War Z
7	Suicide Squad
8	Iron Man 2
9	Thor
10	Spider-Man
11	Captain America: The First Avenger
12	The Hunger Games: Mockingjay - Part 1
13	Prometheus
14	The Hunger Games

- Which is the cheapest platform to watch the show?

```
In [216]: #Which is the cheapest platform to watch the show?
run_query('select `streaming on` as cheapest_platform
          from movie
          group by `streaming on`
          order by cheapest_platform desc limit 1')
```

```
Out[216]:
```

	cheapest_platform
0	prime

- What are the low budget- top rated shows?

```
In [218]: #What are the low budget- top rated show?
run_query('select r.movie_title from ratings r, budget b where r.movie_id=b.movie_id and rating>8 order by b.budget')
```

```
Out[218]:
```

	movie_title
0	The Lord of the Rings: The Return of the King
1	Inception
2	Interstellar
3	The Dark Knight

- What are the shows that are popular in the country outside their origin country?

```
In [222]: #What are the shows that are popular in the country outside their origin country?
run_query('select f.movie_title, f.country from festiveFilm f, earnings e where f.movie_id=e.movie_id and votes>10000')
```

```
Out[222]:
```

	movie_title	country
0	Avatar	United States of America United Kingdom
1	The Avengers	United States of America
2	The Dark Knight	United Kingdom United States of America
3	Interstellar	Canada United States of America United...
4	Inception	United Kingdom United States of America
5	Django Unchained	United States of America

- Who are involved in the cast and crew of the particular show?

In [224]:	#Who are involved in the cast and crew of the particular show? run_query('select tvshows,star_cast from tvshow')																																				
Out[224]:	<table border="1"> <thead> <tr> <th></th> <th>tvshows</th> <th>star_cast</th> </tr> </thead> <tbody> <tr> <td>click to scroll output; double click to hide</td> <td>rth II</td> <td>David Attenborough, Chadden Hunter</td> </tr> <tr> <td>1</td> <td>Breaking Bad</td> <td>Bryan Cranston, Aaron Paul</td> </tr> <tr> <td>2</td> <td>Planet Earth</td> <td>Sigourney Weaver, David Attenborough</td> </tr> <tr> <td>3</td> <td>Band of Brothers</td> <td>Scott Grimes, Damian Lewis</td> </tr> <tr> <td>4</td> <td>Chernobyl</td> <td>Jessie Buckley, Jared Harris</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>95</td> <td>Gullak</td> <td>Jameel Khan, Geetanjali Kulkarni</td> </tr> <tr> <td>96</td> <td>Sarabhai V/S Sarabhai</td> <td>Satish Shah, Ratna Pathak Shah</td> </tr> <tr> <td>97</td> <td>Invincible</td> <td>Steven Yeun, J.K. Simmons</td> </tr> <tr> <td>98</td> <td>Panchayat</td> <td>Jitendra Kumar, Raghuvir Yadav</td> </tr> <tr> <td>99</td> <td>Mad Men</td> <td>Jon Hamm, Elisabeth Moss</td> </tr> </tbody> </table> <p>100 rows × 2 columns</p>		tvshows	star_cast	click to scroll output; double click to hide	rth II	David Attenborough, Chadden Hunter	1	Breaking Bad	Bryan Cranston, Aaron Paul	2	Planet Earth	Sigourney Weaver, David Attenborough	3	Band of Brothers	Scott Grimes, Damian Lewis	4	Chernobyl	Jessie Buckley, Jared Harris	95	Gullak	Jameel Khan, Geetanjali Kulkarni	96	Sarabhai V/S Sarabhai	Satish Shah, Ratna Pathak Shah	97	Invincible	Steven Yeun, J.K. Simmons	98	Panchayat	Jitendra Kumar, Raghuvir Yadav	99	Mad Men	Jon Hamm, Elisabeth Moss
	tvshows	star_cast																																			
click to scroll output; double click to hide	rth II	David Attenborough, Chadden Hunter																																			
1	Breaking Bad	Bryan Cranston, Aaron Paul																																			
2	Planet Earth	Sigourney Weaver, David Attenborough																																			
3	Band of Brothers	Scott Grimes, Damian Lewis																																			
4	Chernobyl	Jessie Buckley, Jared Harris																																			
...																																			
95	Gullak	Jameel Khan, Geetanjali Kulkarni																																			
96	Sarabhai V/S Sarabhai	Satish Shah, Ratna Pathak Shah																																			
97	Invincible	Steven Yeun, J.K. Simmons																																			
98	Panchayat	Jitendra Kumar, Raghuvir Yadav																																			
99	Mad Men	Jon Hamm, Elisabeth Moss																																			

- Which are the most viewed shows?

In [229]:	#Which are the most viewed shows? run_query('select movie_title,`no.of reviews` from movie where (`no.of reviews`)>10000')																					
Out[229]:	<table border="1"> <thead> <tr> <th></th> <th>movie_title</th> <th>no.of reviews</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>Avatar</td> <td>11800</td> </tr> <tr> <td>1</td> <td>The Avengers</td> <td>11776</td> </tr> <tr> <td>2</td> <td>The Dark Knight</td> <td>12002</td> </tr> <tr> <td>3</td> <td>Interstellar</td> <td>10867</td> </tr> <tr> <td>4</td> <td>Inception</td> <td>13752</td> </tr> <tr> <td>5</td> <td>Django Unchained</td> <td>10099</td> </tr> </tbody> </table>		movie_title	no.of reviews	0	Avatar	11800	1	The Avengers	11776	2	The Dark Knight	12002	3	Interstellar	10867	4	Inception	13752	5	Django Unchained	10099
	movie_title	no.of reviews																				
0	Avatar	11800																				
1	The Avengers	11776																				
2	The Dark Knight	12002																				
3	Interstellar	10867																				
4	Inception	13752																				
5	Django Unchained	10099																				

- Can a user filter the show by its durations?

In [233]:	#Can a user filter the show by its durations? run_query('select movie,time_minute as `duration(mins)` from scrapel order by time_minute')																																				
Out[233]:	<table border="1"> <thead> <tr> <th></th> <th>movie</th> <th>duration(mins)</th> </tr> </thead> <tbody> <tr> <td>click to scroll output; double click to hide</td> <td>Sherlock Jr.</td> <td>45</td> </tr> <tr> <td>1</td> <td>Freaks</td> <td>64</td> </tr> <tr> <td>2</td> <td>Battleship Potemkin</td> <td>66</td> </tr> <tr> <td>3</td> <td>The Cabinet of Dr. Caligari</td> <td>67</td> </tr> <tr> <td>4</td> <td>The General</td> <td>67</td> </tr> <tr> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>995</td> <td>Once Upon a Time in America</td> <td>229</td> </tr> <tr> <td>996</td> <td>Gone with the Wind</td> <td>238</td> </tr> <tr> <td>997</td> <td>Zack Snyder's Justice League</td> <td>242</td> </tr> <tr> <td>998</td> <td>Hamlet</td> <td>242</td> </tr> <tr> <td>999</td> <td>Gangs of Wasseypur</td> <td>321</td> </tr> </tbody> </table> <p>1000 rows × 2 columns</p>		movie	duration(mins)	click to scroll output; double click to hide	Sherlock Jr.	45	1	Freaks	64	2	Battleship Potemkin	66	3	The Cabinet of Dr. Caligari	67	4	The General	67	995	Once Upon a Time in America	229	996	Gone with the Wind	238	997	Zack Snyder's Justice League	242	998	Hamlet	242	999	Gangs of Wasseypur	321
	movie	duration(mins)																																			
click to scroll output; double click to hide	Sherlock Jr.	45																																			
1	Freaks	64																																			
2	Battleship Potemkin	66																																			
3	The Cabinet of Dr. Caligari	67																																			
4	The General	67																																			
...																																			
995	Once Upon a Time in America	229																																			
996	Gone with the Wind	238																																			
997	Zack Snyder's Justice League	242																																			
998	Hamlet	242																																			
999	Gangs of Wasseypur	321																																			

- What are the shows that are platform originals? Ex-shows that are netflix original.

```
In [238]: #What are the shows that are platform originals? Ex-shows that are netflix original.
run_query('select tvshows, platform from tvshow where originals="1"')

Out[238]:
      tvshows    platform
0   The Sopranos      HBO
1   Game of Thrones      HBO
2     Sherlock      BBC One
3  Batman: The Animated Series    prime
4 Scam 1992: The Harshad Mehta Story    sling
5       Fargo        hulu
6    TVF Pitchers       tvf
7    Black Mirror       None
8      Narcos      netflix
9   Dragon Ball Z   funimation
10  Dragon Ball Z   funimation
11   Six Feet Under      HBO
12    Kota Factory      netflix
13      Dark      netflix
14    The Boys      prime
15  Stranger Things      netflix
16      Gullak      sling
17   Panchayat      prime
```

- Select top rated TV shows/movies based on their genre.

```
In [241]: #Select top rated TV shows/movies based on their genre.
run_query('select g.movie_title, genre')

Out[241]:
      movie_title           genre
0          Titanic  Drama Romance Thriller
1  The Great Gatsby      Drama Romance
2  The Curious Case of Benjamin Button  Fantasy Drama Thriller Mystery ...
3    Cinderella Man      Romance Drama History
4  Memoirs of a Geisha  Drama History Romance
5        Déjà Vu      Romance Drama
6  A Beautiful Mind      Drama Romance
```

- Select award winning TV shows/movies.

```
In [242]: #Select award winning TV shows/movies.
run_query('select movie_title from movie where award_name="1"')

Out[242]:
      movie_title
0          Avatar
1  The Dark Knight Rises
2        Tangled
3  Harry Potter and the Half-Blood Prince
4        The Avengers
5  The Hobbit: The Desolation of Smaug
6        Titanic
7  X-Men: Days of Future Past
8  Star Trek Into Darkness
9    Edge of Tomorrow
10       Inside Out
11      Interstellar
12      Inception
13  Harry Potter and the Goblet of Fire
14      Ratatouille
15      Batman Begins
16  The Hunger Games: Catching Fire
17  The Lord of the Rings: The Fellowship of the Ring
18      Gladiator
```

