```
> show dbs
B04     0.000GB
admin   0.000GB
config  0.000GB
local   0.000GB
> use B04;
switched to db B04
> db.dropDatabase();
{ "dropped" : "B04", "ok" : 1 }
> use B04;
switched to db B04
>
> db.createCollection("Aadhar");
b.Aadhar.insert({
   Aadharno : 123456,
   Name : "Prathamesh KS",
   MobileNo : 7755922895,
   { "ok" : 1 }
> show collections;
Aadhar
>
> db.Aadhar.insert({
...     Aadharno : 123456,
...     Name : "Prathamesh KS",
...     MobileNo : 7755922895,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 20,
...     Address : "Pune",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
>
> db.Aadhar.insert({
...     Aadharno : 123459,
...     Name : "Sooraj VS",
...     MobileNo : 7755922894,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 21,
...     Address : "Mumbai",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 133455,
...     Name : "Aditya Somani",
...     MobileNo : 7755922893,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 31,
...     Address : "Mumbai",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 125455,
...     Name : "Utkarsh Gurav",
...     MobileNo : 7755922892,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 98,
...     Address : "Nashik",
...     Voted : {"2020" : 0, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 126451,
...     Name : "Henry VS",
...     MobileNo : 7755922824,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 21,
...     Address : "Mumbai",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123465,
...     Name : "Tanvi MD",
...     MobileNo : 7755922214,
...     Gender : "Female",
...     Citizenship : "US",
...     Age : 43,
...     Address : "Newyork",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
>
>
>
>
> db.Aadhar.insert({
...     Aadharno : 123500,
...     Name : "Aakash MD",
...     MobileNo : 7722922214,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 34,
...     Address : "Newyork",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123501,
...     Name : "Rajas S",
...     MobileNo : 775592215,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 23,
...     Address : "London",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123502,
...     Name : "Shreyas MD",
...     MobileNo : 2255922216,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 12,
...     Address : "Indian",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123503,
...     Name : "Chetan LK",
...     MobileNo : 7755922219,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 22,
...     Address : "Indian",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123504,
...     Name : "Hritik S",
...     MobileNo : 7755922405,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 22,
...     Address : "Pune",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 1 }
... });
```

```
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123505,
...     Name : "Shreya LM",
...     MobileNo : 7755922404,
...     Gender : "Female",
...     Citizenship : "US",
...     Age : 15,
...     Address : "Newyork",
...     Voted : {"2020" : 0, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123506,
...     Name : "Sparsh D",
...     MobileNo : 7755922406,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 20,
...     Address : "Mumbai",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123507,
...     Name : "Harshita A",
...     MobileNo : 7755922407,
...     Gender : "Female",
...     Citizenship : "Mexican",
...     Age : 19,
...     Address : "Newyork",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123508,
...     Name : "Bhavika RC",
...     MobileNo : 7755922408,
...     Gender : "Female",
...     Citizenship : "US",
...     Age : 38,
...     Address : "Newyork",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123509,
...     Name : "Ronak M",
...     MobileNo : 7755922409,
...     Gender : "Male",
...     Citizenship : "Indian",
...     Age : 35,
...     Address : "Pune",
...     Voted : {"2020" : 0, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123510,
...     Name : "Aditi LL",
...     MobileNo : 7755922410,
...     Gender : "Female",
...     Citizenship : "US",
...     Age : 48,
...     Address : "Pune",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123511,
...     Name : "Ayush Shah",
...     MobileNo : 7755922411,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 41,
...     Address : "India",
...     Voted : {"2020" : 1, "2019" : 0, "2018" : 1 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123512,
...     Name : "Karandeep H",
...     MobileNo : 7755922412,
...     Gender : "Male",
...     Citizenship : "US",
...     Age : 63,
...     Address : "Mumbai",
...     Voted : {"2020" : 0, "2019" : 0, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.insert({
...     Aadharno : 123513,
...     Name : "Bhumika VC",
...     MobileNo : 7755922413,
...     Gender : "Female",
...     Citizenship : "US",
...     Age : 65,
...     Address : "Pune",
...     Voted : {"2020" : 1, "2019" : 1, "2018" : 0 }
... });
WriteResult({ "nInserted" : 1 })
> db.Aadhar.find().pretty();
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dff5"),
        "Aadharno" : 123456,
        "Name" : "Prathamesh KS",
        "MobileNo" : 7755922895,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 20,
        "Address" : "Pune",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dff6"),
        "Aadharno" : 123459,
        "Name" : "Sooraj VS",
        "MobileNo" : 7755922894,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 21,
        "Address" : "Mumbai",
        "Voted" : {
                "2018" : 1,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dff7"),
        "Aadharno" : 133455,
        "Name" : "Aditya Somani",
        "MobileNo" : 7755922893,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 31,
        "Address" : "Mumbai",
        "Voted" : {
                "2018" : 1,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dff8"),
        "Aadharno" : 125455,
