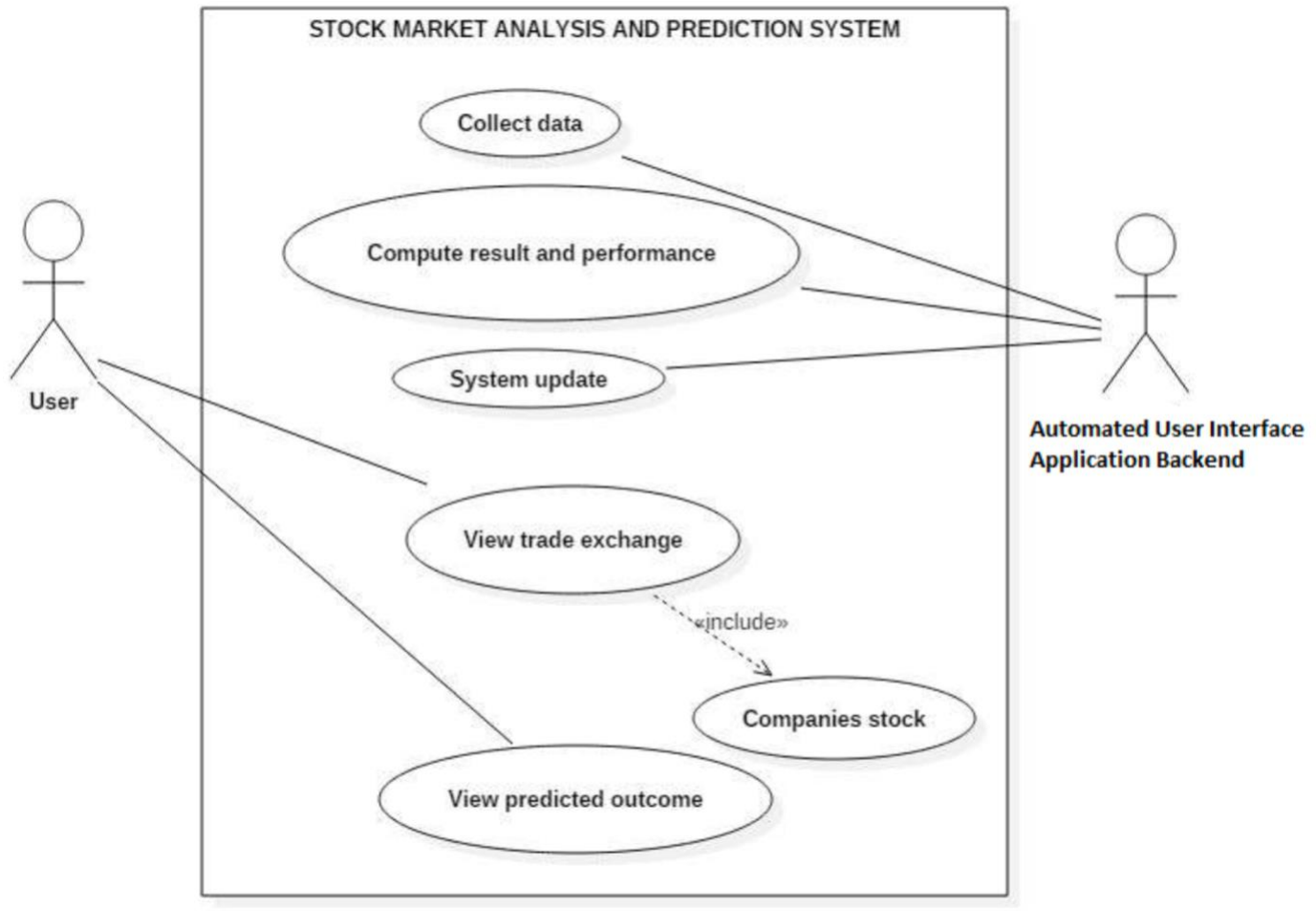


# Quantum Computing Companies Stock Prices Analysis and Forecasting Web Application