In [4]:

```
import json
import pandas as pd
import numpy as np

import re

from sqlalchemy import create_engine
import psycopg2

from config import db_password

import time

# 1. Create a function that takes in three arguments;
# Wikipedia data #Kaggle metadata and MovieLens rating data (from Kaggl
def movies_function():

    # 2. Read in the kaggle metadata and MovieLens ratings CSV files as
    kaggle_metadata = pd.read_csv ('movies_metadata.csv', low_memory =
    ratings = pd.read_csv('ratings.csv')

    kaggle_metadata_df = pd.DataFrame(kaggle_metadata)
    ratings_df = pd.DataFrame(ratings)

    # 3. Open the read the Wikipedia data JSON file.
    file_dir = "/Users/caroline/Documents/Data Boot Camp/Module 8/Movie
    with open(f'{file_dir}/wikipedia-movies.json', mode='r') as file:
        wiki_movies_raw = json.load(file)

    # 4. Read in the raw wiki movie data as a Pandas DataFrame.
    wiki_movies_df = pd.DataFrame(wiki_movies_raw)

    # 5. Return the three DataFrames
    return wiki_movies_df, kaggle_metadata, ratings

# 6 Create the path to your file directory and variables for the three
file_dir = "/Users/caroline/Documents/Data Boot Camp/Module 8/Movies-ET

# Wikipedia data
wiki_file = f'{file_dir}/wikipedia.movies.json'

# Kaggle metadata
kaggle_file = f'{file_dir}/movies_metadata.csv'

# MovieLens rating data.
ratings_file = f'{file_dir}/ratings.csv'

# 7. Set the three variables in Step 6 equal to the function created in
wiki_file, kaggle_file, ratings_file = movies_function()
```

In [5]:

```
# 8. Set the DataFrames from the return statement equal to the file nam
wiki_movies_df = wiki_file
kaggle_metadata = kaggle_file
ratings = ratings_file
```

In [6]:
```
# 9. Check the wiki_movies_df DataFrame.
wiki_movies_df.head()
```

Out[6]:

	url	year	imdb_link	tit
0	https://en.wikipedia.org/wiki/The_Adventures_o...	1990.0	https://www.imdb.com/title/tt0098987/	Th Adventure of Fo Fairlar
1	https://en.wikipedia.org/wiki/After_Dark,_My_S...	1990.0	https://www.imdb.com/title/tt0098994/	After Dar My Swe
2	https://en.wikipedia.org/wiki/Air_America_(film)	1990.0	https://www.imdb.com/title/tt0099005/	A Americ
3	https://en.wikipedia.org/wiki/Alice_(1990_film)	1990.0	https://www.imdb.com/title/tt0099012/	Alic
4	https://en.wikipedia.org/wiki/Almost_an_Angel	1990.0	https://www.imdb.com/title/tt0099018/	Almost a Ang

5 rows × 193 columns

In [7]:
```
# 10. Check the kaggle_metadata DataFrame.
kaggle_metadata.head()
```

Out[7]:

	adult	belongs_to_collection	budget	genres	homepage	id	imdb_i
0	False	{'id': 10194, 'name': 'Toy Story Collection', ...	30000000	[{'id': 16, 'name': 'Animation'}, {'id': 35, '...	http://toystory.disney.com/toy-story	862	tt011470
1	False	NaN	65000000	[{'id': 12, 'name': 'Adventure'}, {'id': 14, '...	NaN	8844	tt011349
2	False	{'id': 119050, 'name': 'Grumpy Old Men Collect...	0	[{'id': 10749, 'name': 'Romance'}, {'id': 35, ...	NaN	15602	tt011322
3	False	NaN	16000000	[{'id': 35, 'name': 'Comedy'}, {'id': 18, 'nam...	NaN	31357	tt011488
4	False	{'id': 96871, 'name': 'Father of the Bride Col...	0	[{'id': 35, 'name': 'Comedy'}]	NaN	11862	tt011304

5 rows × 24 columns

In [9]:
```
# 11. Check the ratings DataFrame.
ratings
```

Out[9]:

	userId	movieId	rating	timestamp
0	1	110	1.0	1425941529
1	1	147	4.5	1425942435
2	1	858	5.0	1425941523
3	1	1221	5.0	1425941546
4	1	1246	5.0	1425941556
...	...	...	...	...
26024284	270896	58559	5.0	1257031564
26024285	270896	60069	5.0	1257032032
26024286	270896	63082	4.5	1257031764
26024287	270896	64957	4.5	1257033990
26024288	270896	71878	2.0	1257031858

26024289 rows × 4 columns