- Select TV shows/movies based on real stories.

```
In [245]: #Select TV shows/movies based on real stories.
run_query('select movie_title,released_date from movie where true_story="1"')
```

Out[245]:

	movie_title	released_date
0	Robin Hood	2010-05-12
1	2012	2009-10-10
2	Gulliver's Travels	2010-12-25
3	Ali	2001-12-11
4	Town & Country	2001-04-27
5	金陵十三釵	2011-12-15
6	Zodiac	2007-03-02
7	The Legend of Zorro	2005-10-24
8	Pompeii	2014-02-18

- Select TV shows/movies according to Directors.

```
In [247]: #Select TV shows/movies according to Directors.
run_query('select movie_title, director from director where director="Andrew Stanton"')
```

Out[247]:

	movie_title	director
0	John Carter	Andrew Stanton
1	Alice Through the Looking Glass	Andrew Stanton

- Select a TV shows/movie which has sequence eg. part 1 or part 2

```
In [255]: #Select a movie which has sequence eg. part 1 or part 2
run_query('select movie_title from movie where movie_title like "Spider-Man%"')
```

Out[255]:

	movie_title
0	Spider-Man 3
1	Spider-Man 2
2	Spider-Man

- Select a TV show/movie which was released in more than one language/country.

```
In [262]: #Select a TV show/movie which was released in more than one language/country.
run_query('select movie_title,country from festiveFilm where country like "%_____United States of America"')
```

Out[262]:

	movie_title	country
0	Spectre	United Kingdom United States of America
1	Harry Potter and the Half-Blood Prince	United Kingdom United States of America
2	Quantum of Solace	United Kingdom United States of America
3	Man of Steel	United Kingdom United States of America
4	The Chronicles of Narnia: Prince Caspian	CzCzech Republic Poland Slovenia U...
...
123	Nine	Italy United States of America
124	Red Planet	Australia United States of America
125	A Sound of Thunder	Czech Republic Germany United Kingdom...
126	Pompeii	Canada Germany United States of America
127	Red Dragon	Germany United States of America

128 rows × 2 columns

- Select a TV show/movie based on the cast and crew.

```
In [263]: #Select a TV show/movie based on the cast and crew.
run_query('select * from cast')

Out[263]:
   movie_id      movie_title          cast
0     25683           Avatar Jake Sully aacaca Sam Worthington ...
1    14531  Pirates of the Caribbean: At World's End Captain Jack Sparrow fecafbd Johnny D...
2     18260            Spectre James Bond fedcaeddb Daniel Craig ...
3    25056  The Dark Knight Rises Bruce Wayne / Batman fecaf Christian ...
4    99197        John Carter John Carter feacafea Taylor Kitsch ...
...
495   75730  Journey 2: The Mysterious Island Hank fecae Dwayne Johnson Sean...
496   78945  Cloudy with a Chance of Meatballs 2 Flint Lockwood (voice) feabcafddf Bill...
497   57476            Red Dragon Hannibal Lecter fecafbbd Anthony Hopkins ...
498   55309            Hidalgo Frank Hopkins fedcaf Viggo Mortensen ...
499   72551        Jack and Jill Jack / Jill fedcaef Adam Sandler ...

500 rows × 3 columns
```

- Select a TV show/movie which was most watched on the released date.

```
In [268]: #Select a TV show/movie which was most watched on the released date.
run_query('select m.movie_title,m.released_date from movie m,earnings e where m.movie_id=e.movie_id and e.opening like')

Out[268]:
   movie_title  released_date
0   Suicide Squad  2016-08-02
1   Jason Bourne  2016-07-27
2   Ghostbusters  2016-07-14
3   Star Trek Beyond  2016-07-07
4   The Legend of Tarzan  2016-06-29
```

- What movies/TV shows require a subscription and on which platform?

```
In [281]: #What movies/TV shows require a subscription and on which platform?
run_query('select movie.title, `streaming on` from movie')

Out[281]:
   movie_title  streaming on
0           Avatar        hulu
1  Pirates of the Caribbean: At World's End        hulu
2           Spectre        hulu
click to scroll output; double click to hide  The Dark Knight Rises        netflix
4           John Carter        hulu
...
495  Journey 2: The Mysterious Island        hulu
496  Cloudy with a Chance of Meatballs 2        hulu
497            Red Dragon        hulu
498            Hidalgo        netflix
499        Jack and Jill        hulu

500 rows × 2 columns
```

- What are the top critic's movies/tv-show?

```
In [283]: #What are the top critic's movies/tv-show?
run_query('select movie_title from director where critics_rating>7')

Out[283]:
      movie_title
0          Avatar
1  The Dark Knight Rises
2        Tangled
3  Avengers: Age of Ultron
4  Harry Potter and the Half-Blood Prince
...
94    Spirit: Stallion of the Cimarron
95       Déjà Vu
96        Oceans
97   A Beautiful Mind
98      The Lion King

99 rows × 1 columns
```

- What are the top movies/tv-show from different Countries?

```
In [288]: #What are the top movies/tv-show from different Countries?
run_query('select movie_title, country from festiveFilm ')

Out[288]:
      movie_title           country
0          Avatar United States of America United Kingdom
1  Pirates of the Caribbean: At World's End           United States of America
2        Spectre  United Kingdom United States of America
3  The Dark Knight Rises           United States of America
4      John Carter           United States of America
...
495  Journey 2: The Mysterious Island           United States of America
496  Cloudy with a Chance of Meatballs 2           United States of America
497      Red Dragon           Germany United States of America
498      Hidalgo           United States of America Morocco
499      Jack and Jill           United States of America
```

- What are the different Companies(Warner Bros/ Phantom etc) and their top movies/Tv-show?

```
In [293]: #What are the different Companies( Warner Bros/ Phantom etc)?
run_query('select movie_title from movie where companies like "%Warner Bros.%"')

Out[293]:
      movie_title
0      The Dark Knight Rises
1  Harry Potter and the Half-Blood Prince
2  Batman v Superman: Dawn of Justice
3      Superman Returns
4      Man of Steel
...
81      Swordfish
82      Battlefield Earth
83  Looney Tunes: Back in Action
84      The Postman
85      Red Planet

86 rows × 1 columns
```

- Which movies/Tv-show were released in theaters or had a platform streaming?

```
In [297]: #Which movies/Tv-show were released on platform streaming?
run_query('select movie_title from movie where `streaming on` = "netflix"')

Out[297]:
      movie_title
0      The Dark Knight Rises
1      Avengers: Age of Ultron
2      Harry Potter and the Half-Blood Prince
3      Pirates of the Caribbean: Dead Man's Chest
4      The Avengers
...
161     Babe: Pig in the City
162     Oceans
163     A Beautiful Mind
164     The Lion King
165     Hidalgo

166 rows × 1 columns
```

- What are recently released movies/tv-shows?

```
In [305]: #What are recently released movies/tv-shows?
run_query('select movie_title, released_date from movie where released_date >2016-12-10 and released_date<2012-01-10')

Out[305]:
      movie_title  released_date
0      Titanic    1997-11-18
1      Wild Wild West    1999-06-29
2      Waterworld    1995-07-28
3      Armageddon    1998-07-01
4      Lethal Weapon 4    1998-07-10
...
71      Battlefield Earth    2000-05-10
72      The Postman    1997-12-25
73      Babe: Pig in the City    1998-11-25
74      Red Planet    2000-11-10
75      The Lion King    1994-06-23

76 rows × 2 columns
```

- What are the top short films/Limited Series?

```
In [314]: #What are the top short films/Limited Series?
run_query('select movie, time_minute, imdb_rating from scrapel where time_minute>60 and time_minute <99 order by imdb_ra')

Out[314]:
      movie  time_minute  imdb_rating
0      12 Angry Men      96      9.0
1      The Lion King      88      8.5
2      Grave of the Fireflies      89      8.5
3      Modern Times      87      8.5
4      City Lights      87      8.5
...
165     The Station Agent      89      7.6
166     The Invisible Man      71      7.6
167     Philomena      98      7.6
168     Eyes Without a Face      90      7.6
169     The 39 Steps      86      7.6

170 rows × 3 columns
```

- Which movies/tv-show were in the Film festival(Cannes)?

```
In [315]: #Which movies/tv-show were in the Film festival(Cannes)?
run_query('select movie_title from festiveFilm where festival = "Cannes"')
```

Out[315]:

	movie_title
0	Avatar
1	Pirates of the Caribbean: At World's End
2	John Carter
3	Tangled
4	Batman v Superman: Dawn of Justice
...	...
256	Oceans
257	A Sound of Thunder
258	Pompeii
259	The Lion King
260	Cloudy with a Chance of Meatballs 2

261 rows x 1 columns

- Which movies/tv-show had a top box office collection?

```
In [320]: #Which movies/tv-show had a top box office collection?
run_query('select movie_title from budget where budget > 100000000')
```

Out[320]:

	movie_title
0	Avatar
1	Pirates of the Caribbean: At World's End
2	Spectre
3	The Dark Knight Rises
4	John Carter
...	...
270	Lara Croft: Tomb Raider
271	The Alamo
272	Atlantis: The Lost Empire
273	Swordfish
274	Pompeii

