Contents

[Summary 1](#_Toc63622251)

[Stakeholders 1](#_Toc63622252)

[Business Rules 1](#_Toc63622253)

[Glossary 2](#_Toc63622254)

[Data Questions 3](#_Toc63622255)

[The Conceptual Model 3](#_Toc63622256)

[The Logical Model 4](#_Toc63622257)

# Summary

The purpose of constructing this database is to create easier access to a diverse array of financial information. I hope through the creation of this database the user will be able to perform various types of technical and fundamental analysis. I hope it will aide in financial research and applications in finding alpha factors/features, alternative trading strategies and risk management. There will be price data, useful for technical and statistical analysis like momentum/stochastic oscillators and other forms of price forecasting. There will also be fundamental data available for analysis, which will be useful for cash flow analysis, earnings forecasts, and valuation. This database will be practical for researching and applying machine and deep learning algorithms/models in finance.

# Stakeholders

The stakeholders for my database will be those who are interested in performing various types of analysis/research on companies traded on the New York Stock Exchange. This includes myself, and others who have interests and curiosity in the financial markets.

# Business Rules

* User chooses a stock symbol
* User chooses to either pull fundamental or price data
* If user chooses fundamental data
  + Select balance sheet
  + Select Income statement
  + Select statement of cash flows
* Select balance sheet pull companies assets, liabilities and owner’s equity
* Select statement of cashflows pull operating activities, investing activities, financing activities
* Select income statement pull revenue, expenses, gains, losses
* If user pulls numeric data
  + Daily price data
  + Daily volume data
* Numeric data includes stocks price open, high, low, close, adjusted close, and volume

# Glossary

A **ticker symbol** is an arrangement of characters (usually letters) representing particular securities listed on an exchange (traded publicly)

An **exchange** is a marketplace where securities, commodities, derivatives and other financial instruments are traded (for the sake of this database we will be only looking at securities)

**Fundamental Data** in financial analysis refers to revenues, earnings, profit margins and other data used to determine a company’s underlying value and potential for future growth

A **balance sheet** is a financial statement that reports a company’s **assets**, **liabilities**, and **shareholder’s equity** at a SPECIFIC point in time

**Assets** area resource owned by a company that is expected to provide a future benefit

**Liabilities** are something a company owes (i.e. loans & accounts payable)

**Shareholder’s equity**, also referred to as **stockholder’s equity** or simply **owner’s equity** is the remaining amount of assets available to shareholders after all liabilities have been paid (comparable to a company’s net worth)

The **income statement** is a financial statement that reports a company’s **revenue, expenses, gains,** and **losses.**

**Revenue** is the income generated by normal operating activities

**Expenses** are the cost of operations a company incurs to generate revenue

**Gains** are a general increase in the value of an asset or property

**Losses** are incurred when an asset decreases in value

The **statement of cash flows** summarizes the amount of cash and cash equivalents that are entering and leaving a company

**Operating Activities** aka **Cash Flow From Operating Activities (CFO**) – indicates the amount of money a company brings in from its regular business activities

**Financing Activities** aka **Cash Flow From Financing Activities (CFF)** – indicates the net cash flows that are used to finance the company, including debt, equity, and dividends

**Investing Activities** aka **Cash Flow From Investing Activities (CFI)** – indicates the amount of cash spent or received from investment activities, such as purchases of physical assets, investments in securities, or the sale of securities or assets

**Technical analysis** is a type of analysis that uses past market data for forecasting the direction of stock prices (typically uses **stock price** and **volume data**)

**Stock price** refers to the current price that a share of a stock is trading for on an exchange

**Volume** the amount of a security that was traded during a given time period

**Alpha factor** is used to refer the excess return or a trading strategy’s ability to beat the market

# Data Questions

What was the closing price of a security on a certain day/time?

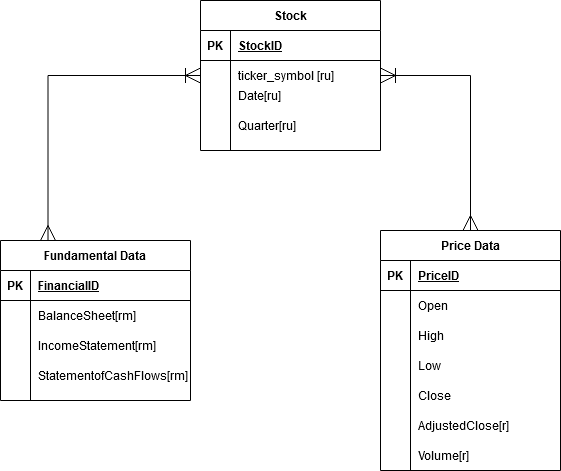
What was the maximum price of a security within a certain time period?

