# **PART 2 - READING ASSIGNMENT (Papers Attached)**

## **Q1** Write a .bat/.sh to import the entire NYSE dataset (stocks A to Z) into MongoDB.

## 

**CODE（store in each collection）:**

{

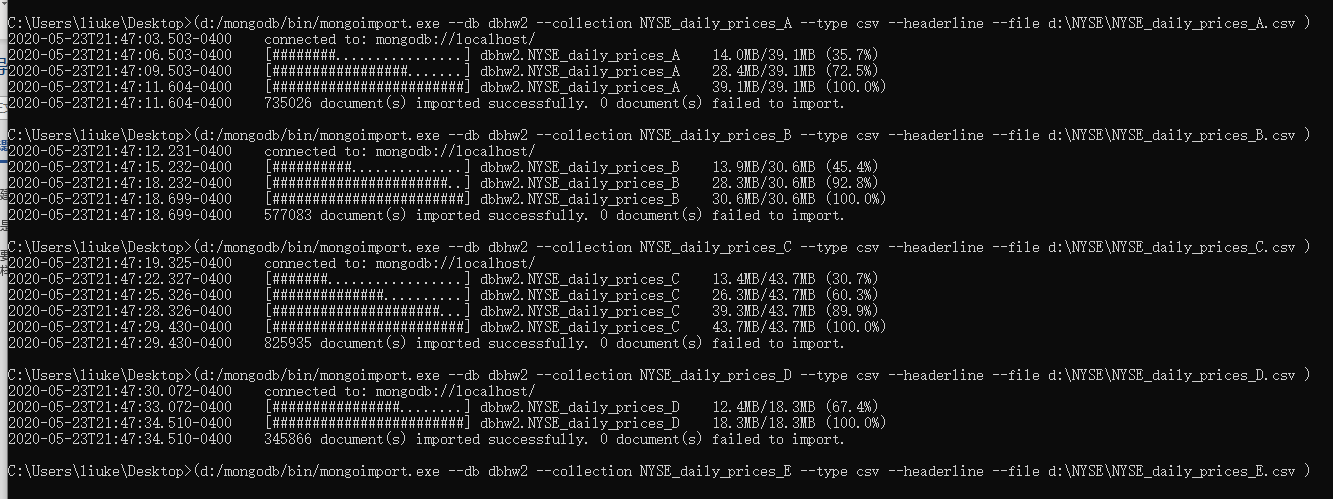
FOR %%C in (d:\NYSE\NYSE\_daily\_prices\_\*) do (d:/mongodb/bin/mongoimport.exe --db dbhw2 --collection %%~nC --type csv --headerline --file %%C)

Pause

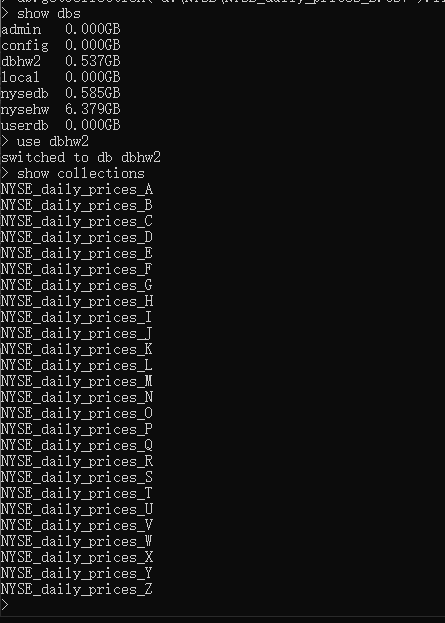
}

**Screenshot**

**Processing: Save in collection named with original data’s file name. (To avoid illegal collection name with symbol “ : ”**