275 rows x 1 columns

DATA VISUALIZATION

```
import pandas as pd
import matplotlib.pyplot as plt

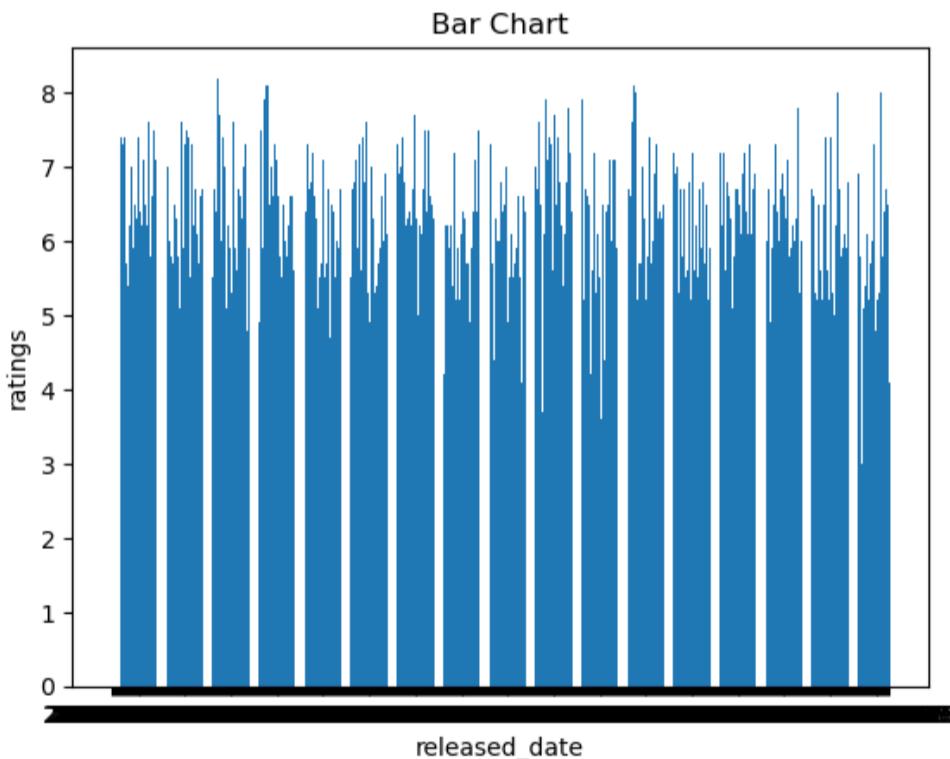
# reading the database
data = pd.read_csv("movie.csv")

# Bar chart with day against tip
plt.bar(data['released_date'], data['ratings'])

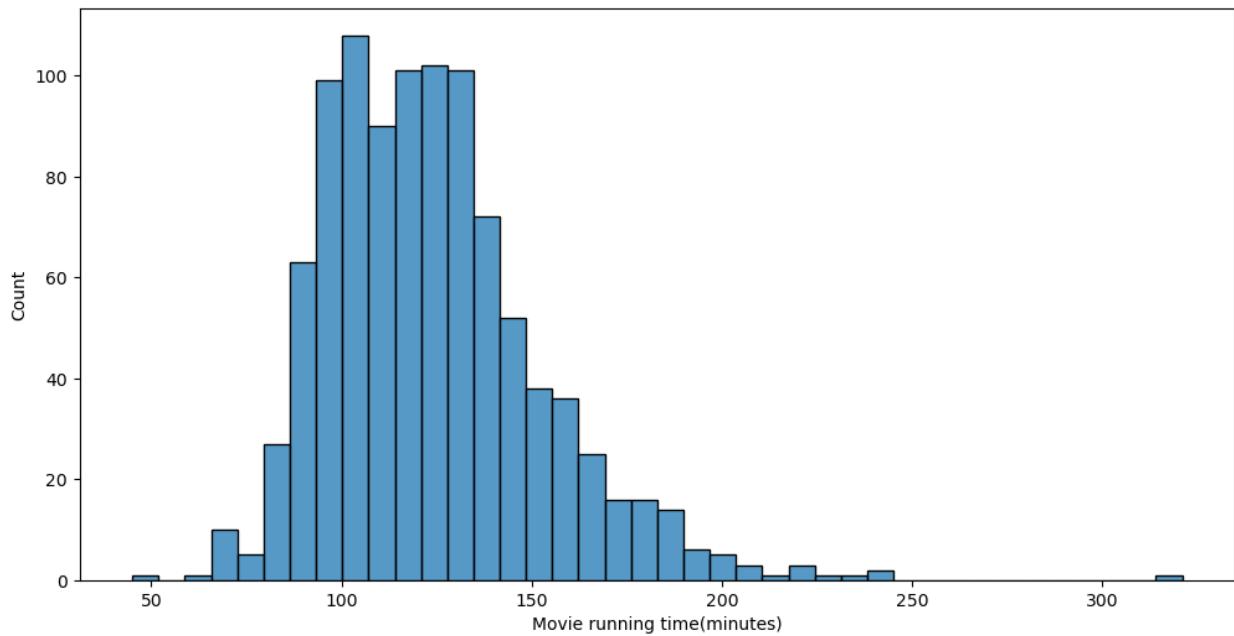
plt.title("Bar Chart")

# Setting the X and Y labels
plt.xlabel('released_date')
plt.ylabel('ratings')

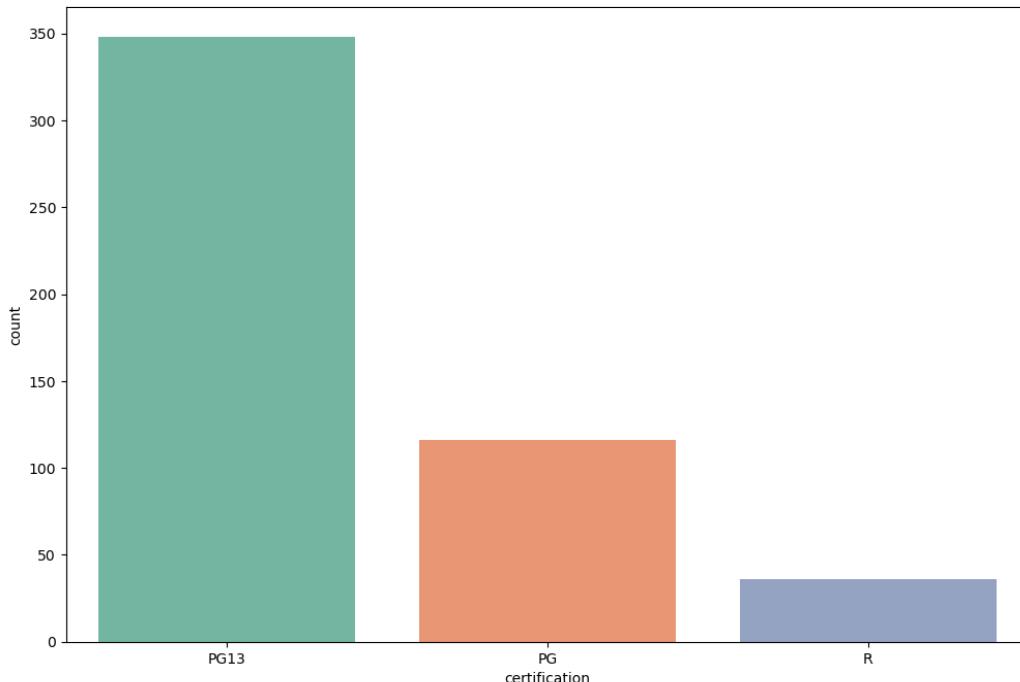
# Adding the legends
plt.show()
```



```
In [333]: #Visualise the data according to Runtime ?
fig, ax = plt.subplots(figsize=(12,6))
sns.histplot(data=movie_df['time_minute'],ax=ax)
plt.xlabel('Movie running time(minutes)')
```



```
In [345]: #Visualise the data according to the certification:
df2 = movie_df.groupby('certification')['movie_id'].mean()
fig, ax1 = plt.subplots(figsize=(12,8))
ax1 = sns.countplot(x="certification", data=movie_df, palette="Set2",
                     order=movie_df['certification'].value_counts().index[0:13])
```



```
click to scroll output; double click to hide
df1 = movie_df.groupby('released_date')['ratings'].mean()
df1.plot( label='Avg IMDB by year')
plt.legend()
plt.xlabel('Released Year')
plt.ylabel('Avg IMDB ')
plt.show()
```

NORMALIZATION

First Normal Form (1NF)

- Each table has a primary key with minimal attributes that can uniquely identify a record.
- The values in each column of a table are atomic.
- There are no repeating groups in the same table.

Example 1: In the genre.csv (image below) the Movie **Avatar** has **4 genres** but in according to 1NF there should be minimal attributes and each record should be uniquely identified.

