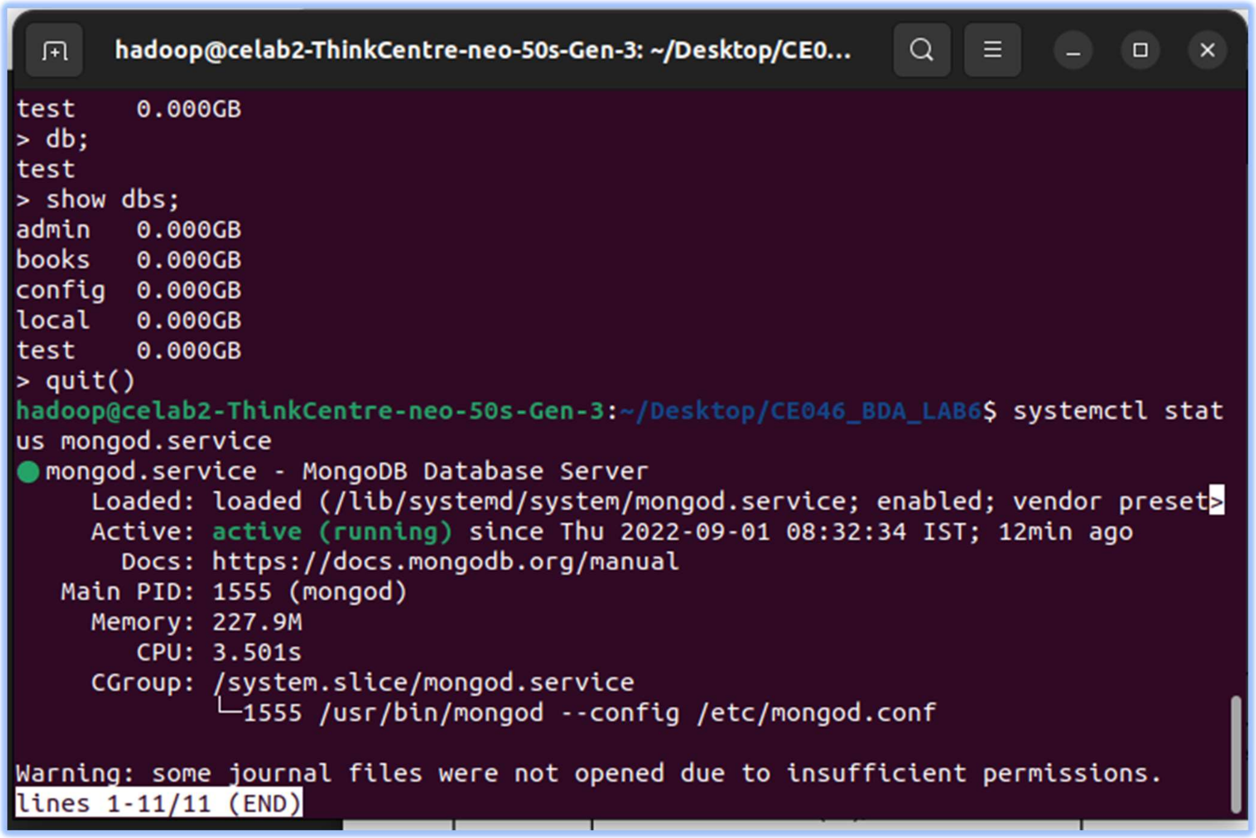
LAB - 6

Name	Keval D Gandevia
Roll Number	CE046
ID	19CEUEG017
Subject	Big Data Analytics

Aim: Connecting to NoSQL database and querying to provide analysis using api like aggregation, etc. To be able to successfully import/export from/to csv.

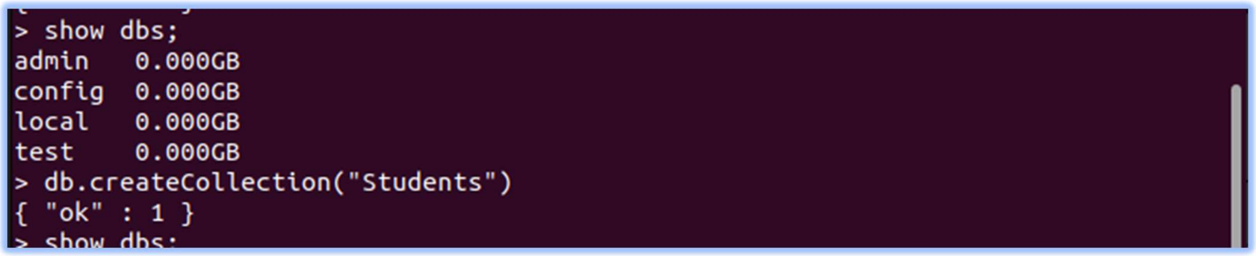
- **Verifying mongod service:**



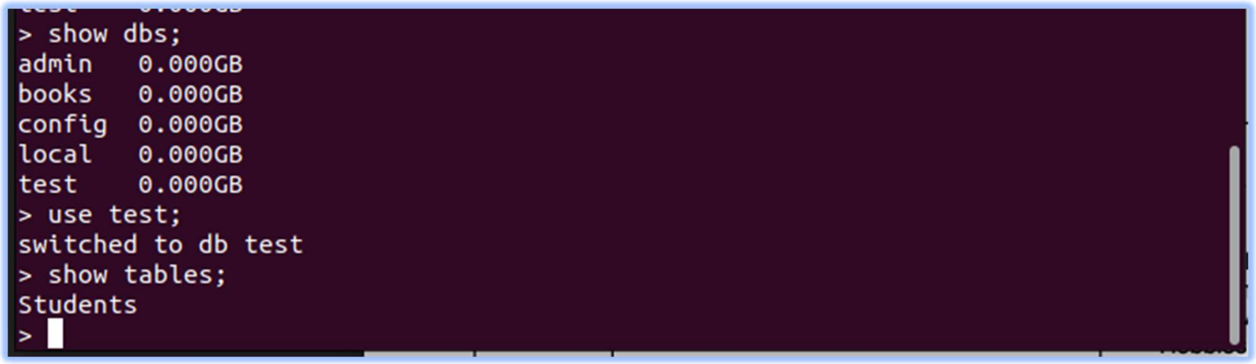
```
test     0.000GB
> db;
test
> show dbs;
admin    0.000GB
books    0.000GB
config   0.000GB
local    0.000GB
test     0.000GB
> quit()
hadoop@celab2-ThinkCentre-neo-50s-Gen-3:~/Desktop/CE046_BDA_LAB6$ systemctl stat
us mongod.service
● mongod.service - MongoDB Database Server
     Loaded: loaded (/lib/systemd/system/mongod.service; enabled; vendor preset>
     Active: active (running) since Thu 2022-09-01 08:32:34 IST; 12min ago
       Docs: https://docs.mongodb.org/manual
   Main PID: 1555 (mongod)
     Memory: 227.9M
        CPU: 3.501s
     CGroup: /system.slice/mongod.service
             └─1555 /usr/bin/mongod --config /etc/mongod.conf

Warning: some journal files were not opened due to insufficient permissions.
lines 1-11/11 (END)
```

❖ Creating a collection:

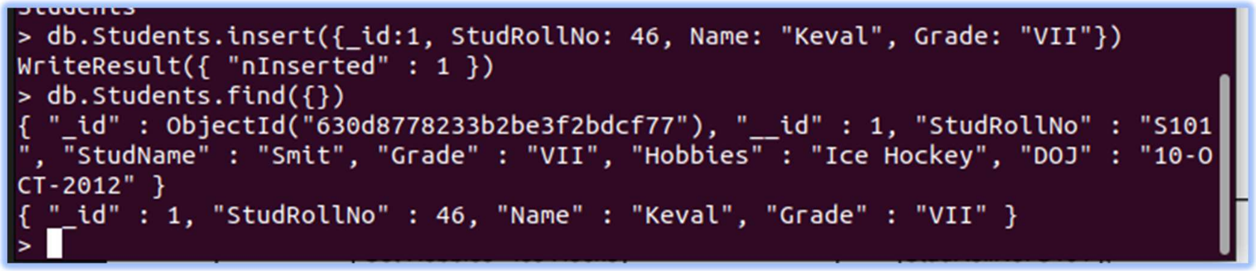


```
> show dbs;
admin    0.000GB
config   0.000GB
local    0.000GB
test     0.000GB
> db.createCollection("Students")
{ "ok" : 1 }
> show dbs;
```



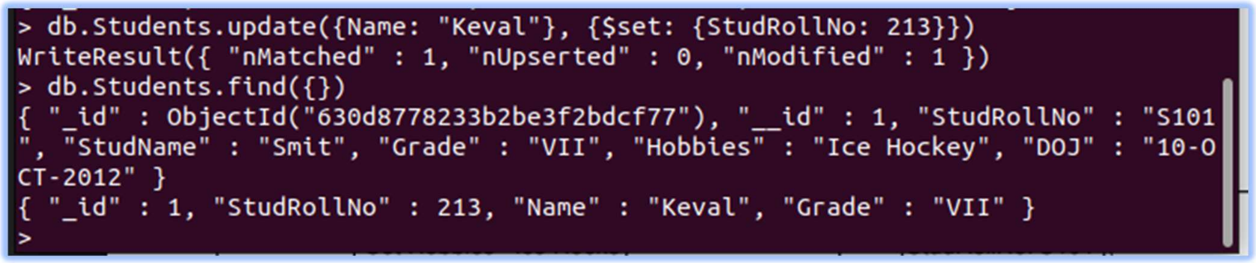
```
> show dbs;
admin    0.000GB
books    0.000GB
config   0.000GB
local    0.000GB
test     0.000GB
> use test;
switched to db test
> show tables;
Students
> 
```

❖ Inserting data into the table:



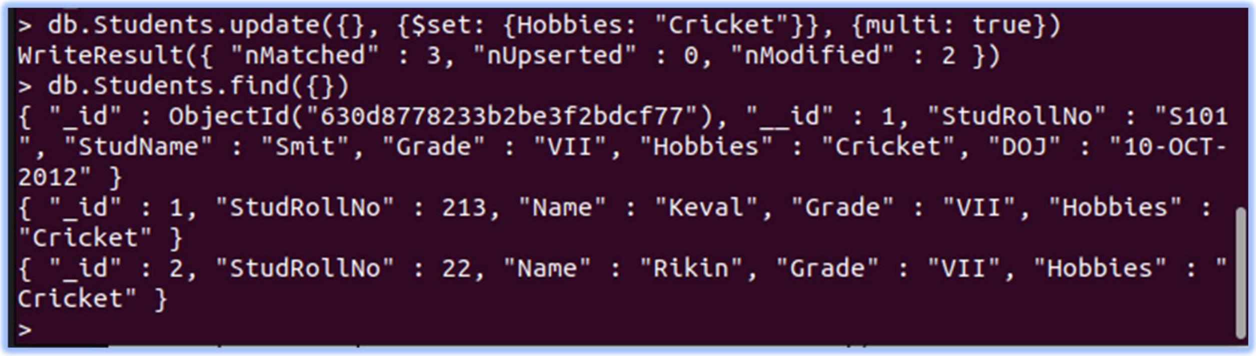
```
> db.Students.insert({_id:1, StudRollNo: 46, Name: "Keval", Grade: "VII"})
WriteResult({ "nInserted" : 1 })
> db.Students.find({})
{ "_id" : ObjectId("630d8778233b2be3f2bdcf77"), "__id" : 1, "StudRollNo" : "S101
", "StudName" : "Smit", "Grade" : "VII", "Hobbies" : "Ice Hockey", "DOJ" : "10-O
CT-2012" }
{ "_id" : 1, "StudRollNo" : 46, "Name" : "Keval", "Grade" : "VII" }
> 
```

