



Entity Sets

Accounts = (ID, username, password, creation date, cash)

Lots = (ID, Lot_num, Ticker, purchase_price, num_shares, purchase_date)

Stocks = (Ticker, Company_name)

Price_history=(Ticker, Date Time, price)

View Tables

Used to build the display on Home

Account_value
ID
AccountValue

Join of Accounts and
Lots_value where
Accounts.ID = Lots_value.ID

Account value:
Cash + sum(TotalValue)
For each lot owned by
The Account

Lots_value
lot_num
ID
date
ticker
shares
Price today
Price yesterday
TotalValue

Join of Lots, price_history,price_history

Used on our Stocks display

price_movement
Ticker
date
price today
Price yesterday
Price three days ago

Join of price history, price_history,price_history

```
create view lot_value(TotalValue,ticker,Price,Previous,Shares,Basis,Lot,LotOwner,l_date) as
SELECT (num_shares*currentPrice),ticker,CurrentPrice,Yesterday,num_shares,(num_shares*purchase_price),lot_num,id,pDate
from lots NATURAL JOIN (select p.ticker, p.price as currentPrice, p.price_date as pDate, o.price as Yesterday
    from price_history as p left outer join price_history as o
    on o.price_date = DATE_SUB(p.price_date,INTERVAL 1 DAY) AND p.ticker = o.ticker) as stockStats
where purchase_date <= stockStats.pDate;
```

```
create view account_value(accVal,id,aDate) as
SELECT IF(TotalValue IS NOT NULL,sum(TotalValue)+cash, cash), id, l_date
from lot_value right outer JOIN account on lot_value.LotOwner=account.id
GROUP BY id,l_date;
```

```
create view price_movement(ticker,price, pDate, one_day, three_day) as
select day_history.Ticker, day_history.today,day_history.pDate,day_history.yesterday, t.price as three_day from
(Select p.ticker as Ticker, p.price as today, p.price_date as pDate, o.price as yesterday
from price_history as p left outer join price_history as o
on o.price_date = DATE_SUB(p.price_date,INTERVAL 1 DAY) AND p.ticker = o.ticker) as day_history left outer join price_history as t
on t.price_date = DATE_SUB(pDate,INTERVAL 3 DAY) and t.ticker = day_history.Ticker;
```