Before 1NF:

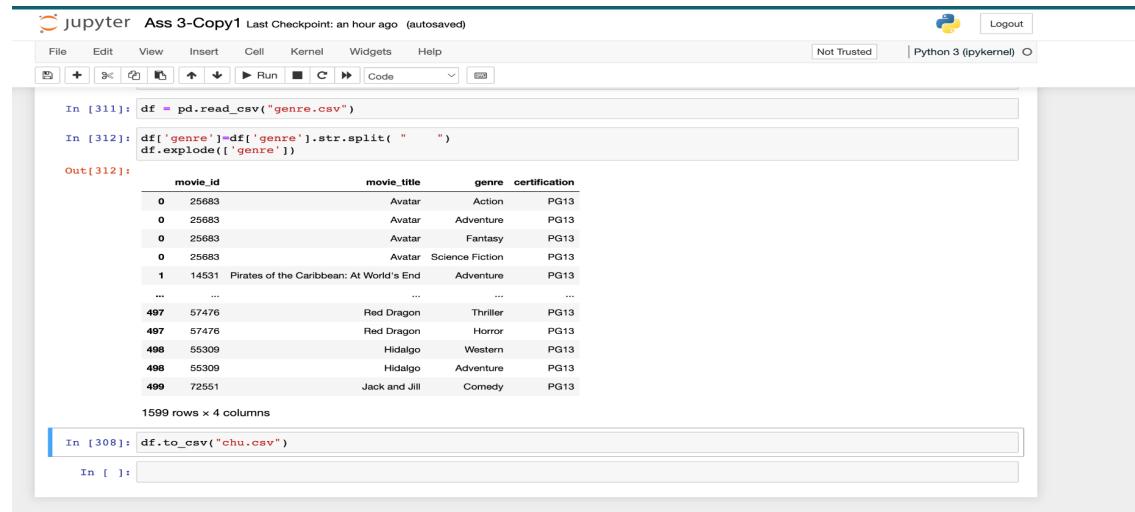
The screenshot shows a Jupyter Notebook interface with the following code and output:

```
In [251]: df = pd.read_csv("genre.csv")
In [252]: df
Out[252]:
   movie_id      movie_title      genre certification
0     25683        Avatar  Action Adventure Fantasy Science F...
1     14531  Pirates of the Caribbean: At World's End  Adventure Fantasy Action PG13
2     18260          Spectre  Action Adventure Crime PG13
3     25056  The Dark Knight Rises  Action Crime Drama Thriller PG13
4     99197       John Carter  Action Adventure Science Fiction PG13
...
495    75730  Journey 2: The Mysterious Island  Adventure Action Science Fiction PG
496    78945  Cloudy with a Chance of Meatballs 2  Animation Family Comedy PG13
497    57476          Red Dragon  Crime Thriller Horror PG13
498    55309          Hidalgo  Western Adventure PG13
499    72551        Jack and Jill  Comedy PG13
500 rows × 4 columns
```

```
In [311]: df = pd.read_csv("genre.csv")
In [312]: df['genre']=df['genre'].str.split( "  ")
df.explode(['genre'])
Out[312]:
   movie_id      movie_title      genre certification
0     25683        Avatar      Action          PG13
```

So in 1NF, the values in each column of a table should be atomic and there should be no repeating groups so we can see that here the movie **Avatar's** genre section is now split into **4 rows** of genres : Action, Adventure, Fantasy and Science Fiction.

After 1NF:



In [311]: df = pd.read_csv("genre.csv")
In [312]: df['genre']=df['genre'].str.split(" ")
df.explode(['genre'])

Out[312]:

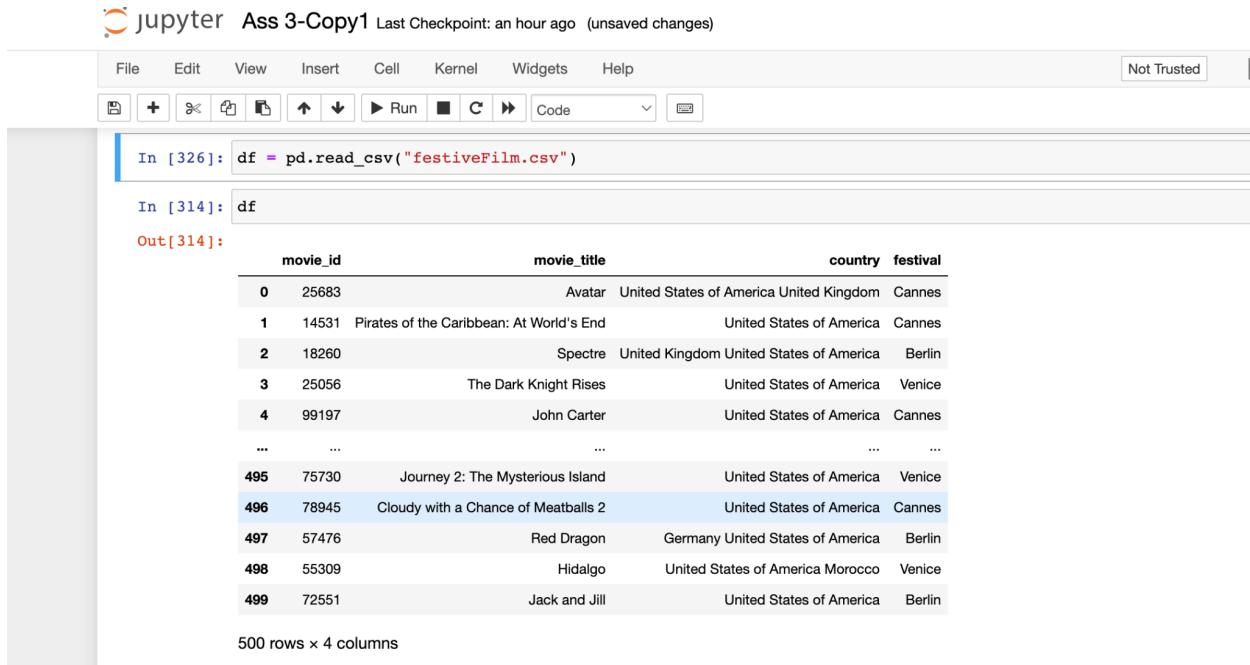
	movie_id	movie_title	genre	certification
0	25683	Avatar	Action	PG13
0	25683	Avatar	Adventure	PG13
0	25683	Avatar	Fantasy	PG13
0	25683	Avatar	Science Fiction	PG13
1	14531	Pirates of the Caribbean: At World's End	Adventure	PG13
...
497	57476	Red Dragon	Thriller	PG13
497	57476	Red Dragon	Horror	PG13
498	55309	Hidalgo	Western	PG13
498	55309	Hidalgo	Adventure	PG13
499	72551	Jack and Jill	Comedy	PG13

1599 rows × 4 columns

In [308]: df.to_csv("chu.csv")
In []:

Example 2: Same thing is implemented for countries. In the country column we have multiple values, so we have applied 1NF and separated it into atomic values.

Before 1NF:



In [326]: df = pd.read_csv("festiveFilm.csv")
In [314]: df

Out[314]:

	movie_id	movie_title	country	festival
0	25683	Avatar	United States of America United Kingdom	Cannes
1	14531	Pirates of the Caribbean: At World's End	United States of America	Cannes
2	18260	Spectre	United Kingdom United States of America	Berlin
3	25056	The Dark Knight Rises	United States of America	Venice
4	99197	John Carter	United States of America	Cannes
...
495	75730	Journey 2: The Mysterious Island	United States of America	Venice
496	78945	Cloudy with a Chance of Meatballs 2	United States of America	Cannes
497	57476	Red Dragon	Germany United States of America	Berlin
498	55309	Hidalgo	United States of America Morocco	Venice
499	72551	Jack and Jill	United States of America	Berlin

500 rows × 4 columns

After 1NF:

497	57476	Red Dragon	Germany United States of America	Berlin
498	55309	Hidalgo	United States of America Morocco	Venice
499	72551	Jack and Jill	United States of America	Berlin

500 rows × 4 columns

```
In [327]: df['country']=df['country'].str.split(" ")
df.explode(['country'])
```

Out[327]:

	movie_id	movie_title	country	festival
0	25683	Avatar	United States of America	Cannes
0	25683	Avatar	United Kingdom	Cannes
1	14531	Pirates of the Caribbean: At World's End	United States of America	Cannes
2	18260	Spectre	United Kingdom	Berlin
2	18260	Spectre	United States of America	Berlin
...
497	57476	Red Dragon	Germany	Berlin
497	57476	Red Dragon	United States of America	Berlin
498	55309	Hidalgo	United States of America	Venice
498	55309	Hidalgo	Morocco	Venice
499	72551	Jack and Jill	United States of America	Berlin

765 rows × 4 columns

```
In [328]: df.to_csv("1NF_festiveFilm.csv")
```

Example 3. In the below example, the “company’s” column is having multiple values and it is violating 1NF. So we have implemented 1NF and now each entry has a separate company name.

Before 1NF:

```
In [332]: df = pd.read_csv("movie.csv")
In [333]: df
```

	movie_id	movie_title	released_date	about	streaming_on	subscription	language	no.of_reviews	ratings	award_name	true_story	companies
0	25683	Avatar	2009-12-10	In the 22nd century, a paraplegic Marine is di...	hulu	yes	English	11800	7.2	True	False	Ingenious Film Partners
1	14531	Pirates of the Caribbean: At World's End	2007-05-19	Captain Barbosa, long believed to be dead, ha...	hulu	yes	English	4500	6.9	False	False	Walt Disney Pictures Jerry Bruckheimer Fil...
2	18260	Spectre	2015-10-26	A cryptic message from Bond's past sends him o...	hulu	yes	English	4466	6.3	False	False	Columbia Pictures Danjaq B
3	25056	The Dark Knight Rises	2012-07-16	Following the death of District Attorney Harve...	netflix	yes	English	9106	7.6	True	False	Legendary Pictures Warner Bros. DC Ente...
4	99197	John Carter	2012-03-07	John Carter is a war-weary, former military ca...	hulu	yes	English	2124	6.1	False	False	Walt Disney Pictures
...
495	75730	Journey 2: The Mysterious Island	2012-01-19	Sean Anderson partners with his mom's boyfriend...	hulu	yes	English	1030	5.8	False	False	New Line Cinema Contra...
496	78945	Cloudy with a Chance of Meatballs 2	2013-09-26	After the disastrous food storm in the first f...	hulu	yes	English	915	6.4	False	False	Columbia Pictures Sony

After 1NF:

```
In [342]: df['companies']=df['companies'].str.split( " ")
df.explode(['companies'])
```

	movie_id	movie_title	released_date	about	streaming_on	subscription	language	no.of_reviews	ratings	award_name	true_story	companies
0	25683	Avatar	2009-12-10	In the 22nd century, a paraplegic Marine is di...	hulu	yes	English	11800	7.2	True	False	Ingenious Film Partners
0	25683	Avatar	2009-12-10	In the 22nd century, a paraplegic Marine is di...	hulu	yes	English	11800	7.2	True	False	Twentieth Century Fox Film Corporation
0	25683	Avatar	2009-12-10	In the 22nd century, a paraplegic Marine is di...	hulu	yes	English	11800	7.2	True	False	Dune Entertainment
0	25683	Avatar	2009-12-10	In the 22nd century, a paraplegic Marine is di...	hulu	yes	English	11800	7.2	True	False	Lightstorm Entertainment
1	14531	Pirates of the Caribbean: At World's End	2007-05-19	Captain Barbosa, long believed to be dead, ha...	hulu	yes	English	4500	6.9	False	False	Walt Disney Pictures
...
498	55309	Hidalgo	2004-03-05	Set in 1890, this is the story of a Pony Expr...	netflix	yes	English	318	6.5	False	False	Touchstone Pictures
498	55309	Hidalgo	2004-03-05	Set in 1890, this is the story of a Pony Expr...	netflix	yes	English	318	6.5	False	False	Dune Films
				Jack Sadelstein,								

Second Normal Form (2NF)

- Each table has a primary key with minimal attributes that can uniquely identify a record.
- The values in each column of a table are atomic.
- There are no repeating groups in the same table.
- There are no Partial dependencies or Calculated data.