❖ Updating the documents of the collection:



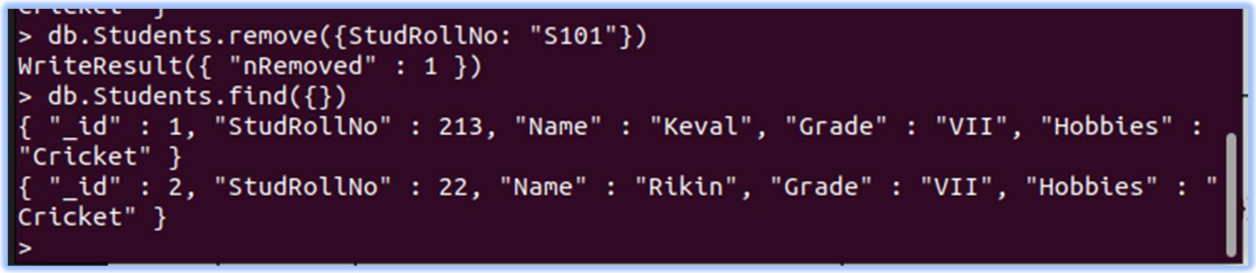
```
> db.Students.update({Name: "Keval"}, {$set: {StudRollNo: 213}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Students.find({})
{ "_id" : ObjectId("630d8778233b2be3f2bdcf77"), "__id" : 1, "StudRollNo" : "S101
", "StudName" : "Smit", "Grade" : "VII", "Hobbies" : "Ice Hockey", "DOJ" : "10-O
CT-2012" }
{ "_id" : 1, "StudRollNo" : 213, "Name" : "Keval", "Grade" : "VII" }
>
```

❖ Updating multiple rows simultaneously:



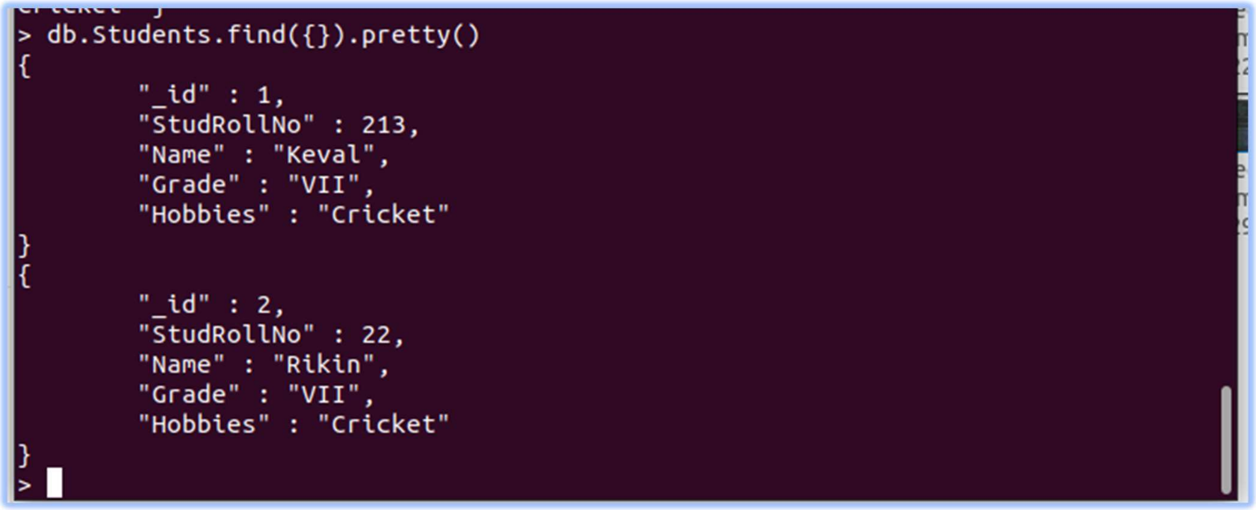
```
> db.Students.update({}, {$set: {Hobbies: "Cricket"}}, {multi: true})
WriteResult({ "nMatched" : 3, "nUpserted" : 0, "nModified" : 2 })
> db.Students.find({})
{ "_id" : ObjectId("630d8778233b2be3f2bdcf77"), "__id" : 1, "StudRollNo" : "S101
", "StudName" : "Smit", "Grade" : "VII", "Hobbies" : "Cricket", "DOJ" : "10-OCT-
2012" }
{ "_id" : 1, "StudRollNo" : 213, "Name" : "Keval", "Grade" : "VII", "Hobbies" :
"Cricket" }
{ "_id" : 2, "StudRollNo" : 22, "Name" : "Rikin", "Grade" : "VII", "Hobbies" : "
Cricket" }
>
```

❖ Removing the documents from the collection:

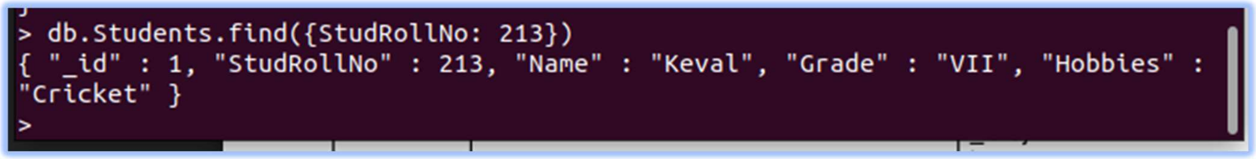


```
> db.Students.remove({StudRollNo: "S101"})
WriteResult({ "nRemoved" : 1 })
> db.Students.find({})
{ "_id" : 1, "StudRollNo" : 213, "Name" : "Keval", "Grade" : "VII", "Hobbies" :
"Cricket" }
{ "_id" : 2, "StudRollNo" : 22, "Name" : "Rikin", "Grade" : "VII", "Hobbies" : "
Cricket" }
>
```

❖ Getting the documents from the collection:



```
> db.Students.find({}).pretty()
{
        "_id" : 1,
        "StudRollNo" : 213,
        "Name" : "Keval",
        "Grade" : "VII",
        "Hobbies" : "Cricket"
}
{
        "_id" : 2,
        "StudRollNo" : 22,
        "Name" : "Rikin",
        "Grade" : "VII",
        "Hobbies" : "Cricket"
}
>
```

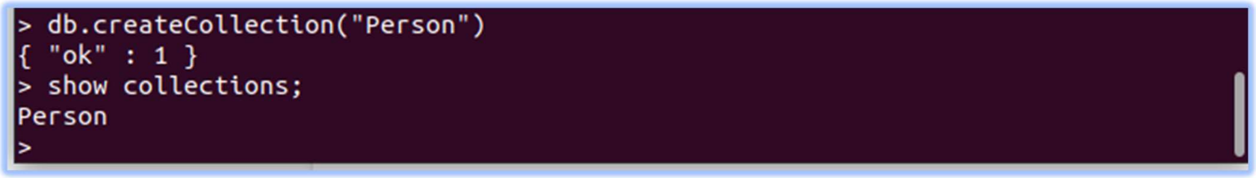


```
> db.Students.find({StudRollNo: 213})
{ "_id" : 1, "StudRollNo" : 213, "Name" : "Keval", "Grade" : "VII", "Hobbies" :
"Cricket" }
>
```

Exercises:

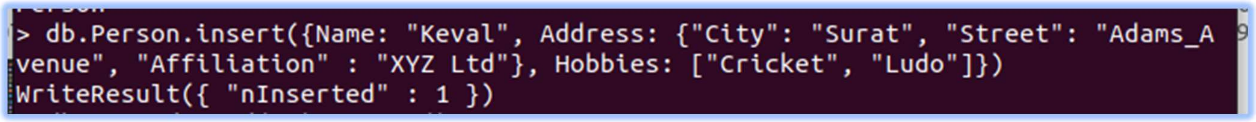
Q. 2: Write the insert method to store the following document in MongoDB.

- ❖ **Creating a collection named 'person':**



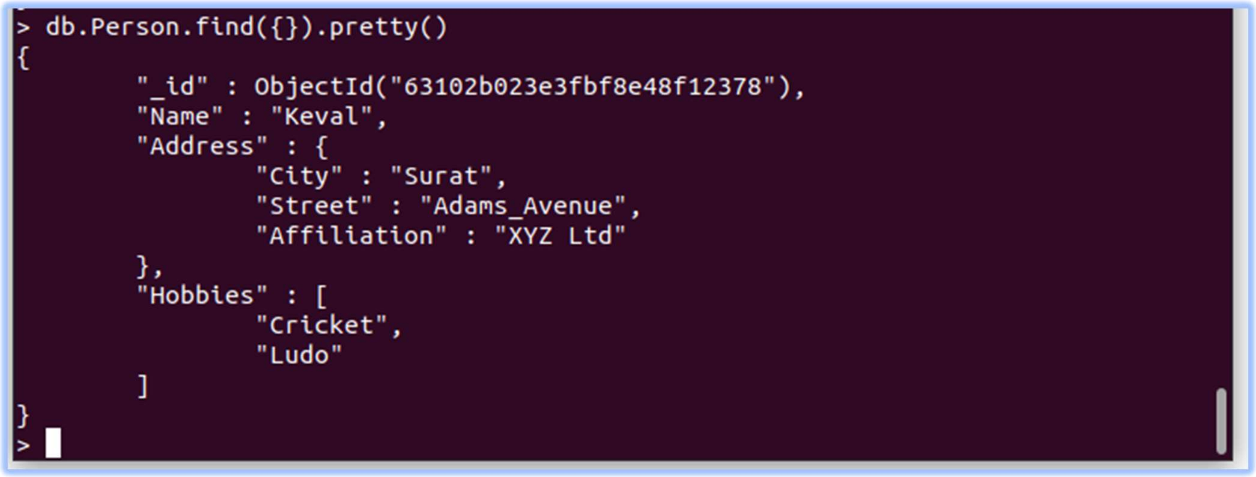
```
> db.createCollection("Person")
{ "ok" : 1 }
> show collections;
Person
>
```

- ❖ **Inserting a data as given in the question into the collection:**



```
> db.Person.insert({Name: "Keval", Address: {"City": "Surat", "Street": "Adams_A
venue", "Affiliation" : "XYZ Ltd"}, Hobbies: ["Cricket", "Ludo"]})
WriteResult({ "nInserted" : 1 })
```

