Assignment - 2
Big Data Systems

(S1-23_CCZG522)

Group - 35

S.No	Group Members	BITS ID
01	Praveen J Anand	2023MT03132
02	Vishnuvarthan N	2023MT03047
03	Shashank A	2023MT03028
04	Giridhara Guptha V R	2023MT03075
05	Vishaal Aravinth	2023MT03115

Table of Contents

1. Data Set

Sample Screenshot in xlsx format :

S.No	Country	Brand	Points	Price	Province	Region 1	Region 2	Variety	Winery
0	US	Martha's Vineyar	96	235	California	Napa Valley	Napa	Cabernet Sauvignon	Heitz
1	Spain	Carodorum Selec	96	110	Northern Spain	Toro		Tinta de Toro	Bodega Carmen RodrÃ-guez
2	US	Special Selected L	96	90	California	Knights Valley	Sonoma	Sauvignon Blanc	Macauley
3	US	Reserve	96	65	Oregon	Willamette Valley	Willamette	Pinot Noir	Ponzi
4	France	La BrÃ»lade	95	66	Provence	Bandol		Provence red blend	Domaine de la BÃ©gude
5	Spain	Numanthia	95	73	Northern Spain	Toro		Tinta de Toro	Numanthia
6	Spain	San RomÃ¡n	95	65	Northern Spain	Toro		Tinta de Toro	Maurodos
7	Spain	Carodorum Ãšnic	95	110	Northern Spain	Toro		Tinta de Toro	Bodega Carmen RodrÃ-guez
8	US	Silice	95	65	Oregon	Chehalem Mounta	Willamette	Pinot Noir	BergstrÃ¶m
9	US	Gap's Crown Vine	95	60	California	Sonoma Coast	Sonoma	Pinot Noir	Blue Farm
10	Italy	Ronco della Chies	95	80	Northeastern It	Collio		Friulano	Borgo del Tiglio
11	US	Estate Vineyard V	95	48	Oregon	Ribbon Ridge	Willamette	Pinot Noir	Patricia Green Cellars

JSON FORMAT :


```
[
    {
        "_id": "0",
        "Country": "US",
        "Brand": "Martha's Vineyard",
        "Rating": "96",
        "Price": "235",
        "State": "California",
        "Province1": "Napa Valley",
        "Province2": "Napa",
        "Variety": "Cabernet Sauvignon",
        "Winery": "Heitz"
    },
    {
        "_id": "1",
        "Country": "Spain",
        "Brand": "Carodorum Selección Especial Reserva",
        "Rating": "96",
        "Price": "110",
        "State": "Northern Spain",
        "Province1": "Toro",
        "Province2": "",
        "Variety": "Tinta de Toro",
        "Winery": "Bodega Carmen Rodríguez"
    },
    {
        "_id": "2",
        "Country": "US",
        "Brand": "Special Selected Late Harvest",
        "Rating": "96",
        "Price": "90",
        "State": "California",
        "Province1": "Knights Valley",
        "Province2": "Sonoma",
        "Variety": "Sauvignon Blanc",
        "Winery": "Macauley"
    },
```


Link to complete dataset :

JSON: - For MongoDB Analysis

https://drive.google.com/file/d/1hLZdAV7iVx3E_K5f1v2s1CVVeP-QbToK/view?usp=sharing

Total Records : 15,347

CSV: - For PIG Analysis

data.csv - Google Drive

Total Records : 1,50,930

2. Analysis using PIG

S.NO	Analysis	Expected output	BITS ID
1	Analysis by country and the total unique number of wine varieties they manufacture	Total No of unique varieties available per country	2023MT03132
2	Analysis by country, province and number of wineries	Total no of wineries w.r.t Province and country	2023MT03047
3	Analysis Minimum and Maximum Price by Country	Minimum and Maximum price of Wine w.r.t Country	2023MT03028
4	Analysis of Average Rating based on Variety	Average Rating for each variety	2023MT03115
5	Analysis of Average price based on variety	Average price value for each variety	2023MT03075

Note: The input is provided as 'csv' file from the above Google drive link

2.1. Analysis 1 - Analysis by country and the total unique number of wine varieties they manufacture

2.1.1. Input & Output Format

Input Format : <ID, Country, Brand, Rating, Price, State, Province1, Province2, Variety, Winery>

Output Format : (**Country**, **Total count of Unique Wine Varieties** they manufacture)

2.1.2. Steps:

- **Start History Server :**

  cd $HADOOP_HOME/sbin
  ./mr-jobhistory-daemon.sh start historyserver

```
[centos@master sbin]$ jps
3602 DataNode
3477 NameNode
4869 NodeManager
794 QuorumPeerMain
4476 SecondaryNameNode
8508 Jps
4734 ResourceManager
[centos@master sbin]$ ls
distribute-exclude.sh  httpfs.sh              start-all.cmd      start-dfs.sh
FederationStateStore   kms.sh                 start-all.sh       start-secure-dns.
hadoop-daemon.sh       mr-jobhistory-daemon.sh  start-balancer.sh  start-yarn.cmd
hadoop-daemons.sh      refresh-namenodes.sh    start-dfs.cmd      start-yarn.sh
[centos@master sbin]$
[centos@master sbin]$ ./mr-jobhistory-daemon.sh start historyserver
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
[centos@master sbin]$ jps
3602 DataNode
3477 NameNode
4869 NodeManager
794 QuorumPeerMain
4476 SecondaryNameNode
4734 ResourceManager
8686 JobHistoryServer
8831 Jps
[centos@master sbin]$
```

- **Open PIG Shell**

  pig -x local

```
[centos@master sbin]$ pig -x local
2023-11-09 19:16:25,483 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2023-11-09 19:16:25,484 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2023-11-09 19:16:25,514 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016,
2023-11-09 19:16:25,514 [main] INFO  org.apache.pig.Main - Logging error messages to: /opt/hadoop-3.2.4/sbin/pig_1699
2023-11-09 19:16:25,635 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/centos/.pigbootup not
2023-11-09 19:16:25,872 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is depreca
2023-11-09 19:16:25,872 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated
2023-11-09 19:16:25,875 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to h
2023-11-09 19:16:25,982 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is depr
2023-11-09 19:16:26,006 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-c423006f-8
2023-11-09 19:16:26,006 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled s
grunt>
```

- **Load Data from CSV to HDFS & Describe & View it:**

```
grunt> wineRatingsData = LOAD '/home/centos/Public/data.csv' USING
PigStorage(',') as
 (ID : int ,
 Country : chararray,
 Brand : chararray,
 Rating : int,
 Price : int,
 State : chararray,
 Province1 : chararray,
 Province2 : chararray,
 Variety : chararray,
 Winery : chararray);
```

```
Details at logfile: /opt/hadoop-3.2.4/sbin/pig_1699557385511.log
grunt> wineRatingsData = LOAD '/home/centos/Public/data.csv' USING PigStorage(',') as (ID : int , Country : chararray, Brand : chararray,
te : chararray, Province1 : chararray, Province2 : chararray, Variety : chararray, Winery : chararray);
2023-11-09 19:21:24,820 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use
2023-11-09 19:21:24,821 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.def
grunt> describe wineRatingsData
wineRatingsData: {ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Province2: char
y: chararray}
grunt>
```

- **View the data**
  dump wineRatingsData

centos@master:/opt/hadoop-3.2.4/sbin

```
150900,Chile,Prima Reserva,81,13,Maipo Valley,,,Cabernet Sauvignon,De Martino)
150901,Chile,Reserva,81,12,Maipo Valley,,,Merlot,Undurraga)
150902,Chile,Estate Bottled,81,10,Maipo Valley,,,Chardonnay,De Martino)
150903,Chile,120,81,7,Rapel Valley,,,Cabernet Sauvignon,Santa Rita)
150904,Chile,,81,10,Maipo Valley,,,Cabernet Sauvignon,De Martino)
150905,Chile,Prima Reserva,80,13,Maipo Valley,,,Merlot,De Martino)
150906,France,Clos des Reas,93,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
150907,France,Les Beaux-Monts,92,52,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Daniel Rion)
150908,France,Aux Brulees,90,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
150909,France,Clos dea Argillieres,89,52,Burgundy,Nuits-St.-Georges,,Pinot Noir,Daniel Rion)
150910,France,,89,38,Burgundy,Chambolle-Musigny,,Pinot Noir,Michel Gros)
150911,France,Les Chaliots,87,37,Burgundy,Nuits-St.-Georges,,Pinot Noir,Michel Gros)
150912,France,Les Charmes,87,65,Burgundy,Chambolle-Musigny,,Pinot Noir,Daniel Rion)
150913,France,,94,30,RhÃ´ne Valley,ChÃ¢teauneuf-du-Pape,,RhÃ´ne-style Red Blend,Le Vieux Donjon)
150914,US,Late Harvest Cluster Select,94,25,California,Anderson Valley,Mendocino/Lake Counties,White Riesling,Navarro)
150915,US,Nightingale,93,30,California,North Coast,North Coast,White Blend,Beringer)
150916,US,J. Schram,93,65,California,Napa Valley,Napa,Champagne Blend,Schramsberg)
150917,France,Brut MosaÃ¯que,92,30,Champagne,Champagne,,Champagne Blend,Jacquart)
150918,France,CuvÃ©e MosaÃ¯que,92,38,Champagne,Champagne,,Champagne Blend,Jacquart)
150919,France,CuvÃ©e President,91,37,Champagne,Champagne,,Champagne Blend,H.Germain)
150920,Italy,Brut Riserva,91,19,Northeastern Italy,Trento,,Champagne Blend,Letrari)
150921,France,Blanc de Blancs Brut MosaÃ¯que,91,38,Champagne,Champagne,,Champagne Blend,Jacquart)
150922,Italy,Superiore,91,,Northeastern Italy,Colli Orientali del Friuli,,Tocai,Ronchi di Manzano)
150923,France,Demi-Sec,91,30,Champagne,Champagne,,Champagne Blend,Jacquart)
150924,France,Diamant Bleu,91,70,Champagne,Champagne,,Champagne Blend,Heidsieck & Co Monopole)
150925,Italy,,91,20,Southern Italy,Fiano di Avellino,,White Blend,Feudi di San Gregorio)
150926,France,CuvÃ©e Prestige,91,27,Champagne,Champagne,,Champagne Blend,H.Germain)
```

- **Group the data w.r.t Countries and store it in groued_data**

  grouped_data = GROUP wineRatingsData BY Country;

- **For Each grouped_data, find unique variety by using DISTINCT operator on wineRatingsData.Variety and store the count of it w.r.t countries**

  ```
  unique_varieties = FOREACH grouped_data {
          uniq_varieties = DISTINCT wineRatingsData.Variety;
          GENERATE group AS Country, COUNT(uniq_varieties) AS VarietyCount;
  }
  ```

- **See the result in unique_varieties**

  dump unique_varieties

```
grunt> grouped_data = GROUP wineRatingsData BY Country;
grunt> unique_varieties = FOREACH grouped_data {uniq_varieties = DISTINCT wineRatingsData.Variety;GENERATE group AS Country, COUNT(uniq_varie
ties) AS VarietyCount;}
grunt> dump unique_varieties
2023-11-09 19:38:45,676 [main] INFO  org.apache.pig.tools.pigstats.ScriptState - Pig features used in the script: GROUP_BY
2023-11-09 19:38:45,689 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaul
tFS
2023-11-09 19:38:45,690 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs
```

2.1.3. Output:

```
2023-11-09 19:38:48,410
tFS
2023-11-09 19:38:48,410
.bytes-per-checksum
2023-11-09 19:38:48,410
2023-11-09 19:38:48,413
2023-11-09 19:38:48,413
(US,229)
(Chile,61)
(China,2)
(Egypt,3)
(India,4)
(Italy,177)
(Japan,1)
(Spain,115)
(Brazil,8)
(Canada,29)
(Cyprus,5)
(France,131)
(Greece,43)
(Israel,37)
(Mexico,18)
(Serbia,6)
(Turkey,21)
(Albania,1)
(Austria,52)
(Croatia,20)
(England,2)
(Georgia,7)
(Germany,36)
(Hungary,29)
(Lebanon,5)
(Moldova,18)
(Morocco,8)
(Romania,19)
(Tunisia,2)
(Ukraine,2)
(Uruguay,11)
(Bulgaria,16)
(Portugal,78)
(Slovakia,1)
(Slovenia,28)
(Argentina,65)
```

2.2. Analysis 2 - **Analysis by Country, Province and Number of Wineries**

2.2.1. Input **:**

<ID, Country, Brand, Rating, Price, State, Province1, Province2, Variety, Winery>

2.2.2. Output :

<Country, Province1, Number of wineries>

2.2.3. Steps :

- **Check the status of history server**

  Command : jps

  Output :

```
[centos@master ~]$ jps
3968 NameNode
5216 ResourceManager
4962 SecondaryNameNode
807 QuorumPeerMain
7832 Jps
5354 NodeManager
4093 DataNode
[centos@master ~]$
```

- **Start history server**

  Command : cd $HADOOP_HOME/sbin
  ./mr-jobhistory-daemon.sh start historyserver

  Output :

```
[centos@master ~]$ cd $HADOOP_HOME/sbin
[centos@master sbin]$ ./mr-jobhistory-daemon.sh start historyserver
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
[centos@master sbin]$ jps
3968 NameNode
5216 ResourceManager
4962 SecondaryNameNode
8197 Jps
807 QuorumPeerMain
8055 JobHistoryServer
5354 NodeManager
4093 DataNode
[centos@master sbin]$
```

- **Open PIG shell**

  Command : pig -x local

  Output :

```
[centos@master sbin]$ pig -x local
2023-11-09 04:43:32,559 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2023-11-09 04:43:32,559 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2023-11-09 04:43:32,588 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2023-11-09 04:43:32,588 [main] INFO  org.apache.pig.Main - Logging error messages to: /opt/hadoop-3.2.4/sbin/pig_1699505012586.log
2023-11-09 04:43:32,720 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/centos/.pigbootup not found
2023-11-09 04:43:32,883 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-11-09 04:43:32,883 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-11-09 04:43:32,885 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: file:///
2023-11-09 04:43:32,994 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-11-09 04:43:33,016 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-63907f59-057d-4e4a-b968-22f4ab0e1bfa
2023-11-09 04:43:33,016 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt>
```

- ## Load File

Command : wineData = LOAD '/home/centos/Downloads/data.csv' USING PigStorage(',') as
(ID : int ,
Country : chararray,
Brand : chararray,
Rating : int,
Price : int,
State : chararray,
Province1 : chararray,
Province2 : chararray,
Variety : chararray,
Winery : chararray);

Output :

```
grunt> wineData = LOAD '/home/centos/Downloads/data.csv' USING PigStorage(',') as (ID : int , Country : chararray, Brand : chararray, Rating : int, Price : int, St
ate : chararray, Province1 : chararray, Province2 : chararray, Variety : chararray, Winery : chararray);
2023-11-09 04:45:15,426 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-11-09 04:45:15,426 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt>
```

- ## View Data

Command : dump wineData;

Output :

```
(150898,Chile,,82,7,Maipo Valley,,,Pinot Noir,Undurraga)
(150899,Chile,Medalla Real Special Reserve,82,15,Maipo Valley,,,Cabernet Sauvignon,Santa Rita)
(150900,Chile,Prima Reserva,81,13,Maipo Valley,,,Cabernet Sauvignon,De Martino)
(150901,Chile,Reserva,81,12,Maipo Valley,,,Merlot,Undurraga)
(150902,Chile,Estate Bottled,81,10,Maipo Valley,,,Chardonnay,De Martino)
(150903,Chile,120,81,7,Rapel Valley,,,Cabernet Sauvignon,Santa Rita)
(150904,Chile,,81,10,Maipo Valley,,,Cabernet Sauvignon,De Martino)
(150905,Chile,Prima Reserva,80,13,Maipo Valley,,,Merlot,De Martino)
(150906,France,Clos des Reas,93,65,Burgundy,Vosne-Romanée,,Pinot Noir,Michel Gros)
(150907,France,Les Beaux-Monts,92,52,Burgundy,Vosne-Romanée,,Pinot Noir,Daniel Rion)
(150908,France,Aux Brulees,90,65,Burgundy,Vosne-Romanée,,Pinot Noir,Michel Gros)
(150909,France,Clos dea Argillieres,89,52,Burgundy,Nuits-St.-Georges,,Pinot Noir,Daniel Rion)
(150910,France,,89,38,Burgundy,Chambolle-Musigny,,Pinot Noir,Michel Gros)
(150911,France,Les Chaliots,87,37,Burgundy,Nuits-St.-Georges,,Pinot Noir,Michel Gros)
(150912,France,Les Charmes,87,65,Burgundy,Chambolle-Musigny,,Pinot Noir,Daniel Rion)
(150913,France,,94,30,Rhône Valley,Châteauneuf-du-Pape,,Rhône-style Red Blend,Le Vieux Donjon)
(150914,US,Late Harvest Cluster Select,94,25,California,Anderson Valley,Mendocino/Lake Counties,White
(150915,US,Nightingale,93,30,California,North Coast,North Coast,White Blend,Beringer)
(150916,US,J. Schram,93,65,California,Napa Valley,Napa,Champagne Blend,Schramsberg)
(150917,France,Brut Mosaïque,92,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150918,France,Cuvée Mosaïque,92,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150919,France,Cuvée President,91,37,Champagne,Champagne,,Champagne Blend,H.Germain)
(150920,Italy,Brut Riserva,91,19,Northeastern Italy,Trento,,Champagne Blend,Letrari)
(150921,France,Blanc de Blancs Brut Mosaïque,91,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150922,Italy,Superiore,91,,Northeastern Italy,Colli Orientali del Friuli,,Tocai,Ronchi di Manzano)
(150923,France,Demi-Sec,91,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150924,France,Diamant Bleu,91,70,Champagne,Champagne,,Champagne Blend,Heidsieck & Co Monopole)
(150925,Italy,,91,20,Southern Italy,Fiano di Avellino,,White Blend,Feudi di San Gregorio)
(150926,France,Cuvée Prestige,91,27,Champagne,Champagne,,Champagne Blend,H.Germain)
(150927,Italy,Terre di Dora,91,20,Southern Italy,Fiano di Avellino,,White Blend,Terredora)
(150928,France,Grand Brut Rosé,90,52,Champagne,Champagne,,Champagne Blend,Gosset)
(150929,Italy,,90,15,Northeastern Italy,Alto Adige,,Pinot Grigio,Alois Lageder)
```

- **Group data by country and province**

  Command : grpData = GROUP wineData by (Country, Province1);

  Output :

```
grunt> grpData = GROUP wineData by (Country, Province1);
grunt>
```

- **Find Number of wineries by country and province**

  Command : count = FOREACH grpData GENERATE FLATTEN(group) AS
  (Country:chararray,
  Province1:chararray),
  COUNT(wineData) AS row_count;

  Output :

```
grunt> count = FOREACH grpData GENERATE
grunt>
grunt>     FLATTEN(group) AS (Country:chararray, Province1:chararray),
>>
>>     COUNT(wineData) AS row_count;
grunt>
grunt>
```

- **View Output**

  Command : dump count;

  Output :

```
(France,Vin de Pays de la Méditerranée,11)
(France,Vin de Pays des Côtes Catalanes,18)
(France,Bourgogne Hautes Côtes de Beaune,18)
(France,Muscat de Saint-Jean de Minervois,3)
(France,Vin de Pays des Côtes de Thongue,1)
(France,Côtes de Provence Sainte-Victoire,64)
(France,Vin de Pays des Coteaux de Murviel,1)
(France,Vin de Pays des Coteaux de Peyriac,4)
(France,Vin de Pays des Côtes de Gascogne,97)
(France,Vin de Pays du Jardin de la France,16)
(France,Coteaux du Languedoc Pic Saint Loup,14)
(France,Vin de Pays des Collines de la Moure,2)
(France,Vin de Pays des Coteaux de Bessilles,2)
(France,Vin de Pays des Coteaux de L'Ardeche,1)
(France,Vin de Pays des Pyrénees Orientales,1)
(France,Vin de Pays de la Vallée de l’Aude,1)
(France,Vin de Pays des Collines Rhodaniennes,9)
(France,Côtes du Roussillon Villages Caramany,1)
(France,Côtes du Roussillon Villages Tautavel,1)
(France,Vin de Pays de Sainte-Marie la Blanche,1)
(France,Vin de Pays des Sables du Golfe du Lion,1)
(France,Vin de Pays des Portes de Méditerranée,4)
(France,Vin de Pays de la Haute Vallée de l’Aude,2)
(France,,15)
(Greece,,884)
(Israel,,630)
(Mexico,,63)
(Serbia,,14)
(Turkey,,52)
(Albania,,2)
(Austria,Niederösterreich,2)
(Austria,,3055)
(Country,Province1,0)
```

2.3. Analysis 3 - **Analysis Minimum and Maximum Price by Country**

2.3.1. Input :

ID, Country, Brand, Rating, Price, State, Province1, Province2, Variety, Winery

2.3.2. Output :

Country, Minimum Price, Maximum Price

2.3.3. Steps :

- **Check the status of history server**

Command: jps
Output:

```
[centos@master Downloads]$ jps
8976 JobHistoryServer
887 QuorumPeerMain
17143 Jps
[centos@master Downloads]$
```

- **Start history server**

Command: cd $HADOOP_HOME/sbin
./mr-jobhistory-daemon.sh start historyserver
Output:

```
[centos@master sbin]$ ./mr-jobhistory-daemon.sh start historyserver
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
[centos@master sbin]$
```

- **Open PIG shell**

Command: pig -x local
Output:

```
[centos@master sbin]$ pig -x local
2023-11-09 06:59:13,737 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2023-11-09 06:59:13,738 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2023-11-09 06:59:13,768 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2023-11-09 06:59:13,768 [main] INFO  org.apache.pig.Main - Logging error messages to: /opt/hadoop-3.2.4/sbin/pig_1699513153766.log
2023-11-09 06:59:13,791 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/centos/.pigbootup not found
2023-11-09 06:59:13,884 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use
mapreduce.jobtracker.address
2023-11-09 06:59:13,885 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.
defaultFS
2023-11-09 06:59:13,887 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system
at: file:///
2023-11-09 06:59:14,002 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, u
se dfs.bytes-per-checksum
2023-11-09 06:59:14,024 [main] INFO  org.apache.pig.PigServer - Pig Script ID for the session: PIG-default-63b6457a-ff5c-4d0e-beb9-25d4
5eb0cb8b
2023-11-09 06:59:14,024 [main] WARN  org.apache.pig.PigServer - ATS is disabled since yarn.timeline-service.enabled set to false
grunt>
```

- **Load the Data from csv**

Command: wine_data = LOAD '/home/centos/Downloads/winedataset.csv' USING PigStorage(',') as

(ID : int ,
Country : chararray,
Brand : chararray,
Rating : int,
Price : int,
State : chararray,
Province1 : chararray,
Province2 : chararray,
Variety : chararray,
Winery : chararray);

Output:

```
grunt> wine_data = LOAD '/home/centos/Downloads/winedataset.csv' USING PigStorage(',') as
>>
>> (ID : int ,
>>
>> Country : chararray,
>>
>> Brand : chararray,
>>
>> Rating : int,
>>
>> Price : int,
>>
>> State : chararray,
>>
>> Province1 : chararray,
>>
>> Province2 : chararray,
>>
>> Variety : chararray,
>>
>> Winery : chararray);
2023-11-09 07:06:47,122 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, u
se dfs.bytes-per-checksum
2023-11-09 07:06:47,123 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.
defaultFS
grunt>
```

Command: describe wine_data;

Output:

```
grunt> describe wine_data;
wine_data: {ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Province2: chararr
ay,Variety: chararray,Winery: chararray}
grunt> _
```

Command: dump wine_data;