What was the maximum trading Volume during a certain period of time?

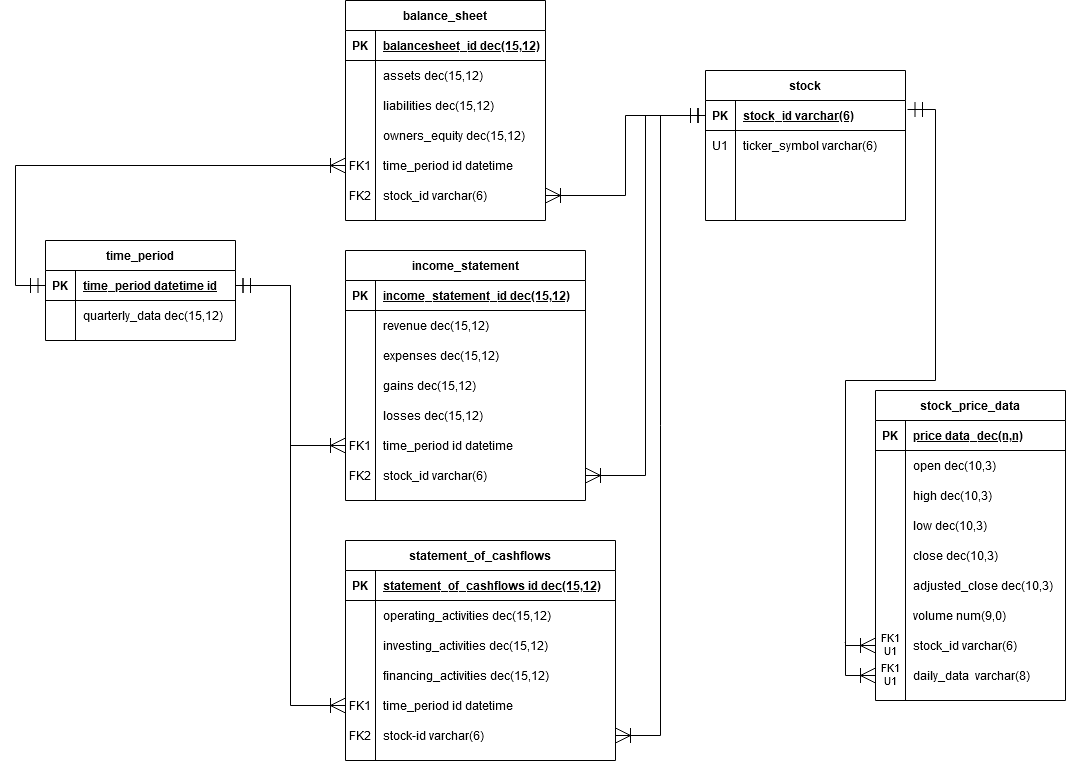
What is the minimum price of a security over a certain time period?

What was the minimum trading volume over a certain period of time?

# The Conceptual Model



# The Normalized Logical Model



**Physical Database / Warehouse Design**

/\*

Author : Ryan Dean

Course : IST659 M401

Term : January, 2021

\*/

-- Drop Tables

Drop Table if Exists statement\_of\_cashflows\_

Drop Table if Exists income\_statement\_

Drop Table if Exists balance\_sheet\_

Drop Table if Exists stock\_price\_data

Drop Table if Exists stock

--Creating Stock Table

Create Table stock (

--Columns for the User Table

stock\_id varchar(6),

--Constraints on the User Table

Constraint pk\_stock\_id Primary Key (stock\_id),

)

--Creating Stock Price Data Table

Create Table stock\_price\_data (

--Columns for the stock\_price\_data

price\_data decimal(10,3),

p\_open decimal(10,3),

p\_high decimal(10,3),

p\_low decimal (10,3),

p\_close decimal (10,3),

p\_adjusted\_close decimal(10,3) not null,

volume numeric (9,0) not null,

ticker\_symbol varchar(6) not null,

ticker\_day datetime not null,

-- Constraints on the Stock Price Data Table

Constraint pk\_price\_data Primary Key (price\_data),

Constraint u1\_stock Unique (ticker\_symbol, ticker\_day),

Constraint fk1\_stock Foreign Key (ticker\_symbol) References stock(stock\_id),

)