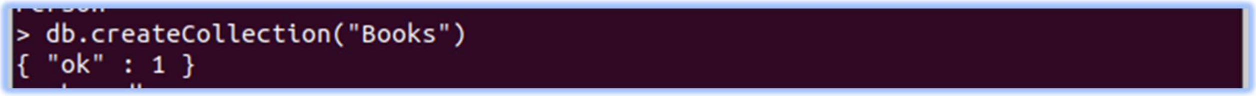
- ❖ **Output using find () method:**



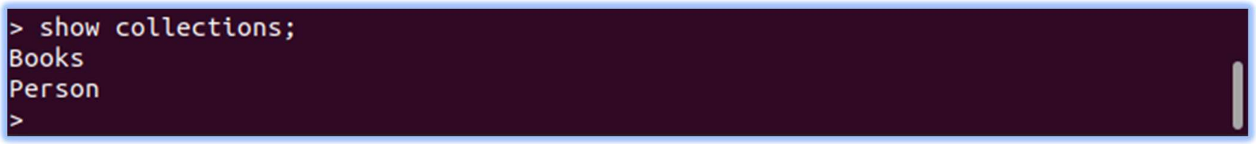
```
> db.Person.find({}).pretty()
{
        "_id" : ObjectId("63102b023e3fbf8e48f12378"),
        "Name" : "Keval",
        "Address" : {
                "City" : "Surat",
                "Street" : "Adams_Avenue",
                "Affiliation" : "XYZ Ltd"
        },
        "Hobbies" : [
                "Cricket",
                "Ludo"
        ]
}
>
```

Q. 3: Practice MapReduce programming in MongoDB.

❖ Creating a collection named ‘Books’:

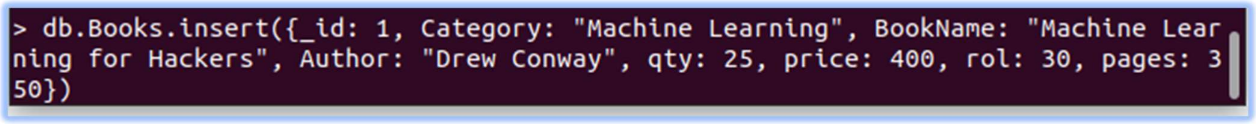


```
> db.createCollection("Books")
{ "ok" : 1 }
```

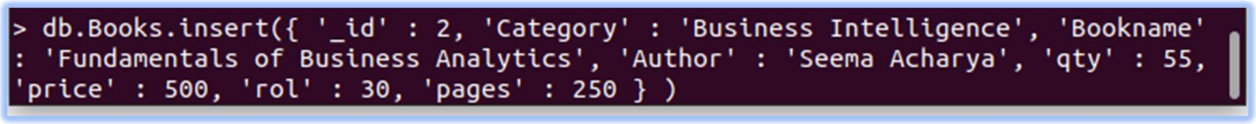


```
> show collections;
Books
Person
>
```

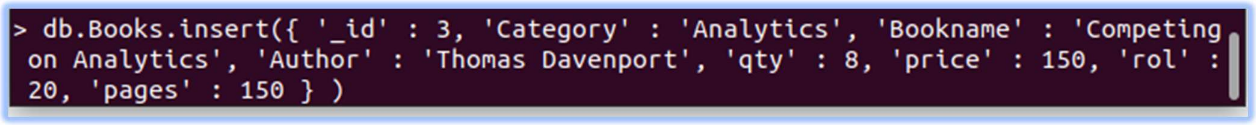
❖ Inserting demo data into the collection ‘Books’:



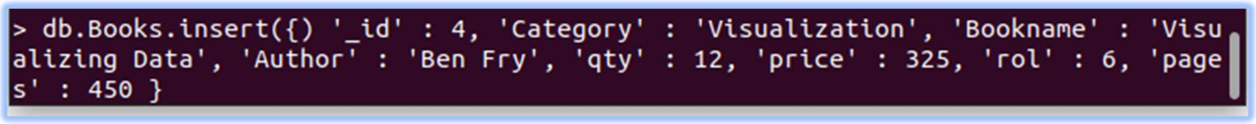
```
> db.Books.insert({_id: 1, Category: "Machine Learning", BookName: "Machine Lear
ning for Hackers", Author: "Drew Conway", qty: 25, price: 400, rol: 30, pages: 3
50})
```



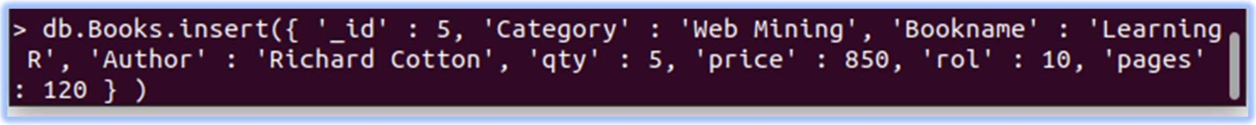
```
> db.Books.insert({ '_id' : 2, 'Category' : 'Business Intelligence', 'Bookname'
: 'Fundamentals of Business Analytics', 'Author' : 'Seema Acharya', 'qty' : 55,
'price' : 500, 'rol' : 30, 'pages' : 250 } )
```



```
> db.Books.insert({ '_id' : 3, 'Category' : 'Analytics', 'Bookname' : 'Competing
 on Analytics', 'Author' : 'Thomas Davenport', 'qty' : 8, 'price' : 150, 'rol' :
 20, 'pages' : 150 } )
```