Output:

```
(150905,Chile,Prima Reserva,80,13,Maipo Valley,,,Merlot,De Martino)
(150906,France,Clos des Reas,93,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150907,France,Les Beaux-Monts,92,52,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Daniel Rion)
(150908,France,Aux Brulees,90,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150909,France,Clos dea Argillieres,89,52,Burgundy,Nuits-St.-Georges,,Pinot Noir,Daniel Rion)
(150910,France,,89,38,Burgundy,Chambolle-Musigny,,Pinot Noir,Michel Gros)
(150911,France,Les Chaliots,87,37,Burgundy,Nuits-St.-Georges,,Pinot Noir,Michel Gros)
(150912,France,Les Charmes,87,65,Burgundy,Chambolle-Musigny,,Pinot Noir,Daniel Rion)
(150913,France,,94,30,RhÃ´ne Valley,ChÃ¢teauneuf-du-Pape,,RhÃ´ne-style Red Blend,Le Vieux Donjon)
(150914,US,Late Harvest Cluster Select,94,25,California,Anderson Valley,Mendocino/Lake Counties,White Riesling,Navarro)
(150915,US,Nightingale,93,30,California,North Coast,North Coast,White Blend,Beringer)
(150916,US,J. Schram,93,65,California,Napa Valley,Napa,Champagne Blend,Schramsberg)
(150917,France,Brut MosaÃ¯que,92,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150918,France,CuvÃ©e MosaÃ¯que,92,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150919,France,CuvÃ©e President,91,37,Champagne,Champagne,,Champagne Blend,H.Germain)
(150920,Italy,Brut Riserva,91,19,Northeastern Italy,Trento,,Champagne Blend,Letrari)
(150921,France,Blanc de Blancs Brut MosaÃ¯que,91,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150922,Italy,Superiore,91,,Northeastern Italy,Colli Orientali del Friuli,,Tocai,Ronchi di Manzano)
(150923,France,Demi-Sec,91,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150924,France,Diamant Bleu,91,70,Champagne,Champagne,,Champagne Blend,Heidsieck & Co Monopole)
(150925,Italy,,91,20,Southern Italy,Fiano di Avellino,,White Blend,Feudi di San Gregorio)
(150926,France,CuvÃ©e Prestige,91,27,Champagne,Champagne,,Champagne Blend,H.Germain)
(150927,Italy,Terre di Dora,91,20,Southern Italy,Fiano di Avellino,,White Blend,Terredora)
(150928,France,Grand Brut RosÃ©,90,52,Champagne,Champagne,,Champagne Blend,Gosset)
(150929,Italy,,90,15,Northeastern Italy,Alto Adige,,Pinot Grigio,Alois Lageder)
grunt>
```

- ## Group data with Country

Command: group_data = GROUP wine_data BY Country;

Output:

```
grunt> group_data = GROUP wine_data BY Country;
grunt>
```

Command: describe group_data;
Output:

```
grunt> describe group_data;
group_data: {group: chararray,wine_data: {(ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province
1: chararray,Province2: chararray,Variety: chararray,Winery: chararray)}}
grunt>
```

- **Find Minimum and Maximum Price based on Country**

Command: result = FOREACH group_data {
min_price = MIN(wine_data.Price);
max_price = MAX(wine_data.Price);
GENERATE group AS Country, min_price AS MinPrice, max_price AS MaxPrice;
}
Output:

```
grunt>
grunt> result = FOREACH group_data {
>>
>>     min_price = MIN(wine_data.Price);
>>
>>     max_price = MAX(wine_data.Price);
>>
>>     GENERATE group AS Country, min_price AS MinPrice, max_price AS MaxPrice;
>>
>> }
grunt>
```

Command: describe result;
Output:

```
grunt> describe result;
result: {Country: chararray,MinPrice: int,MaxPrice: int}
grunt>
```

- **View Output**

Command: dump count;
Output:

```
(Lebanon,12,51)
(Moldova,8,42)
(Morocco,6,35)
(Romania,4,320)
(Tunisia,,)
(Ukraine,13,13)
(Uruguay,7,60)
(Bulgaria,7,28)
(Portugal,4,980)
(Slovakia,15,16)
(Slovenia,7,90)
(Argentina,4,250)
(Australia,5,850)
(Lithuania,10,10)
(Macedonia,12,25)
(US-France,50,50)
(Luxembourg,36,50)
(Montenegro,10,10)
(New Zealand,7,125)
(South Korea,11,16)
(Switzerland,19,38)
(South Africa,5,145)
(Czech Republic,15,25)
(Bosnia and Herzegovina,12,13)
```

2.4. Analysis 4 - Analysis of Average Rating based on Variety

2.4.1. Input:

< ID, Country, Brand, Rating, Price, State, Province1, Province2, Variety, Winery>

2.4.2. Output :

< Variety, Average Rating>

2.4.3. Steps :

- **Check the status of history server**

  Command : jps

  Output

```
File  Edit  View  Search  Terminal  Help
[centos@master ~]$ jps
801 QuorumPeerMain
4707 ResourceManager
4452 SecondaryNameNode
6536 Jps
3579 DataNode
4845 NodeManager
3454 NameNode
[centos@master ~]$
```

- **Start history server**

Command: cd $HADOOP_HOME/sbin
./mr-jobhistory-daemon.sh start historyserver

Output:

```
File  Edit  View  Search  Terminal  Help
[centos@master ~]$ cd $HADOOP_HOME/sbin
[centos@master sbin]$ ./mr-jobhistory-daemon.sh start historyserver
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
[centos@master sbin]$ jps
801 QuorumPeerMain
4707 ResourceManager
4452 SecondaryNameNode
7524 Jps
3579 DataNode
4845 NodeManager
3454 NameNode
7454 JobHistoryServer
[centos@master sbin]$
```

- **Open PIG shell**

Command: pig -x local

Output:

```
[centos@master sbin]$ pig -x local
2023-11-11 04:06:46,573 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2023-11-11 04:06:46,573 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2023-11-11 04:06:46,605 [main] INFO  org.apache.pig.Main - Apache Pig version 0.
16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2023-11-11 04:06:46,605 [main] INFO  org.apache.pig.Main - Logging error message
s to: /opt/hadoop-3.2.4/sbin/pig_1699675606602.log
2023-11-11 04:06:46,742 [main] INFO  org.apache.pig.impl.util.Utils - Default bo
otup file /home/centos/.pigbootup not found
2023-11-11 04:06:46,858 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.addr
ess
2023-11-11 04:06:46,859 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-11-11 04:06:46,861 [main] INFO  org.apache.pig.backend.hadoop.executionengi
ne.HExecutionEngine - Connecting to hadoop file system at: file:///
2023-11-11 04:06:46,971 [main] INFO  org.apache.hadoop.conf.Configuration.deprec
ation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-11-11 04:06:46,995 [main] INFO  org.apache.pig.PigServer - Pig Script ID fo
r the session: PIG-default-5378bb18-772f-4a75-9f7d-5083d33c6b2c
2023-11-11 04:06:46,995 [main] WARN  org.apache.pig.PigServer - ATS is disabled
since yarn.timeline-service.enabled set to false
grunt>
```

- **Load the Data from csv**

Command: wineData = LOAD '/home/centos/data.csv' USING PigStorage(',') as
(ID : int ,
Country : chararray,
Brand : chararray,
Rating : int,
Price : int,
State : chararray,
Province1 : chararray,
Province2 : chararray,
Variety : chararray,
Winery : chararray);
Output:

```
grunt> wineData = LOAD '/home/centos/data.csv' USING PigStorage(',') as
>>
>> (ID : int ,
>>
>> Country : chararray,
>>
>> Brand : chararray,
>>
>> Rating : int,
>>
>> Price : int,
>>
>> State : chararray,
>>
>> Province1 : chararray,
>>
>> Province2 : chararray,
>>
>> Variety : chararray,
>>
>> Winery : chararray);
2023-11-11 04:14:34,638 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-pe
r-checksum
2023-11-11 04:14:34,639 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt>
grunt>
```

Command: describe wineData;
Output:

```
grunt> describe wineData;
wineData: {ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Province2: chararray,Variety: char
array,Winery: chararray}
grunt>
```

Command: dump wineData;
Output:

```
(150902,Chile,Estate Bottled,81,10,Maipo Valley,,,Chardonnay,De Martino)
(150903,Chile,120,81,7,Rapel Valley,,,Cabernet Sauvignon,Santa Rita)
(150904,Chile,,81,10,Maipo Valley,,,Cabernet Sauvignon,De Martino)
(150905,Chile,Prima Reserva,80,13,Maipo Valley,,,Merlot,De Martino)
(150906,France,Clos des Reas,93,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150907,France,Les Beaux-Monts,92,52,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Daniel Rion)
(150908,France,Aux Brulees,90,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150909,France,Clos dea Argillieres,89,52,Burgundy,Nuits-St.-Georges,,Pinot Noir,Daniel Rion)
(150910,France,,89,38,Burgundy,Chambolle-Musigny,,Pinot Noir,Michel Gros)
(150911,France,Les Chaliots,87,37,Burgundy,Nuits-St.-Georges,,Pinot Noir,Michel Gros)
(150912,France,Les Charmes,87,65,Burgundy,Chambolle-Musigny,,Pinot Noir,Daniel Rion)
(150913,France,,94,30,RhÃ´ne Valley,ChÃ¢teauneuf-du-Pape,,RhÃ´ne-style Red Blend,Le Vieux Donjon)
(150914,US,Late Harvest Cluster Select,94,25,California,Anderson Valley,Mendocino/Lake Counties,White Riesling,Navarro)
(150915,US,Nightingale,93,30,California,North Coast,North Coast,White Blend,Beringer)
(150916,US,J. Schram,93,65,California,Napa Valley,Napa,Champagne Blend,Schramsberg)
(150917,France,Brut MosaÃ¯que,92,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150918,France,CuvÃ©e MosaÃ¯que,92,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150919,France,CuvÃ©e President,91,37,Champagne,Champagne,,Champagne Blend,H.Germain)
(150920,Italy,Brut Riserva,91,19,Northeastern Italy,Trento,,Champagne Blend,Letrari)
(150921,France,Blanc de Blancs Brut MosaÃ¯que,91,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150922,Italy,Superiore,91,,Northeastern Italy,Colli Orientali del Friuli,,Tocai,Ronchi di Manzano)
(150923,France,Demi-Sec,91,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150924,France,Diamant Bleu,91,70,Champagne,Champagne,,Champagne Blend,Heidsieck & Co Monopole)
(150925,Italy,,91,20,Southern Italy,Fiano di Avellino,,White Blend,Feudi di San Gregorio)
(150926,France,CuvÃ©e Prestige,91,27,Champagne,Champagne,,Champagne Blend,H.Germain)
(150927,Italy,Terre di Dora,91,20,Southern Italy,Fiano di Avellino,,White Blend,Terredora)
(150928,France,Grand Brut RosÃ©,90,52,Champagne,Champagne,,Champagne Blend,Gosset)
(150929,Italy,,90,15,Northeastern Italy,Alto Adige,,Pinot Grigio,Alois Lageder)
grunt>
```

- **Group data with Variety**

Command: groupData = GROUP wineData BY Variety;
Output:

```
grunt> groupData = GROUP wineData BY Variety;
grunt>
```

Command:
Output:

```
grunt> describe groupData;
groupData: {group: chararray,wineData: {(ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Prov
ince2: chararray,Variety: chararray,Winery: chararray)}}
grunt>
```

- **Find Average Rating based on Variety**

Command: result = FOREACH groupData {
avg_rating = AVG(wineData.Rating);
GENERATE group AS Variety, avg_rating AS AvgRating;
}
Output:

```
grunt> result = FOREACH groupData {
>>
>>     avg_rating = AVG(wineData.Rating);
>>
>>     GENERATE group AS Variety, avg_rating AS AvgRating;
>>
>> }
grunt>
```

Command: describe result;
Output:

```
grunt> describe result;
result: {Variety: chararray,AvgRating: double}
grunt>
```

- **View Output**

Command: dump result;
Output:

```
File  Edit  View  Search  Terminal  Help
(Merlot-Cabernet Sauvignon,87.17073170731707)
(Moscatel de AlejandrÃ a,83.0)
(Shiraz-Cabernet Sauvignon,86.73195876288659)
(Bordeaux-style White Blend,89.47343378271214)
(Cabernet Franc-Tempranillo,87.0)
(Chardonnay-Sauvignon Blanc,87.6)
(GewÃ¼rztraminer-Riesling,86.66666666666667)
(RhÃ´ne-style White Blend,87.92156862745098)
(Sauvignon Blanc-Chardonnay,86.45454545454545)
(Cabernet Franc-CarmenÃ¨re,86.0)
(Cabernet Sauvignon Grenache,86.5)
(Chenin Blanc-Sauvignon Blanc,85.0)
(Pinot Grigio-Sauvignon Blanc,87.0)
(Sauvignon Blanc-Chenin Blanc,88.0)
(Cabernet Sauvignon-Sangiovese,88.46153846153847)
(Muscat Blanc Ã  Petit Grain,86.33333333333333)
(Sangiovese-Cabernet Sauvignon,86.33333333333333)
(Cabernet Sauvignon-Tempranillo,86.66666666666667)
(Sauvignon Blanc-Sauvignon Gris,86.0)
(Tempranillo-Cabernet Sauvignon,85.43529411764706)
(Cabernet Sauvignon-CarmenÃ¨re,86.3103448275862)
(CarmenÃ¨re-Cabernet Sauvignon,87.22727272727273)
(Cabernet Sauvignon-Merlot-Shiraz,87.66666666666667)
(Cabernet Franc-Cabernet Sauvignon,89.33333333333333)
(Cabernet Sauvignon-Cabernet Franc,87.65217391304348)
(Cabernet Sauvignon and Tinta Roriz,86.0)
(Touriga Nacional-Cabernet Sauvignon,87.67857142857143)
```