--Creating the Balance Sheet Data Table

Create Table balance\_sheet\_ (

--Columns for the Balance Sheet

balance\_sheet int identity,

assets decimal(15,12) not null,

liabilities decimal(15,12) not null,

owners\_equity decimal(15,12) not null,

time\_period datetime not null,

stock\_id varchar(6) not null,

--Constraints on the Balance Sheet

Constraint pk\_balance\_sheet Primary Key (balance\_sheet),

Constraint fk2\_stock\_id1 Foreign Key (stock\_id) References stock(stock\_id),

)

--Creating the Income Statement Data Table

Create Table income\_statement\_ (

--Columns for the Income Statement

income\_statement int identity,

revenue decimal(15,12) not null,

expenses decimal(15,12) not null,

gains decimal(15,12) not null,

losses decimal(15,12) not null,

time\_period datetime not null,

stock\_id varchar(6) not null,

--Constraints on the Balance Sheet

Constraint pk\_income\_statement Primary Key (income\_statement),

Constraint fk2\_stock\_id2 Foreign Key (stock\_id) References stock(stock\_id),

)

--Creating the Statement of Cashflows

Create Table statement\_of\_cashflows\_ (

--Columns for the Statement of Cashflows

statement\_of\_cashflows int identity,

operating\_activities decimal(15,12) not null,

investing\_activities decimal(15,12) not null,

financing\_activities decimal(15,12) not null,

time\_period datetime not null,

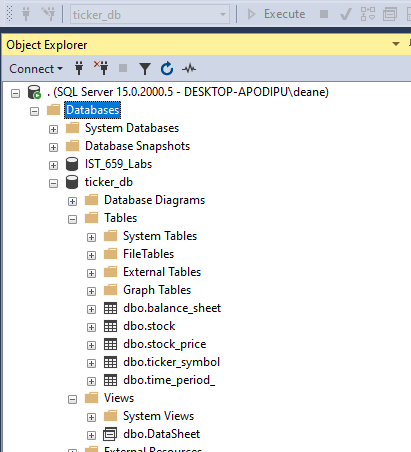
stock\_id varchar(6) not null,

--Constraints on the Statement of Cashflows

Constraint pk\_statement\_of\_cashflows Primary Key (statement\_of\_cashflows),

Constraint fk2\_stock\_id3 Foreign Key (stock\_id) References stock(stock\_id),

)



The above code represents the construction of the physical database/data warehouse I designed. The comments describe what I was attempting to do while constructing the db.

**Data Creation**

Drop Table if Exists stock\_price

Drop Table if Exists ticker\_symbol

-- Creating Tables as proxy to insert data into

Create Table stock\_price (

Date\_ datetime not null,

Open\_ decimal(10,3),

High\_ decimal(10,3),

Low\_ decimal(10,3),

Close\_ decimal(10,3),

AdjClose\_ decimal(10,3),

Volume\_ decimal(10,3),

)

Create Table ticker\_symbol (

ticker varchar(6) not null

)

Go

--Inserting ticker symbol manually

Insert Into ticker\_symbol

Values ('CRNC')

--Bulk insert from CSV file

Bulk Insert stock\_price

From 'C:\Random\crnc\_3\_5\_21.csv'

With (

Format ='CSV',

Firstrow = 2,

Fieldterminator =',',

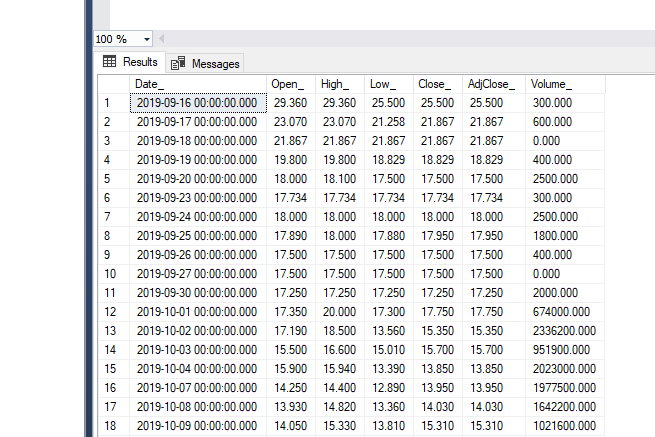
Rowterminator = '\n'

