**Assignment 2**

**PART 2 - PROGRAMMING ASSIGNMENT**

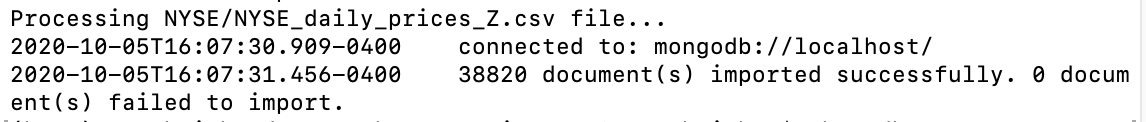
 Write a .bat/.sh to import the entire NYSE dataset (stocks A to Z) into MongoDB.

NYSE Dataset Link: http://msis.neu.edu/nyse/

Here is my shell script:

1. !/bin/bash
2. FILES=NYSE/NYSE\_daily\_prices\_\*.csv
3. **for** f in $FILES
4. **do**
5. echo "Processing $f file..."
6. # ls -l $f
7. mongoimport --db nysedb --collection stocks --type csv --headerline --file $f
8. done

Here is the output

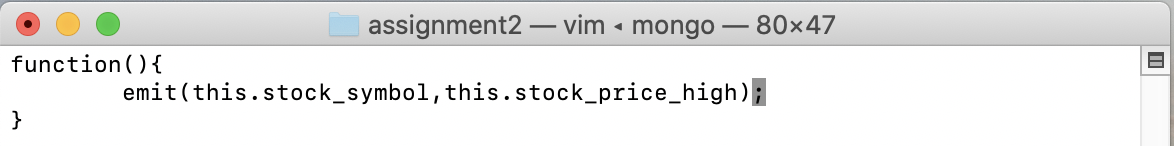


After import use show dbs and count to confirm results:

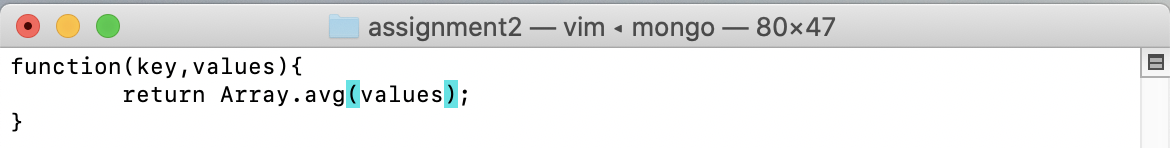


**PART 3.1. Use the NYSE database to find the average price of stock\_price\_high values for each stock using MapReduce.**

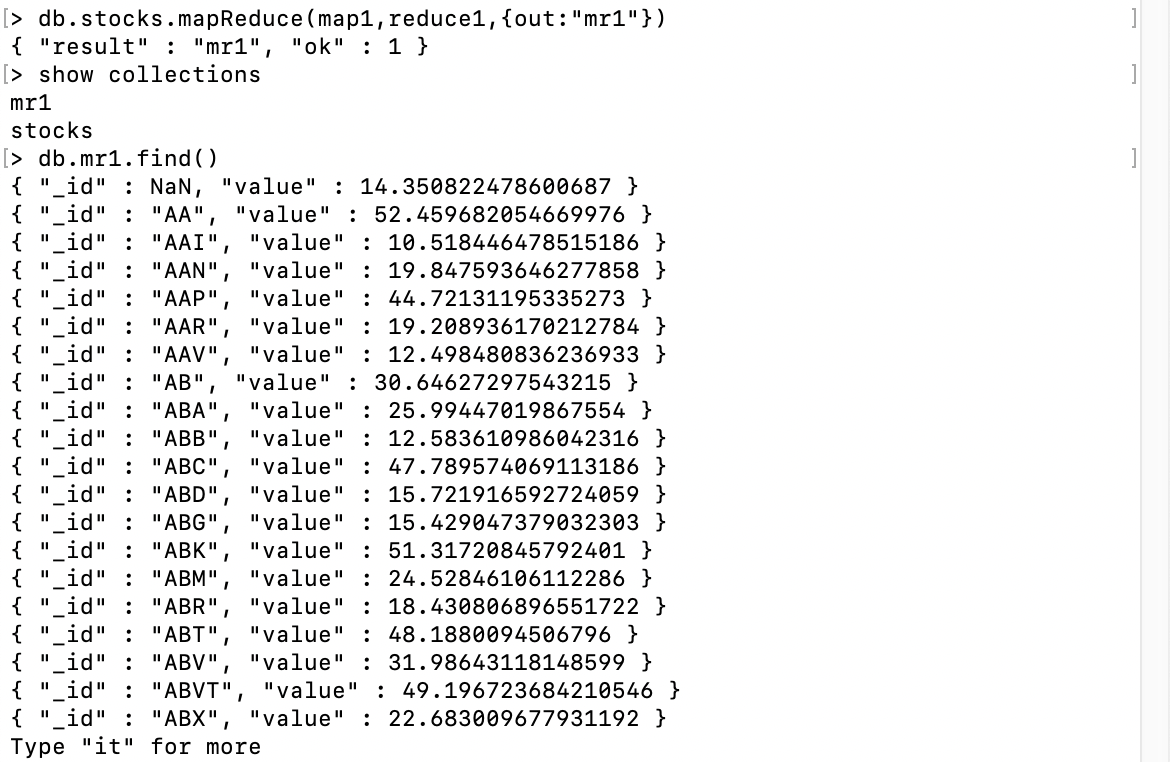
 First the map function:



And the reduce function:



This is part of results:

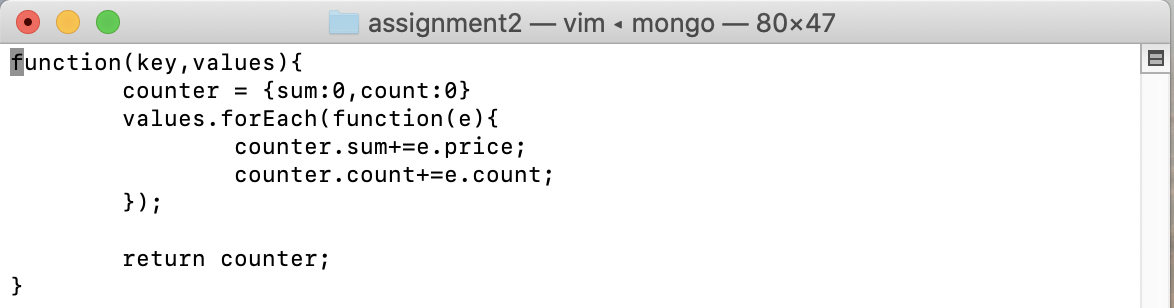


**PART 3.2. Part 3.1 result will not be correct as AVERAGE is a commutative operation but nor associative. Use a FINALIZER to find the correct average. (Hint: pass sum and count from the reducer)**

