In [30]:
```
# Import necessary module
import pandas as pd
import numpy as np
import mysql.connector
from mysql.connector import errorcode
```

In [16]:
```
# from https://dev.mysql.com/doc/connector-python/en/connector-python-example-
connecting.html
try:
    db = mysql.connector.connect(
        host="localhost",
        user="root",
        password="",
        database="netflixstudy"
        )
except mysql.connector.Error as err:
  if err.errno == errorcode.ER_ACCESS_DENIED_ERROR:
    print("Something is wrong with your user name or password")
  elif err.errno == errorcode.ER_BAD_DB_ERROR:
    print("Database does not exist")
  else:
    print(err)
# else:
#   cnx.close()
# db.close()   # at some point need to close the connection with this instruct
ion

cursor = db.cursor(buffered=True)  # to avoid [error](https://stackoverflow.co
m/questions/29772337/python-mysql-connector-unread-result-found-when-using-fet
chone)
```

In [ ]:
```
cursor.close()
db.close()
```

In [ ]:
```
# Build select statement for ratings table: stmt
query = 'SELECT * FROM members'

# Execute the statement and fetch the results: results
cursor.execute(query)

rows = cursor.fetchall()    # get all selected rows
for r in rows:
    print(r)
```

In [3]:
```
df = pd.read_csv('../data/processed/df.csv')
```

In [4]:
```
df1= df[0:10000]
```

In [5]:
```
df1.head()
```

Out[5]:

	Unnamed: 0	Cust_Id	Rating	Movie_Id
0	696	712664	5.0	3
1	697	1331154	4.0	3
2	698	2632461	3.0	3
3	699	44937	5.0	3
4	700	656399	4.0	3

In [6]:
```
df1.describe()
```

Out[6]:

	Unnamed: 0	Cust_Id	Rating	Movie_Id
count	10000.000000	1.000000e+04	10000.000000	10000.000000
mean	10459.371300	1.328241e+06	3.207300	7.253500
std	5199.107895	7.688886e+05	1.267552	1.782009
min	696.000000	7.000000e+00	1.000000	3.000000
25%	6739.000000	6.538500e+05	2.000000	8.000000
50%	10778.000000	1.335578e+06	3.000000	8.000000
75%	14871.250000	1.999638e+06	4.000000	8.000000
max	18862.000000	2.649336e+06	5.000000	8.000000

In [7]:
```
df1.groupby('Cust_Id').describe()
```

Out[7]:

	Unnamed: 0								Rating	
	count	mean	std	min	25%	50%	75%	max	count	me
Cust_Id										
7	1.0	12549.0	NaN	12549.0	12549.00	12549.0	12549.00	12549.0	1.0	
695	1.0	8718.0	NaN	8718.0	8718.00	8718.0	8718.00	8718.0	1.0	
1333	2.0	3160.5	3377.849094	772.0	1966.25	3160.5	4354.75	5549.0	2.0	
2133	1.0	7297.0	NaN	7297.0	7297.00	7297.0	7297.00	7297.0	1.0	
3184	1.0	13595.0	NaN	13595.0	13595.00	13595.0	13595.00	13595.0	1.0	
...	...	...	...	...	...	...	...	...	...	
2648583	1.0	15795.0	NaN	15795.0	15795.00	15795.0	15795.00	15795.0	1.0	
2648694	1.0	13918.0	NaN	13918.0	13918.00	13918.0	13918.00	13918.0	1.0	
2648781	1.0	16367.0	NaN	16367.0	16367.00	16367.0	16367.00	16367.0	1.0	
2648956	1.0	14924.0	NaN	14924.0	14924.00	14924.0	14924.00	14924.0	1.0	
2649336	1.0	17447.0	NaN	17447.0	17447.00	17447.0	17447.00	17447.0	1.0	

9796 rows × 24 columns

In [12]:
```
df.groupby('Cust_Id').Rating.mean()
```

Out[12]:
```
Cust_Id
6          3.425957
7          4.019563
10         3.434263
79         3.557012
97         3.225207
             ...
2649370    3.873984
2649378    3.273273
2649388    3.297203
2649426    4.069444
2649429    4.183908
Name: Rating, Length: 144380, dtype: float64
```

In [14]:
```
df.groupby('Cust_Id').Rating.std()
```

Out[14]:
```
Cust_Id
6          0.835619
7          0.899736
10         1.034728
79         1.064994
97         1.164034
             ...
2649370    1.024383
2649378    1.006204
2649388    0.878072
2649426    0.719589
2649429    0.951057
Name: Rating, Length: 144380, dtype: float64
```

In [13]:
```
df.groupby('Movie_Id').Rating.mean()
```

Out[13]:
```
Movie_Id
3        3.620228
8        3.140967
16       3.080652
17       2.914113
18       3.768554
           ...
17761    2.913339
17762    3.613454
17763    3.391178
17764    3.844434
17769    2.496705
Name: Rating, Length: 5332, dtype: float64
```

In [15]:
```
df.groupby('Movie_Id').Rating.std()
```

Out[15]:
```
Movie_Id
3        0.982988
8        1.294535
16       0.982483
17       0.972030
18       0.938449
           ...
17761    0.969598
17762    0.924843
17763    1.095431
17764    0.964141
17769    1.049344
Name: Rating, Length: 5332, dtype: float64
```

Homework assignment Steven Bowler 20562494 UTRGV CSCI6370 Dr. Lei

Submit: a report in word doc format answering 4 questions:

1. Describe the dataset
2. Describe how the data is loaded
3. What is the average rating for movie ID 1001?
4. What is the average rating that user ID 20001 gives to movies?

Homework Question 1

Netflix study data is provided in three principal parts

```
1.Training Data - in for files titled 'combined_data_*.txt total 100MM records
2.Test Data - in the file probe.txt
3.Qualifying Data - used for the competition to provide predictions against
```

The above 3 files are in the same general format, that each need to be parsed out into a table: Movie_Id: Cust_Id,Rating,Date

There is also a Movie Titles file that contains Movie_Id and Movie_Title.

Homework Question 2

Currently the data is loaded using two Jupyter notebooks, see Github repo (https://github.com/stevenbowler/netflixstudy) stevenbowler/netflixstudy:

1. Data-Wrangling: Load, clean, store as .csv see Github Data Wrangling (https://github.com/stevenbowler/netflixstudy/blob/master/reports/netflixstudyDataWranglingForCSCI6370.pdf)
2. Preliminary EDA: this same file, see Github Preliminary EDA (https://github.com/stevenbowler/netflixstudy/blob/master/reports/netflixstudyEDAforCSCI6370.pdf)

Homework Question 3

Pick a movie id and show average rating for that movie

In [39]:
```
Movie_id = 1001
movie_average = df[df['Movie_Id']==Movie_id].Rating.mean()
print('Movie ',Movie_id,'had an average rating of',np.round(movie_average,2))
```

```
Movie  1001 had an average rating of 3.29
```

Homework Question 4

pick a customer id and show average rating for that customer

In [38]:
```
# per homework question 4, pick a customer id and show average rating for that
customer
Customer_id = 97
customer_average = df[df['Cust_Id']==Customer_id].Rating.mean()
print('Customer ',Customer_id,'had an average rating of',np.round(customer_ave
rage,2))
```

```
Customer  97 had an average rating of 3.23
```