**Result:**



## Code（store all .csv in one collection）

{

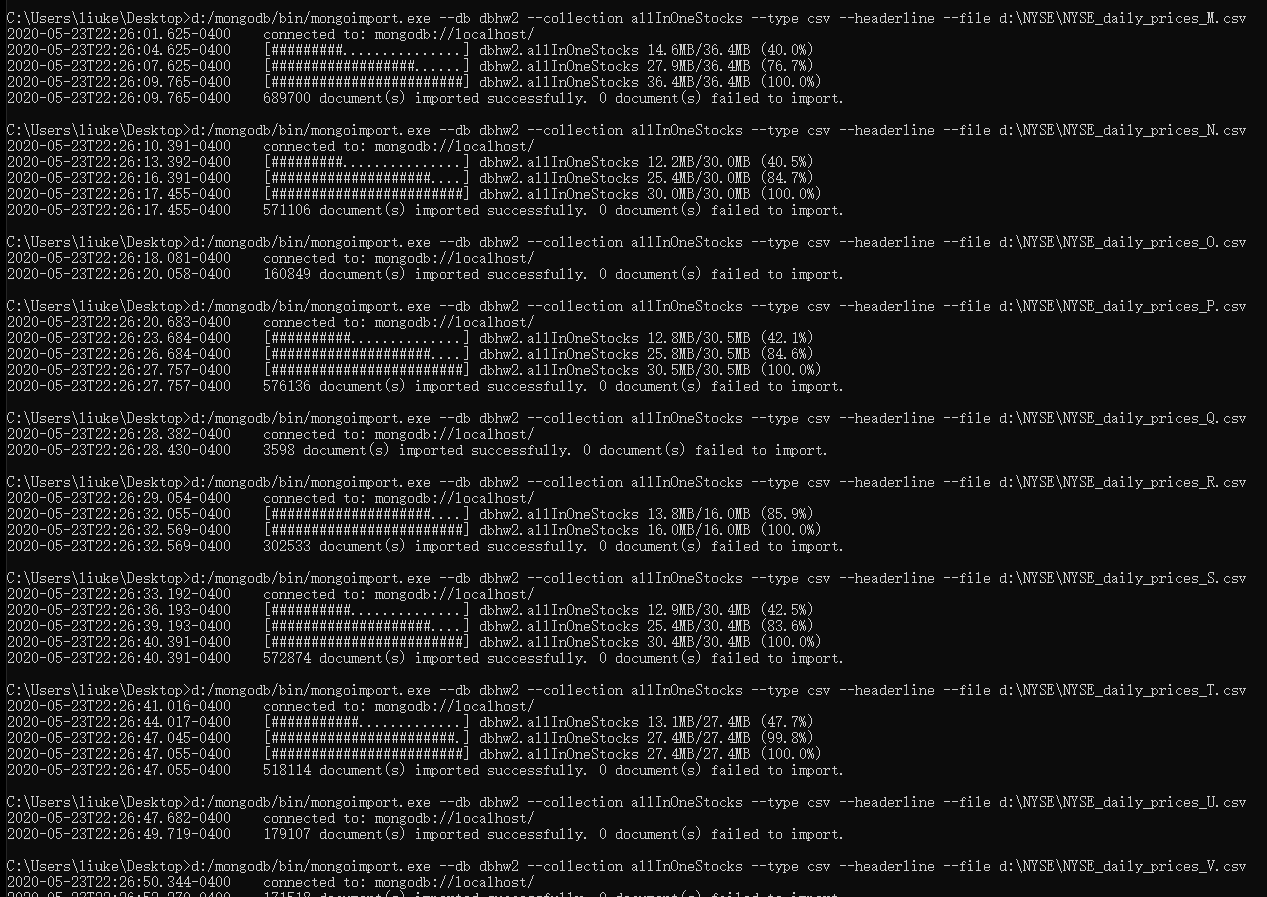
FOR %%C in (d:\NYSE\NYSE\_daily\_prices\_\*) do d:/mongodb/bin/mongoimport.exe --db dbhw2 --collection allInOneStocks --type csv --headerline --file %%C

Pause

}



**Screenshot:**



Result: 

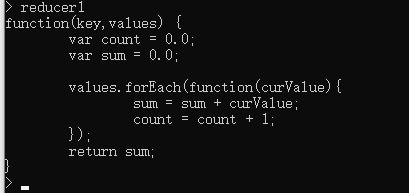
# **PART 3.1 - PROGRAMMING ASSIGNMENT**

# Use the NYSE database to find the average price of stock\_price\_high values for each stock using MapReduce.

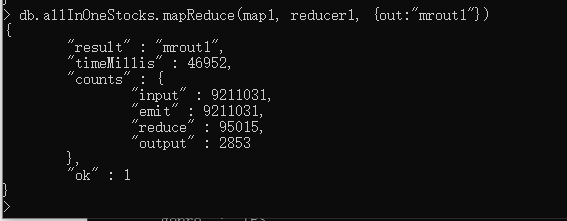
## Map function :