)

Go

--Shows all data from date CSV was Created

select \* from stock\_price



--Inserting data manually to update rows and columns

Insert into stock\_price

Values ('3/17/2021', '105.1500015', '110.8899994', '102.8600006', '109.3199997', '109.3199997', '378600')

Insert into stock\_price

Values ('3/18/2021', '106.2900009', '107.2399979', '100.3700027','101.0899963', '101.0899963', '359800')

Insert into stock\_price

Values ('3/19/2021', '99.91999817', '105.5800018', '98.73000336', '102.5899963', '102.5899963', '730200')

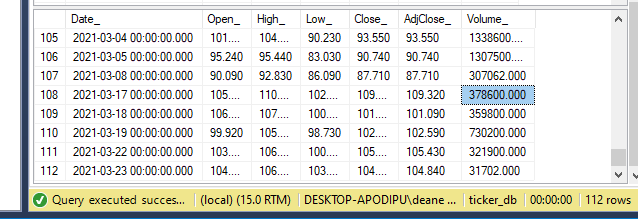
Insert into stock\_price

Values ('3/22/2021', '103.25', '106.8000031', '100.3550034', '105.4300003', '105.4300003', '321900')

Insert into stock\_price

Values ('3/23/2021', '104.4000015', '106.2399979', '103.9759979', '104.8399963', '104.8399963', '31702')

Select \* From stock\_price



**Data Manipulation/Functions**

--Added this statement to ensure code would run

Drop View If Exists [DataSheet]

Go

--Creating a view where adjusted close is above $55

Create View DataSheet As

Select \* From stock\_price

Where AdjClose\_ > 55

Go

Select \* From [DataSheet];

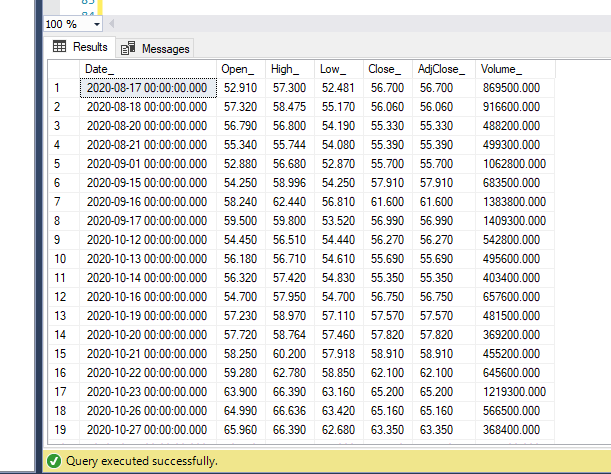
Go

--Update CRUD example

Update stock\_price

SET AdjClose\_ = '101.00'

Where Date\_ = '2020-08-18 00:00:00.000';



**Answering Business/Data Questions**

-- Creating data table that answers proposed business questions

Select

Max(AdjClose\_) as AllTimeHigh,

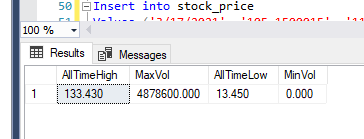
Max(Volume\_) as MaxVol,

Min(AdjClose\_) as AllTimeLow,

Min(Volume\_) as MinVol

From stock\_price

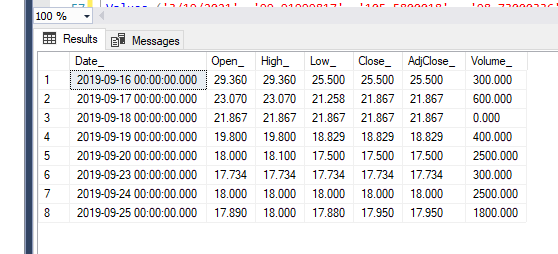
The above code finds and executes four of the business questions provided in Part 1. I used the maximum and minimum function to pull the maximum value of trading volume as well as the minimum value for trading volume. I renamed the column names for the output.