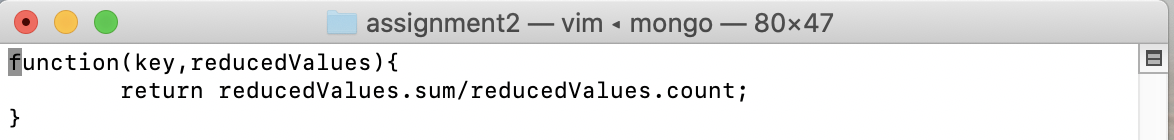
The count is for counting prices



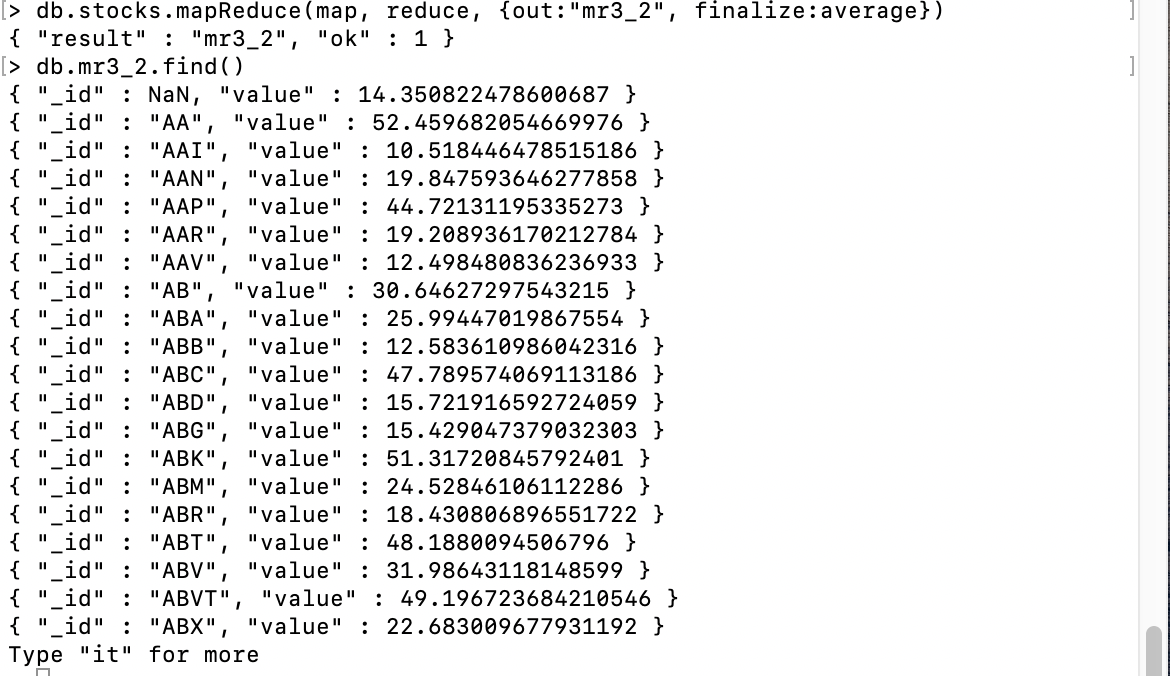
Add price



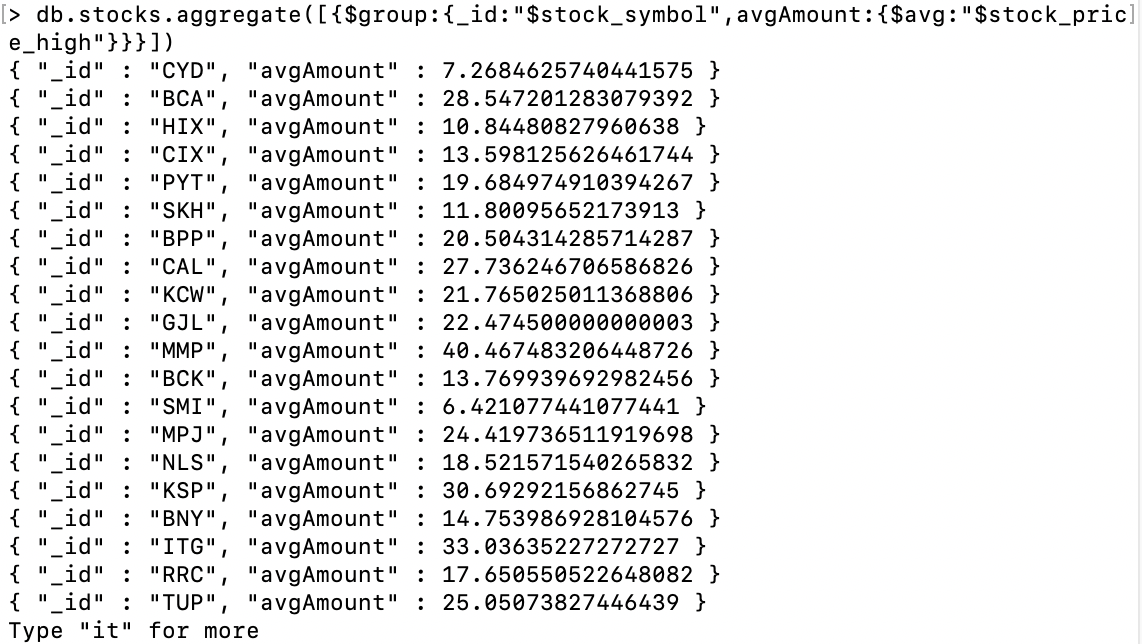
Here is finalizer:



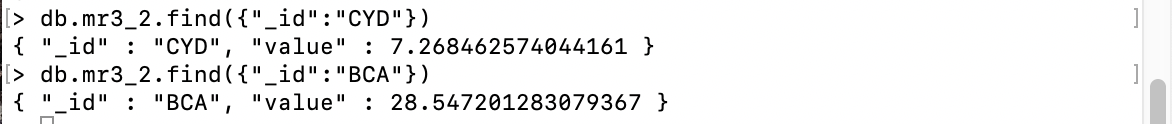
Here is part of results



**PART 4. Calculate the average stock price of each price of all stocks using $avg aggregation.**

****

Compare with previous calculation

****

**PART 5 - PROGRAMMING ASSIGNMENT**

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

[**https://grouplens.org/datasets/movielens/1m/ (Links to an external site.)**](https://grouplens.org/datasets/movielens/1m/)**[you may replace :: in the dateset with comma or tab to import]**

Import to databace :lens

1. **import** java.io.File;
2. **import** java.io.FileNotFoundException;
3. **import** java.util.Map;
4. **import** java.util.Scanner;
5. **import** java.util.function.Consumer;
6. **import** com.mongodb.\*;
7. **import** com.mongodb.MongoClient;
8. **import** com.mongodb.client.MongoCollection;
9. **import** com.mongodb.client.MongoDatabase;
10. **import** com.mongodb.MongoClientURI;
11. **import** com.mongodb.ServerAddress;
12. **import** com.mongodb.MongoCredential;
13. **import** com.mongodb.MongoClientOptions;
14. **import** org.bson.Document;
16. **public** **class** MovieImport {
17. **private** **static** **final** String host = "localhost";
18. **private** **static** **final** **int** port = 27017;
19. **private** **static** **final** String databaseName = "lens";
20. **public** **static** **int** importToMongo(String collectionName, String fileFullPath, String header, String delimeter) {
21. MongoClient client = **new** MongoClient(host, port);
22. MongoDatabase database = client.getDatabase(databaseName);
23. MongoCollection<Document> collection = database.getCollection(collectionName);
24. Document document = **new** Document();
26. **int** counter = 0;
27. Scanner data;
28. **try** {
29. data = **new** Scanner(**new** File(fileFullPath));
30. **if** (header == **null**)
31. header = data.nextLine();
32. String[] names = header.split(delimeter);
34. **while** (data.hasNextLine()) {
35. String[] columns = data.nextLine().split(delimeter);
36. **for** (**int** i = 0; i < names.length; i++) {
37. document.put(names[i], columns[i]);
38. }
39. collection.insertOne(document);
40. document.clear();
41. counter++;
42. }
43. data.close();
44. } **catch** (FileNotFoundException e) {
45. counter = -1;
46. e.printStackTrace();
47. }
49. client.close();
51. **return** counter;

54. }
55. **private** **static** **void** loadUsers() {
56. String header = "UserID::Gender::Age::Occupation::Zip-code";
57. String collection = "users";
58. **int** n = importToMongo(collection, "/Users/wangbaichao/Desktop/INFO7250/Assignments/assignment2/ml-1m/users.dat", header, "::");
59. System.out.println("Successfully imported: " + n + " documents to " + collection);
60. }
62. **private** **static** **void** loadRatings() {
63. String header = "UserID::MovieID::Rating::Timestamp";
64. String collection = "ratings";
65. **int** n = importToMongo(collection, "/Users/wangbaichao/Desktop/INFO7250/Assignments/assignment2/ml-1m/ratings.dat", header, "::"); // data file path
66. System.out.println("Successfully imported: " + n + " documents to " + collection);
67. }

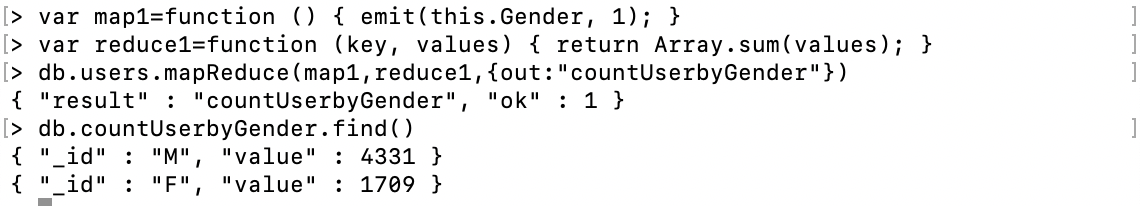

71. **private** **static** **void** loadMovies() {
72. String header = "MovieID::Title::Year::Genres";
73. String collection = "movies";
74. **int** n = importToMongo(collection, "/Users/wangbaichao/Desktop/INFO7250/Assignments/assignment2/ml-1m/movies.dat", header, "::");
75. System.out.println("Successfully imported: " + n + " documents to " + collection);
76. }
78. **public** **static** **void** loadDataToMongo() {
79. loadRatings();
80. loadUsers();
81. loadMovies();
82. }
84. }

