

Assignment 1

use mydb;

```
db.books.save({_id:1,Category:"Machine Learning",Bookname:"Machine Learning for Hackers",Author:"David Conway",qty:25,price:400,rol:30,pages:350
});
db.books.save({_id:2,Category:"Buisness Intelligence",Bookname:"Fundamental of Buisness Analytics",Author:"Seeema Acharya",qty:55,price:500,rol:30,pages:250 });
db.books.save({_id:3,Category:"Analytics",Bookname:"Competing on Analytics",Author:"Thomas Davensport",qty:8,price:150,rol:20,pages:150 });
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
db.books.save({_id:4,Category:"Visualization",Bookname:"Visualizing Data",Author:"Ben Fry",qty:12,price:325,rol:6,pages:450 });
db.books.save({_id:5,Category:"Web Mining",Bookname:"Learning R",Author:"Richard C",qty:55,price:850,rol:10,pages:120 });
STEP 1-
```

```
Command Prompt - mongo
switched to db mydb
> db.books.save({_id:1,Category:"Machine Learning",Bookname:"Machine Learning for Hackers",Author:"David Conway",qty:25,price:400,rol:30,pass:350 })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.books.save({_id:2,Category:"Business Intelligence",Bookname:"Fundamental of Business Analytics",Author:"Seeema Acharya",qty:55,price:500,rol:30,pass:250 })
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 0 })
> db.books.save({_id:3,Category:"Analytics",Bookname:"Competing on Analytics",Author:"Thomas Davensport",qty:8,price:150,rol:20,pass:150 });
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
> db.books.save({_id:4,Category:"Visualization",Bookname:"Visualizing Data",Author:"Ben Fry",qty:12,price:325,rol:6,pass:450 });
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 4 })
> db.books.find().pretty()

  "_id" : 1,
  "Category" : "Machine Learning",
  "Bookname" : "Machine Learning for Hackers",
  "Author" : "David Conway",
  "qty" : 25,
  "price" : 400,
  "rol" : 30,
  "pass" : 350

  "_id" : 2,
  "Category" : "Business Intelligence",
  "Bookname" : "Fundamental of Business Analytics",
  "Author" : "Seeema Acharya",
  "qty" : 55,
  "price" : 500,
  "rol" : 30,
  "pass" : 250

  "_id" : 3,
  "Category" : "Analytics",
  "Bookname" : "Competing on Analytics",
  "Author" : "Thomas Davensport",
  "qty" : 8,
  "price" : 150,
  "rol" : 20,
  "pass" : 150

  "_id" : 4,
  "Category" : "Visualization",
  "Bookname" : "Visualizing Data",
  "Author" : "Ben Fry",
  "qty" : 12,
  "price" : 325,
  "rol" : 6,
  "pass" : 450

db.books.find()
```

STEP 2-

```
Command Prompt - mongo
> db.books.find()
{ "_id" : 1, "Category" : "Machine Learning", "Bookname" : "Machine Learning for Hackers", "Author" : "David Conway", "qty" : 25, "price" : 400, "rol" : 30, "pages" : 350 }
{ "_id" : 2, "Category" : "Business Intelligence", "Bookname" : "Fundamental of Business Analytics", "Author" : "Seeema Acharya", "qty" : 55, "price" : 500, "rol" : 30, "pages" : 250 }
{ "_id" : 3, "Category" : "Analytics", "Bookname" : "Competing on Analytics", "Author" : "Thomas Davensport", "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 C", "qty" : 55, "price" : 850, "rol" : 10, "pages" : 120 }
```

STEP 3&4&5;

```
db.books.mapReduce( function()
{let key=null,value=null;
  if(this.pages>=300)
  {
    key="Big Books";
```

```

        value=this.pages;
    }
    else{
        key="Small Books";
        value=this.pages;
    }
    emit(key,value);
},
function(key,value)
{ return value.length;
},
{
    out:"Book_Records"}
);

```

```

Command Prompt - mongo
> db.books.mapReduce( function(){ let key=null,value=null; if(this.pages>=300){ key="Big Books"; value=this.pages; } else{ key
="Small Books"; value=this.pages; } emit(key,value);},function(key,value){ return value.length; }, {out:"Book_Records"} );
{
  "result" : "Book_Records",
  "timeMillis" : 41,
  "counts" : {
    "input" : 5,
    "emit" : 5,
    "reduce" : 2,
    "output" : 2
  },
  "ok" : 1
}
> db.Book_Records.find();
{ "_id" : "Big Books", "value" : 2 }
{ "_id" : "Small Books", "value" : 3 }
>