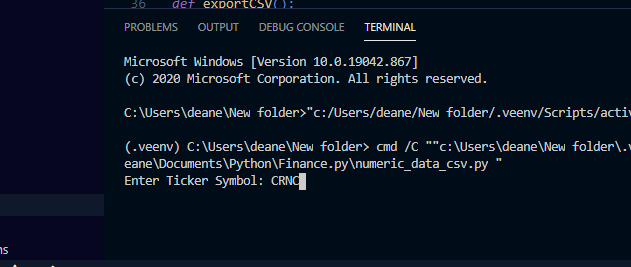
-- Answering more business questions using a between clause

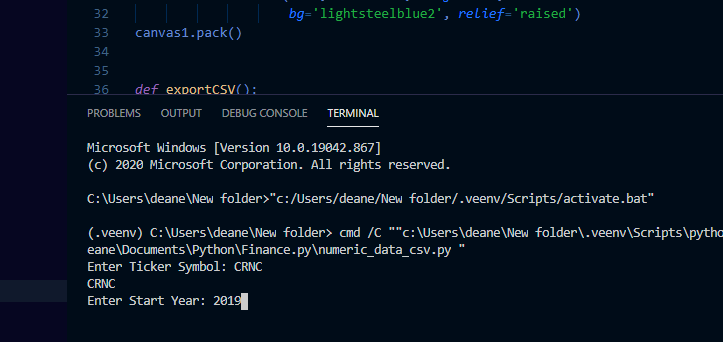
Select \* From stock\_price

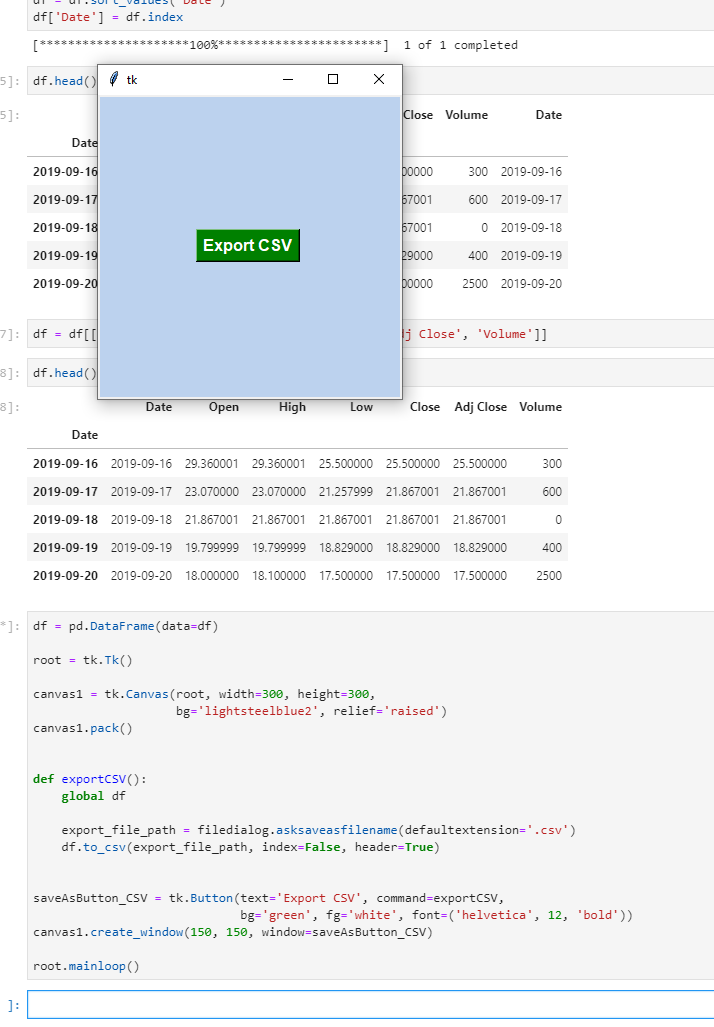
Where date\_ Between '2019-09-16 00:00:00.000' and '2019-09-25 00:00:00.000'