Then check the import:

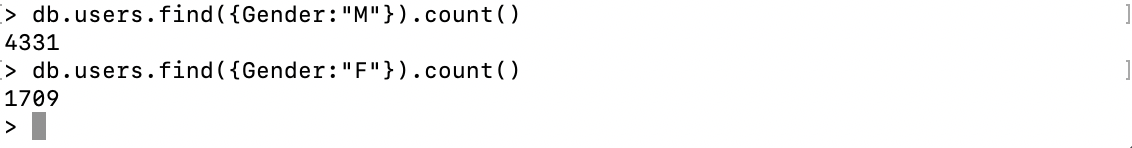


* **Find the number Females and Males from the users collection using MapReduce. Do the same thing using count() to compare the results.**

Use MapReduce:

****

Use count()

****

The results are same;

* **Find the number of Movies per year using MapReduce**

Map3:

function () { emit(this.Year,{'MovieID':this.MovieID,'count':1}); }

Reduce3:

function(key,values){

var res={count:0}

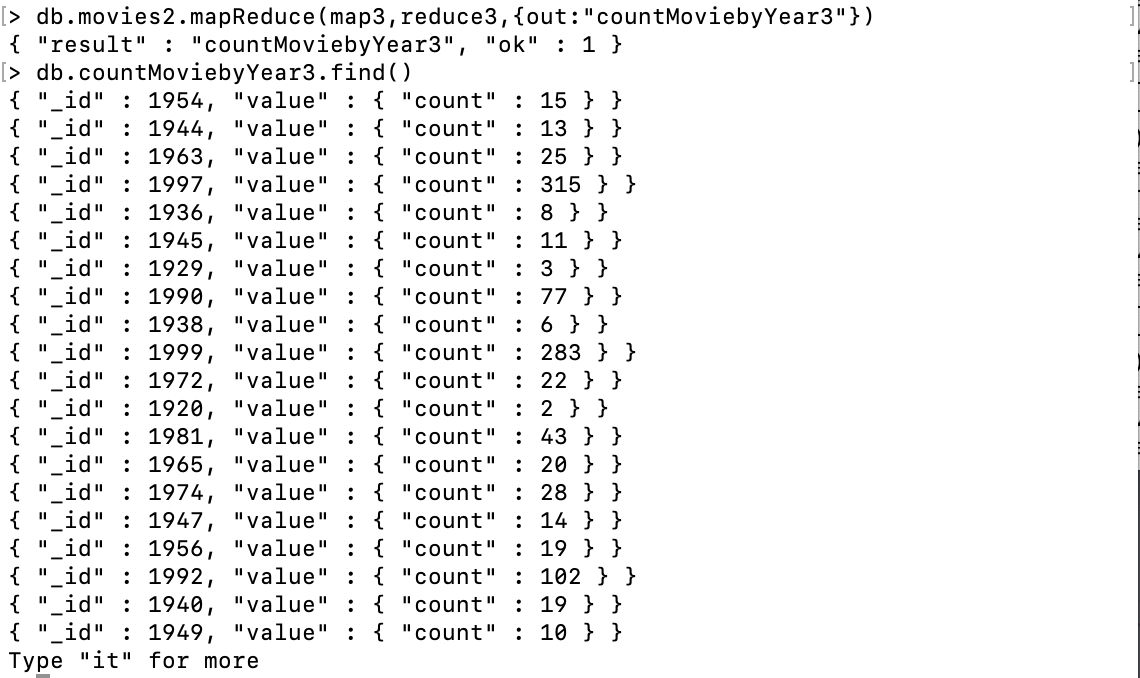
for(i=0;i<values.length;i++){

res.count+=values[i].count

}

return res;

}

****

* **Find the number of Movies per rating using MapReduce**

Map2:

function () { emit(this.Rating,{'MovieID':this.MovieID,'count':1}); }

reduce2:

function(key,values){

var res={count:0}

for(i=0;i<values.length;i++){

res.count+=values[i].count

}

return res;

}

Result:

