1. Install mongodb --------- run msi file ----download a zip file -------extract that file in some folder
2. Create a folder c:\mydata\mongodbdemos
3. To start mongodb server

C:\program file\mongodb\server\3.6\bin>mongod -dbpath c:\mydata\mongodbdemos

- c:\mydata\mongodbdemos this is folder for storing database
4. To connect to server
   C:\program file\mongodb\server\3.6\bin> mongo.exe <portno>   --------27017 default port
5. To import data to mongodb server

C:\program file\mongodb\server\3.6\bin> mongoimport --db test --collection restaurants --file c:\mydata\mongodb\samplerestaurent.json

6. To connect to client
   bin>mongo <ip address> <port>

Commands

> use employee

>db.emptab.insert({empid:123,ename:'kishori',sal:234567})

-show dbs   -----------------show databases which contains atleast one collection

-show collections

>db.emptab.find() ------------list all documents and all key value pairs

>db.emp.find({"deptno":10});   -------------list all employees from dept 10 and all key value pairs

---------to list all employees name and designation who are working in department 10

> db.emp.find({"deptno":10},{"ename":1,"designation":1});

---------to list all employees name and designation

> db.emp.find({},{"ename":1,"designation":1});

Updatemany

db.restaurant.updateMany(

   { violations: { $gt: 4 } },

```
    { $set: { "Review" : true } }

  );
```

```
{ "_id" : 1, "item" : "abc", "price" : 10, "fee" : 2, date: ISODate("20
14-03-01T08:00:00Z") }
{ "_id" : 2, "item" : "jkl", "price" : 20, "fee" : 1, date: ISODate("20
14-03-01T09:00:00Z") }
{ "_id" : 3, "item" : "xyz", "price" : 5,  "fee" : 0, date: ISODate("20
14-03-15T09:00:00Z") }
```

```
db.sales.aggregate(
   [
     { $project: { item: 1, total: { $add: [ "$price", "$fee" ] } } }
   ]
)
```

Db.employees.aggregate({$project:{"total_pay":{$add:{$salary,$bonus}}}})

Db.employees.aggregate({$project:{"experience":{$subtract:}}})

How to use mapreduce

{custid:a123,amt:200;status:'a'}

{custid:a222,amt:300;status:'a'}

{custid:a123,amt:500;status:'a'}

{custid:a123,amt:300;status:'b'}

{custid:a222,amt:300;status:'a'}

{custid:a253,amt:500;status:'a'}

Db.orders.mapreduce(function(){emit(this.custid,this.amt)

},function(){

  Return Arrays.sum(values)

},

{

Query:{status:'a'},

Out:ordercoll

})

Finalize: function to send reduce output to
keeptemp : Boolean result of collection u need after connection is close

Steps for replication

1. Use enterprise mongoDB version
2. The machine should be in network
3. Create 3 folder for storing 3 different databses for 3 instances
4. Open cmd prompt. Change folder to bin folder and specify following command
   C:\program files\....\bin>mongod –dbpath c:\mydata\testdb1 –port 7000 –replSet "rs1"
5. Do this to start 2 more instances ;
   a. dbpath - c:\mydata\testdb2   port -8181   replSet "rs1"
   b. dbpath - c:\mydata\testdb3   port -8000   replSet "rs1"
6. start one more instance of cmd. Change the folder to C:\program files\....server\3.6\bin
7. to start client
   a. mongo –port 7000
8. >rs.initiate()
9. Rs1.secondary>rs.config()
10. Rs1.primary>rs.status()
11. Rs1.primary>rs.add("as123:8000");
12. >rs.status()
13. Add some collection on primary server then the data should get sync on secondary server
14. If not happening use command
    a. rs.slaveOk() on secondary servers
15. to check the data open different clients and connect to secondary server and chek the collection