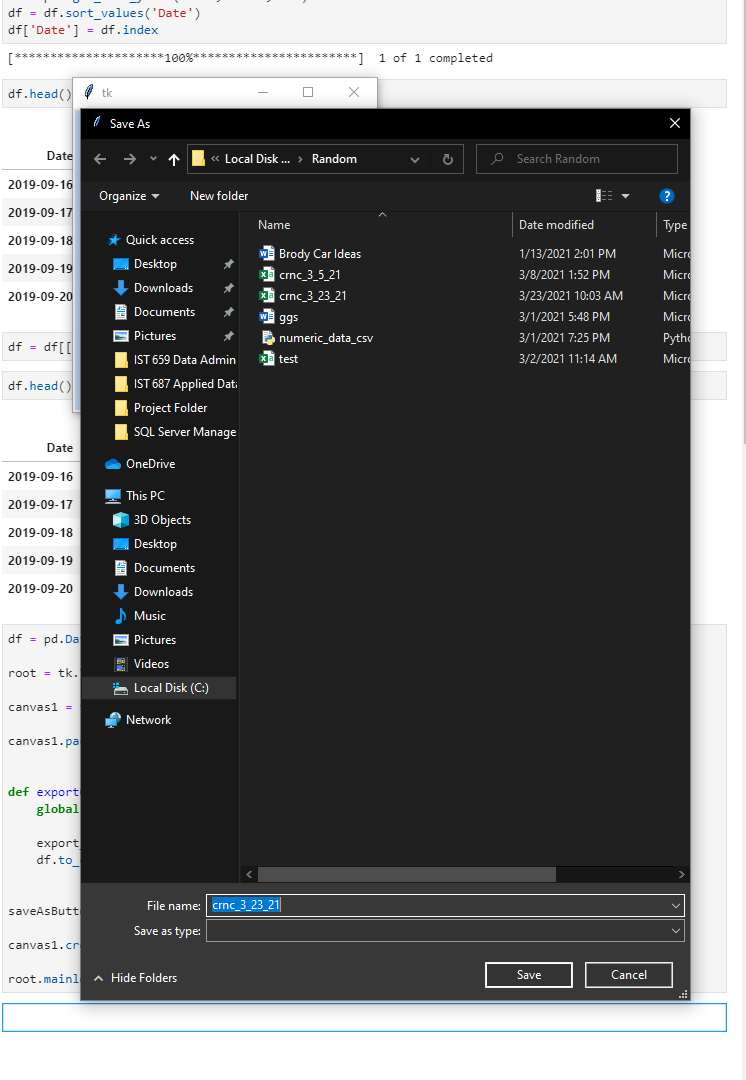


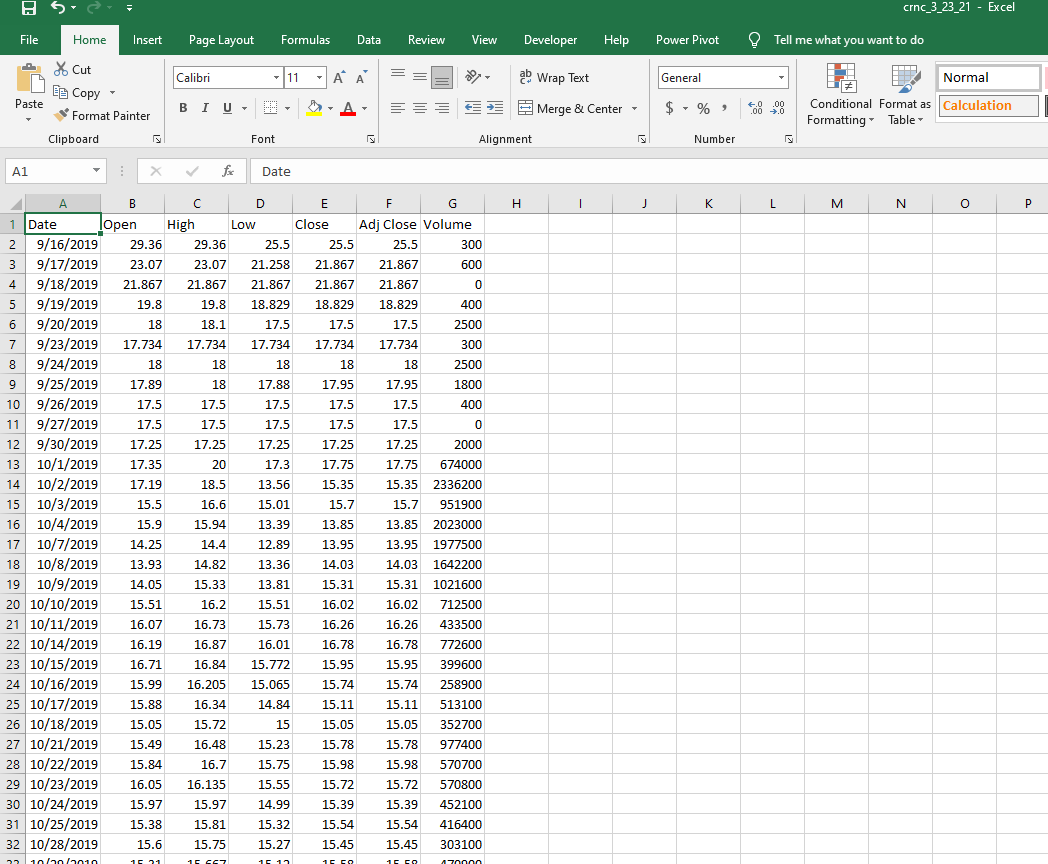
**Implementation : Thinking of the User**