## 

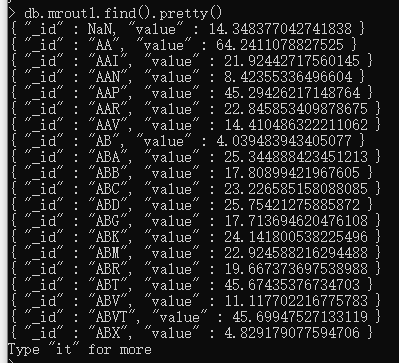
## Reducer function:



## MapReduce:



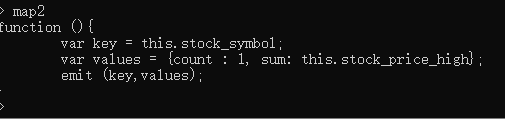
## Mrout1 collection:



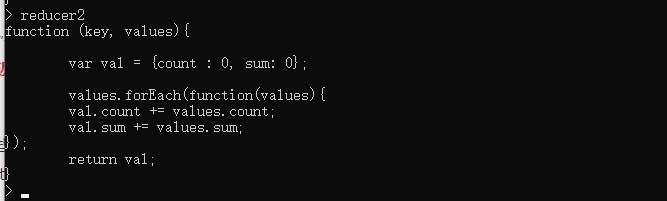
# **PART 3.2- PROGRAMMING ASSIGNMENT**

# Part 3.1 result will not be correct as AVERAGE is a commutative operation but nor associative. Use a FINALIZER to find the correct average. (Hint: pass sum and count from the reducer)

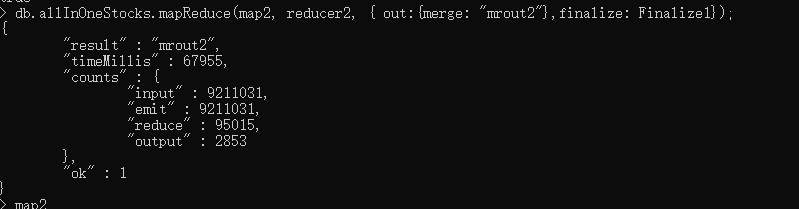
## Map function :



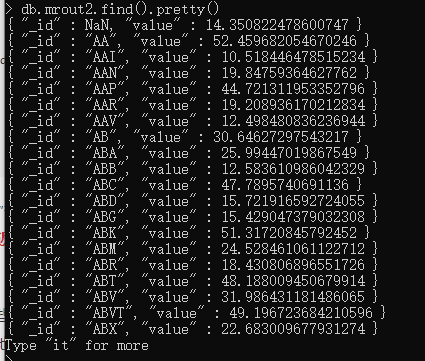
## Reducer function:



## MapReduce:



## Mrout2 collection:



# Conclusion :

It’s easy to see that there are many different values between Mrout1 and Mrout2. For example, the average value of stock\_price\_high in Mrout1 for Stock “AA” is 64.2. And the average value of stock\_price\_high in Mrout2 for Stock with the same name is 52.4. So we can prove that average operation is either not associative or commutative (we can confirm it is not associative by mathematic way). And if there are other operation who have same performance like average operation, the reducer can not be combiner. A finalizer is necessary.

# PART 5 - PROGRAMMING ASSIGNMENT

# Import the Movielens dataset into MongoDB. Refer to README about file contents and headings.

## .bat code :

{

d:/mongodb/bin/mongoimport.exe --db dbhw2 --collection movies --type csv --fields="MovieID.int32(),Title.string(),Genres.string()" --file d:/ml-1m/movies.dat

d:/mongodb/bin/mongoimport.exe --db dbhw2 --collection users --type csv --fields="UserID.int32(),Gender.string(),Age.int32(),Occupation.int32(),Zip-code.int32()" --file d:/ml-1m/users.dat

d:/mongodb/bin/mongoimport.exe --db dbhw2 --collection ratings --type csv --fields="UserID.int32(),MovieID.int32(),Rating.int32(),Timestamp.int32()"--file d:/ml-1m/ratings.dat

