Ford Neild and Ahmed Elzokm 04/16/2020
Database Project Part C Process

1. This is the link where I extracted the stock market data: https://github.com/CNuge/kaggle-code/blob/master/stock_data/individual_stocks_5yr.zi This download is a collection of csvs. Each cvs is a security in the S&P500 index. The contents of the csv are date,open,high,low,close,volume, Name. Note: the "Name" field is the actually security symbol.
2. Then I will extract all of the symbols names for each of the securities.

```
ls -l ./individual_stocks_5yr >> security-names-raw.txt
```

3. Then I cleaned the data to just get the symbol

```
file1 = open('security-names-raw.txt', 'r')
file2 = open('security-names-cleaned.txt', 'w')
Lines = file1.readlines()
cleanedLines = [""] * len(Lines);
count = 0
# Strips the newline character
for line_index, line in enumerate(Lines):
   splitLine = line.split(" ")
   fileName = splitLine[-1].split("_")[0]
   cleanedLines[line_index] = fileName + '\n'
file2.writelines(cleanedLines)
file1.close()
file2.close()
```

4. Now I need to get the company's name and sector given the symbol. This download has a csv with the symbol, name, and sector of each one of the stocks in the S&P 500. https://datahub.io/core/s-and-p-500-companies
5. Now I need to extract all of the unique sectors from this csv with a sectorID to make the sector-relation.txt file. The relation looks like Sector(SectorID, Sector Name)

```
file1 = open('constituents_csv.csv', 'r')
file2 = open('sector-relation.txt', 'w')
Lines = file1.readlines()
sectors = []
cleanedLines= []
# Strips the newline character
for line_index, line in enumerate(Lines):
   sector = line.split(",")[-1]
   if sector not in sectors:
       sectors.append(sector)
       cleanedLines.append(str(len(sectors)) +","+ sector)
```

```

file2.writelines(cleanedLines)
```

6. Now I need to make the securities-relation data which looks like Securities(Symbol, SectorID, Company Name)

```
file1 = open('constituents_csv.csv', 'r')
file2 = open('security-relation.txt', 'w')
Lines = file1.readlines()
sectors = {}
count = 0
cleanedLines= [None] * len(Lines)
for line_index, line in enumerate(Lines):
   line_split = line.split(",")
   sector = line_split[-1].strip()
   if sector not in sectors:
       sectors[sector] = len(sectors.keys())+1

   cleanedLines[line_index] = line_split[0] + ',' + str(sectors[sector]) + ',' + line_split[1] + '\n'

file2.writelines(cleanedLines)
```

7. Now I need to make the date-relation.txt and the trades-relation.txt. The Date relation looks like Date(DateID, Month, Day, Year). Since we got more data than just closing price we are changing the Closing Price(DateID, Symbol, Security Price) relation into Trade(DateID, Symbol, Open, High, Low,Close, Volume)

```
import os
directory = './individual_stocks_5yr/'
tradeFile = open('trade-relation.txt', 'w')
dateFile = open('date-relation.txt', 'w')

dates ={}
trades=[]
datesCleanedLines = []
tradesCleanedLines = []

for filename in os.listdir(directory):
   if filename.endswith(".csv"):
       path = (directory + '/' + filename)
       f = open(path)
```

```
        lines = f.readlines()
        lines.pop(0) #remove head of csv file with attribute names
        for line_index, line in enumerate(lines):
            line_split = line.split(",")
            date=line_split[0]
            if date not in dates:
                dates[date] = len(dates) + 1
                [year, month, day] = date.split('-')
                datesCleanedLines.append(str(len(dates)) + "," + str(int(month)) + ","
+ str(int(day)) + "," + str(int(year))+'\n')
            tradesCleanedLines.append(str(dates[date])+"," +
line_split[-1].strip()+","+line_split[1] +','+ line_split[2]+','+
line_split[3]+','+line_split[4] +','+line_split[5] + "\n")
dateFile.writelines(datesCleanedLines)
tradeFile.writelines(tradesCleanedLines)
```

8. Now that we have all of the stock market data we need to get the weather data. The following website https://www.ncdc.noaa.gov/data-access allows you to write a query to their database. Due to limitations on the result sets size I was only able to get 3 years of data for the NY area.
9. I will limit the scope our a weather to a single weather station in central park.

```
grep -hnr "NY CITY" weather.csv > city_weather.csv
```

10. I will parse the weather data and match it with the correct dateID to create the following relation. Note we will have more attributes in the Forecast relation because they came with the download and they may be interesting.
11. After parsing the data, I realized the data from that was garbage so I have to find new data. For some reason average temp is always blank so I had to add min and max temp. Then I reparsed the better data with

```
import os
directory = './individual_stocks_5yr/'
forcastFile = open('forcast-relation.txt', 'w')



file1 = open('date-relation.txt','r')
lines = file1.readlines()
dates ={}
for line_index, line in enumerate(lines):
    [dateId, month, day, year] = line.split(",")
    if(len(month) == 1):
        month = '0' + month
    if(len(day) == 1):
        day = '0' + day
    date = year.strip()+'-'+month+'-'+day
    if date not in dates:
        dates[date] = len(dates) + 1

file2 = open('weather2.csv')
lines = file2.readlines()
lines.pop(0) #remove head of csv file with attribute names
cleanedForcastLines = []
for line_index, line in enumerate(lines):

    line_split = line.split(",")[1:-1]
    line_split.pop(2)
    date = line_split[0][1:-1]

    # print(date)
    if date in dates:
        line_split[0] = str(dates[date])
        line_split[1] = line_split[1][1:-1]
        line_split[2]= line_split[2][1:-1]
        line_split[3]= line_split[3][1:-1]
        cleanedForcastLines.append(",".join(line_split) + '\n')

forcastFile.writelines(cleanedForcastLines)
```

12. We can create the small files by doing

```
head -10 forecast-relation.txt >> forecast-relation-small.txt
```