2.5. Analysis 5 - **Analysis of Average price based on variety**

2.5.1. Input :

< ID, Country, Brand, Rating, Price, State, Province1, Province2, Variety, Winery>

2.5.2. Output :

< Variety, Average Price>

2.5.3. Steps :

- **Check the status of history server**

Command: jps
Output:

```
File  Edit  View  Search  Terminal  Help
[centos@master ~]$ jps
4786 SecondaryNameNode
5027 ResourceManager
6727 Jps
3897 DataNode
5162 NodeManager
3772 NameNode
782 QuorumPeerMain
```

- **Start history server**

Command: cd $HADOOP_HOME/sbin
./mr-jobhistory-daemon.sh start historyserver

Output:

```
[centos@master ~]$ cd $HADOOP_HOME/sbin
[centos@master sbin]$
[centos@master sbin]$ ./mr-jobhistory-daemon.sh start historyserver
WARNING: Use of this script to start the MR JobHistory daemon is deprecated.
WARNING: Attempting to execute replacement "mapred --daemon start" instead.
[centos@master sbin]$ jps
4786 SecondaryNameNode
5027 ResourceManager
7272 JobHistoryServer
3897 DataNode
5162 NodeManager
3772 NameNode
7356 Jps
782 QuorumPeerMain
```

- **Open PIG shell**

Command: pig -x local

Output:

```
[centos@master sbin]$ pig -x local
2023-11-09 12:16:26,001 INFO pig.ExecTypeProvider: Trying ExecType : LOCAL
2023-11-09 12:16:26,002 INFO pig.ExecTypeProvider: Picked LOCAL as the ExecType
2023-11-09 12:16:26,038 [main] INFO  org.apache.pig.Main - Apache Pig version 0.16.0 (r1746530) compiled Jun 01 2016, 23:10:49
2023-11-09 12:16:26,038 [main] INFO  org.apache.pig.Main - Logging error messages to: /opt/hadoop-3.2.4/sbin/pig_1699532186035.log
2023-11-09 12:16:26,136 [main] INFO  org.apache.pig.impl.util.Utils - Default bootup file /home/centos/.pigbootup not found
2023-11-09 12:16:26,240 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - mapred.job.tracker is deprecated. Instead, use mapreduce.jobtracker.address
2023-11-09 12:16:26,240 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
2023-11-09 12:16:26,243 [main] INFO  org.apache.pig.backend.hadoop.executionengine.HExecutionEngine - Connecting to hadoop file system at: file:///
2023-11-09 12:16:26,353 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
```

- **Load the Data from csv**

Command: wineData = LOAD '/home/centos/Downloads/winedataset.csv' USING
PigStorage(',') as
(ID : int ,
Country : chararray,
Brand : chararray,
Rating : int,

Price : int,
State : chararray,
Province1 : chararray,
Province2 : chararray,
Variety : chararray,
Winery : chararray);
Output:

```
grunt> wineData = LOAD '/home/centos/Downloads/data.csv' USING PigStorage(',') as
>>
>> (ID : int ,
>>
>> Country : chararray,
>>
>> Brand : chararray,
>>
>> Rating : int,
>>
>> Price : int,
>>
>> State : chararray,
>>
>> Province1 : chararray,
>>
>> Province2 : chararray,
>>
>> Variety : chararray,
>>
>> Winery : chararray);
2023-11-09 12:20:10,802 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - io.bytes.per.checksum is deprecated. Instead, use dfs.bytes-per-checksum
2023-11-09 12:20:10,803 [main] INFO  org.apache.hadoop.conf.Configuration.deprecation - fs.default.name is deprecated. Instead, use fs.defaultFS
grunt>
```

Command: describe wineData;
Output:

```
grunt> describe wineData;
wineData: {ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Province2: chararray,Variety: chararray,Winery: chara
rray}
grunt>
```

Command: dump wineData;
Output:

```
(150900,Chile,Prima Reserva,81,13,Maipo Valley,,,Cabernet Sauvignon,De Martino)
(150901,Chile,Reserva,81,12,Maipo Valley,,,Merlot,Undurraga)
(150902,Chile,Estate Bottled,81,10,Maipo Valley,,,Chardonnay,De Martino)
(150903,Chile,120,81,7,Rapel Valley,,,Cabernet Sauvignon,Santa Rita)
(150904,Chile,,81,10,Maipo Valley,,,Cabernet Sauvignon,De Martino)
(150905,Chile,Prima Reserva,80,13,Maipo Valley,,,Merlot,De Martino)
(150906,France,Clos des Reas,93,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150907,France,Les Beaux-Monts,92,52,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Daniel Rion)
(150908,France,Aux Brulees,90,65,Burgundy,Vosne-RomanÃ©e,,Pinot Noir,Michel Gros)
(150909,France,Clos dea Argillieres,89,52,Burgundy,Nuits-St.-Georges,,Pinot Noir,Daniel Rion)
(150910,France,,89,38,Burgundy,Chambolle-Musigny,,Pinot Noir,Michel Gros)
(150911,France,Les Chaliots,87,37,Burgundy,Nuits-St.-Georges,,Pinot Noir,Michel Gros)
(150912,France,Les Charmes,87,65,Burgundy,Chambolle-Musigny,,Pinot Noir,Daniel Rion)
(150913,France,,94,30,RhÃ´ne Valley,ChÃ¢teauneuf-du-Pape,,RhÃ´ne-style Red Blend,Le Vieux Donjon)
(150914,US,Late Harvest Cluster Select,94,25,California,Anderson Valley,Mendocino/Lake Counties,White Riesling,Navarro)
(150915,US,Nightingale,93,30,California,North Coast,North Coast,White Blend,Beringer)
(150916,US,J. Schram,93,65,California,Napa Valley,Napa,Champagne Blend,Schramsberg)
(150917,France,Brut MosaÃ¯que,92,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150918,France,CuvÃ©e MosaÃ¯que,92,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150919,France,CuvÃ©e President,91,37,Champagne,Champagne,,Champagne Blend,H.Germain)
(150920,Italy,Brut Riserva,91,19,Northeastern Italy,Trento,,Champagne Blend,Letrari)
(150921,France,Blanc de Blancs Brut MosaÃ¯que,91,38,Champagne,Champagne,,Champagne Blend,Jacquart)
(150922,Italy,Superiore,91,,Northeastern Italy,Colli Orientali del Friuli,,Tocai,Ronchi di Manzano)
(150923,France,Demi-Sec,91,30,Champagne,Champagne,,Champagne Blend,Jacquart)
(150924,France,Diamant Bleu,91,70,Champagne,Champagne,,Champagne Blend,Heidsieck & Co Monopole)
(150925,Italy,,91,20,Southern Italy,Fiano di Avellino,,White Blend,Feudi di San Gregorio)
(150926,France,CuvÃ©e Prestige,91,27,Champagne,Champagne,,Champagne Blend,H.Germain)
(150927,Italy,Terre di Dora,91,20,Southern Italy,Fiano di Avellino,,White Blend,Terredora)
(150928,France,Grand Brut RosÃ©,90,52,Champagne,Champagne,,Champagne Blend,Gosset)
(150929,Italy,,90,15,Northeastern Italy,Alto Adige,,Pinot Grigio,Alois Lageder)
```

- ## Group data with Variety

Command: groupData = GROUP wineData BY Variety;
Output:

```
grunt> groupData = GROUP wineData BY Variety;
```

Command: describe groupData;
Output:

```
grunt> describe groupData;
groupData: {group: chararray,wineData: {(ID: int,Country: chararray,Brand: chararray,Rating: int,Price: int,State: chararray,Province1: chararray,Province2: chararray,Va
riety: chararray,Winery: chararray)}}
grunt>
```

- ## Find Average Price based on Variety

Command: result = FOREACH groupData {
avg_price = AVG(wineData.Price);
GENERATE group AS Variety, avg_price AS AvgPrice;
}
Output:

```
grunt> result = FOREACH groupData {
>>
>>     avg_price = AVG(wineData.Price);
>>
>>     GENERATE group AS Variety, avg_price AS AvgPrice;
>>
>> }
grunt>
grunt>
```

Command: describe result;
Output:

```
grunt>
grunt> describe result;
result: {Variety: chararray,AvgPrice: double}
grunt>
grunt>
```

- ## View Output

Command: dump result;
Output:

```
(Macabeo-GewÃ¼rztraminer,10.0)
(Malbec-Cabernet Sauvignon,31.517241379310345)
(Merlot-Cabernet Sauvignon,29.84722222222222)
(Moscatel de AlejandrÃ a,15.0)
(Shiraz-Cabernet Sauvignon,20.105263157894736)
(Bordeaux-style White Blend,36.720689655172414)
(Cabernet Franc-Tempranillo,18.0)
(Chardonnay-Sauvignon Blanc,13.6)
(GewÃ¼rztraminer-Riesling,17.666666666666668)
(RhÃ´ne-style White Blend,30.0935960591133)
(Sauvignon Blanc-Chardonnay,18.272727272727273)
(Cabernet Franc-CarmenÃ¨re,18.5)
(Cabernet Sauvignon Grenache,14.25)
(Chenin Blanc-Sauvignon Blanc,10.428571428571429)
(Pinot Grigio-Sauvignon Blanc,35.0)
(Sauvignon Blanc-Chenin Blanc,13.666666666666666)
(Cabernet Sauvignon-Sangiovese,43.90909090909091)
(Muscat Blanc Ã  Petit Grain,17.666666666666668)
(Sangiovese-Cabernet Sauvignon,27.666666666666668)
(Cabernet Sauvignon-Tempranillo,13.666666666666666)
(Sauvignon Blanc-Sauvignon Gris,)
(Tempranillo-Cabernet Sauvignon,25.141176470588235)
(Cabernet Sauvignon-CarmenÃ¨re,23.603448275862068)
(CarmenÃ¨re-Cabernet Sauvignon,16.05)
(Cabernet Sauvignon-Merlot-Shiraz,19.333333333333332)
(Cabernet Franc-Cabernet Sauvignon,34.0)
(Cabernet Sauvignon-Cabernet Franc,41.35)
(Cabernet Sauvignon and Tinta Roriz,8.0)
(Touriga Nacional-Cabernet Sauvignon,13.956521739130435)
```

3. Analysis using MongoDB

3.1. Student 1 - 2023MT03132

Command used to import document :

- mongosh
- mongoimport --collection='wineratings1' --file='/home/centos/praveenanand/assignment2/winedataset.json' --jsonArray

3.1.1. Analysis 1 : Find the count of wines by variety and sort them in descending order.

command:

```
db.wineratings1.aggregate([
    { $group: { _id: "$Variety", count: { $sum: 1 } } },
    { $sort: { count: -1 } }
])
```

Output:

```
test> db.wineratings.aggregate([
...     { $group: { _id: "$Variety", count: { $sum: 1 } } },
...     { $sort: { count: -1 } }
... ])
[
  { _id: 'Pinot Noir', count: 1906 },
  { _id: 'Chardonnay', count: 1293 },
  { _id: 'Red Blend', count: 1075 },
  { _id: 'Cabernet Sauvignon', count: 934 },
  { _id: 'Bordeaux-style Red Blend', count: 813 },
  { _id: 'Riesling', count: 713 },
  { _id: 'Rosé', count: 587 },
  { _id: 'Sauvignon Blanc', count: 559 },
  { _id: 'Syrah', count: 462 },
  { _id: 'Sangiovese', count: 433 },
  { _id: 'Nebbiolo', count: 339 },
  { _id: 'White Blend', count: 323 },
  { _id: 'Malbec', count: 309 },
  { _id: 'Zinfandel', count: 302 },
  { _id: 'Merlot', count: 278 },
  { _id: 'Portuguese Red', count: 276 },
  { _id: 'Tempranillo', count: 270 },
  { _id: 'Pinot Gris', count: 197 },
  { _id: 'Gewürztraminer', count: 156 },
  { _id: 'Rhône-style Red Blend', count: 154 }
```