In order to insert data into my database I created a python script using the Yahoo Finance API in python. I then exported it to a CSV where I could then use SQLs bulk insert to insert into my db. The python script allows the user to enter any ticker symbol and any start year.

import pandas as pd

import numpy as np

import yfinance as yf

import datetime as dt

from pandas\_datareader import data as pdr

from pandas import DataFrame

import tkinter as tk

from tkinter import filedialog

yf.pdr\_override()

stock = input("Enter Ticker Symbol: ")

print(stock)

startyear = int(input("Enter Start Year: "))

startmonth = 1

startday = 1

start = dt.datetime(startyear, startmonth, startday)

now = dt.datetime.now()

df = pdr.get\_data\_yahoo(stock, start, now)

df = df.sort\_values('Date')

df['Date'] = df.index

df = df[['p\_date', 'p\_open', 'p\_high', 'p\_low', 'p\_close', 'p\_adjclose', 'p\_volume']]

df = pd.DataFrame(data=df)

root = tk.Tk()

canvas1 = tk.Canvas(root, width=300, height=300,

bg='lightsteelblue2', relief='raised')

canvas1.pack()

def exportCSV():

global df

export\_file\_path = filedialog.asksaveasfilename(defaultextension='.csv')

df.to\_csv(export\_file\_path, index=False, header=True)

saveAsButton\_CSV = tk.Button(text='Export CSV', command=exportCSV,

bg='green', fg='white', font=('helvetica', 12, 'bold'))

canvas1.create\_window(150, 150, window=saveAsButton\_CSV)

root.mainloop()

**Reflection:**

When I first started this project I had absolutely zero experience using SQL. I did not understand certain vernacular and parts of the process, and even where to begin with databases. I struggled with the conceptual and logical model, as far as cardinality and how to interpret them abstractly. While working through the project and creating the physical database design I began to understand the importance and use of those diagrams, and ultimately understood them much better. I struggled with the scope of my project, it was fairly large and went beyond my skill level in may aspects. However I was able to pursue through some aspects as far as creating a script in python to fill a table in my database. Now that I have more experience writing in the SQL language I would like to spend more time on the design phase since I now have a better idea of the use of them. I would also like to explore the pyodbc package in python to see if I could automate updating my tables as new information comes available.

I understand the issues you arose back when we first met after turning in part one. I was creating a data warehouse rather than a database. This did create some problems for me.

As a “data professional” I see the importance of spending a lot of time in the planning phase and learning the ability to be flexible, troubleshoot, and learn and explore new topics. I believe it will help me be more humble in the scope of projects and time management as well as how I am able to convey the information I find in a meaningful way.

I’m excited I was able to download, use and understand SQL on my machine and complete the labs and the project without using the Syracuse virtual portal. It has helped me tremendously on my continuing efforts to understand and use data science tools and each of their importance. I am wholly impressed by the concept of relational algebra and the natural ebb and flow of data creation, organization and retrieval.

Hey Chad, I know this probably isn’t exactly how you wanted it, but I tried my damnedest on it. I learned a lot of SQL (obviously) and python as well, and how they connect. Thanks again for your time and help outside of class.

**Summary:**

This database provides stock price information that can be selected based on a specific date and time. It includes major attributes such as open, high, low, close and volume that are crucial for time series analysis of stock prices. While the database is not complete, and will need some automation to actually be useful, it provides a great template that could be added to and worked on in the future.