        "Name" : "Utkarsh Gurav",
        "MobileNo" : 7755922892,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 98,
        "Address" : "Nashik",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 0
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dff9"),
        "Aadharno" : 126451,
        "Name" : "Henry VS",
        "MobileNo" : 7755922824,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 21,
        "Address" : "Mumbai",
        "Voted" : {
                "2018" : 1,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dffa"),
        "Aadharno" : 123465,
        "Name" : "Tanvi MD",
        "MobileNo" : 7755922214,
        "Gender" : "Female",
        "Citizenship" : "US",
        "Age" : 43,
        "Address" : "Newyork",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dffb"),
        "Aadharno" : 123500,
        "Name" : "Aakash MD",
        "MobileNo" : 7722922214,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 34,
        "Address" : "Newyork",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dffc"),
        "Aadharno" : 123501,
        "Name" : "Rajas S",
        "MobileNo" : 775592215,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 23,
        "Address" : "London",
        "Voted" : {
                "2018" : 0,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dffd"),
        "Aadharno" : 123502,
        "Name" : "Shreyas MD",
        "MobileNo" : 2255922216,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 12,
        "Address" : "Indian",
        "Voted" : {
                "2018" : 1,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dffe"),
        "Aadharno" : 123503,
        "Name" : "Chetan LK",
        "MobileNo" : 7755922219,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 22,
        "Address" : "Indian",
        "Voted" : {
                "2018" : 1,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e60029afd08c76dfff"),
        "Aadharno" : 123504,
        "Name" : "Hritik S",
        "MobileNo" : 7755922405,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 22,
        "Address" : "Pune",
        "Voted" : {
                "2018" : 1,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e000"),
        "Aadharno" : 123505,
        "Name" : "Shreya LM",
        "MobileNo" : 7755922404,
        "Gender" : "Female",
        "Citizenship" : "US",
        "Age" : 15,
        "Address" : "Newyork",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 0
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e001"),
        "Aadharno" : 123506,
        "Name" : "Sparsh D",
        "MobileNo" : 7755922406,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 20,
        "Address" : "Mumbai",
        "Voted" : {
                "2018" : 0,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e002"),
        "Aadharno" : 123507,
        "Name" : "Harshita A",
        "MobileNo" : 7755922407,
        "Gender" : "Female",
        "Citizenship" : "Mexican",
        "Age" : 19,
        "Address" : "Newyork",
        "Voted" : {
                "2018" : 0,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e003"),
        "Aadharno" : 123508,
        "Name" : "Bhavika RC",
        "MobileNo" : 7755922408,
        "Gender" : "Female",
        "Citizenship" : "US",
        "Age" : 38,
        "Address" : "Newyork",
        "Voted" : {
                "2018" : 1,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e004"),
        "Aadharno" : 123509,
        "Name" : "Ronak M",
        "MobileNo" : 7755922409,
        "Gender" : "Male",
        "Citizenship" : "Indian",
        "Age" : 35,
        "Address" : "Pune",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 0
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e005"),
        "Aadharno" : 123510,
        "Name" : "Aditi LL",
        "MobileNo" : 7755922410,
        "Gender" : "Female",
        "Citizenship" : "US",
        "Age" : 48,
        "Address" : "Pune",
        "Voted" : {
                "2018" : 0,
                "2019" : 1,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e006"),
        "Aadharno" : 123511,
        "Name" : "Ayush Shah",
        "MobileNo" : 7755922411,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 41,
        "Address" : "India",
        "Voted" : {
                "2018" : 1,
                "2019" : 0,
                "2020" : 1
        }
}
{
        "_id" : ObjectId("5fa8c5e70029afd08c76e007"),
        "Aadharno" : 123512,
        "Name" : "Karandeep H",
        "MobileNo" : 7755922412,
        "Gender" : "Male",
        "Citizenship" : "US",
        "Age" : 63,
        "Address" : "Mumbai",
        "Voted" : {
                "2018" : 0,
                "2019" : 0,
                "2020" : 0
        }
}
{
        "_id" : ObjectId("5fa8c5e80029afd08c76e008"),
        "Aadharno" : 123513,
        "Name" : "Bhumika VC",
        "MobileNo" : 7755922413,
        "Gender" : "Female",
        "Citizenship" : "US",
        "Age" : 65,
        "Address" : "Pune",
        "Voted" : {
                "2018" : 0,
                "2019" : 1,
                "2020" : 1
        }
}
>

#### using mapreduce ####

> var mapf = function(){emit(this.Gender,this.Age)};
> var reducef = function(key,values){return Array.avg(values)};
> db.Aadhar.mapReduce(mapf,reducef,{query:{Aadharno:{$gt:123456}},out:'result1'});
ult1.find().pretty();{ "result" : "result1", "ok" : 1 }
> db.result1.find().pretty();
{ "_id" : "Male", "value" : 34.07692307692308 }
{ "_id" : "Female", "value" : 38 }

#### using mapreduce ####


> db.Aadhar.mapReduce(
...                     function(){emit(this.Citizenship, this.Age)},
...                     function(key,values){return Array.avg(values)},
...                     {out:'result2'}
...                 )
{ "result" : "result2", "ok" : 1 }
> db.result2.find().pretty();
{ "_id" : "Indian", "value" : 33.625 }
{ "_id" : "US", "value" : 36.63636363636363 }
{ "_id" : "Mexican", "value" : 19 }
```