A relation is 2NF when it is in the First Normal Form but has no non-prime attribute functionality. There should be no partial dependencies or Calculated data in a relation. So here the country and the festival have dependencies so if in the future a festival wants to shift to another country so the changes should only be made in the country column only.

Before 2NF:

The screenshot shows a Jupyter Notebook interface with the following content:

- In [344]:** `df = pd.read_csv("countryFest.csv")`
- In [345]:** `df`
- Out[345]:** A table showing movie data with columns: movie_id, movie_title, country, festival. The data includes rows for Avatar, Pirates of the Caribbean: At World's End, Spectre, The Dark Knight Rises, John Carter, Journey 2: The Mysterious Island, Cloudy with a Chance of Meatballs 2, Red Dragon, Hidalgo, and Jack and Jill.
- In [358]:** `df1=df[['movie_id','movie_title','country']]
df2=df[['country','festival']]`
- In [359]:** `print(f"Unique FN: {df['country'].unique()}")`
Output: Unique FN: ['United States of America' 'Germany' 'Spain' 'Italy' 'Australia' 'Canada']
- In [376]:** `df4=(f"Unique Values from 2 Columns:\n{pd.concat([df['country'],df['festival']]).unique()}")`

After 2NF:

jupyter Ass 3-Copy1 Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel) | Logout

```
In [358]: df1=df[['movie_id','movie_title','country']]
df2=df[['country','festival']]

In [359]: print(f"Unique FN: {df['country'].unique()}")
Unique FN: ['United States of America' 'Germany' 'Spain' 'Italy' 'Australia' 'Canada']

In [376]: df4=(f"Unique Values from 2 Columns:\n{pd.concat([df['country'],df['festival']]).unique()}"
```

In [377]: df4

Out[377]: "Unique Values from 2 Columns:['United States of America' 'Germany' 'Spain' 'Italy' 'Australia' 'Canada'\n 'Cannes' 'Berlin' 'Venice' 'Sundance' 'Rotterdam' 'Toronto']"

```
In [378]: df1
```

Out[378]:

	movie_id	movie_title	country
0	25683	Avatar	United States of America
1	14531	Pirates of the Caribbean: At World's End	United States of America
2	18260	Spectre	Germany
3	25056	The Dark Knight Rises	Spain
4	99197	John Carter	United States of America
...
495	75730	Journey 2: The Mysterious Island	Spain
496	78945	Cloudy with a Chance of Meatballs 2	United States of America
497	57476	Red Dragon	Germany
498	55309	Hidalgo	Spain

After 2NF:

jupyter Ass 3-Copy1 Last Checkpoint: an hour ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3 (ipykernel) | Logout

```
... ... ... ...
495 75730 Journey 2: The Mysterious Island Spain
496 78945 Cloudy with a Chance of Meatballs 2 United States of America
497 57476 Red Dragon Germany
498 55309 Hidalgo Spain
499 72551 Jack and Jill Germany
```

500 rows x 3 columns

```
In [379]: df.to_csv("2NF_Table1.csv")
In [380]: df = pd.read_csv("2NF_Table2.csv")
In [381]: df
```

Out[381]:

	country	festival
0	United States of America	Cannes
1	Germany	Berlin
2	Spain	Venice
3	Italy	Sundance
4	Australia	Rotterdam
5	Canada	Toronto

In []:

Third Normal Form (3NF)

- Each table has a primary key with minimal attributes that can uniquely identify a record.
- The values in each column of a table are atomic.
- There are no repeating groups in the same table.
- There are no Partial dependencies or Calculated data.
- Eliminate fields that do not directly depend on the primary key; that is no transitive dependencies

3NF is a relational database that uses normalizing principles to reduce the duplication of data, avoid data anomalies, ensure referential integrity, and simplify data management. In 2NF the attributes should be functionally dependent but in 3NF the attributes should be directly (non-transitively) dependent. In 3NF A is dependent on B and B is dependent on C like here genre is dependent on movie and age restriction is dependent on genre. So we have made 2 tables that separate genre and age_restriction columns. Thus, we eliminated transitive dependency and now our data is in 3NF.

Before 3NF:

The screenshot shows a Jupyter Notebook interface with the title "jupyter Ass 3-Copy1 Last Checkpoint: 14 minutes ago (autosaved)". The menu bar includes File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. The toolbar includes icons for file operations like Open, Save, and Run. The status bar indicates "Not Trusted" and "Python 3 (ipykernel)".

In [384]: `df = pd.read_csv("restriction.csv")`

In [385]: `df`

Out[385]:

	movie_id	movie_title	genre	age_restriction
0	25683	Avatar	Action	PG13
1	14531	Pirates of the Caribbean: At World's End	Action	PG13
2	18260	Spectre	Action	PG13
3	25056	The Dark Knight Rises	Action	PG13
4	99197	John Carter	Action	PG13
...
495	75730	Journey 2: The Mysterious Island	Family	PG
496	78945	Cloudy with a Chance of Meatballs 2	Action	PG13
497	57476	Red Dragon	Action	PG13
498	55309	Hidalgo	Action	PG13
499	72551	Jack and Jill	Action	PG13

500 rows x 4 columns

In [386]: `df1=df[['movie_id','movie_title','genre']]
df2=df[['genre','age_restriction']]`

In [387]: `df1`

Out[387]:

	movie_id	movie_title	genre
0	25683	Avatar	Action

After 3NF:

```
In [387]: df1
Out[387]:
      movie_id          movie_title   genre
0       25683             Avatar   Action
1     14531  Pirates of the Caribbean: At World's End   Action
2       18260            Spectre   Action
3       25056  The Dark Knight Rises   Action
4       99197        John Carter   Action
...     ...
495    75730  Journey 2: The Mysterious Island   Family
496    78945  Cloudy with a Chance of Meatballs 2   Action
497    57476           Red Dragon   Action
498    55309           Hidalgo   Action
499    72551         Jack and Jill   Action
500 rows × 3 columns

In [389]: print(f"Unique FN: {df['genre'].unique()}")
Unique FN: ['Action' 'Family' 'Romance']

In [390]: df4=(f"Unique Values from 2 Columns:\n{pd.concat([df['genre'],df['age_restriction']]).unique()}")
In [391]: df4
Out[391]: "Unique Values from 2 Columns:['Action' 'Family' 'Romance' 'PG13' 'PG' 'R']"
```

After 3NF:

```
In [389]: print(f"Unique FN: {df['genre'].unique()}")
Unique FN: ['Action' 'Family' 'Romance']

In [390]: df4=(f"Unique Values from 2 Columns:\n{pd.concat([df['genre'],df['age_restriction']]).unique()}")
In [391]: df4
Out[391]: "Unique Values from 2 Columns:['Action' 'Family' 'Romance' 'PG13' 'PG' 'R']"

In [392]: df1.to_csv("3NF_Table1.csv")
In [393]: df = pd.read_csv("3NF_Table2.csv")
In [394]: df
Out[394]:
      genre age_restriction
0   Action        PG13
1   Family         PG
2 Romance          R

In [ ]:
```

CREATING VIEWS

PREVIOUS ASSIGNMENTS USE CASES

(Use-cases Views 5*3= 15 Views)

5 USE-CASES (SMITI AGRAWAL)

1.What are the family movies that are released during the Cannes film festival ?

```
CREATE VIEW [familyfest] AS select f.movie_title,g.genre,f.festival from restriction  
g,festiveFilm f where g.movie_id=f.movie_id and festival="Cannes" and genre like  
"%Family%"
```

2.What are the top 10 movies having higher profits?

```
CREATE VIEW [profitablemovies] AS select e.movie_title,(e.gross-b.budget) as profit  
from earnings e, budget b where e.movie_id=b.movie_id order by profit desc limit 10
```

3.Which director's movie has earned maximum money ?

```
CREATE VIEW [directorearning] AS select distinct d.director,sum(e.gross) as total_gross  
from director d, earnings e  
where d.movie_id=e.movie_id  
group by d.director  
order by e.gross desc limit 1
```

4.Highest rated movies released in the year 2000, USA?

```
CREATE VIEW [topmovie2000] AS select m.movie_title,m.release_date, f.country,  
m.ratings  
from movie m,festiveFilm f  
where m.movie_id=f.movie_id and ratings>7 and m.release_date like "%2000%"  
and f.country like "%United States of America%"  
order by ratings desc
```

5.What are real story based movies that have earned the highest votes?

```
CREATE VIEW [realstory] AS select distinct(m.movie_title), e.votes  
from movie m,earnings e where e.movie_id=m.movie_id and m.biograpgy="1"  
order by votes desc
```

