**Database Design**

**Stock Emulator classes:**

* Class Account:
* **ID**: bigint **(primary key)**
* Investment: float
* AvailableCash: float
* Total Trans: int
* PositiveTrans: int
* NegativeTrans: int
* Username: varchar
* Password\_MD5: varchar
* Class Stock:
* **Ticker**: varchar (primary key)
* Name: varchar
* Price: float
* PrevClosePrice: float (the last close price)
* PriceChange: float (Price – PrevClosePrice)
* PriceChangePercent: float
* Class Transactions:
* **ID**: bigint (**primary key**)
* User: Account
* Ticker: varchar
* Date: date
* Type: varchar (Buy/Sell) (0/1: bit)
* Num: bigint
* Price: float
* **AvrBuyPrice (Cost in Porfolio table)**: float

Given:

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

If **Type** is Buy:

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

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

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

Or update:

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

If **Type** is Sell:

* + Total = Price \* Num + 10
  + **G\L = (Price – AvrBuyPrice) \* Num**
  + **G\L % = (Price – AvrBuyPrice) / (AvrBuyPrice)**
* Class WatchStock: list of stocks that the user is paying attention to
* **ID**: bigint (primary key)
* Stock: Stock
* User: Account
* Class Porfolio: update information on stocks that the user is possessing
* **ID**: bigint (primary key)
* User: Account
* Cost (**AvrBuyPrice**): float
* Num: bigint
* Stock: Stock
* Class InsiderTrades: list of illegal trades
* **ID**: bigint (primary key)
* Tick: varchar
* InsiderDetail: varchar
* CompanyName: varchar
* Type: varchar
* Quant: bigint
* Price: float
* Total: float
* Time: datetime
* Class Settings:
* **AccountID**: bigint (primary key)
* AutoUpdateTimer: int
* EnforceMarketTiming: true/false
* AllowThemeSupport: true/false