3.1.2. Analysis 2 : Calculate the average price of wines by country.

command:

```
db.wineratings1.aggregate([
    { $group: { _id: "$Country", avgPrice: { $avg: { $toDouble: "$Price" } } } },
    { $project: { _id: 0, Country: "$_id", avgPrice: 1 } }
])
```

Output:

```
test> db.wineratings1.aggregate([
...     { $group: { _id: "$Country", avgPrice: { $avg: { $toDouble: "$Price" } } } },
...     { $project: { _id: 0, Country: "$_id", avgPrice: 1 } }
... ])
[
  { avgPrice: 24.5, Country: 'Switzerland' },
  { avgPrice: 11.833333333333334, Country: 'India' },
  { avgPrice: 23.5, Country: '' },
  { avgPrice: 16.055555555555557, Country: 'Bulgaria' },
  { avgPrice: 30, Country: 'Turkey' },
  { avgPrice: 21.210526315789473, Country: 'Moldova' },
  { avgPrice: 25.714285714285715, Country: 'Croatia' },
  { avgPrice: 15, Country: 'Brazil' },
  { avgPrice: 28.63659147869674, Country: 'Argentina' },
  { avgPrice: 38.16185616329134, Country: 'US' },
  { avgPrice: 62.10618692408951, Country: 'Italy' },
  { avgPrice: 20, Country: 'Albania' },
  { avgPrice: 30.275, Country: 'Canada' },
  { avgPrice: 19.444444444444443, Country: 'Morocco' },
  { avgPrice: 21.63395225464191, Country: 'Chile' },
  { avgPrice: 65.52173913043478, Country: 'Austria' },
  { avgPrice: 24.226666666666667, Country: 'Greece' },
  { avgPrice: 31.7, Country: 'Uruguay' },
  { avgPrice: 34.21450459652707, Country: 'Spain' },
  { avgPrice: 12, Country: 'Bosnia and Herzegovina' }
]
Type "it" for more
test>
```

3.1.3. Analysis 3 : countries with no. of wines with a rating of 90 or higher

Command :

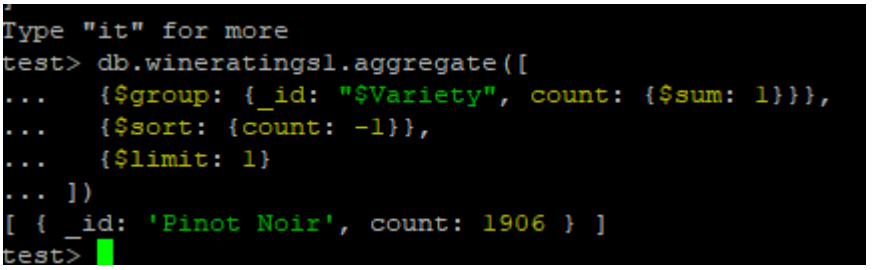
```
db.wineratings1.aggregate([
 {$group: {_id: "$Country", count: {$sum: {$cond: [{$gt: ["$Rating", 89]}, 1, 0]}}}},
 {$sort: {count: -1}}
])
```

Output :

```
test> db.wineratings1.aggregate([
...   {$group: {_id: "$Country", count: {$sum: {$cond: [{$gt: ["$Rating", 89]}, 1
...   {$sort: {count: -1}}
... ])
[
  { _id: 'US', count: 6271 },
  { _id: 'France', count: 2887 },
  { _id: 'Italy', count: 2279 },
  { _id: 'Spain', count: 979 },
  { _id: 'Portugal', count: 577 },
  { _id: 'Argentina', count: 399 },
  { _id: 'Austria', count: 391 },
  { _id: 'Chile', count: 377 },
  { _id: 'Germany', count: 373 },
  { _id: 'Australia', count: 234 },
  { _id: 'New Zealand', count: 136 },
  { _id: 'South Africa', count: 81 },
  { _id: 'Greece', count: 75 },
  { _id: 'Israel', count: 55 },
  { _id: 'Canada', count: 40 },
  { _id: 'Slovenia', count: 33 },
  { _id: 'Turkey', count: 27 },
  { _id: 'Moldova', count: 19 },
  { _id: 'Bulgaria', count: 18 },
  { _id: 'Romania', count: 16 }
```

3.1.4. Analysis 4 : Most popular wine variety by no of places it is manufactured.

```
db.wineratings1.aggregate([
 {$group: {_id: "$Variety", count: {$sum: 1}}},
 {$sort: {count: -1}},
 {$limit: 1}
])
```



3.1.5. Analysis 5: Identify the top 5 wines by rating.

```
db.wineratings1.find().sort({Rating: -1}).limit(5)
```

```
est> db.wineratings1.find().sort({Rating:

{
  _id: '2146',
  Country: 'Italy',
  Brand: 'Madonna delle Grazie',
  Rating: '99',
  Price: '200',
  State: 'Tuscany',
  Province1: 'Brunello di Montalcino',
  Province2: '',
  Variety: 'Sangiovese',
  Winery: 'Il Marroneto'
},
{
  _id: '10006',
  Country: 'US',
  Brand: 'Westside Road Neighbors',
  Rating: '98',
  Price: '69',
  State: 'California',
  Province1: 'Russian River Valley',
  Province2: 'Sonoma',
  Variety: 'Pinot Noir',
  Winery: 'Williams Selyem'
},
{
  _id: '2148',
  Country: 'Australia',
  Brand: 'The Factor',
  Rating: '98',
  Price: '125',
  State: 'South Australia',
  Province1: 'Barossa Valley',
  Province2: '',
```

3.2. Student 2 - 2023MT03047

Command used to import document :

- mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray

```
[centos@master ~]$ mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray
2023-11-08T08:48:32.693+0000    connected to: mongodb://localhost/
2023-11-08T08:48:33.185+0000    15346 document(s) imported successfully. 0 document(s) failed to import.
[centos@master ~]$
```

Command used to connect to the mongo shell script:

- mongosh

```
[centos@master ~]$ mongosh
Current Mongosh Log ID: 654b4be07025f67f36918784
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:          6.0.2
Using Mongosh:          1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/
```

3.2.1. Analysis 1 : Find Count of wines by Country and State

Command :

```
db.wineratings1.aggregate([{
                    $group : {
                            _id : {column1 : "$Country", column2 : "$State"},
                            count : {$sum : 1}
                            }
                    }]);
```

Output :

```
test> db.wineratings1.aggregate([{ $group : { _id : {column1 : "$Country", column2 : "$State"}, count : {$sum : 1}}}])
[
  {
    _id: { column1: 'Austria', column2: 'Niederösterreich' },
    count: 78
  },
  { _id: { column1: 'Spain', column2: 'Spanish Islands' }, count: 6 },
  { _id: { column1: 'Germany', column2: 'Rheinhessen' }, count: 42 },
  { _id: { column1: 'Brazil', column2: 'Serra Gaúcha' }, count: 1 },
  {
    _id: { column1: 'Austria', column2: 'Wagram-Donauland' },
    count: 6
  },
  {
    _id: { column1: 'Portugal', column2: 'Beira Atlantico' },
    count: 2
  },
  { _id: { column1: 'Chile', column2: 'Loncomilla Valley' }, count: 2 },
  { _id: { column1: 'Chile', column2: 'Chile' }, count: 5 },
  {
    _id: { column1: 'Greece', column2: 'Vin de Pays de Macédoine' },
    count: 2
  },
  { _id: { column1: 'Slovenia', column2: 'Dolenjska' }, count: 1 },
  { _id: { column1: 'Greece', column2: 'Rapsani' }, count: 2 },
  { _id: { column1: 'US', column2: 'Pennsylvania' }, count: 1 },
  { _id: { column1: 'Albania', column2: 'Mirditë' }, count: 2 },
  { _id: { column1: 'Germany', column2: 'Baden' }, count: 15 },
  { _id: { column1: 'New Zealand', column2: 'Nelson' }, count: 5 },
  { _id: { column1: 'Romania', column2: 'Dealu Mare' }, count: 1 },
  { _id: { column1: 'Portugal', column2: 'Dão' }, count: 32 },
  { _id: { column1: 'US', column2: 'California' }, count: 4223 },
  { _id: { column1: 'Greece', column2: 'Attica' }, count: 1 },
  { _id: { column1: 'South Africa', column2: 'Overberg' }, count: 1 }
```

3.2.2. Analysis 2 : Find States in each Country with more than 100 wine varieties

Command :

```
db.wineratings1.aggregate([{
                    $group : {
                            _id : {column1 : "$Country", column2 : "$State"},
                            count : {$sum : 1}}
                            },
                    { $match : { count :{ $gte : 100 }}   }])
```

Output :

```
test> db.wineratings1.aggregate([{ $group : { _id : {column1 : "$Country", column2 : "$State"}, count : {$sum : 1}} }, { $match : { count :{ $gte : 100 }}}])
[
  { _id: { column1: 'US', column2: 'California' }, count: 4223 },
  { _id: { column1: 'Portugal', column2: 'Alentejano' }, count: 136 },
  { _id: { column1: 'US', column2: 'New York' }, count: 242 },
  { _id: { column1: 'France', column2: 'Alsace' }, count: 505 },
  {
    _id: { column1: 'Italy', column2: 'Northeastern Italy' },
    count: 215
  },
  { _id: { column1: 'Italy', column2: 'Southern Italy' }, count: 182 },
  { _id: { column1: 'France', column2: 'Bordeaux' }, count: 720 },
  { _id: { column1: 'Italy', column2: 'Veneto' }, count: 200 },
  { _id: { column1: 'Spain', column2: 'Catalonia' }, count: 172 },
  {
    _id: { column1: 'Australia', column2: 'South Australia' },
    count: 165
  },
  { _id: { column1: 'France', column2: 'Provence' }, count: 283 },
  { _id: { column1: 'Spain', column2: 'Northern Spain' }, count: 581 },
  { _id: { column1: 'Italy', column2: 'Tuscany' }, count: 765 },
  { _id: { column1: 'Germany', column2: 'Mosel' }, count: 188 },
  { _id: { column1: 'France', column2: 'Rhône Valley' }, count: 155 },
  {
    _id: { column1: 'Argentina', column2: 'Mendoza Province' },
    count: 347
  },
  { _id: { column1: 'US', column2: 'Oregon' }, count: 903 },
  { _id: { column1: 'France', column2: 'Burgundy' }, count: 457 },
  {
    _id: { column1: 'Italy', column2: 'Sicily & Sardinia' },
    count: 358
  },
```

3.2.3. Analysis 3 : Find wines in country US with Rating greater than 90 or Price greater than 100

Command :

```
db.wineratings1.find({
                        $and : [ { "Country" : "US" },
                                 { $or : [{ "Rating" : { $gt : "90" }},
                                          { "Price" : { $gt : "100" }}]
                        } ]
                })
```

Output :

```
test> db.wineratings1.find({ $and : [ { "Country" : "US" }, { $or : [{ "Rating" : { $gt : "90" }}, { "Price" : { $gt : "100" }}]} ]})
[
  {
    _id: '3',
    Country: 'US',
    Brand: 'Reserve',
    Rating: '96',
    Price: '65',
    State: 'Oregon',
    Province1: 'Willamette Valley',
    Province2: 'Willamette Valley',
    Variety: 'Pinot Noir',
    Winery: 'Ponzi'
  },
  {
    _id: '8',
    Country: 'US',
    Brand: 'Silice',
    Rating: '95',
    Price: '65',
    State: 'Oregon',
    Province1: 'Chehalem Mountains',
    Province2: 'Willamette Valley',
    Variety: 'Pinot Noir',
    Winery: 'Bergström'
  },
  {
    _id: '9',
    Country: 'US',
    Brand: "Gap's Crown Vineyard",
    Rating: '95',
    Price: '60',
    State: 'California'
```

3.2.4. Analysis 4 : Calculate the average price of all wines.

Command :

```
db.wineratings1.aggregate([
        { $group: { _id: null, avgPrice: { $avg: { $toDouble: "$Price" } } } }
        ])
```

Output :

```
test> db.wineratings1.aggregate([
...
...     { $group: { _id: null, avgPrice: { $avg: { $toDouble: "$Price" } } } }
...
... ])
[ { _id: null, avgPrice: 47.875993744298185 } ]
test>

test>
```

3.2.5. Analysis 5 : Countries with average wine price greater than 50

Command:

```
db.wineratings1.aggregate([
                                { $group: { _id: "$Country",
```

avgPrice: { $avg: { $toDouble: "$Price" } } } },
{ $match : { avgPrice : {$gt : 50} }}
])

Output :

```
test> db.wineratings1.aggregate([ { $group: { _id: "$Country", avgPrice: { $avg: { $toDouble: "$Price" } } } }, { $match : { avgPrice : {$gt : 50} }}])
[
  { _id: 'Italy', avgPrice: 62.10618692408951 },
  { _id: 'Austria', avgPrice: 65.52173913043478 },
  { _id: 'France', avgPrice: 67.17180464149637 },
  { _id: 'Australia', avgPrice: 54.47008547008547 },
  { _id: 'Portugal', avgPrice: 62.24956672443674 }
]
test>
```

3.3. Student 3 - 2023MT03028

Command used to import document :

- mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray

```
[centos@master Downloads]$ mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray
2023-11-07T19:39:57.362+0000    connected to: mongodb://localhost/
2023-11-07T19:39:57.799+0000    15346 document(s) imported successfully. 0 document(s) failed to import.
[centos@master Downloads]$
```

Command used to connect to the mongo shell script:

- mongosh

```
[centos@master Downloads]$ mongosh
Current Mongosh Log ID: 654a929bb10fbfd329e3788e
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:          6.0.2
Using Mongosh:          1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-11-07T19:25:15.315+00:00: Access control is not enabled for the database. Read and write access to data and configuration is un
restricted
   2023-11-07T19:25:15.315+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
   2023-11-07T19:25:15.315+00:00: /sys/kernel/mm/transparent_hugepage/defrag is 'always'. We suggest setting it to 'never'
   2023-11-07T19:25:15.315+00:00: vm.max_map_count is too low
------
```

3.3.1. Analysis 1 : Find the most expensive wine.

Command:

db.wineratings1.find().sort({ Price: -1 }).limit(1)

Output:

```
test> db.wineratings1.find().sort({ Price: -1 }).limit(1)
[
  {
    _id: '1998',
    Country: 'Italy',
    Brand: 'Boiolo',
    Rating: '88',
    Price: '99',
    State: 'Piedmont',
    Province1: 'Barolo',
    Province2: '',
    Variety: 'Nebbiolo',
    Winery: 'Camparo'
  }
]
test>
```

3.3.2. Analysis 2 : Find wines produced in California.

Command:

db.wineratings1.find({ "State": "California" })

Output:

```
test> db.wineratings1.find({ "State": "California" })
[
  {
    _id: '0',
    Country: 'US',
    Brand: "Martha's Vineyard",
    Rating: '96',
    Price: '235',
    State: 'California',
    Province1: 'Napa Valley',
    Province2: 'Napa',
    Variety: 'Cabernet Sauvignon',
    Winery: 'Heitz'
  },
  {
    _id: '2',
    Country: 'US',
    Brand: 'Special Selected Late Harvest',
    Rating: '96',
    Price: '90',
    State: 'California',
    Province1: 'Knights Valley',
    Province2: 'Sonoma',
    Variety: 'Sauvignon Blanc',
    Winery: 'Macauley'
  },
```

3.3.3. Analysis 3 : Calculate the average rating of all wines.

Command:

```
db.wineratings1.aggregate([
   { $group: { _id: null, avgRating: { $avg: { $toDouble: "$Rating" } } } }
])
```

Output:

```
test> db.wineratings1.aggregate([ { $group: { _id: null, avgRating: { $avg: { $toDouble: "$Rating" } } } }])
[ { _id: null, avgRating: 88.74064902906295 } ]
test>
```

3.3.4. Analysis 4 : Find wines with a rating less than or equal to 50.

Command:

db.wineratings1.find({ "Rating": { $lte: "50" } })

Output:

```
test> db.wineratings1.find({ "Rating": { $lte: "50" } })
[
  {
    _id: '2145',
    Country: 'France',
    Brand: '',
    Rating: '100',
    Price: '848',
    State: 'Bordeaux',
    Province1: 'Pessac-Léognan',
    Province2: '',
    Variety: 'Bordeaux-style White Blend',
    Winery: 'Château Haut-Brion'
  }
]
test>
```

3.3.5. Analysis 5: Calculate the total price of all wines from the US.

Command:

```
db.wineratings1.aggregate([
  { $match: { "Country": "US" } },
  { $group: { _id: null, totalUSPrice: { $sum: { $toDouble: "$Price" } } } }
])
```

Output:

```
test> db.wineratings1.aggregate([ { $match: { "Country": "US" } }, { $group: { _id: null, totalUSPrice: { $sum: { $toDouble: "$Price" }
 } } }])
[ { _id: null, totalUSPrice: 239313 } ]
test>
```

3.4. Student 4 - 2023MT03115

Command used to import document :

- mongoimport --collection='wineratings1' --file='/home/centos/winedataset.json' --jsonArray

Command used to connect to the mongo shell script:

- mongosh

3.4.1. Analysis 1 : Find the cheapest wine

Command:

```
db.wineratings1.find().sort({ Price: 1 }).limit(1)
```

Output:

```
test> db.wineratings1.find().sort({ Price: 1 }).limit(1)
[
  {
    _id: '90',
    Country: 'US',
    Brand: '',
    Rating: '86',
    Price: '10',
    State: 'California',
    Province1: 'California',
    Province2: 'California Other',
    Variety: 'Cabernet Sauvignon',
    Winery: 'Belle Ambiance'
  }
]
test>
```

3.4.2. Analysis 2 : Find wines with a rating equal to 100

Command:

```
db.wineratings1.find({ “Rating”: “100” })
```

Output:

```
test> db.wineratings1.find({ "Rating": "100" })
[
  {
    _id: '2145',
    Country: 'France',
    Brand: '',
    Rating: '100',
    Price: '848',
    State: 'Bordeaux',
    Province1: 'Pessac-Léognan',
    Province2: '',
    Variety: 'Bordeaux-style White Blend',
    Winery: 'Château Haut-Brion'
  }
]
test>
```

3.4.3. Analysis 3 : Find wines which are produced in Germany.

Command:

```
db.wineratings1.find({ Country: "Germany" })
```

Output:

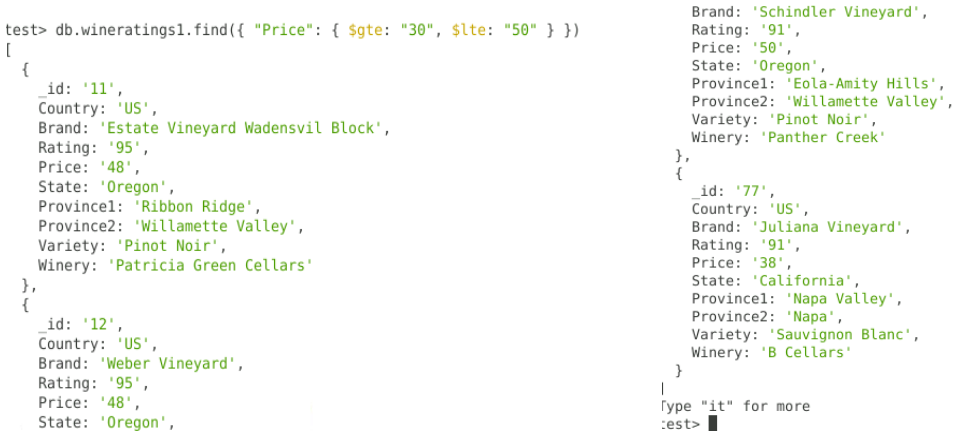
```
test> db.wineratings1.find({ Country: "Germany" })
[
  {
    _id: '331',
    Country: 'Germany',
    Brand: 'Graacher Himmelreich Kabinett',
    Rating: '91',
    Price: '23',
    State: 'Mosel',
    Province1: '',
    Province2: '',
    Variety: 'Riesling',
    Winery: 'Friedrich-Wilhelm-Gymnasium'
  },
  {
    _id: '348',
    Country: 'Germany',
    Brand: 'Saulheimer Probstey Trocken',
    Rating: '91',
    Price: '39',
    State: 'Rheinhessen',
    Province1: '',
    Province2: '',
    Variety: 'Silvaner',
    Winery: 'Thörle'
  },
  {
    _id: '367',
    Country: 'Germany',
    Brand: 'Dry',
    Brand: 'Trittenheimer Apotheke Auslese',
    Rating: '91',
    Price: '23',
    State: 'Mosel',
    Province1: '',
    Province2: '',
    Variety: 'Riesling',
    Winery: 'Bollig-Lehnert'
  },
  {
    _id: '610',
    Country: 'Germany',
    Brand: 'Ürziger Würzgarten Spätlese',
    Rating: '91',
    Price: '28',
    State: 'Mosel',
    Province1: '',
    Province2: '',
    Variety: 'Riesling',
    Winery: 'C.H. Berres'
  }
]
Type "it" for more
test>
```

3.4.4. Analysis 4 : Find wines with a price between $30 and $50

Command:

```
db.wineratings1.find({ “Price”: { $gte: 30, $lte: 50 } })
```

Output:



```
test> db.wineratings1.find({ "Price": { $gte: "30", $lte: "50" } })
[
  {
    _id: '11',
    Country: 'US',
    Brand: 'Estate Vineyard Wadensvil Block',
    Rating: '95',
    Price: '48',
    State: 'Oregon',
    Province1: 'Ribbon Ridge',
    Province2: 'Willamette Valley',
    Variety: 'Pinot Noir',
    Winery: 'Patricia Green Cellars'
  },
  {
    _id: '12',
    Country: 'US',
    Brand: 'Weber Vineyard',
    Rating: '95',
    Price: '48',
    State: 'Oregon',
    Brand: 'Schindler Vineyard',
    Rating: '91',
    Price: '50',
    State: 'Oregon',
    Province1: 'Eola-Amity Hills',
    Province2: 'Willamette Valley',
    Variety: 'Pinot Noir',
    Winery: 'Panther Creek'
  },
  {
    _id: '77',
    Country: 'US',
    Brand: 'Juliana Vineyard',
    Rating: '91',
    Price: '38',
    State: 'California',
    Province1: 'Napa Valley',
    Province2: 'Napa',
    Variety: 'Sauvignon Blanc',
    Winery: 'B Cellars'
  }
]
Type "it" for more
test>
```

3.4.5. Analysis 5 : Identify the total price of Bordeaux-style White Blend variety wines

Command:

```
db.wineratings1.aggregate([
  {
    $match: { "Variety": "Bordeaux-style White Blend" }
  },
  {
    $group: {
      _id: null,
      totalPrice: { $sum: { $toDouble: "$Price" } }
    }
  }
])
```

Output:

```
test> db.wineratings1.aggregate([
...
...     {
...
...         $match: { "Variety": "Bordeaux-style White Blend" }
...
...     },
...
...     {
...
...         $group: {
...
...             _id: null,
...
...             totalPrice: { $sum: { $toDouble: "$Price" } }
...
...         }
...
...     }
...
... ])
[ { _id: null, totalPrice: 10830 } ]
test>
```

3.5. Student 5 - 2023MT03075

Command used to import document :

- mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray

```
[centos@master Downloads]$ mongoimport --collection='wineratings1' --file='/home/centos/Downloads/winedataset.json' --jsonArray
2023-11-08T10:14:46.623+0000    connected to: mongodb://localhost/
2023-11-08T10:14:47.181+0000    15346 document(s) imported successfully. 0 document(s) failed to import.
[centos@master Downloads]$
[centos@master Downloads]$
```

Command used to connect to the mongo shell script:

- mongosh

```
[centos@master Downloads]$ mongosh
Current Mongosh Log ID: 654b60134788c1ca50c13b3c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.6.0
Using MongoDB:          6.0.2
Using Mongosh:          1.6.0

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------
   The server generated these startup warnings when booting
   2023-11-08T10:05:21.526+00:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
   2023-11-08T10:05:21.526+00:00: /sys/kernel/mm/transparent_hugepage/enabled is 'always'. We suggest setting it to 'never'
   2023-11-08T10:05:21.527+00:00: /sys/kernel/mm/transparent_hugepage/defrag is 'always'. We suggest setting it to 'never'
   2023-11-08T10:05:21.527+00:00: vm.max_map_count is too low
------

------
   Enable MongoDB's free cloud-based monitoring service, which will then receive and display
   metrics about your deployment (disk utilization, CPU, operation statistics, etc).

   The monitoring data will be available on a MongoDB website with a unique URL accessible to you
   and anyone you share the URL with. MongoDB may use this information to make product
   improvements and to suggest MongoDB products and deployment options to you.

   To enable free monitoring, run the following command: db.enableFreeMonitoring()
   To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
------

test>
```

3.5.1. Analysis 1 : Find wines with a variety of "Sauvignon Blanc."

Command:

```
db.wineratings1.find({ "Variety": "Sauvignon Blanc" })
```

Output:

```
test> db.wineratings1.find({ "Variety": "Sauvignon Blanc" })
[
  {
    _id: '2',
    Country: 'US',
    Brand: 'Special Selected Late Harvest',
    Rating: '96',
    Price: '90',
    State: 'California',
    Province1: 'Knights Valley',
    Province2: 'Sonoma',
    Variety: 'Sauvignon Blanc',
    Winery: 'Macauley'
  },
  {
    _id: '52',
    Country: 'France',
    Brand: 'Les 7 Hommes',
    Rating: '90',
    Price: '42',
    State: 'Loire Valley',
    Province1: 'Sancerre',
    Province2: '',
    Variety: 'Sauvignon Blanc',
    Winery: 'Cherrier Frères'
  },
  {
    _id: '56',
    Country: 'France',
    Province2: '',
    Variety: 'Sauvignon Blanc',
    Winery: 'Francis Blanchet'
  },
  {
    _id: '672',
    Country: 'France',
    Brand: 'Cuvée C.M.',
    Rating: '88',
    Price: '28',
    State: 'Loire Valley',
    Province1: 'Sancerre',
    Province2: '',
    Variety: 'Sauvignon Blanc',
    Winery: 'Jean-Max Roger'
  },
  {
    _id: '702',
    Country: 'US',
    Brand: 'D',
    Rating: '90',
    Price: '36',
    State: 'California',
    Province1: 'Happy Canyon of Santa Barbara',
    Province2: 'Central Coast',
    Variety: 'Sauvignon Blanc',
    Winery: 'Margerum'
  }
]
Type "it" for more
test>
```

3.5.2. Analysis 2 : Calculate the total price of all wines from Spain.

Command:

```
db.wineratings1.aggregate([
  {
    $match: { "Country": "Spain" }
  },
  {
    $group: {
      _id: null,
      totalPrice: { $sum: { $toDouble: "$Price" } }
    }
  }
])
```

Output:

```
test> db.wineratings1.aggregate([
...
...     {
...
...         $match: { "Country": "Spain" }
...
...     },
...
...     {
...
...         $group: {
...
...             _id: null,
...
...             totalPrice: { $sum: { $toDouble: "$Price" } }
...
...         }
...
...     }
...
... ])
[ { _id: null, totalPrice: 33496 } ]
```

3.5.3. Analysis 3 : Find wines with a brand name containing "Late Harvest."

Command:

db.wineratings1.find({ "Brand": { $regex: "Late Harvest", $options: "i" } })

Output:

```
test> db.wineratings1.find({ "Brand": { $regex: "Late Harvest", $options: "i" } })
[
  {
    _id: '2',
    Country: 'US',
    Brand: 'Special Selected Late Harvest',
    Rating: '96',
    Price: '90',
    State: 'California',
    Province1: 'Knights Valley',
    Province2: 'Sonoma',
    Variety: 'Sauvignon Blanc',
    Winery: 'Macauley'
  },
  {
    _id: '104',
    Country: 'US',
    Brand: 'Tears of Dew Late Harvest',
    Rating: '86',
    Price: '20',
    State: 'California',
    Province1: 'Paso Robles',
    Province2: 'Central Coast',
    Variety: 'Moscato',
    Winery: 'EOS'
  },
  {
    _id: '625',
    Country: 'US',
    Brand: 'Griva Vineyard Late Harvest'
    _id: '14724',
    Country: 'US',
    Brand: 'Late Harvest',
    Rating: '88',
    Price: '20',
    State: 'California',
    Province1: 'Anderson Valley',
    Province2: 'Mendocino/Lake Counties',
    Variety: 'Gewürztraminer',
    Winery: 'Husch'
  },
  {
    _id: '14801',
    Country: 'US',
    Brand: "Special Select Late Harvest Saralee's Vineyards 375 ml",
    Rating: '94',
    Price: '35',
    State: 'California',
    Province1: 'Russian River Valley',
    Province2: 'Sonoma',
    Variety: 'White Riesling',
    Winery: 'Arrowood'
  }
]
Type "it" for more
```

3.5.4. Analysis 4 : Find wines with a price greater than $100.

Command:

db.wineratings1.find({ "Price": { $gt: "100" } })

Output:

```
test> db.wineratings1.find({ "Price": { $gt: "100" } })
[
  {
    _id: '0',
    Country: 'US',
    Brand: "Martha's Vineyard",
    Rating: '96',
    Price: '235',
    State: 'California',
    Province1: 'Napa Valley',
    Province2: 'Napa',
    Variety: 'Cabernet Sauvignon',
    Winery: 'Heitz'
  },
  {
    _id: '1',
    Country: 'Spain',
    Brand: 'Carodorum Selección Especial Reserva',
    Rating: '96',
    Price: '110',
    State: 'Northern Spain',
    Province1: 'Toro',
    Province2: '',
    Variety: 'Tinta de Toro',
    Winery: 'Bodega Carmen Rodríguez'
  },
  {
    _id: '2',
    Country: 'US',
```

3.5.5. Analysis 5: Group wines by brand and calculate the average price for each brand.

Command:

```
db.wineratings1.aggregate([
  {
    $group: {
      _id: "$Brand",
      averagePrice: { $avg: { $toDouble: "$Price" } }
    }
  }
])
```

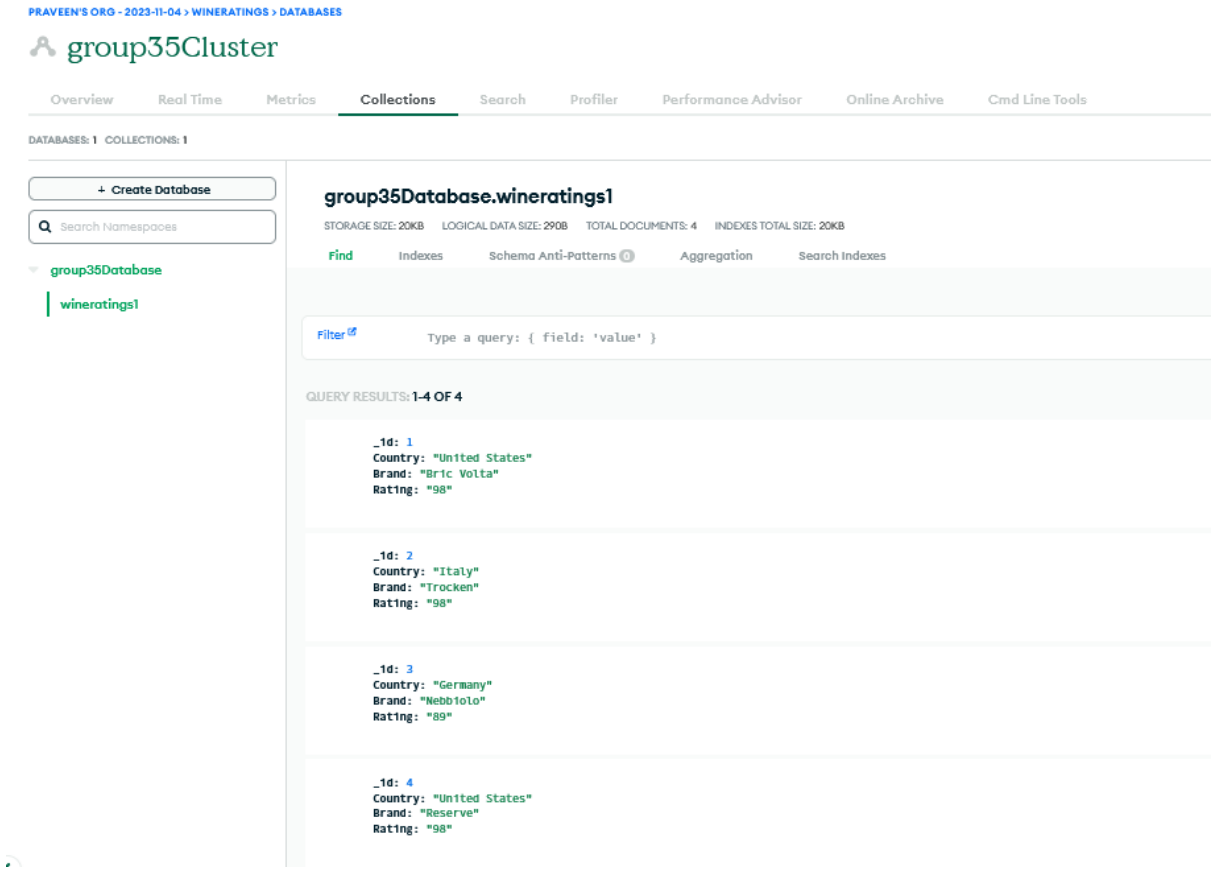
Output:

```
test> db.wineratings1.aggregate([
...
...     {
...
...         $group: {
...
...             _id: "$Brand",
...
...             averagePrice: { $avg: { $toDouble: "$Price" } }
...
...         }
...
...     }
...
... ])
[
  { _id: 'Latifeh', averagePrice: 100 },
  { _id: 'Somers Ranch', averagePrice: 38 },
  { _id: "Jane's Vineyard", averagePrice: 35 },
  { _id: 'Varvàra', averagePrice: 28 },
  {
    _id: 'Terroir Series Finca Coletto Single Vineyard',
    averagePrice: 55
  },
  { _id: 'Finca del Castillo Joven', averagePrice: 9 },
  { _id: 'Espero', averagePrice: 40 },
  { _id: 'Intense', averagePrice: 116.5 },
  { _id: 'Ramos Vineyard', averagePrice: 78 },
  { _id: 'HD', averagePrice: 16 },
  { _id: 'Seventeen Forty Reserve', averagePrice: 56 },
  { _id: 'Occhio di Pernice', averagePrice: 65 },
```

4. Connecting MongoDBAtlas

4.1. MongoDB Cluster using Mongo Atlas Service :

Cluster Name : **group35Cluster**
Database Name : group35Database
Collection Name : WineRatings1
Connection URL :
mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.net



4.2. CRUD operations using PYTHON Programming language :

```
import pymongo

# MongoDB connection string
connection_string =
"mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.
net/?retryWrites=true"
db_name = "group35Database"

# Create a MongoDB client
client = pymongo.MongoClient(connection_string)

try:
    print("\nSuccessfully connected to MongoDB!")
    database = client[db_name]
    print(f"\nCreated / Retrieved {db_name} Database to MongoDB!")

    collection = database["wineratings1"]
    # Delete the collection and start fresh - add before the initial inserts
    collection.drop()
    print("\nCreated / Retrieved wineratings1 Collection to MongoDB!")
```

```
    # Create operation
    documents = [
        {"_id": 1, "Country": "United States", "Brand": "Bric Volta", "Rating": "98"},
        {"_id": 2, "Country": "Italy", "Brand": "Trocken", "Rating": "98"},
        {"_id": 3, "Country": "Germany", "Brand": "Nebbiolo", "Rating": "98"},
        {"_id": 4, "Country": "United States", "Brand": "Reserve", "Rating": "98"},
        {"_id": 5, "Country": "United States", "Brand": "Nebbiolo", "Rating": "98"},
    ]

    collection.insert_many(documents)
    print("\nInserted records to MongoDB!")

    # Read the collection
    cursor = collection.find()
    for document in cursor:
        print(document)

    # Update a document
    # Print the third document before update
    third = collection.find_one({"_id": 3})
    print("\nBefore update:", third)

    collection.update_one(
        {"_id": 3},
        {
            "$set": {
                "Country": "Germany",
                "Brand": "Nebbiolo",
                "Rating": "89"
            }
        }
    )

    print("\nUpdated third document:")
    updated_doc = collection.find_one({"_id": 3})
    print("\nAfter update : third document:", updated_doc)

    # Delete Operation
    print("\nCollection size before delete operation:", collection.count_documents({}),
"documents.\n")
    fifth = collection.find_one({"_id": 5})
    collection.delete_one(fifth)
    print("\nCollection size after delete operation:", collection.count_documents({}),
"documents.\n")
    print("\nRecords after delete operation:")
    cursor = collection.find()
    for document in cursor:
        print(document)

except pymongo.errors.PyMongoError as e:
    print(f"An error occurred: {e}")

finally:
    client.close()
```