OUTPUTS-

1.What are the family movies that are released during the Cannes film festival ?

```

5      #What are the family movies that are released during the Cannes film festival ?
6      CREATE VIEW [familyfest] AS select f.movie_title,g.genre,f.festival from restriction g,festiveFilm f where g.movie_id=f.movie_id and festival="Cannes" and genre like "%Family%";
7
8
9
10     SELECT * FROM familyfest;
11

```

	movie_title	genre	festival
1	Tangled	Family	Cannes
2	Monsters University	Family	Cannes
3	Cars 2	Family	Cannes
4	Toy Story 3	Family	Cannes
5	Jack the Giant Slayer	Family	Cannes
6	The Good Dinosaur	Family	Cannes
7	The Chronicles of Narnia: The Lion, the Witch and the ...	Family	Cannes
8	Hugo	Family	Cannes
9	Evan Almighty	Family	Cannes
10	Inside Out	Family	Cannes
11	Maleficent	Family	Cannes
12	Big Hero 6	Family	Cannes
13	Wreck-It Ralph	Family	Cannes
14	How to Train Your Dragon	Family	Cannes
15	Shrek the Third	Family	Cannes
16	Ratatouille	Family	Cannes

Execution finished without errors.
Result: 63 rows returned in 10ms
At line 10:
SELECT * FROM familyfest;

2.What are the top 10 movies having higher profits?

```

1      #What are the top 10 movies having higher profits?
2      CREATE VIEW [profitablemovies] AS select e.movie_title,(e.gross-b.budget) as profit from earnings e, budget b where e.movie_id=b.movie_id order by profit desc limit 10;
3
4      SELECT * FROM profitablemovies;

```

	movie_title	profit
1	Avatar	2550965087
2	Titanic	1645034188
3	Jurassic World	1363528810
4	Furious 7	1316249360
5	The Avengers	1299557910
6	Avengers: Age of Ultron	1125403694
7	Frozen	1124219009
8	The Lord of the Rings: The Return of the King	1024888979
9	Iron Man 3	1015439994
10	Transformers: Dark of the Moon	928746996

Execution finished without errors.
Result: 10 rows returned in 13ms
At line 4:
SELECT * FROM profitablemovies;

3.Which director's movie has earned maximum money ?

```
6  
7      #Which directors movie has earned the maximum money ?  
8      CREATE VIEW [directorearning] AS select distinct d.director,sum(e.gross) as total_gross from director d,earnings e  
9      where d.movie_id=e.movie_id  
10     group by d.director  
11     order by e.gross desc limit 1;  
12  
13  
14      SELECT * FROM directorearning;
```

director	total_gross
James cameron	2787965087

Execution finished without errors.
Result: 1 rows returned in 20ms
At line 14:
SELECT * FROM directorearning;

4.Highest rated movies released in the year 2000, USA?

```
1  
2  
3      #Highest rated movies released in year 2000, USA?  
4      CREATE VIEW [topmovie2000] AS select m.movie_title,m.release_date,f.country, m.ratings  
5      from movie m,festiveFilm f  
6      where m.movie_id=f.movie_id and ratings>7 and m.release_date like "%2000%" and f.country like "%United States of America%"  
7      order by ratings desc;  
8  
9      SELECT * FROM topmovie2000;
```

	movie_title	ratings
1	Avatar	7.2
2	Pirates of the Caribbean: At World's End	6.9
3	Spectre	6.3
4	The Dark Knight Rises	7.6
5	John Carter	6.1
6	Spider-Man 3	5.9
7	Tangled	7.4
8	Avengers: Age of Ultron	7.3
9	Harry Potter and the Half-Blood Prince	7.4
10	Batman v Superman: Dawn of Justice	5.7
11	Superman Returns	5.4
12	Quantum of Solace	6.1
13	Pirates of the Caribbean: Dead Man's Chest	7
14	The Lone Ranger	5.9
15	Man of Steel	6.5
16	The Chronicles of Narnia: Prince Caspian	6.3

Execution finished without errors.
Result: 500 rows returned in 6ms
At line 12:

5.What are real story based movies that have earned the highest votes?

```
1 #What are real story based movies that has earned highest votes?
2
3  CREATE VIEW [realstory] AS select distinct(m.movie_title), e.votes
4     from movie m,earnings e where e.movie_id=m.movie_id and m.biograpgy="FALSE"
5         order by votes desc
6
7 SELECT * from realstory;
8
9
```

	movie_title	votes
1	Inception	13752
2	The Dark Knight	12002
3	Avatar	11800
4	The Avengers	11776
5	Interstellar	10867
6	Django Unchained	10099
7	Guardians of the Galaxy	9742
8	The Hunger Games	9455
9	Mad Max: Fury Road	9427
10	The Dark Knight Rises	9106
11	Iron Man 3	8806
12	Iron Man	8776
13	The Lord of the Rings: The Fellowship of the Ring	8705
14	Jurassic World	8662
15	The Hobbit: An Unexpected Journey	8297
16	The Lord of the Rings: The Return of the King	8064

Execution finished without errors.
Result: 500 rows returned in 27ms
At line 1:
select distinct(m.movie_title), e.votes
from movie m,earnings e where e.movie_id=m.movie_id and m.biograpgy="FALSE"
order by votes desc

5 USE-CASES (SHREYAS RAI)

1.What are top rated movies for children above age 13?

```
CREATE VIEW [parentsguide] AS select g.movie_title, g.certification, r.rating
    from genre g, ratings r
    where g.movie_id=r.movie_id and g.certification = "PG13"
        order by rating desc limit 20
```

2.List award winning movies for Action packed

```
CREATE VIEW [actionpacked] AS select m.movie_title, g.genre, m.ratings
    from genre g, movie m
    where g.movie_id=m.movie_id
        and m.award LIKE "%TRUE%"
        and g.genre like "%Action%"
        order by m.ratings desc
```

3.List the movies which has less ratings but earned more

```
CREATE VIEW [underrated] AS select e.movie_title
    from earnings e, ratings r
    where e.movie_id=r.movie_id
```

and r.rating<5
order by e.gross desc

4. Movies which were superhit in more than one country

```
CREATE VIEW [superhit] AS select r.movie_title as superhit_movies, f.country
from ratings r, festiveFilm f
where r.rating>7.1
group by r.movie_title limit 10
```

5. Cast and directors with maximum hit movies

```
CREATE VIEW [bestdirector] AS select c.movie_title as Movie, d.director, c.cast
from cast c, director d, ratings r
where c.movie_id=d.movie_id
and d.movie_id=r.movie_id
and r.rating>8
order by r.rating desc
```

OUTPUT-

1. What are top rated movies for children above age 13?

```
9
0
1#Top rated movies for children above age 13
2CREATE VIEW [parentsguide] AS select g.movie_title, g.certification, r.rating
3    from genre g, ratings r
4    where g.movie_id=r.movie_id and g.certification = "PG13"
5    order by rating desc limit 20
6
7SELECT * from parentsguide;
```

movie_title	certification	rating
The Dark Knight	PG13	8.2
Interstellar	PG13	8.1
Inception	PG13	8.1
The Lord of the Rings: The Return of the King	PG13	8.1
The Lord of the Rings: The Fellowship of the Ring	PG13	8
The Lord of the Rings: The Two Towers	PG13	8
Guardians of the Galaxy	PG13	7.9
Gladiator	PG13	7.9
The Wolf of Wall Street	PG13	7.9
The Departed	PG13	7.9
Shutter Island	PG13	7.8
Terminator 2: Judgment Day	PG13	7.7
The Dark Knight Rises	PG13	7.6
The Hobbit: The Desolation of Smaug	PG13	7.6
Edge of Tomorrow	PG13	7.6
Captain America: The Winter Soldier	PG13	7.6

Execution finished without errors.
Result: 20 rows returned in 18ms
at line 17:
SELECT * from parentsguide;

2. List award winning movies for Action packed

```

19      #List award winning movies for Action packed
20
21
22  ┌─ CREATE VIEW [actionpacked] AS select m.movie_title, g.genre, m.ratings
23  │   from genre g, movie m
24  │   where g.movie_id=m.movie_id
25  │   and m.award LIKE "%TRUE%"
26  │   and g.genre like "%Action%"
27  │   order by m.ratings desc
28
29  ┌─ SELECT * from actionpacked;
30

```

	movie_title	genre	ratings
1	Inception	Action Thriller Science Fiction Mystery Adventure	8.1
2	The Lord of the Rings: The Return of the King	Adventure Fantasy Action	8.1
3	The Lord of the Rings: The Fellowship of the Ring	Adventure Fantasy Action	8
4	The Lord of the Rings: The Two Towers	Adventure Fantasy Action	8
5	Gladiator	Action Drama Adventure	7.9
6	Terminator 2: Judgment Day	Action Thriller Science Fiction	7.7
7	The Dark Knight Rises	Action Crime Drama Thriller	7.6
8	Edge of Tomorrow	Action Science Fiction	7.6
9	X-Men: Days of Future Past	Action Adventure Fantasy Science Fiction	7.5
10	Batman Begins	Action Crime Drama	7.5
11	The Avengers	Science Fiction Action Adventure	7.4
12	Star Trek Into Darkness	Action Adventure Science Fiction	7.4
13	The Hunger Games: Catching Fire	Adventure Action Science Fiction	7.4
14	Avatar	Action Adventure Fantasy Science Fiction	7.2

Execution finished without errors.
Result: 14 rows returned in 10ms
At line 29:
SELECT * from actionpacked;

3.List the movies which has less ratings but earned more

```

32      #Movies which has less ratings but earned more
33
34
35  ┌─ CREATE VIEW [underrated] AS select e.movie_title
36  │   from earnings e, ratings r
37  │   where e.movie_id=r.movie_id
38  │   and r.rating<6
39  │   order by e.gross desc
40
41  ┌─ select * from underrated;
42
43
44
45

```

	movie_title
1	Independence Day: Resurgence
2	The Last Airbender
3	Eragon
4	Batman & Robin
5	Gulliver's Travels
6	Fantastic Four
7	Speed 2: Cruise Control
8	Jack and Jill
9	Nutty Professor II: The Klumps
10	Cats & Dogs 2 : The Revenge of Kitty Galore
11	Catwoman
12	Stealth
13	How Do You Know
14	Battlefield Earth
15	Town & Country
16	The Adventures of Pluto Nash

Execution finished without errors.
Result: 20 rows returned in 5ms
At line 41:
select * from underrated;

4.Movies which were superhit in more than one country

```

34
35      #Movies which were superhit in more than one country
36      CREATE VIEW [superhit] AS select r.movie_title as superhit_movies, f.country
37          from ratings r, festiveFilm f
38          where r.rating>7.1
39          group by r.movie_title limit 10
40
41      select * from superhit;
42
43

```

	superhit_movies	country
1	A Beautiful Mind	United States of America United Kingdom
2	All That Jazz	United States of America United Kingdom
3	American Gangster	United States of America United Kingdom
4	Avatar	United States of America United Kingdom
5	Avengers: Age of Ultron	United States of America United Kingdom
6	Batman Begins	United States of America United Kingdom
7	Big Hero 6	United States of America United Kingdom
8	Black Hawk Down	United States of America United Kingdom
9	Blood Diamond	United States of America United Kingdom
10	Captain America: The Winter Soldier	United States of America United Kingdom

Execution finished without errors.
Result: 10 rows returned in 54ms
At line 41:
select * from superhit;

5.Cast and directors with maximum hit movies

```

42
43      #Cast and directors with maximum hit movies
44
45      CREATE VIEW [bestdirector] AS select c.movie_title as Movie, d.director, c.cast
46          from cast c, director d, ratings r
47          where c.movie_id=d.movie_id
48          and d.movie_id=r.movie_id
49          and r.rating>8
50          order by r.rating desc
51
52      select * from bestdirector;
53

```

	Movie	director	cast
1	The Dark Knight	Roberto Benigni	Bruce Wayne fecafdf Christian Bale Joker fefcafcbf ...
2	Interstellar	Makoto Shinkai	Joseph Cooper febbfceb Matthew McConaughey ...
3	Inception	Nitesh Tiwari	Dom Cobb fecaeade Leonardo DiCaprio Arthur ...
4	The Lord of the Rings: The Return of the King	Darius Marder	Frodo Baggins febcabb Elijah Wood Gandalf the Whi...

Execution finished without errors.
Result: 4 rows returned in 28ms
At line 52:
select * from bestdirector;

5 USE-CASES (SARTHAK SHRIVASTAV)

1.What are the best movies available on Hulu?

```
CREATE VIEW [tophulu] AS select m.movie_title  
    from review r, movie m  
    where r.movie_title=m.movie_title and m.streaming_on like "%hulu%" and r.review  
like "%Best%"
```

2.Which directors has age restrictions as PG?

```
CREATE VIEW [directorrestriction] AS select d.movie_title,d.director  
    from director d , genre g where d.movie_id=g.movie_id and g.certification = "PG"
```

3.What movies of directors are in the Berlin Festival?

```
CREATE VIEW [dirberlin] AS select d.movie_title, d.director, f.festival  
    from director d , festiveFilm f  
    where d.movie_id =f.movie_id and festival like "%Berlin%"
```

4. Which actor were active in which year?

```
CREATE VIEW [activeactor] AS select m.movie_title, c.cast, m.release_date  
    from movie m, cast c  
    where m.movie_id=c.movie_id and c.cast like "%Robert Downey Jr.%"
```

5.Which actor have done a Romantic Movie?

```
CREATE VIEW [romantic] AS select c.cast from cast c,genre g where  
c.movie_id=g.movie_id and genre like "%Romance"
```

OUTPUT-

1.What are the best movies available on Hulu?

```
54 #what are the Best movies available on Hulu?
55  CREATE VIEW [tophulu] AS select m.movie_title
56     from review r, movie m
57     where r.movie_title=m.movie_title and m.streaming_on like "%hulu%" and r.review like "%Best%"
58
59
60
61 select * from tophulu;
```

movie_title
1 Avatar
2 The Hobbit: The Desolation of Smaug
3 The Hobbit: The Desolation of Smaug
4 X-Men: Days of Future Past
5 Star Trek Into Darkness
6 The Great Gatsby
7 The Dark Knight
8 Edge of Tomorrow
9 Maleficent
10 Dawn of the Planet of the Apes
11 Captain America: The Winter Soldier
12 How to Train Your Dragon
13 Guardians of the Galaxy
14 The Curious Case of Benjamin Button
15 X-Men: First Class
16 Ratatouille

Execution finished without errors.
Result: 16 rows returned in 9ms
At line 61:
select * from tophulu;

2.Which directors has age restrictions as PG?

```
63
64 #which directors has age restriction as PG
65
66  CREATE VIEW [directorrestriction] AS select d.movie_title,d.director
67     from director d , genre g where d.movie_id=g.movie_id and g.certification = "PG"
68
69
70 select * from directorrestriction;
```

movie_title	director
1 Tangled	Nathan Greno
2 Harry Potter and the Half-Blood Prince	David Yates
3 The Chronicles of Narnia: Prince Caspian	Andrew Adamson
4 Monsters University	Dan Scanlon
5 Oz: The Great and Powerful	Sam Raimi
6 Cars 2	Francis Ford Coppola
7 Toy Story 3	Francis Ford Coppola
8 Jack the Giant Slayer	David Fincher
9 The Good Dinosaur	Martin Scorsese
10 Brave	Irvin Kershner
11 WALL-E	Thomas Kail
12 The Chronicles of Narnia: The Lion, the Witch and the ...	Steven Spielberg
13 Up	David Fincher
14 Hugo	Masaki Kobayashi
15 Evan Almighty	Damien Chazelle
16 Inside Out	Roman Polanski

Execution finished without errors.
Result: 16 rows returned in 13ms
At line 70:
select * from directorrestriction;

3.What movies of directors are in the Berlin Festival?

```
73  
74  
75 #What movies of directors are in Berlin Festival?  
76  
77 ┌─ CREATE VIEW [dirberlin] AS select d.movie_title, d.director, f.festival  
78   from director d, festiveFilm f  
79   where d.movie_id =f.movie_id and festival like "%Berlin%"  
80  
81 └─ select * from dirberlin;  
82
```

	movie_title	director	festival
1	Spectre	Sam mendes	Berlin
2	Harry Potter and the Half-Blood Prince	David Yates	Berlin
3	Quantum of Solace	Marc Forster	Berlin
4	Robin Hood	Otto Bathurst	Berlin
5	King Kong	Peter Jackson	Berlin
6	Skyfall	Sam Mendes	Berlin
7	X-Men: Days of Future Past	Steven Spielberg	Berlin
8	Up	David Fincher	Berlin
9	Monsters vs Aliens	Jonathan Demme	Berlin
10	Iron Man	George Lucas	Berlin
11	G.I. Joe: The Rise of Cobra	Martin Scorsese	Berlin
12	The Jungle Book	Ridley Scott	Berlin
13	Terminator 3: Rise of the Machines	Can Ulkay	Berlin
14	Inception	Nitesh Tiwari	Berlin
15	Alice Through the Looking Glass	Andrew Stanton	Berlin
16	Alexander	Stanley Kubrick	Berlin

Execution finished without errors.
Result: 49 rows returned in 12ms
At line 81:
select * from dirberlin;

4. Which actor were active in which year?

```
83  
84  
85 #Which actor were active in which year?  
86  
87 ┌─ CREATE VIEW [activeactor] AS select m.movie_title, c.cast, m.release_date  
88   from movie m, cast c  
89   where m.movie_id=c.movie_id and c.cast like "%Robert Downey Jr%"  
90  
91  
92 └─ select * from activeactor;  
93  
94
```

	movie_title	cast	release_date
1	Avengers: Age of Ultron	Tony Stark / Iron Man edcde Robert Downey Jr. Tho...	4/22/15
2	The Avengers	Tony Stark / Iron Man fecaeb Robert Downey Jr. ...	4/25/12
3	Captain America: Civil War	Steve Rogers / Captain America ecbeadb Chris Evans ...	4/27/16
4	Iron Man 3	Tony Stark / Iron Man fecaedff Robert Downey Jr. ...	4/18/13
5	Iron Man	Tony Stark / Iron Man fecafe Robert Downey Jr. Lt...	4/30/08
6	Iron Man 2	Tony Stark / Iron Man fec Robert Downey Jr. Virgin...	4/28/10
7	The Incredible Hulk	Bruce Banner / The Hulk fecafc Edward Norton Bett...	6/12/08
8	Sherlock Holmes: A Game of Shadows	Sherlock Holmes fecafba Robert Downey Jr. Dr. Jo...	11/22/11
9	Tropic Thunder	Tugg Speedman fedcae Ben Stiller Jeff Portnoy ...	8/9/08
10	Sherlock Holmes	Sherlock Holmes fecef Robert Downey Jr. Dr. John ...	12/23/09
11	Zodiac	Robert Graysmith fecafddd Jake Gyllenhaal Paul ...	3/2/07

Execution finished without errors.
Result: 11 rows returned in 11ms
At line 92:
select * from activeactor;

5.Which actor have done a Romantic Movie?

```
91
92
93
94
95 #Which actor have done a Romantic Movie?
96 CREATE VIEW [romantic] AS select c.cast from cast c,genre g where c.movie_id=g.movie_id and genre like "%Romance"
97
98 select * from romantic;
99
```

	cast		
1	Jay Gatsby feacea	Leonardo DiCaprio	Nick ...
2	Prince Dastan fecafbe	Jake Gyllenhaal	Tamina ...
3	Maleficent feacafbc	Angelina Jolie	Princess Auro...
4	James Stewart / Jay Fennel feacea	Josh Hartnett	...
5	Daisy fecabf	Cate Blanchett	Benjamin Button ...
6	Alexander fecafed	Colin Farrell	Olympias fecafef ...
7	Isabella 'Bella' Swan feccaff	Kristen Stewart	Edwa...
8	Lisa Jorgenson fecafe	Reese Witherspoon	George...
9	Porter Stoddard fecae	Warren Beatty	Ellie Stoddard...
10	Jack Byrnes feaca	Robert De Niro	Greg Focker fea...
11	Frank Tupelo fec	Johnny Depp	Elise Clifton-Ward ...
12	Carrie Bradshaw feecc	Sarah Jessica Parker	...
13	Giselle feccaff	Amy Adams	Robert Philip feccaffb ...
14	Amanda Woods fecaf	Cameron Diaz	Iris Simpkins ...
15	Jane Adler fecaedc	Meryl Streep	Jake Adler fecae...
16	Savuri fecacf	Zhang Ziyi	Hatsumomo fecacf...