Assignment B04

- Date of Completion : 28-10-20

- Date of Submission: 4-11-20

- Title - Map Reduce

- Problem Statement:
Write & implement example of Map Reduce as data processing paradigm with suitable collection using MongoDB

- Learning Objectives:
1. To understand concept of Map Reduce as data preprocessing paradigm
2. To compare Map Reduce & aggregation pipeline in MongoDB

- Learning Outcomes:
1) Learn & implement map-reduce for condensing large volumes of data into useful aggregation results
2) To know the difference between the use of map reduce & aggregation pipline in MongoDB.

- Software & Hardware Req.
1) MongoDB 4.4* server
2) 64 bit OS: Ubuntu 20.14, 8GB RAM, intel i5

- Theory:

Map reduce is data processing paradigm for condensing large volumes of data into useful results.

Syntax:

```
db. collections.mapReduce (
      mapFunction () {
            emit (
                  <key>,
                  <value>
                  )
      },
      reduce function (
            key,
            value
            ){
            //operation & return
      };    }
      {
            query : {
               <query document>
            }
            out : <collection name>
      }
)
```

Map & Reduce phases

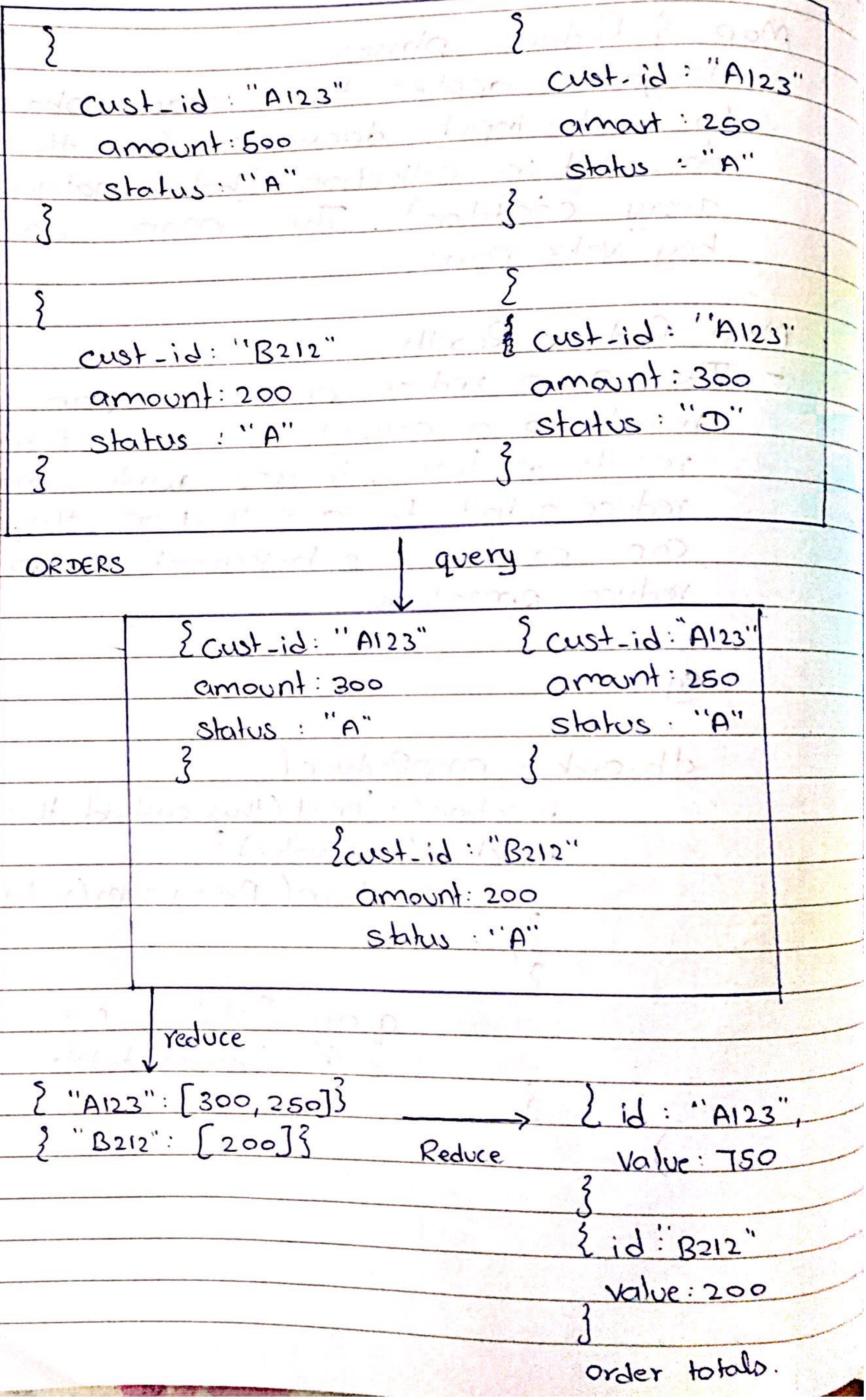
- MongoDB applies the "map phase" to each input document (ie: the document in collection that matches the query condition). The map emits key value pairs.