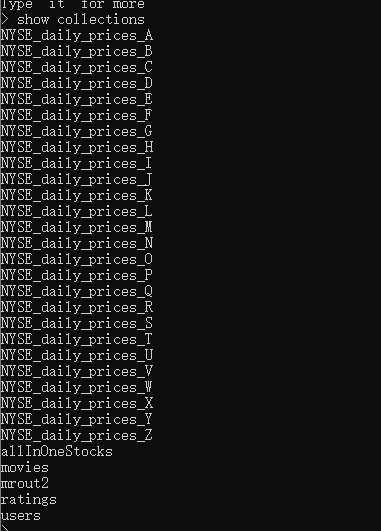
pause

}

## Screenshot

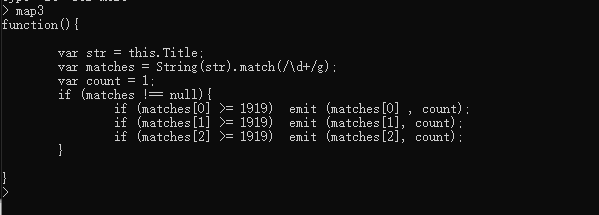


## Result



# Q1 Find the number Females and Males from the users collection using MapReduce. Do the same thing using count() to compare the results.

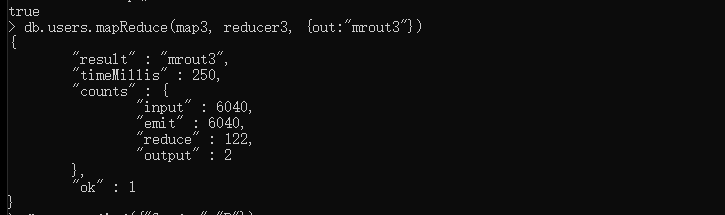
## MAP:



## Reduce：

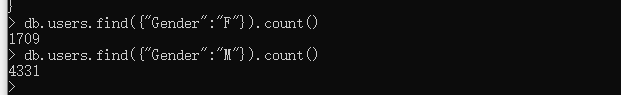


## MapReduce



MRout3：

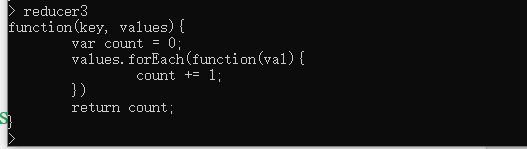
## Using Function count（）



# Q2 Find the number of Movies per year using MapReduce

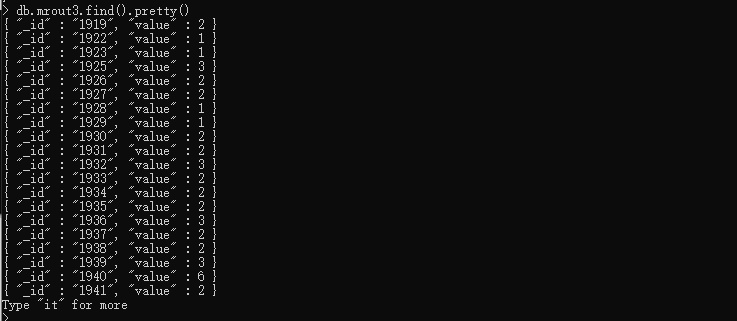
## MAP:

## Reduce：



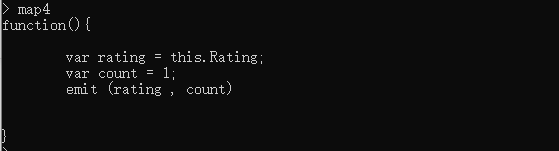
## MapReduce

MRout

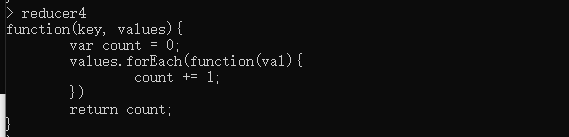


# Q3 Find the number of Movies per rating using MapReduce

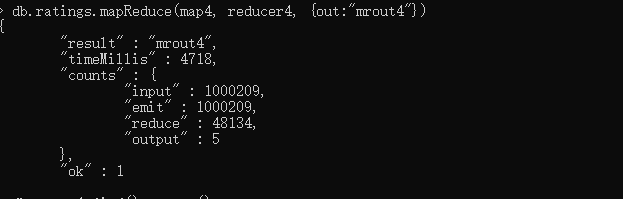
## Map：



## Reduce:



## MapReduce:



## MRout4：