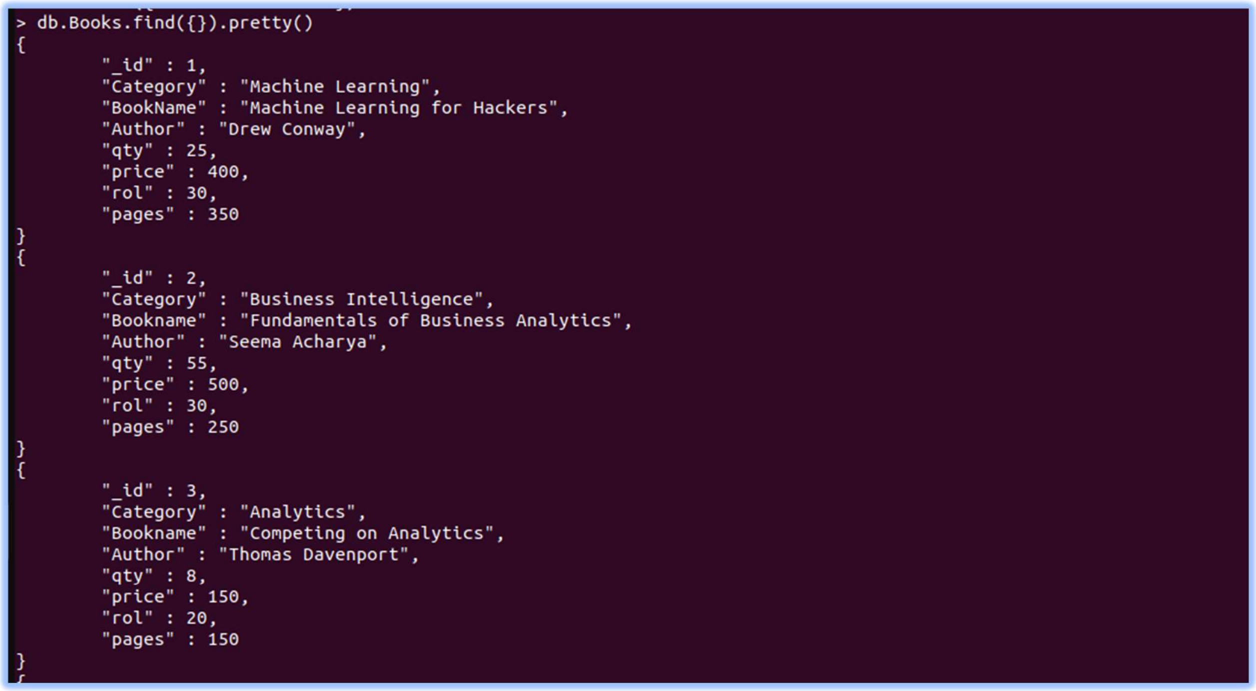


```
> db.Books.insert({) '_id' : 4, 'Category' : 'Visualization', 'Bookname' : 'Visu
alizing Data', 'Author' : 'Ben Fry', 'qty' : 12, 'price' : 325, 'rol' : 6, 'page
s' : 450 }
```

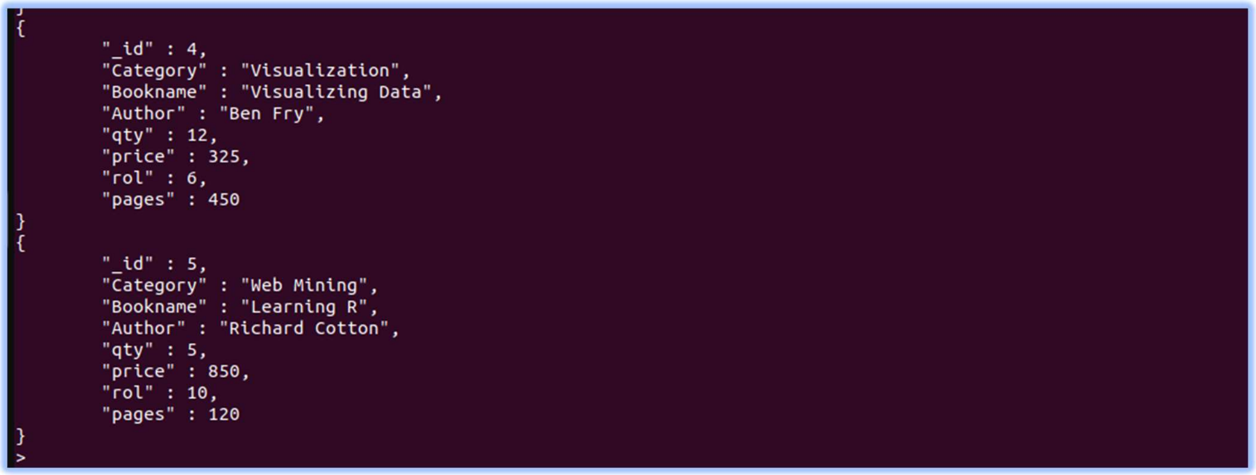


```
> db.Books.insert({ '_id' : 5, 'Category' : 'Web Mining', 'Bookname' : 'Learning
 R', 'Author' : 'Richard Cotton', 'qty' : 5, 'price' : 850, 'rol' : 10, 'pages'
: 120 } )
```

❖ **Data is added successfully.**

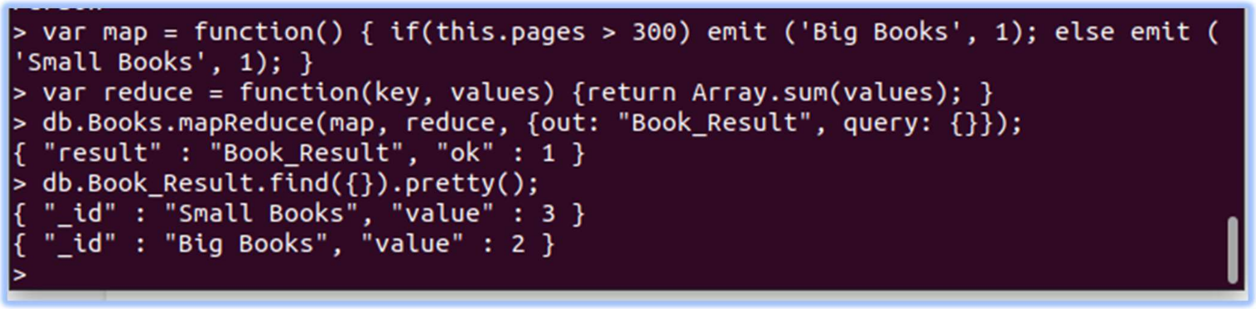


```
> db.Books.find({}).pretty()
{
        "_id" : 1,
        "Category" : "Machine Learning",
        "BookName" : "Machine Learning for Hackers",
        "Author" : "Drew Conway",
        "qty" : 25,
        "price" : 400,
        "rol" : 30,
        "pages" : 350
}
{
        "_id" : 2,
        "Category" : "Business Intelligence",
        "Bookname" : "Fundamentals of Business Analytics",
        "Author" : "Seema Acharya",
        "qty" : 55,
        "price" : 500,
        "rol" : 30,
        "pages" : 250
}
{
        "_id" : 3,
        "Category" : "Analytics",
        "Bookname" : "Competing on Analytics",
        "Author" : "Thomas Davenport",
        "qty" : 8,
        "price" : 150,
        "rol" : 20,
        "pages" : 150
}
```



```
{
        "_id" : 4,
        "Category" : "Visualization",
        "Bookname" : "Visualizing Data",
        "Author" : "Ben Fry",
        "qty" : 12,
        "price" : 325,
        "rol" : 6,
        "pages" : 450
}
{
        "_id" : 5,
        "Category" : "Web Mining",
        "Bookname" : "Learning R",
        "Author" : "Richard Cotton",
        "qty" : 5,
        "price" : 850,
        "rol" : 10,
        "pages" : 120
}
>
```

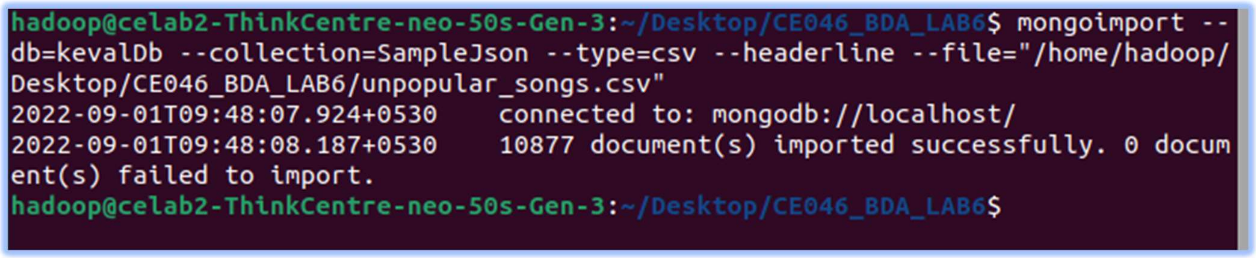
❖ **Creating functions for map and reduce and using mapReduce method of the mongoDB and storing the result into the collection ‘Book_Result’.**



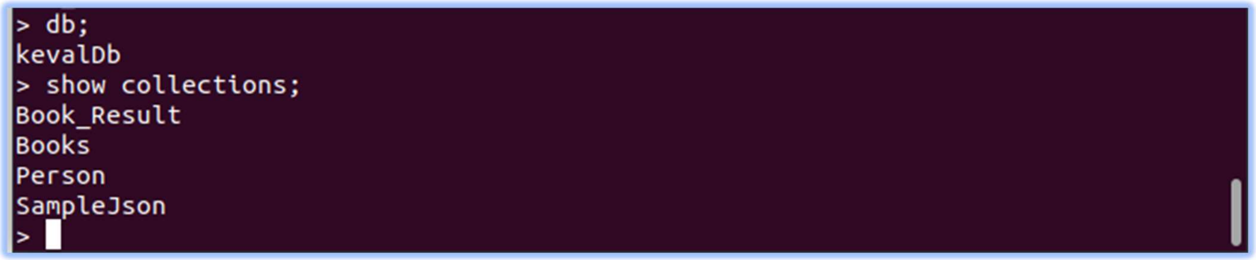
```
> var map = function() { if(this.pages > 300) emit ('Big Books', 1); else emit (
'Small Books', 1); }
> var reduce = function(key, values) {return Array.sum(values); }
> db.Books.mapReduce(map, reduce, {out: "Book_Result", query: {}});
{ "result" : "Book_Result", "ok" : 1 }
> db.Book_Result.find({}).pretty();
{ "_id" : "Small Books", "value" : 3 }
{ "_id" : "Big Books", "value" : 2 }
>
```