```

Assignment 2

```

Command Prompt - mongo
Microsoft Windows [Version 10.0.18363.1082]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\riyan>cd /d C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d mongoDB -c titanic --type csv --headerline --file "C:\Users\riyan\Documents\titanic.csv"
2020-10-15T16:06:42.729+0530   connected to: localhost
2020-10-15T16:06:42.752+0530   imported 891 documents

```

```
Command Prompt - mongo
> db.MongoDBHandsOn.aggregate([{$group: {_id:null, sum: {$sum: "$Survived"}}}])
{ "_id" : null, "sum" : 342 }
> db.MongoDBHandsOn.aggregate([{$group: {_id:"Average Age", avg: {$avg: "$Age"}}}])
{ "_id" : "Average Age", "avg" : 29.69911764705882 }
>
```

```
Command Prompt - mongo
> db.Country.find()
{ "_id" : 1, "Cities" : [ "Visakhapatnam", "Hyderabad" ] }
{ "_id" : 2, "Cities" : [ "Pune", "Mumbai" ] }
{ "_id" : 3, "Cities" : [ "Bangalore", "Chennai" ] }
{ "_id" : 4, "Cities" : [ "Mysore", "Tirupathi" ] }
> db.Country.find().limit(1)
{ "_id" : 1, "Cities" : [ "Visakhapatnam", "Hyderabad" ] }
> db.Country.find().skip(2).limit(2)
{ "_id" : 3, "Cities" : [ "Bangalore", "Chennai" ] }
{ "_id" : 4, "Cities" : [ "Mysore", "Tirupathi" ] }
> db.Country.update({Cities:"Pune"},{$pull:Cities:"Pune"})
2020-10-15T16:40:53.676+0530 E QUERY [js] SyntaxError: missing } after property list @(shell):1:47
> db.Country.update({Cities:"Pune"},{$pull:Cities:"Pune"})
2020-10-15T16:41:13.641+0530 E QUERY [js] SyntaxError: missing } after property list @(shell):1:47
> db.Country.update({Cities:"Pune"},{$pull:{Cities:"Pune"}})
2020-10-15T16:41:43.818+0530 E QUERY [js] SyntaxError: unterminated string literal @(shell):1:49
> db.Country.update({Cities:"Pune"},{$pull:{Cities:"Pune"}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Country.find()
{ "_id" : 1, "Cities" : [ "Visakhapatnam", "Hyderabad" ] }
{ "_id" : 2, "Cities" : [ "Mumbai" ] }
{ "_id" : 3, "Cities" : [ "Bangalore", "Chennai" ] }
{ "_id" : 4, "Cities" : [ "Mysore", "Tirupathi" ] }
> db.Country.update({_id:3},{$pop:{Cities:1}})
2020-10-15T16:44:33.151+0530 E QUERY [js] SyntaxError: expected expression, got ':' @(shell):1:26
> db.Country.update({_id:3},{$pop:{Cities:1}})
2020-10-15T16:44:41.454+0530 E QUERY [js] SyntaxError: missing ) after argument list @(shell):1:30
> db.Country.update({_id:3},{$pop:{Cities:1}})
2020-10-15T16:45:00.945+0530 E QUERY [js] SyntaxError: missing ) after argument list @(shell):1:30
> db.Country.update({_id:3},{$pop:{Cities:1}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
>
```

```
Command Prompt - mongo
> db.Country.update({'_id':1},{set: {'Cities:2': 'Srinagar'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Country.find()
{ "_id" : 1, "Cities" : [ "Visakhapatnam", "Hyderabad" ], "Cities:2" : "Srinagar" }
{ "_id" : 2, "Cities" : [ "Mumbai" ] }
{ "_id" : 3, "Cities" : [ "Bangalore" ] }
{ "_id" : 4, "Cities" : [ "Mysore", "Tirupathi" ] }
> db.Country.update({'_id':2},{addToSet: {'Cities': 'Jaipur'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Country.find()
{ "_id" : 1, "Cities" : [ "Visakhapatnam", "Hyderabad" ], "Cities:2" : "Srinagar" }
{ "_id" : 2, "Cities" : [ "Mumbai", "Jaipur" ] }
{ "_id" : 3, "Cities" : [ "Bangalore" ] }
{ "_id" : 4, "Cities" : [ "Mysore", "Tirupathi" ] }
>
```