4.3. CRUD operations using JAVA Programming language :

```
package com.mongodb.atlas;

import com.mongodb.ConnectionString;
import com.mongodb.MongoClientSettings;
import com.mongodb.MongoException;
import com.mongodb.ServerApi;
import com.mongodb.ServerApiVersion;
import com.mongodb.client.MongoClient;
import com.mongodb.client.MongoClients;
import com.mongodb.client.MongoCollection;
import com.mongodb.client.MongoCursor;
import com.mongodb.client.MongoDatabase;
import com.mongodb.client.model.Filters;

import java.util.ArrayList;
import java.util.List;

import org.bson.Document;

public class ConnectToMongoAtlas {

   public static void main(String[] args) {
        String connectionString =
"mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.net/?retryWrites=true&w=majority";
        String dbName = "group35Database";

        ServerApi serverApi =
ServerApi.builder().version(ServerApiVersion.V1).build();

        MongoClientSettings settings = MongoClientSettings.builder()
                .applyConnectionString(new
ConnectionString(connectionString)).serverApi(serverApi).build();

        try (MongoClient mongoClient = MongoClients.create(settings)) {
            try {
                System.out.println("\n Successfully connected to
MongoDB!");

                MongoDatabase database =
mongoClient.getDatabase(dbName);
                System.out.println("\n Created / Retrieved
group35Database Database to MongoDB!");


                MongoCollection collection =
database.getCollection("wineratings1");
                 // Delete the collection and start fresh - add before the
initial inserts
        collection.drop();
                System.out.println("\n Created / Retrieved wineratings1
Collection to MongoDB!");

                //Create operation
```

```
                        List<Document> documents = new
ArrayList<Document>();
                        documents.add(new Document("_id",
1).append("Country", "United States").append("Brand", "Bric Volta")
                                        .append("Rating", "98"));
                        documents.add(new Document("_id",
2).append("Country", "Italy").append("Brand", "Trocken")
                                        .append("Rating", "98"));
                        documents.add(new Document("_id",
3).append("Country", "Germany").append("Brand", "Nebbiolo")
                                        .append("Rating", "98"));
                        documents.add(new Document("_id",
4).append("Country", "United States").append("Brand", "Reserve")
                                        .append("Rating", "98"));
                        documents.add(new Document("_id",
5).append("Country", "United States").append("Brand", "Nebbiolo")
                                        .append("Rating", "98"));
                        collection.insertMany(documents);
                         System.out.println("\n Inserted records to MongoDB!");

                        // Read the collection

                        try (MongoCursor<Document> cursor =
collection.find().iterator()) {
                                while (cursor.hasNext()) {
                                        Document currentDoc = cursor.next();

System.out.printf(currentDoc.toJson()+"\n");
                                }
                        } catch (MongoException me) {
                                System.err.println("Unable to find any records in
MongoDB due to an error: " + me);
                        }

                        // Update a document
                        // print the third document before update.
                        Document third = (Document)
collection.find(Filters.eq("_id", 3)).first();
                        System.out.println("\n Before update : "+third.toJson());

                        collection.updateOne(new Document("_id", 3), new
Document("$set", new Document("_id", 3)
                                        .append("Country",
"Germany").append("Brand", "Nebbiolo").append("Rating", "89")));

                        System.out.println("\n Updated third document:");
                        Document updatedDocObj = (Document)
collection.find(Filters.eq("_id", 3)).first();
                        System.out.println("\n After update : third
document:"+updatedDocObj.toJson());

                        //Delete Operation
                        System.out.println("\n Collection size before delete
operation: " + collection.countDocuments() + " documents. \n");
                        Document fifth = (Document)
collection.find(Filters.eq("_id", 5)).first();
```

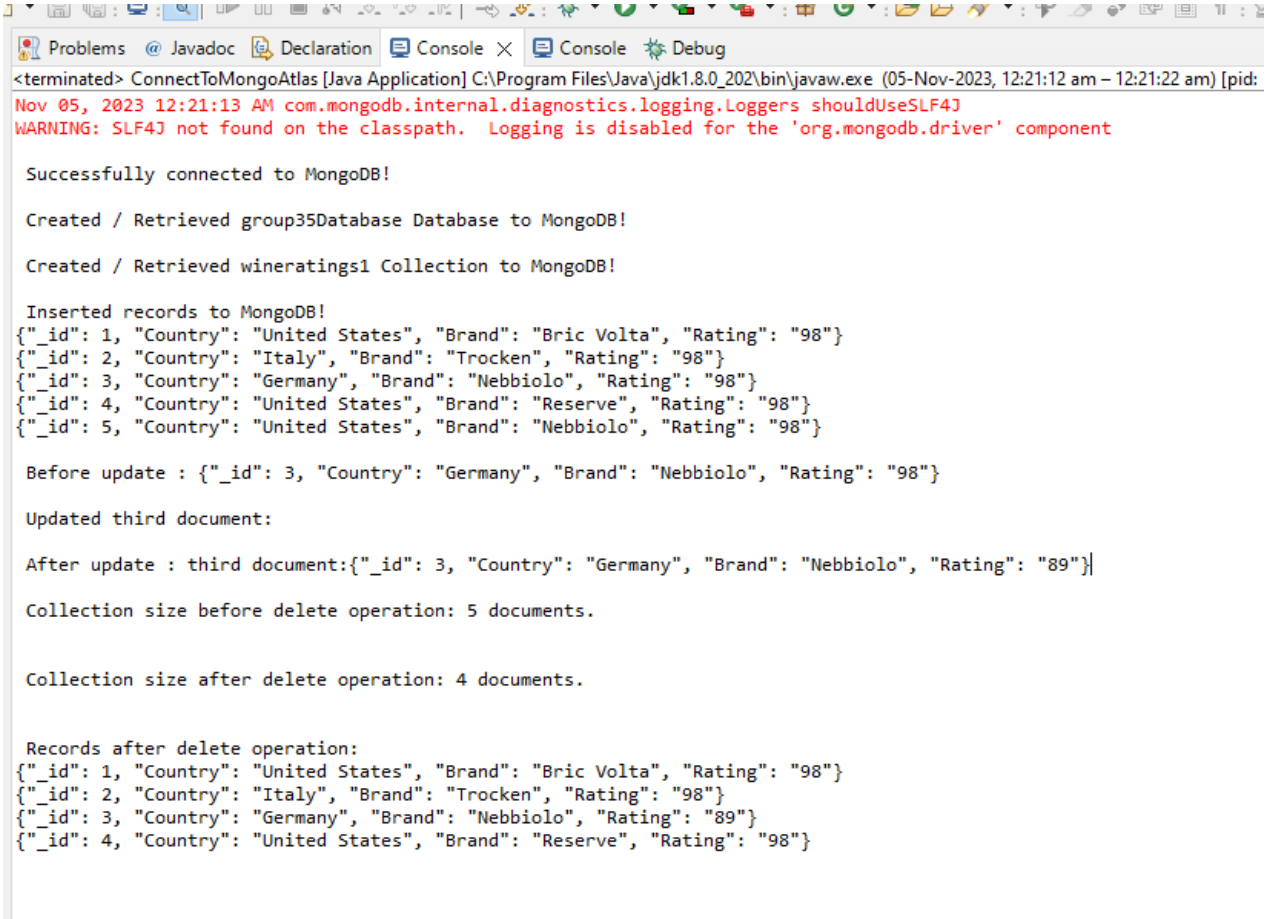
```
                    collection.deleteOne(fifth);
                    System.out.println("\n Collection size after delete
operation: " + collection.countDocuments() + " documents. \n");
                    System.out.println("\n Records after delete operation:");
                    try (MongoCursor<Document> cursor =
collection.find().iterator()) {
                        while (cursor.hasNext()) {
                            Document currentDoc = cursor.next();

System.out.printf(currentDoc.toJson()+"\n");
                        }
                    } catch (MongoException me) {
                        System.err.println("Unable to find any records in
MongoDB due to an error: " + me);
                    }

                } catch (MongoException e) {
                    e.printStackTrace();
                }
            }
        }

}
```

4.4. Output



4.5. Analysis on performance of read/write operations with respect to different consistency configurations

4.6.1. READ - MAJORITY & WRITE - MAJORITY

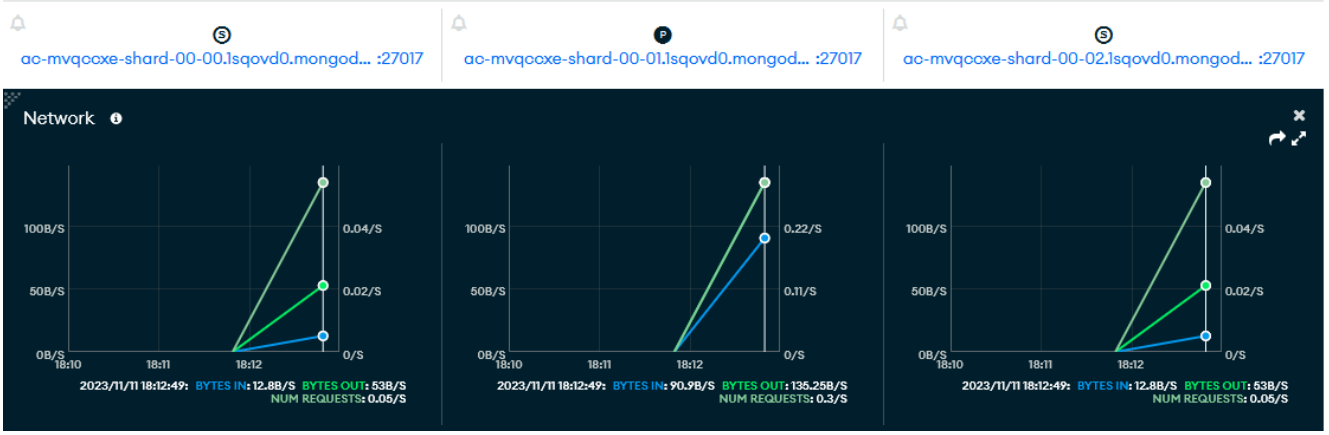
In this case, the READ operation will only return data that has been acknowledged by a majority of the replica set members. This ensures a higher level of consistency but may introduce some latency.
Similarly, the WRITE operation will only be considered successful if it is acknowledged by a majority of the replica set members.

code to configure :

String connectionString = "mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.net/?retryWrites=true";

MongoDatabase database = mongoClient.getDatabase(dbName);
database.withReadConcern(ReadConcern.MAJORITY);
database.withWriteConcern(WriteConcern.MAJORITY);

Metrics : Network Performance for a WRITE & READ operation for READ - MAJORITY & WRITE - MAJORITY:



4.6.2. READ - MAJORITY & WRITE - W1

In this case, the WRITE operation will be considered successful, if the operation has been written to the primary node of the replica set. This ensures that at least one node in the replica set has acknowledged the write, providing a basic level of durability.
But, the READ operation will only return data that has been acknowledged by a majority of the replica set members.

code to configure :

String connectionString = "mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.net/?retryWrites=true";

MongoDatabase database = mongoClient.getDatabase(dbName);

database.withReadConcern(ReadConcern.MAJORITY);
database.withWriteConcern(WriteConcern.W1);

Metrics : Network Performance for a WRITE & READ operation for READ - MAJORITY & WRITE - W1:

4.6.3. READ - LOCAL & WRITE - W1

In this case, read operations will return the most recent data available on the node receiving the read request. This might not reflect the most recent write operations across the entire replica set.
WRITE operation will be considered successful, if the operation has been written to the primary node of the replica set.

code to configure :

String connectionString = "mongodb+srv://jpraveenanand87:<password>@group35cluster.1sqovd0.mongodb.net/?retryWrites=true";

MongoDatabase database = mongoClient.getDatabase(dbName);
database.withReadConcern(ReadConcern.LOCAL);
database.withWriteConcern(WriteConcern.W1);

Metrics : Network Performance for a WRITE & READ operation for READ - LOCAL & WRITE - W1:

Conclusion :

The Atlas Monitoring - Network Chart reveals significant variations in data both entering and exiting MongoDB w.r.t the various consistencies along with the performance of secondary and primary clusters at a network level.