Project Part 2

Pushkar D, Sharanbasav S, Vignesh K

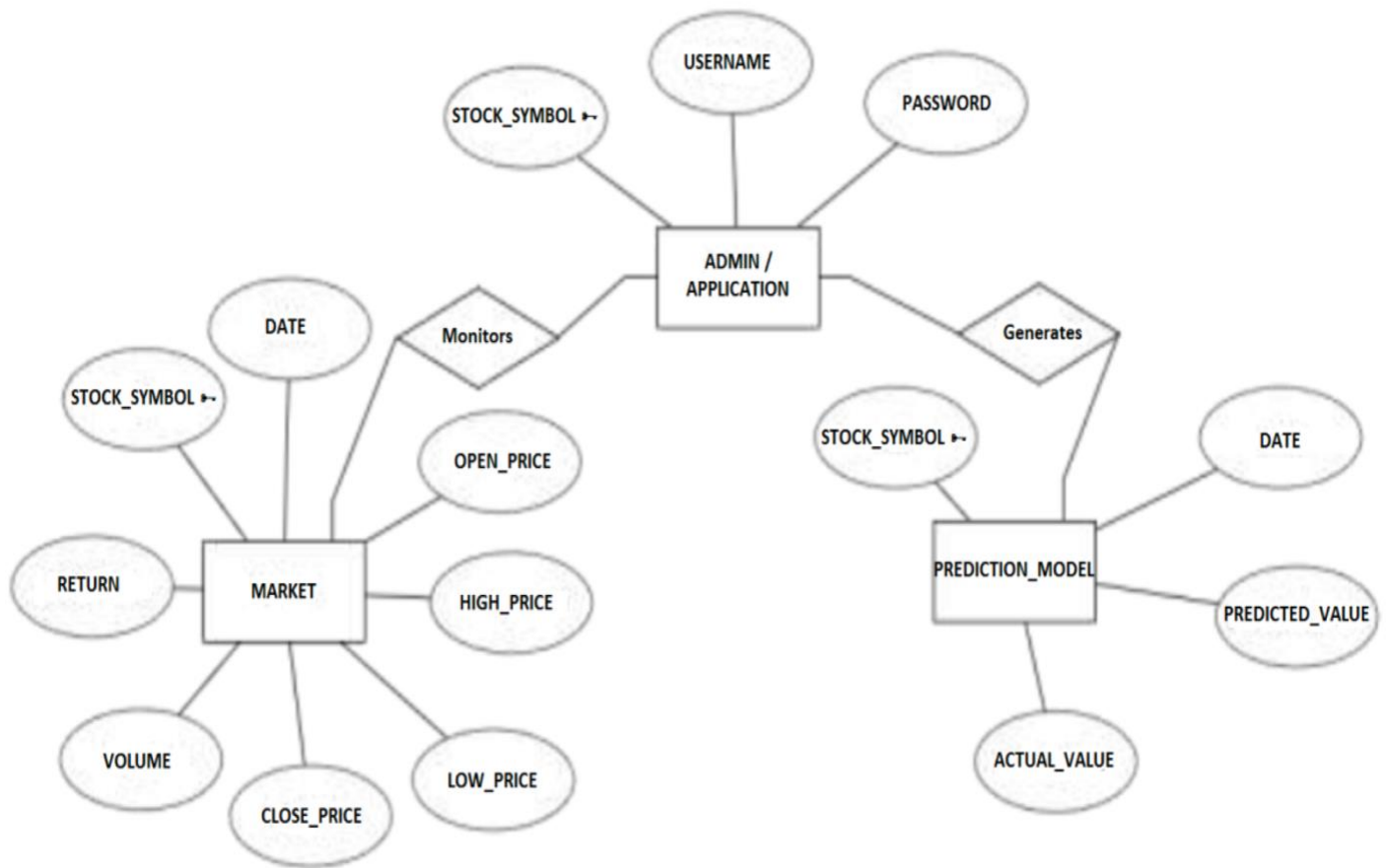
## System Diagram:



The Main Objective of this database application would be to provide quick insights into the stock performance of the Quantum computing driven companies, by appropriate visualizations and Statistics of the stocks. It would also provide forecast of the stocks using ML algorithms (time-series forecasting). Stock Price Prediction using machine learning helps the user discover the future value of company stock and other financial assets traded on an exchange. The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy. By the development of this web-application we try to maximize the accuracy and easy to use interface for the user to analyze the Quantum computing stocks present for investing.

The above system diagram depicts the interaction between the user and the system. The Stock Market is a complex and dynamical system & is influenced by many factors that are subject to uncertainty. So, it is a difficult task to forecast stock price movements. Due to technology and globalization of business & financial markets it is important to predict the stock prices more quickly & accurately. User-friendly Trading application can be developed based on financial predictive indicator algorithms & machine learning techniques to predict the performance of stocks.

## ER Diagram:



An entity–relationship model describes interrelated things of interest in a specific domain of knowledge. A basic ER model is composed of entity types and specifies relationships that can exist between instances of those entity types.

### Entities and their Attributes:

- **Market Entity:** Attributes of Market Entity are Stock Symbol (or) Ticker, High Price, Low Price, Open Price, Volume, Date, Close Price, and Return.
- **Prediction Model Entity:** Attributes of Prediction Model Entity are Actual Value, Predicted Value, Date and Stock Symbol.
- **Admin Entity:** Attributes of Admin Entity are Username, Stock Symbol, and Password

### Relationships:

- Admin allows the user to monitor and analyze the Market and its entities, so that the user can determine if the market is **Bullish** or **Bearish**
- With the Prediction Model entity, the admin can generate the forecast or predictions of the stock data
- The username and Password attributes contribute to the security of the system

## Describe Data Types:

In this project we are using Quantum Stock dataset where we have 9 attributes where ID is the primary key which describes the uniqueness. Apart from that we have Date which is varchar and shouldn't be NULL. As the values of stock are recorded on daily basis Date shouldn't be NULL. If on particular day the prices of stocks are not recorded then the value is either set to 'NA' or '-', which is indeed a particular record for that date. In stock trading, the high and low refer to the maximum and minimum prices in a given time period. Open and close are the prices at which a stock began and ended trading in the same period. Volume is the total amount of trading activity. Adjusted values factor in corporate actions such as dividends, stock splits, and new share issuance. Financial periodicals and websites often include a stock's "high" and "low" price. The high is the highest price at which a stock traded during a period. The low is the lowest price of the period. A stock's high and low points for the day are often called its intraday high and low. As these represent a number value hence we assigned open, high, low, close, adjclose, volume as numeric value. A stock symbol is a unique series of letters assigned to a security for trading purposes. Stocks listed on the New York Stock Exchange (NYSE) can have four or fewer letters. Nasdaq-listed securities can have up to five characters. Symbols are just a shorthand way of describing a company's stock, so there is no significant difference between those that have three letters and those that have four or five. Stock symbols are also known as ticker symbols. As these values represent letters or symbols, we used VARCHAR as data type.

## Describe Constraints:

SQL constraints are used to specify rules for the data in a table. Constraints are used to limit the type of data that can go into a table. This ensures the accuracy and reliability of the data in the table. If there is any violation between the constraint and the data action, the action is aborted. We have enforced constraints to 3 attributes: ID, Date, Ticker. As Date is the important factor in stock market as it records all the values for the ticker and ID. If the stock market is closed for one day the values for attributes OCHL is either '-' or 'NA'. Similarly, as we record a value for that date it is necessary to include ticker and ID as we record the values for that ticker and has unique ID. For the rest of the attributes, we haven't enforced any constraints.

