**Database Design**

**Stock Emulator classes:**

* Class Account:
* Investment: decimal
* AvailableCash: decimal
* Total Trans: int
* PositiveTrans: int
* NegativeTrans: int
* **Username**: varchar
* Password\_MD5: varchar
* Fullname: varchar
* FirstSecurityQuestion: varchar
* FirstSecurityAnswer: varchar
* SecondSecurityQuestion: varchar
* SecondSecurityAnswer: varchar
* Class Stock:
* **Ticker**: varchar (primary key)
* Name: varchar
* Price: decimal
* PrevClosePrice: decimal (the last close price)
* HighPrice: decimal
* LowPrice: decimal
* OpenPrice: decimal
* Volume: decimal
* Change: decimal
* MarketCap: decimal
* [52-week\_High]: decimal
* [52-week\_Low]: decimal
* AskPrice: decimal
* BidPrice: decimal
* AskSize: decimal
* BidSize: decimal
* [1-Year\_Return]: decimal
* Beta: decimal
* PE\_Ratio: decimal
* Dividend: decimal
* DividendPercent: decimal
* UpdateChecker: int
* Class Transactions:
* **ID**: bigint (**primary key**)
* User: Account
* Ticker: varchar
* Date: date
* Type: varchar (Buy/Sell) (0/1: bit)
* Num: bigint
* Price: decimal
* **AvgBuyPrice (not Cost in Portfolio table)**: decimal
* **GainLoss**: decimal
* **GLPercent**: decimal

Given:

* + Price (current price, $\share)
  + Num (number of stocks traded on a transaction)

If **Type** is Buy:

* + Total = Price \* Num – 10
  + Compute **AgrBuyPrice:**

**AvgBuyPrice = (∑ BuyPricei \* Numi) / ∑Numi**

* + - BuyPricei : buy price on transaction i
    - Numi : number of stocks purchased on transaction i

Or update:

**AvgBuyPrice <- (Numalready-possessed \* AvrBuyPrice + Numpurchased \* Price) / (Numalready-possessed + Numpurchased**)

If **Type** is Sell:

* + Total = Price \* Num + 10
  + **GainLoss = (Price – AvgBuyPrice) \* Num**
  + **GLPercent = (Price – AvgBuyPrice) / (AvgBuyPrice)**
* Class WatchStock: list of stocks that the user is paying attention to
* Stock (**Ticker**): Stock
* **Username**: Account
* Class Porfolio: update information on stocks that the user is possessing
* **Username**: Account
* **Ticker**: Stock
* Cost (**AvgBuyPrice**): decimal
* Num: bigint
* Class InsiderTrades: list of illegal trades
* **ID**: bigint (primary key)
* Tick: varchar
* InsiderDetail: varchar
* CompanyName: varchar
* Type: varchar
* Quant: bigint
* Price: decimal
* Total: decimal
* Time: datetime
* Class Settings:
* AutoUpdateTimer: int
* EnforceMarketTiming: true/false
* Class History:
* **Ticker**: varchar
* **Time**: datetime
* HistoryPrice: decimal