Q. 4: Practice import and export and aggregation in MongoDB.

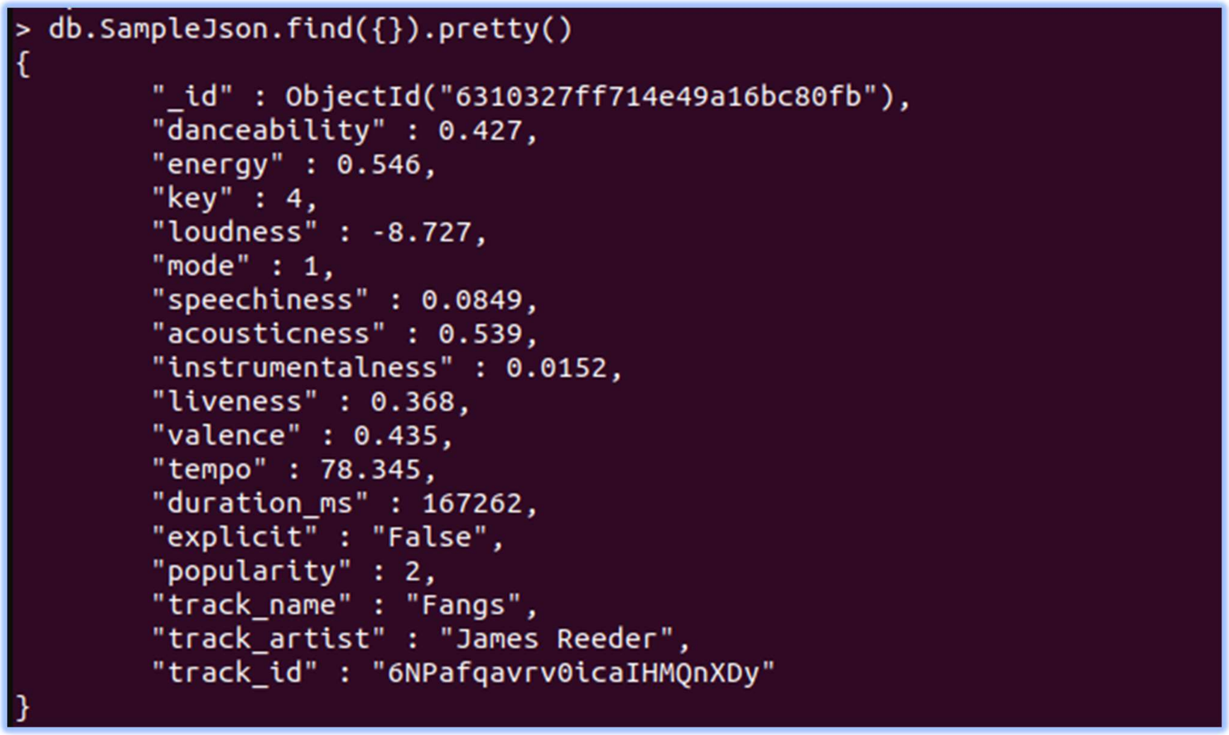
❖ Importing a csv file into collection using a command mongoimport.



```
hadoop@celab2-ThinkCentre-neo-50s-Gen-3:~/Desktop/CE046_BDA_LAB6$ mongoimport --
db=kevalDb --collection=SampleJson --type=csv --headerline --file="/home/hadoop/
Desktop/CE046_BDA_LAB6/unpopular_songs.csv"
2022-09-01T09:48:07.924+0530    connected to: mongodb://localhost/
2022-09-01T09:48:08.187+0530    10877 document(s) imported successfully. 0 docum
ent(s) failed to import.
hadoop@celab2-ThinkCentre-neo-50s-Gen-3:~/Desktop/CE046_BDA_LAB6$
```

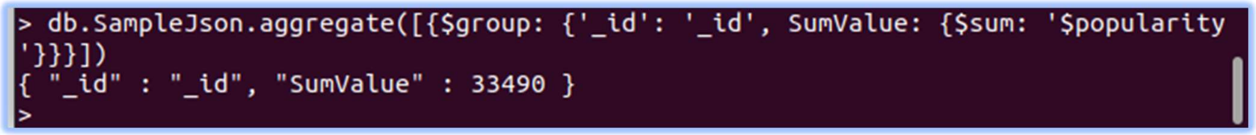


```
> db;
kevalDb
> show collections;
Book_Result
Books
Person
SampleJson
>
```



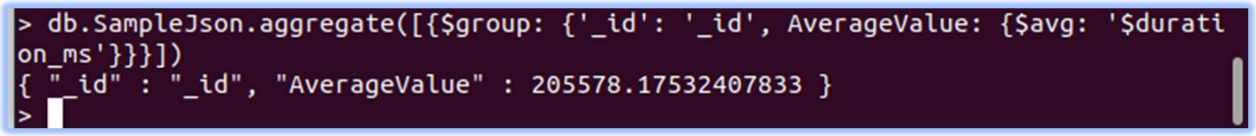
```
> db.SampleJson.find({}).pretty()
{
        "_id" : ObjectId("6310327ff714e49a16bc80fb"),
        "danceability" : 0.427,
        "energy" : 0.546,
        "key" : 4,
        "loudness" : -8.727,
        "mode" : 1,
        "speechiness" : 0.0849,
        "acousticness" : 0.539,
        "instrumentalness" : 0.0152,
        "liveness" : 0.368,
        "valence" : 0.435,
        "tempo" : 78.345,
        "duration_ms" : 167262,
        "explicit" : "False",
        "popularity" : 2,
        "track_name" : "Fangs",
        "track_artist" : "James Reeder",
        "track_id" : "6NPafqavrv0icaIHMQnXDy"
}
```

❖ Compute the sum of the values in the first numeric column.



```
> db.SampleJson.aggregate([{$group: {'_id': '_id', SumValue: {$sum: '$popularity
'}}}])
{ "_id" : "_id", "SumValue" : 33490 }
>
```

❖ Compute the average of the values in the second numeric column.



```
> db.SampleJson.aggregate([{$group: {'_id': '_id', AverageValue: {$avg: '$durati
on_ms'}}}])
{ "_id" : "_id", "AverageValue" : 205578.17532407833 }
> 
```