Map Reduce Results:

- The map reduce operation can write results to a collection on return the resuts in line, if you write map reduce output to a collection they can produce sub-sequent map reduce. operations.

eg:-

```
db.orders. mapReduce(
        function() {emit(this.cust_id, this.amaunt);}
        function(key, value) {
                return( Array.sum(values)
        },
        {
                query : {status: "A"},
                out : "order_totals"
        }
)
```



{
cust_id : "A123"
amount : 500
status : "A"
}

{
cust_id : "A123"
amont : 250
status : "A"
}

{
cust_id : "B212"
amount : 200
status : "A"
}

{
cust_id : "A123"
amount : 300
status : "D"
}

ORDERS

query

{cust_id : "A123"
amount : 300
status : "A"
}

{cust_id : "A123"
amount : 250
status : "A"
}

{cust_id : "B212"
amount : 200
status : "A"

reduce

{ "A123" : [300, 250]}
{ "B212" : [200]}

Reduce

{ id : "A123",
Value : 750
}
{ id : "B212"
value : 200
}

order totals.

MongoDB:

- Average CPU utilization during Map Reduce is high.
- Use of aggregation is faster than Map reduce.
- Output of single pipeline cannot surpass 16MB.
- parallely aggregation pipeline can be used for smaller datasets.

- Conclusion

In this assignment we studied the use of MongoDB Map Reduce function & implemented it successfully.