## Code to Create the Database:

```
1 • CREATE DATABASE STOCK;
2 • DROP TABLE IF EXISTS QSTOCK;
3 • ⊖ CREATE TABLE IF NOT EXISTS QSTOCK(
4
5     Date      VARCHAR(8) NOT NULL
6     ,open     VARCHAR(18)
7     ,high     VARCHAR(18)
8     ,low      VARCHAR(18)
9     ,close    VARCHAR(18)
10    ,adjclose VARCHAR(18)
11    ,volume   NUMERIC(12,1)
12    ,ticker   VARCHAR(6) NOT NULL
13    ,ID       VARCHAR(14) NOT NULL PRIMARY KEY
14 );
```

```
1 • DROP TABLE IF EXISTS QSTOCK;
2 • ⊖ CREATE TABLE IF NOT EXISTS QSTOCK(
3     FIELD1  INTEGER
4     ,Date   DATE NOT NULL
5     ,open   VARCHAR(18)
6     ,high   VARCHAR(18)
7     ,low    VARCHAR(18)
8     ,close  VARCHAR(18)
9     ,adjclose VARCHAR(18)
10    ,volume NUMERIC(12,1)
11    ,ticker  VARCHAR(6) NOT NULL
12    ,ID     VARCHAR(14) NOT NULL PRIMARY KEY
13 );
14 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-04','11.125','11.25','10.5','10.800000190734863','10.80000
15 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-05','10.850000381469728','11.829999923706056','10.75','11.
16 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-06','11.5','11.5','11.0','11.050000190734863','11.05000019
17 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-07','11.5','11.5','11.050000190734863','11.109999656677246
18 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-08','11.649999618530272','12.350000381469728','11.0','11.0
19 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-11','11.800000190734863','11.800000190734863','10.80000019
20 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-12','11.800000190734863','11.800000190734863','11.06999969
21 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-13','11.34000015258789','11.369999885559082','10.970000267
22 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-14','11.3100004196167','11.699999809265137','11.2250003814
23 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-15','11.56999969482422','11.789999961853027','11.109999656
24 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-19','11.399999618530272','11.479999542236328','11.09000015
25 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-20','11.329999923706056','11.449999809265137','11.25','11.
26 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-21','11.390000343322754','11.720000267028809','11.23999977
27 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-22','11.550000190734863','12.199999809265137','11.5','12.0
28 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-25','12.5','12.579999923706056','11.800000190734863','12.0
29 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-26','12.380000114440918','12.449999809265137','11.82999992
30 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-27','11.520000457763672','11.68000030517578','11.260000228
31 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-28','11.5','11.68000030517578','11.350000381469728','11.36
32 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-01-29','11.420000076293944','11.449999809265137','11.10000038
33 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-02-01','11.449999809265137','11.68000030517578','11.399999618
34 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-02-02','11.770000457763672','12.119999885559082','11.72000026
35 • INSERT INTO QSTOCK(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2021-02-03','11.970000542236328','12.310000381469728','11.96000003
```

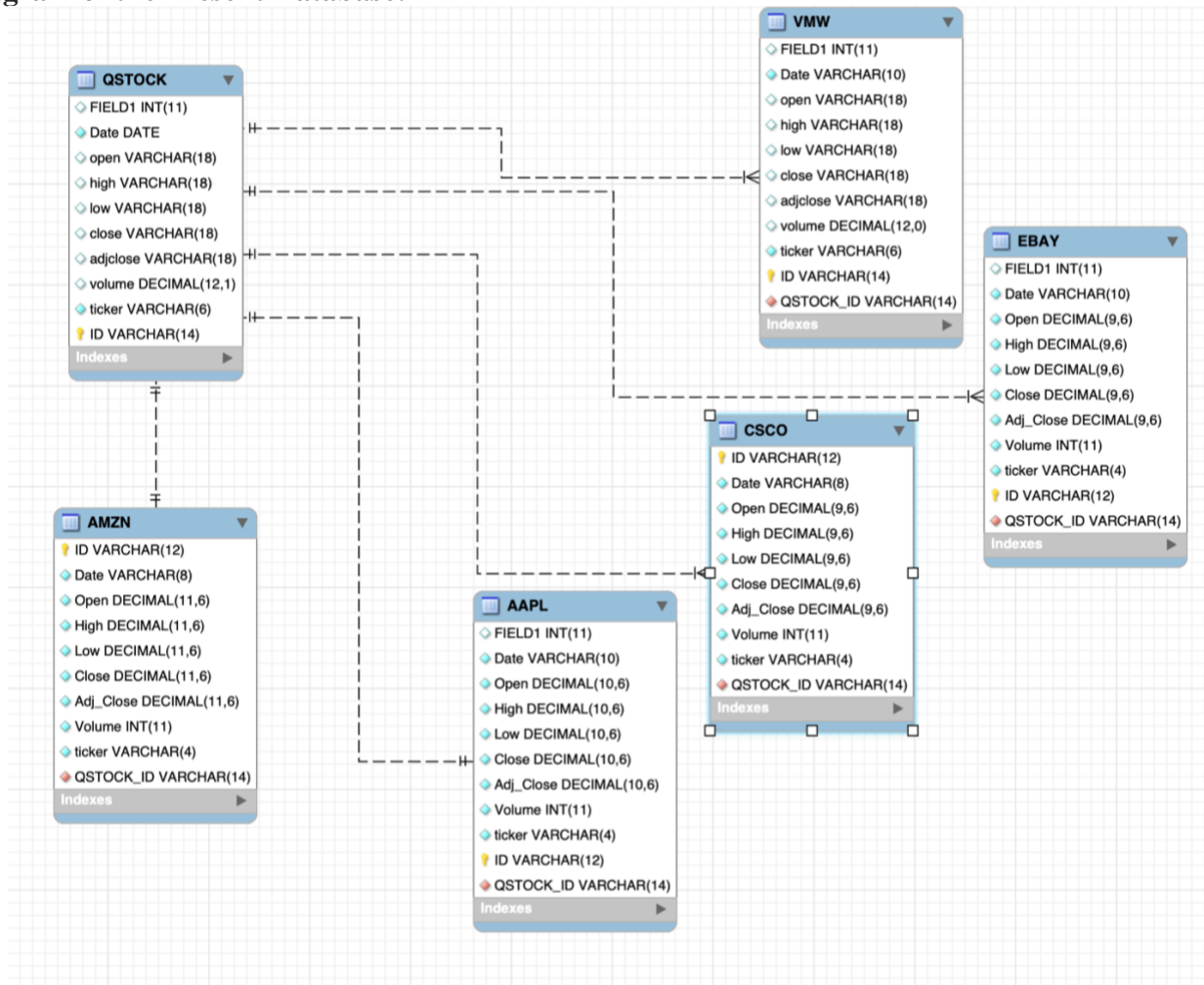


```

1 DROP TABLE IF EXISTS CSC0;
2 CREATE TABLE IF NOT EXISTS CSC0(
3     ID VARCHAR(12) NOT NULL PRIMARY KEY
4     ,Date VARCHAR(8) NOT NULL
5     ,Open NUMERIC(9,6) NOT NULL
6     ,High NUMERIC(9,6) NOT NULL
7     ,Low NUMERIC(9,6) NOT NULL
8     ,Close NUMERIC(9,6) NOT NULL
9     ,Adj_Close NUMERIC(9,6) NOT NULL
10    ,Volume INTEGER NOT NULL
11    ,ticker VARCHAR(4) NOT NULL
12 );
13 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004032000','4/3/00',76.875,77.72,72.9375,53.107918,61832400,'CSC
14 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004042000','4/4/00',74.74,5.64,73.125,53.244457,10846100,'CSC0')
15 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004052000','4/5/00',70.25,74.875,70.125,72.125,52.516327,6497540
16 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004062000','4/6/00',73.875,74.25,70.75,71.8125,52.288784,4382390
17 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004072000','4/7/00',73.75,71.625,74.9375,54.564171,44893800,'CSC
18 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004102000','4/10/00',75.4375,76.72,5.72.5625,52.834805,42264200,'CSC
19 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004112000','4/11/00',72.125,73.6875,68.5,70.50.969048,58865900,'CSC
20 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004122000','4/12/00',69.75,70.6875,64.75,65.47.3284,94231500,'CSC
21 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004132000','4/13/00',66.109375,66.5,61.125,61.125,44.506886,1040
22 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004142000','4/14/00',60.875,61.0625,55.0625,57.41.503365,1457530
23 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004172000','4/17/00',56.9375,66.625,56.671875,66.5,48.420582,120
24 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004182000','4/18/00',68.125,69.5,67.69.25,50.422951,80723900,'CSC
25 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004192000','4/19/00',69.75,70.65.875,66.125,48.147541,51863900,'CSC
26 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004202000','4/20/00',66.75,67.6875,64.5625,65.125,47.419415,3655
27 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004242000','4/24/00',61.5625,63.75,59.25,63.4375,46.190712,58933
28 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004252000','4/25/00',65.125,66.0625,63.0625,66.48.056534,5435760
29 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004262000','4/26/00',66.3125,69.65.875,66.75,48.602631,51055200,'CSC
30 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004272000','4/27/00',64.25,69.125,64.69,50.240917,51540300,'CSC0
31 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC004282000','4/28/00',69.375,70.68.0625,69.328125,50.479832,34352
32 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC005012000','5/1/00',69.875,71.875,69.5,71.4375,52.015736,4069420
33 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC005022000','5/2/00',71.71.6875,67.625,68.49.512806,31282300,'CSC
34 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC005032000','5/3/00',67.5625,67.625,63.625,66.0625,48.102055,4731
35 INSERT INTO CSC0(ID,Date,Open,High,Low,Close,Adj_Close,Volume,ticker) VALUES ('CSC005042000','5/4/00',65.4375,65.5,63.5,63.635,46.327221,40026500

```

## ER Diagram of the Present Database:



## Define Primary Keys:

A primary key is a special relational database table column (or combination of columns) designated to uniquely identify each table record. A primary key is used as a unique identifier to quickly parse data within the table. A table cannot have more than one primary key.

	Date	open	high	low	close	adjclose	volume	ticker
0	2021-01-04	11.125	11.25	10.500	10.80	10.80	150500.0	IONQ
1	2021-01-05	10.850	11.83	10.750	11.11	11.11	165300.0	IONQ
2	2021-01-06	11.500	11.50	11.000	11.05	11.05	175600.0	IONQ
3	2021-01-07	11.500	11.50	11.050	11.11	11.11	94200.0	IONQ
4	2021-01-08	11.650	12.35	11.000	11.01	11.01	137500.0	IONQ
5	2021-01-11	11.800	11.80	10.800	11.17	11.17	162600.0	IONQ
6	2021-01-12	11.800	11.80	11.070	11.18	11.18	210700.0	IONQ
7	2021-01-13	11.340	11.37	10.970	11.20	11.20	386100.0	IONQ
8	2021-01-14	11.310	11.70	11.225	11.44	11.44	291000.0	IONQ
9	2021-01-15	11.570	11.79	11.110	11.22	11.22	269600.0	IONQ

From the above figure we can see that we have date which can be used as a primary key, But the issue was that we have repeating dates as the data has the information on several repeating dates with respect to the different ticker values, So we combine the date and the ticker values to create a new column and called 'ID' which will serve as our Primary key in our database and will be unique with each record :

```
def format(date):  
    date = date.strftime('%m%d%Y')  
    return date
```

```
data['ID'] = data['ticker'] + data['Date'].map(format)
```

```
data.head(10)
```

	Date	open	high	low	close	adjclose	volume	ticker	ID
0	2021-01-04	11.125	11.25	10.500	10.80	10.80	150500.0	IONQ	IONQ01042021
1	2021-01-05	10.850	11.83	10.750	11.11	11.11	165300.0	IONQ	IONQ01052021
2	2021-01-06	11.500	11.50	11.000	11.05	11.05	175600.0	IONQ	IONQ01062021
3	2021-01-07	11.500	11.50	11.050	11.11	11.11	94200.0	IONQ	IONQ01072021
4	2021-01-08	11.650	12.35	11.000	11.01	11.01	137500.0	IONQ	IONQ01082021
5	2021-01-11	11.800	11.80	10.800	11.17	11.17	162600.0	IONQ	IONQ01112021
6	2021-01-12	11.800	11.80	11.070	11.18	11.18	210700.0	IONQ	IONQ01122021
7	2021-01-13	11.340	11.37	10.970	11.20	11.20	386100.0	IONQ	IONQ01132021
8	2021-01-14	11.310	11.70	11.225	11.44	11.44	291000.0	IONQ	IONQ01142021
9	2021-01-15	11.570	11.79	11.110	11.22	11.22	269600.0	IONQ	IONQ01152021

We can see a new column in our data frame called ID that had been created which will serve as our Primary Key in the database ID column contains a unique value for each row of data. It does not contain null values. Every row has a primary key value now.

## How the data is Imported:

For analysis we used Pandas to import the csv and load the data into a data frame:

```
import pandas as pd
import datetime as dt
```

```
data = pd.read_csv('quantumstock1.csv')
```

```
data['Date'] = pd.to_datetime(data['Date'])
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 136518 entries, 0 to 136517
Data columns (total 8 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Date        136518 non-null  datetime64[ns]
1   open        136473 non-null  float64
2   high        136473 non-null  float64
3   low         136473 non-null  float64
4   close       136473 non-null  float64
5   adjclose    136473 non-null  float64
6   volume      136473 non-null  float64
7   ticker      136518 non-null  object
dtypes: datetime64[ns](1), float64(6), object(1)
memory usage: 8.3+ MB
```

After the analysis and creation of Primary Key we dropped the 19 duplicate values in ID to maintain uniqueness in the data and dropped NULL values in the ID column which when checked were none but to be sure checked and dropped if any remained after alterations in the data frame.

```
data.head()
```

	Date	open	high	low	close	adjclose	volume	ticker	ID
0	2021-01-04	11.125	11.25	10.50	10.80	10.80	150500.0	IONQ	IONQ01042021
1	2021-01-05	10.850	11.83	10.75	11.11	11.11	165300.0	IONQ	IONQ01052021
2	2021-01-06	11.500	11.50	11.00	11.05	11.05	175600.0	IONQ	IONQ01062021
3	2021-01-07	11.500	11.50	11.05	11.11	11.11	94200.0	IONQ	IONQ01072021
4	2021-01-08	11.650	12.35	11.00	11.01	11.01	137500.0	IONQ	IONQ01082021

```
data = data.drop_duplicates(subset='ID')
```

```
data = data.dropna(subset=['ID'])
```

```
data.isnull().sum()
```

```
Date      0
open      45
high      45
low       45
close     45
adjclose  45
volume    45
ticker    0
ID        0
dtype: int64
```

```
data.duplicated().sum()
```

```
0
```

To insert the data into the database we wrote queries and saved them as .sql scripts, through which we will enter the data into the Database, Following is a picture of one such table data being ingested into database :

```

1 DROP TABLE IF EXISTS VMW;
2 CREATE TABLE IF NOT EXISTS VMW(
3     FIELD1    INTEGER
4     ,Date      VARCHAR(10) NOT NULL
5     ,open      VARCHAR(18)
6     ,high      VARCHAR(18)
7     ,low       VARCHAR(18)
8     ,close     VARCHAR(18)
9     ,adjclose  VARCHAR(18)
10    ,volume    NUMERIC(12,0)
11    ,ticker    VARCHAR(6) NOT NULL
12    ,ID        VARCHAR(14) NOT NULL PRIMARY KEY
13 );
14 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-03','42.900002','48.060001','42.330002','47.970001','32.68756
15 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-04','47.0','52.0','46.279999','50.490002','34.404732',2589700
16 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-07','51.950001','52.25','49.709999','50.59','34.472874',15676
17 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-08','49.939999','54.139999','49.0','53.27','36.299072',228140
18 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-09','54.0','54.889999','51.669998','52.599998','35.842522',20
19 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-10','52.029999','56.150002','52.029999','53.91','36.735176',1
20 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-11','53.0','53.759998','51.799999','52.5','35.77438',1059300,
21 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-14','52.009998','52.48','51.200001','51.849998','35.331459',1
22 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-15','51.849998','52.5','51.349998','51.720001','35.242874',12
23 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-16','52.799999','55.0','52.799999','53.959999','36.769245',19
24 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-17','53.490002','55.619999','53.0','55.02','37.49155',1994400
25 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-18','56.849998','60.75','56.849998','58.470001','39.842438',3
26 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-21','58.990002','59.119999','55.630001','56.07','38.207035',2
27 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-22','56.09','59.299999','54.220001','58.02','39.535797',64555
28 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-23','64.629997','64.879997','60.549999','60.84','41.457394',7
29 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-24','60.34','61.75','58.349998','61.52','41.920757',2938300,'
30 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-25','61.549999','65.480003','61.16','64.839996','44.18306',25
31 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-28','65.199997','69.470001','65.07','67.330002','45.879791',2
32 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-29','67.970001','68.449997','66.5','68.440002','46.636166',17
33 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-04-30','68.410004','68.980003','66.5','66.639999','45.409615',33
34 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-05-01','66.639999','70.730003','65.550003','70.239998','47.86270
35 INSERT INTO VMW(Date,open,high,low,close,adjclose,volume,ticker,ID) VALUES ('2008-05-02','71.51','71.51','67.520000','68.760002','46.954218',1444100

```

## Data dictionary:

### QStock:

Attributes	PK	FK	Nullable
High	No	No	Yes
Low	No	No	Yes
Open	No	No	Yes
Close	No	No	Yes
Date	Yes	No	No
Ticker	Yes	No	No
ID	Yes	No	Yes

QSTOCK	◆ Date	varchar(8)		NO	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ open	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ high	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ low	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ close	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ adjclose	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ volume	decimal(12,1)		YES			select,insert,update,references
QSTOCK	◆ ticker	varchar(6)		NO	utf8	utf8_general_ci	select,insert,update,references
QSTOCK	◆ ID	varchar(14)		NO	utf8	utf8_general_ci	select,insert,update,references



**VMW:**

Attributes	PK	FK	Nullable
High	No	No	Yes
Low	No	No	Yes
Open	No	No	Yes
Close	No	No	Yes
Date	Yes	No	No
Ticker	Yes	No	No
ID	Yes	No	Yes

VMW	◆ open	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ high	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ low	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ close	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ adjclose	varchar(18)		YES	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ volume	decimal(12,0)		YES			select,insert,update,references
VMW	◆ ticker	varchar(6)		NO	utf8	utf8_general_ci	select,insert,update,references
VMW	◆ ID	varchar(14)		NO	utf8	utf8_general_ci	select,insert,update,references

Qstock is the table which contains the stock information all the companies which have invested in Quantum computing and VMW table is contain stock of VMware company which does not invest in Quantum computing along with have tables such as AAPL, CSCO, EBAY and EA as other companies we can compare with .