Execution finished without errors.
Result: 25 rows returned in 19ms
At line 97:
select * from romantic;

Creating Stored Procedure For Some Use Cases:

1. Find top rated family films released in Cannes Festival

```
USE `moviedb`;
DROP procedure IF EXISTS `cannesFilms`;

DELIMITER $$
USE `moviedb`$$
CREATE PROCEDURE `cannesFilms` ()
BEGIN
select g.movie_title,g.genre,f.festival from genre g,festiveFilm f where
g.movie_id=f.movie_id and festival="Cannes" and genre like "%family%";

SELECT r.movie_title, r.rating , f.festival from ratings r, festiveFilm f where
r.movie_id=f.movie_id AND festival="Cannes" ORDER BY rating desc;

END$$
```

DELIMITER ;

Output:

The screenshot shows two MySQL Workbench result grids side-by-side. Both grids have the following structure:

movie_title	genre	festival
Tangled	Animation Family	Cannes
Alice in Wonderland	Family Fantasy Adventure	Cannes
Monsters University	Animation Family	Cannes
Cars 2	Animation Family Adventure Comedy	Cannes
Toy Story 3	Animation Family Comedy	Cannes
Jack the Giant Slayer	Action Family Fantasy	Cannes
The Good Dinosaur	Adventure Animation Family	Cannes
The Chronicles of Narnia: The Lion, the Witch a...	Adventure Family Fantasy	Cannes
Hugo	Adventure Drama Family	Cannes
Evan Almighty	Fantasy Comedy Family	Cannes
Inside Out	Drama Comedy Animation Family	Cannes
Maleficent	Fantasy Adventure Action Family...	Cannes
Big Hero 6	Adventure Family Animation Action...	Cannes
Wreck-It Ralph	Family Animation Comedy Adventure	Cannes
How to Train Your Dragon	Fantasy Adventure Animation Family	Cannes
Shrek the Third	Fantasy Adventure Animation Com...	Cannes
Ratatouille	Animation Comedy Family Fantasy	Cannes
Madagascar: Escape 2 Africa	Family Animation	Cannes

The right grid has an additional column 'rating' and a different set of data:

movie_title	rating	festival
The Dark Knight	8	Cannes
Inside Out	8	Cannes
Inside Out	8	Cannes
The Lord of the Rings: The Return of the King	8	Cannes
The Lord of the Rings: The Two Towers	8	Cannes
Déjà Vu	8	Cannes
The Lion King	8	Cannes
Avatar	7	Cannes
Avatar	7	Cannes
Tangled	7	Cannes
Tangled	7	Cannes
Pirates of the Caribbean: Dead Man's Chest	7	Cannes
Pirates of the Caribbean: Dead Man's Chest	7	Cannes
Pirates of the Caribbean: Dead Man's Chest	7	Cannes
The Avengers	7	Cannes

2. What are the top 10 movies having higher profits?

```
USE `moviedb`;
DROP procedure IF EXISTS `top10RatedEarningMovies`;
```

```
DELIMITER $$
USE `moviedb`$$
CREATE PROCEDURE `top10RatedEarningMovies` ()
BEGIN
    select e.movie_title,(e.gross-b.budget) as profit from earnings e, budget b where
e.movie_id=b.movie_id order by profit desc limit 10;
END$$
```

```
DELIMITER ;
```

Output:

The screenshot shows the MySQL Workbench interface. The left sidebar under 'SCHEMAS' for 'root (moviedb)' lists various database objects. The 'Stored Procedures' section is expanded, showing procedures like 'cannesFilms', 'mostEarnedDirector', 'top10EarningMovies', and 'tvshowStarCast'. The 'top10EarningMovies' procedure is currently selected. The main pane contains a query editor with the SQL command 'CALL top10EarningMovies();' and a result grid displaying the output of this call. The result grid has columns 'movie_title' and 'profit', listing movies and their earnings.

movie_title	profit
Avatar	2550965087
Titanic	1645034188
Jurassic World	1363528810
Furious 7	1316249360
The Avengers	1299557910
Avengers: Age of Ultron	1125403694
Frozen	1124219009
The Lord of the Rings: The Return of the King	1024888979
Iron Man 3	1015439994
Transformers: Dark of the Moon	928746996

3. Who are involved in the cast and crew of the particular show?

```
USE `moviedb`;
DROP procedure IF EXISTS `tvShowStarCast`;
```

```
DELIMITER $$
USE `moviedb`$$
CREATE PROCEDURE `tvShowStarCast` ()
BEGIN
select tvshows,star_cast from tvshow;
END$$
```

```
DELIMITER ;
```

SQL SQL i Schemas Query 1 tvshowStarCast - Routine cannesFilms - Routine

SCHEMAS

Filter objects

- > Foreign Keys
- > Triggers
- > movie
- > ratings
 - Columns
 - movie_id
 - movie_title
 - rating
 - Indexes
- > Foreign Keys
- > Triggers
- > review
- > scrape1
- > tvshow
- Views
- Stored Procedures
 - cannesFilms
 - mostEarnedDirector
 - top10RatedEarning...
 - tvshowStarCast
- Functions

Object Info Session

Procedure: tvshowStarCast

```

1
2 • CALL tvshowStarCast();
3

```

100% 23:2

Result Grid Filter Rows: Search Export:

tvshows	star_cast
Planet Earth II	David Attenborough, Chadden Hunter
Breaking Bad	Bryan Cranston, Aaron Paul
Planet Earth	Sigourney Weaver, David Attenborough
Band of Brothers	Scott Grimes, Damian Lewis
Chernobyl	Jessie Buckley, Jared Harris
The Wire	Dominic West, Lance Reddick
Blue Planet II	David Attenborough, Peter Drost
Avatar: The Last Airbender	Dee Bradley Baker, Zach Tyler Eisen
Cosmos: A Spacetime Odyssey	Neil deGrasse Tyson, Christopher Emerson
The Sopranos	James Gandolfini, Lorraine Bracco
Cosmos	Carl Sagan, Jaromír Hanzlík
Our Planet	David Attenborough
Game of Thrones	Emilia Clarke, Peter Dinklage
Rick and Morty	Justin Roiland, Chris Parnell
The World at War	Laurence Olivier, Averell Harriman
Fullmetal Alchemist: Brotherh...	Kent Williams, Iemasa Kayumi
The Last Dance	Michael Jordan, Phil Jackson
Life	Oprah Winfrey, David Attenborough

4. What are the relevant shows for the given age group?

```
USE `moviedb`;
DROP procedure IF EXISTS `parentGuidedMovies`;

DELIMITER $$
USE `moviedb`$$
CREATE PROCEDURE `parentGuidedMovies` ()
BEGIN
select movie_title, certification from genre where certification="PG13" OR certification="PG";
END$$

DELIMITER ;
```

The screenshot shows the MySQL Workbench interface. On the left, the Schemas tree displays the `moviedb` schema with various tables, views, and procedures. The `parentGuidedMovies` procedure is highlighted in the 'Stored Procedures' section. In the center, the Query Editor window contains the SQL code for calling the procedure:

```
1
2 • CALL parentGuidedMovies();
3
```

The result grid below shows the output of the procedure, listing movie titles and their certifications:

movie_title	certificati...
Avatar	PG13
Pirates of the Caribbean: At World's End	PG13
Spectre	PG13
The Dark Knight Rises	PG13
John Carter	PG13
Spider-Man 3	PG13
Tangled	PG
Avengers: Age of Ultron	PG13
Harry Potter and the Half-Blood Prince	PG
Batman v Superman: Dawn of Justice	PG13
Superman Returns	PG13
Quantum of Solace	PG13
Pirates of the Caribbean: Dead Man's...	PG13
The Lone Ranger	PG13
Man of Steel	PG13
The Chronicles of Narnia: Prince Caspi...	PG
The Avengers	PG13
Pirates of the Caribbean: On Stranger...	PG13

CONCLUSIONS-

In this project we have achieved our goal to develop a system that recommends and answers queries related to TVShows and Movies. The tool which we have used is Jupyter Notebook for implementing python codes and SQLite3 Database. Before starting with the implementation of this project we designed thirty use cases that we would be covering in the project. We created the Entity Relationship diagram and Use case diagram to get the proper understanding of the project. On the basis of our use cases, we started gathering data by performing web scraping on various websites, like Twitter, IMDB. We also scanned through may data available on Kaggle. The next task we achieved was to perform the cleaning of data which we had gather through various platforms. While cleaning we had finalized our tables according to the use cases. After cleaning of data we started implementing the data normalization. We did this to avoid data redundancy and to achieve the data atomicity. Following this we had started writing Structured Query Language queries and Relational Algebra queries. After this we performed our use case queries and were able to achieve the desired results. Following this we had also used the matplotlib libraries to get the visualization of the queries. Visualisations helped in understanding the data more easily. Later, We created views and Stored procedures to reduce the time of writing the query repeatedly. Now our system is able to solve the user queries with the help of our database.