PROBLEM STATEMENT

The script we have to generate is for the stock recommendation application. Here, when the customer is visiting our website and creating his/her account. Then there will be an option for them to generate/update their portfolio. We have to make a script which can take customer's preferences as an input. Check for his required basic details in our database and will display his/her portfolio with the risk factor (beta) associated with their profile.

What had we done here? Our very first step here is to import the necessary liabraries. After that the connection was set-up for Python with MYSQL. This we had achieved using sqlalchemy through creating an engine.

After this initial steps, we had created table and inserted few dummy values for our reference.

Further, we had generated a class named "beta" and defined certain methods inside it.

This methods are: 1) Updating existing information regarding customer's portfolio. 2) Writing new information if no portfolio exist for a customer. 3) Reading the information from our database in order to display it at our output. 4) A most important function which will take customer's input as json and interact with database and finally give output in the form of json itself.

In our final step we had initiated by methods by calling our class and thus our output was generated in required json format.

Purpose: To display customer's portfolio with associated risk on their profile. Input/ Output file format: JSON.

In [1]:

```
import json
import pandas as pd
import mysql.connector
from sqlalchemy import create_engine

engine=create_engine("mysql+pymysql://root:120450109009@localhost:3306/df")

conn = mysql.connector.connect(host="localhost", user="root", passwd="120450109009", database="df" ,charset="utf8")
cur= conn.cursor(dictionary= True)

cur.execute("CREATE TABLE cust_portfolio(cust_id BIGINT, port_id BIGINT, add_date VARCHAR(50))")
```

```python
cur.execute("CREATE TABLE portfolio_det(port_id BIGINT, trdsym VARCHAR(50), qty INT, pp INT)")

cur.execute("create table beta_tab(trdsym VARCHAR(50), beta FLOAT)")
cur.execute("INSERT INTO beta_tab(trdsym, beta) values('AMAZON',1.43)")
cur.execute("INSERT INTO beta_tab(trdsym, beta) values('WALMART',0.63)")
cur.execute("INSERT INTO beta_tab(trdsym, beta) values('NETFLIX',1.51)")
cur.execute("INSERT INTO beta_tab(trdsym, beta) values('P&G',0.6)")

conn.commit()

class beta:
    def __init__(self,p_id, data):
        #self.cur = cur,
        #self.conn = conn,
        self.p_id = p_id
        self.data = data

    def update(self,p_id, data):
        s= f"delete from portfolio_det where port_id='{self.p_id}'"
        cur.execute(s)
        conn.commit()
        dtfm = pd.DataFrame(data['dt'])
        dtfm['port_id']= data['port_id']
        date= data['add_date']
        dtfm.to_sql(name="portfolio_det", con= engine,if_exists = "append", index=False)
        a= f"update cust_portfolio set add_date = '{date}' where port_id= '{self.p_id}'"
        cur.execute(a)
        conn.commit()

    def write(self,p_id, data):
        dtfm = pd.DataFrame(data['dt'])
        dtfm['port_id']= data['port_id']
        dtfm.to_sql(name="portfolio_det", con= engine,if_exists = "append", index=False)
        c_id= data["cust_id"]
        date= data['add_date']
        b= (c_id,self.p_id,date)
        s="insert into cust_portfolio (cust_id, port_id, add_date) values(%s,%s,%s)"
        cur.execute(s,b)
        conn.commit()

    def read(self,p_id, data):
        c_id= data["cust_id"]
        query = (f"select * from portfolio_det where port_id='{self.p_id}'")
        tab=pd.read_sql(query,engine)
        query2= (f"select sum(portfolio_det.qty*portfolio_det.pp*beta_tab.beta)/sum(portfolio_det.qty*portfolio_det.p
        b= pd.read_sql(query2,engine)
        conn.commit()
        df= [b,tab]
        final={}
        final["beta"] = df[0].beta[0].round(2)
        final["dt"] = df[1].to_dict(orient="records")
        with open("sample.json", "w") as outfile:
            return json.dump(final, outfile)

    def output(self, data):
        list = pd.read_sql("select port_id from cust_portfolio", engine)
        p_id = data['port_id']
        l=[]
        for i in list.port_id:
            l.append(i)
        if p_id in l:
            return beta.update(self, p_id, data)
        else:
            return beta.write(self, p_id, data)
        return beta.read(self, p_id, data)
```

In [3]:

```
with open("x.json") as json_file:
    data = json.load(json_file)

bt= beta(data['port_id'], data)
bt.output(data)
bt.read(data['port_id'],data)
```

In [ ]: