BIG DATA ANALYTICS LAB RECORD

Submitted by

ROHITH Y V
(1BM17CS080)

in partial fulfillment for the award of the degree of
BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING

Under the Guidance of
Pallavi G B
Associate Professor, BMSCE

B. M. S. COLLEGE OF ENGINEERING
(Autonomous Institution under VTU)
BENGALURU-560019
2020-2021

LAB 1

To Create

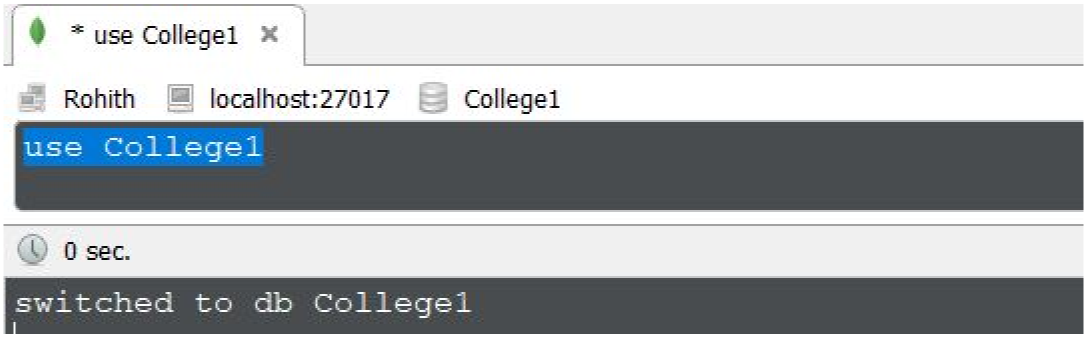
1. College Database

Collections

1. Student
2. Faculty
3. COE
4. Library
5. Admission
6. College Fest - Minimum 6 attributes (4 values with different set of attribute -value pair) - A set of 4 documents

Creating a College Database

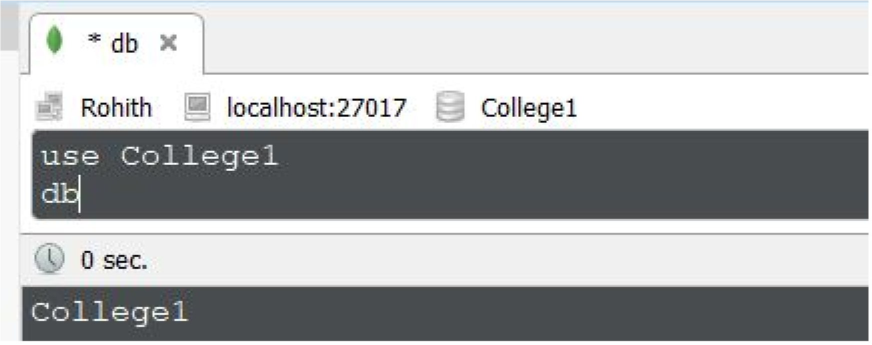
use College1



```
use College1
```

```
switched to db College1
```

db

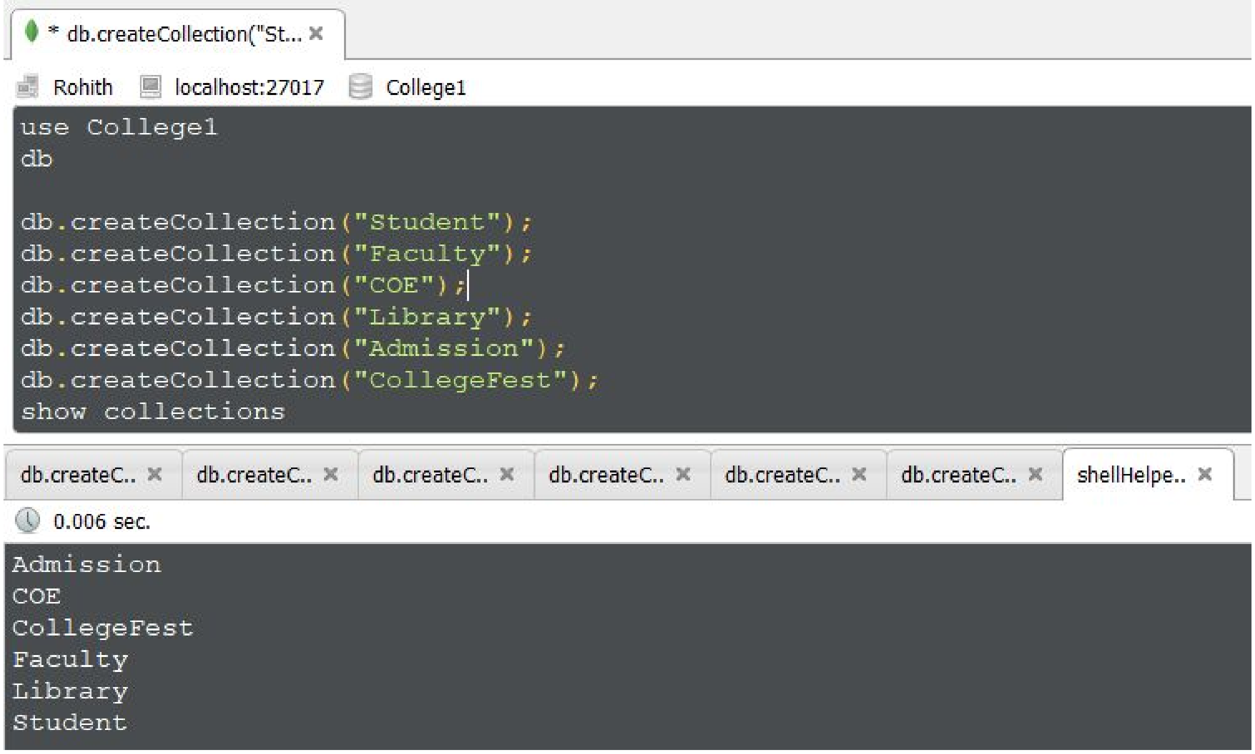


```
use College1
db
```

```
College1
```

db.createCollection("Student");
db.createCollection("Faculty");
db.createCollection("COE");

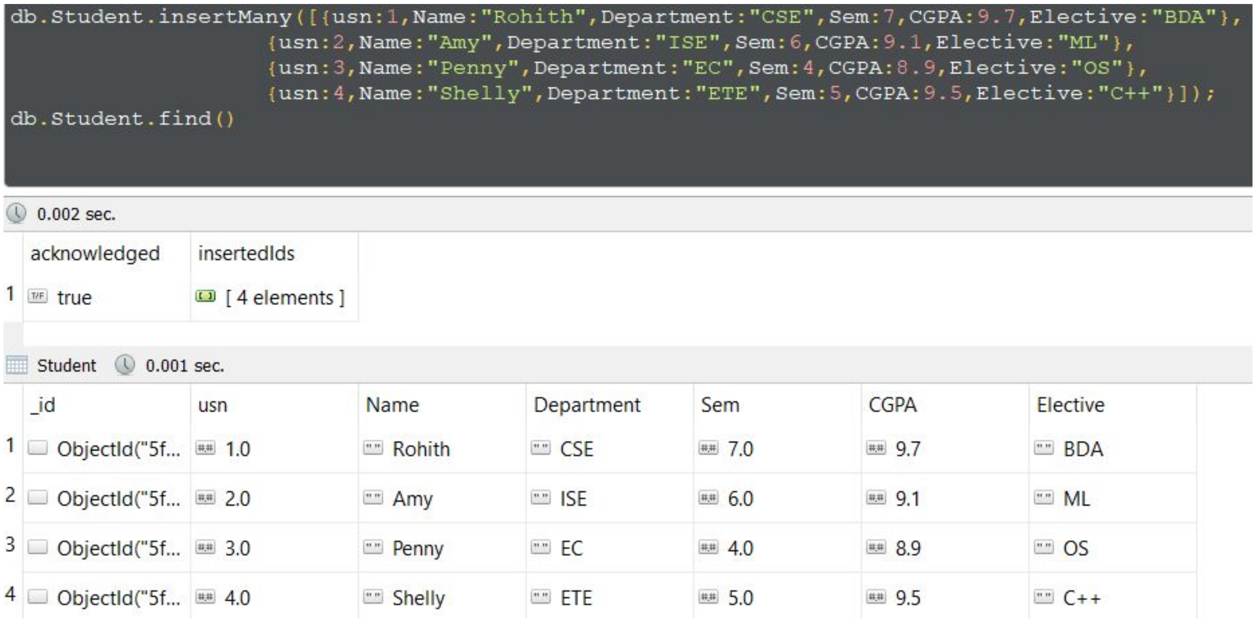
```
db.createCollection("Library");
db.createCollection("Admission");
db.createCollection("CollegeFest");
show collections
```



```
use College1
db

db.createCollection("Student");
db.createCollection("Faculty");
db.createCollection("COE");
db.createCollection("Library");
db.createCollection("Admission");
db.createCollection("CollegeFest");
show collections
```

```
Admission
COE
CollegeFest
Faculty
Library
Student
```

```
db.Student.insertMany([{usn:1,Name:"Rohith",Department:"CSE",Sem:7,CGPA:9.7,Elective:"BDA"},

{usn:2,Name:"Amy",Department:"ISE",Sem:6,CGPA:9.1,Elective:"ML"},

{usn:3,Name:"Penny",Department:"EC",Sem:4,CGPA:8.9,Elective:"OS"},

{usn:4,Name:"Shelly",Department:"ETE",Sem:5,CGPA:9.5,Elective:"C++"}]);
db.Student.find()
```



```
db.Student.insertMany([{usn:1,Name:"Rohith",Department:"CSE",Sem:7,CGPA:9.7,Elective:"BDA"},
                      {usn:2,Name:"Amy",Department:"ISE",Sem:6,CGPA:9.1,Elective:"ML"},
                      {usn:3,Name:"Penny",Department:"EC",Sem:4,CGPA:8.9,Elective:"OS"},
                      {usn:4,Name:"Shelly",Department:"ETE",Sem:5,CGPA:9.5,Elective:"C++"}]);
db.Student.find()
```

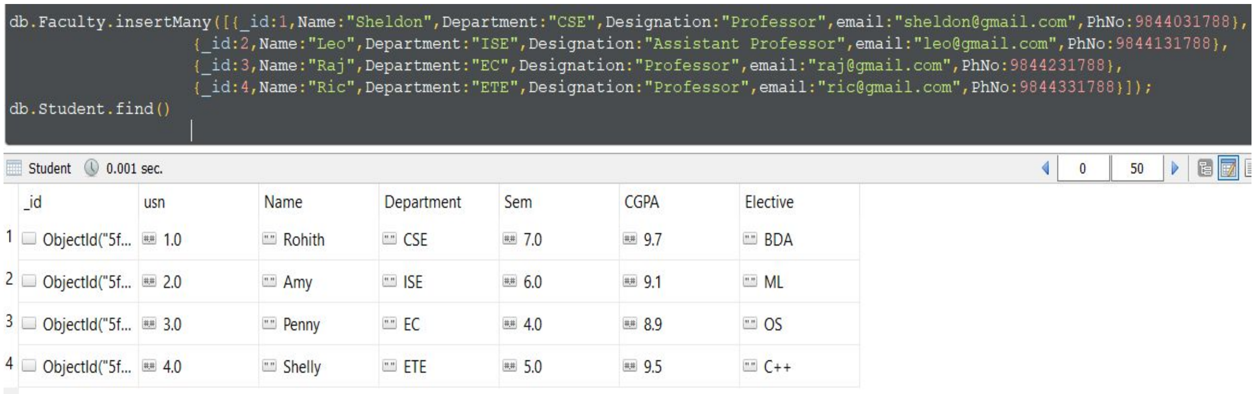
0.002 sec.

	acknowledged	insertedIds
1	true	[4 elements]

Student 0.001 sec.

	_id	usn	Name	Department	Sem	CGPA	Elective
1	ObjectId("5f...	1.0	Rohith	CSE	7.0	9.7	BDA
2	ObjectId("5f...	2.0	Amy	ISE	6.0	9.1	ML
3	ObjectId("5f...	3.0	Penny	EC	4.0	8.9	OS
4	ObjectId("5f...	4.0	Shelly	ETE	5.0	9.5	C++

```
db.Faculty.insertMany([{_id:1,Name:"Sheldon",Department:"CSE",Designation:"
Professor",email:"sheldon@gmail.com",PhNo:9844031788},
                 {_id:2,Name:"Leo",Department:"ISE",Designation:"Assistant
Professor",email:"leo@gmail.com",PhNo:9844131788},

{_id:3,Name:"Raj",Department:"EC",Designation:"Professor",email:"raj@gmail.c
om",PhNo:9844231788},

{_id:4,Name:"Ric",Department:"ETE",Designation:"Professor",email:"ric@gmail.
com",PhNo:9844331788}]);
db.Student.find()
```

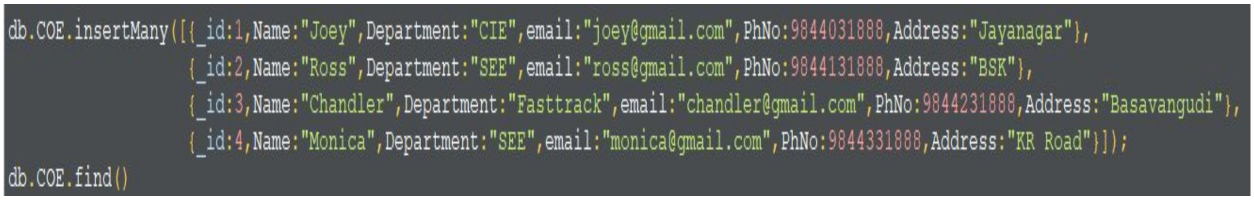


```
db.Faculty.insertMany([{_id:1,Name:"Sheldon",Department:"CSE",Designation:"Professor",email:"sheldon@gmail.com",PhNo:9844031788},
                      {_id:2,Name:"Leo",Department:"ISE",Designation:"Assistant Professor",email:"leo@gmail.com",PhNo:9844131788},
                      {_id:3,Name:"Raj",Department:"EC",Designation:"Professor",email:"raj@gmail.com",PhNo:9844231788},
                      {_id:4,Name:"Ric",Department:"ETE",Designation:"Professor",email:"ric@gmail.com",PhNo:9844331788}]);
db.Student.find()
```

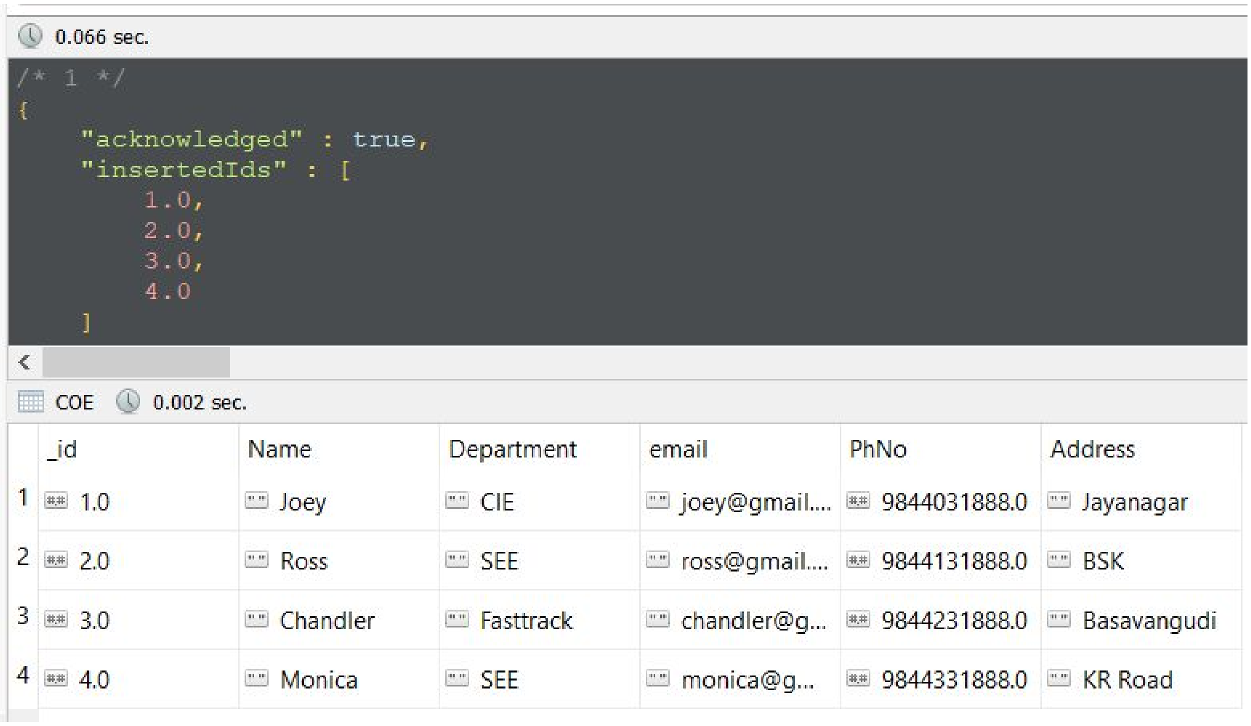
Student 0.001 sec.

	_id	usn	Name	Department	Sem	CGPA	Elective
1	ObjectId("5f...	1.0	Rohith	CSE	7.0	9.7	BDA
2	ObjectId("5f...	2.0	Amy	ISE	6.0	9.1	ML
3	ObjectId("5f...	3.0	Penny	EC	4.0	8.9	OS
4	ObjectId("5f...	4.0	Shelly	ETE	5.0	9.5	C++

```
db.COE.insertMany([{_id:1,Name:"Joey",Department:"CIE",email:"joey@gmail.com",PhNo:9844031888,Address:"Jayanagar"},

{_id:2,Name:"Ross",Department:"SEE",email:"ross@gmail.com",PhNo:9844131888,Address:"BSK"},

{_id:3,Name:"Chandler",Department:"Fasttrack",email:"chandler@gmail.com",PhNo:9844231888,Address:"Basavangudi"},

{_id:4,Name:"Monica",Department:"SEE",email:"monica@gmail.com",PhNo:9844331888,Address:"KR Road"}]);
db.COE.find()
```



```
db.COE.insertMany([{_id:1,Name:"Joey",Department:"CIE",email:"joey@gmail.com",PhNo:9844031888,Address:"Jayanagar"},
                   {_id:2,Name:"Ross",Department:"SEE",email:"ross@gmail.com",PhNo:9844131888,Address:"BSK"},
                   {_id:3,Name:"Chandler",Department:"Fasttrack",email:"chandler@gmail.com",PhNo:9844231888,Address:"Basavangudi"},
                   {_id:4,Name:"Monica",Department:"SEE",email:"monica@gmail.com",PhNo:9844331888,Address:"KR Road"}]);
db.COE.find()
```



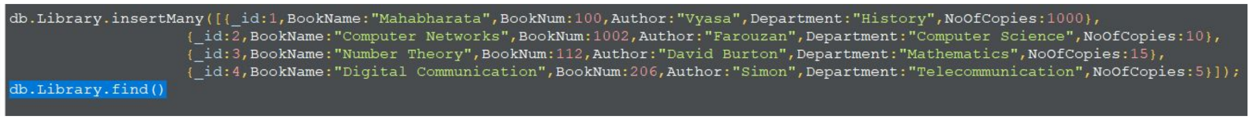
0.066 sec.

```
/* 1 */
{
    "acknowledged" : true,
    "insertedIds" : [
        1.0,
        2.0,
        3.0,
        4.0
    ]
```

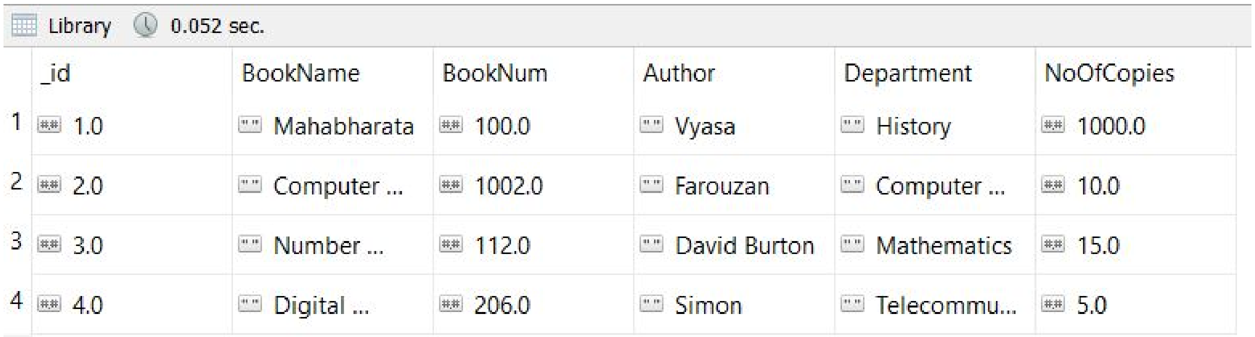
COE 0.002 sec.

	_id	Name	Department	email	PhNo	Address
1	1.0	Joey	CIE	joey@gmail....	9844031888.0	Jayanagar
2	2.0	Ross	SEE	ross@gmail....	9844131888.0	BSK
3	3.0	Chandler	Fasttrack	chandler@g...	9844231888.0	Basavangudi
4	4.0	Monica	SEE	monica@g...	9844331888.0	KR Road

```
db.Library.insertMany([{_id:1,BookName:"Mahabharata",BookNum:100,Author:"Vyasa",Department:"History",NoOfCopies:1000},
                {_id:2,BookName:"Computer Networks",BookNum:1002,Author:"Farouzan",Department:"Computer Science",NoOfCopies:10},
                {_id:3,BookName:"Number Theory",BookNum:112,Author:"David Burton",Department:"Mathematics",NoOfCopies:15},
                {_id:4,BookName:"Digital Communication",BookNum:206,Author:"Simon",Department:"Telecommunication",NoOfCopies:5}]);
db.Library.find()
```



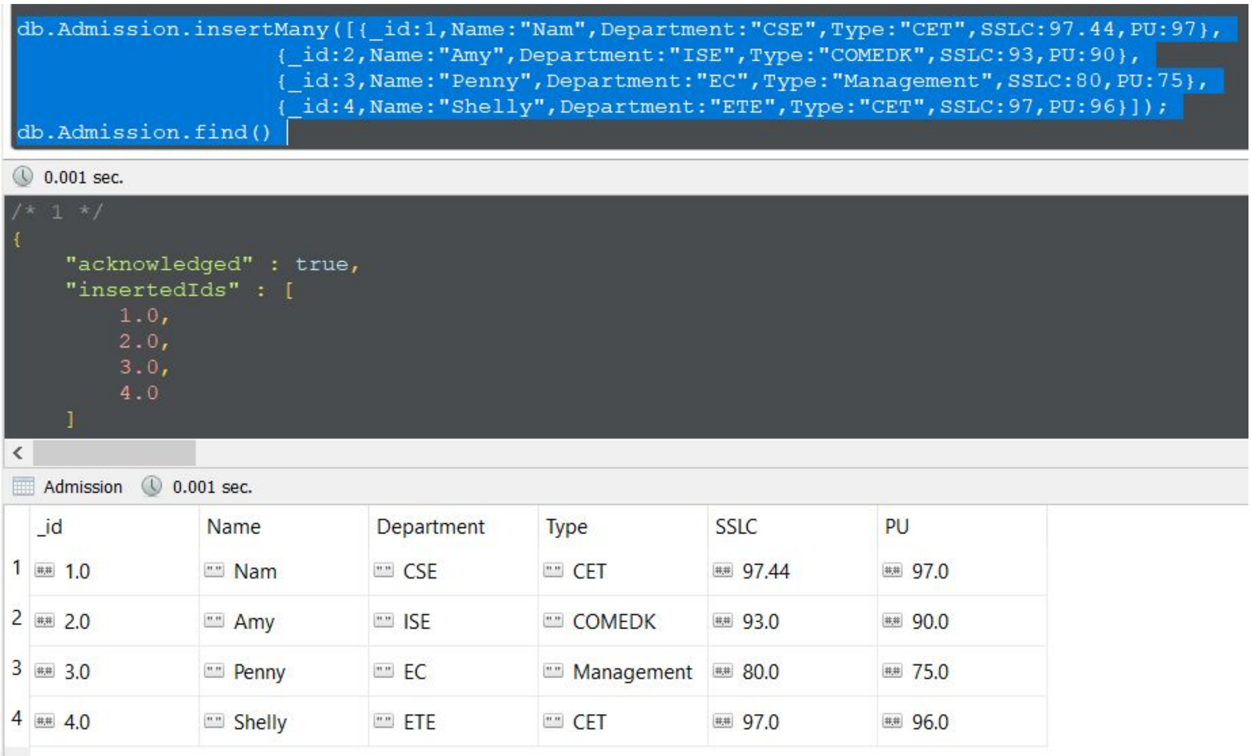
```
db.Library.insertMany([{_id:1,BookName:"Mahabharata",BookNum:100,Author:"Vyasa",Department:"History",NoOfCopies:1000},
                     {_id:2,BookName:"Computer Networks",BookNum:1002,Author:"Farouzan",Department:"Computer Science",NoOfCopies:10},
                     {_id:3,BookName:"Number Theory",BookNum:112,Author:"David Burton",Department:"Mathematics",NoOfCopies:15},
                     {_id:4,BookName:"Digital Communication",BookNum:206,Author:"Simon",Department:"Telecommunication",NoOfCopies:5}]);
db.Library.find()
```



Library 0.052 sec.

	_id	BookName	BookNum	Author	Department	NoOfCopies
1	1.0	Mahabharata	100.0	Vyasa	History	1000.0
2	2.0	Computer ...	1002.0	Farouzan	Computer ...	10.0
3	3.0	Number ...	112.0	David Burton	Mathematics	15.0
4	4.0	Digital ...	206.0	Simon	Telecommu...	5.0

```
db.Admission.insertMany([{_id:1,Name:"Nam",Department:"CSE",Type:"CET",SSLC:97.44,PU:97},

{_id:2,Name:"Amy",Department:"ISE",Type:"COMEDK",SSLC:93,PU:90},

{_id:3,Name:"Penny",Department:"EC",Type:"Management",SSLC:80,PU:75},

{_id:4,Name:"Shelly",Department:"ETE",Type:"CET",SSLC:97,PU:96}]);
db.Admission.find()
```



```
db.Admission.insertMany([{_id:1,Name:"Nam",Department:"CSE",Type:"CET",SSLC:97.44,PU:97},
                   {_id:2,Name:"Amy",Department:"ISE",Type:"COMEDK",SSLC:93,PU:90},
                   {_id:3,Name:"Penny",Department:"EC",Type:"Management",SSLC:80,PU:75},
                   {_id:4,Name:"Shelly",Department:"ETE",Type:"CET",SSLC:97,PU:96}]);
db.Admission.find()
```

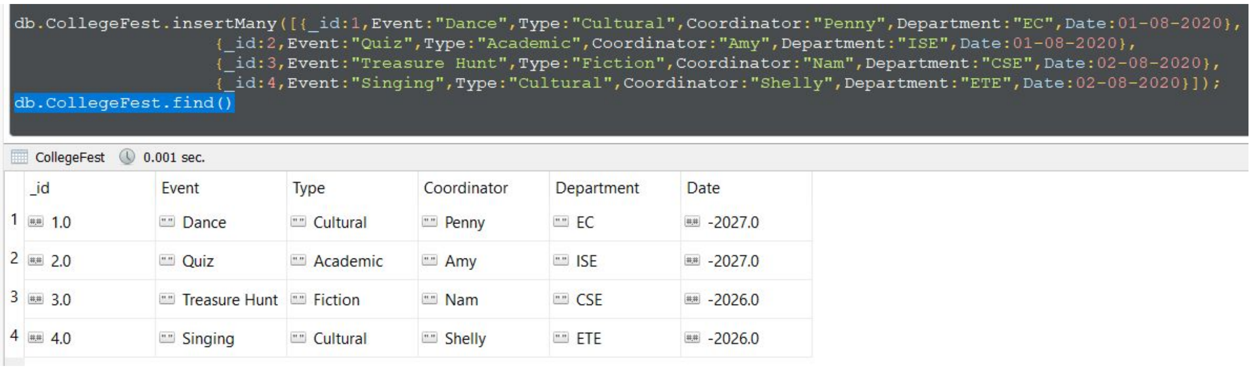
0.001 sec.

```
/* 1 */
{
    "acknowledged" : true,
    "insertedIds" : [
        1.0,
        2.0,
        3.0,
        4.0
    ]
```

Admission 0.001 sec.

	_id	Name	Department	Type	SSLC	PU
1	1.0	Nam	CSE	CET	97.44	97.0
2	2.0	Amy	ISE	COMEDK	93.0	90.0
3	3.0	Penny	EC	Management	80.0	75.0
4	4.0	Shelly	ETE	CET	97.0	96.0

```
db.CollegeFest.insertMany([{_id:1,Event:"Dance",Type:"Cultural",Coordinator:"P
enny",Department:"EC",Date:01-08-2020},

{_id:2,Event:"Quiz",Type:"Academic",Coordinator:"Amy",Department:"ISE",Dat
e:01-08-2020},
                  {_id:3,Event:"Treasure
Hunt",Type:"Fiction",Coordinator:"Nam",Department:"CSE",Date:02-08-2020},

{_id:4,Event:"Singing",Type:"Cultural",Coordinator:"Shelly",Department:"ETE",
Date:02-08-2020}]);
db.CollegeFest.find()
```



```
db.CollegeFest.insertMany([{_id:1,Event:"Dance",Type:"Cultural",Coordinator:"Penny",Department:"EC",Date:01-08-2020},
                    {_id:2,Event:"Quiz",Type:"Academic",Coordinator:"Amy",Department:"ISE",Date:01-08-2020},
                    {_id:3,Event:"Treasure Hunt",Type:"Fiction",Coordinator:"Nam",Department:"CSE",Date:02-08-2020},
                    {_id:4,Event:"Singing",Type:"Cultural",Coordinator:"Shelly",Department:"ETE",Date:02-08-2020}]);
db.CollegeFest.find()
```

CollegeFest 0.001 sec.

	_id	Event	Type	Coordinator	Department	Date
1	1.0	Dance	Cultural	Penny	EC	-2027.0
2	2.0	Quiz	Academic	Amy	ISE	-2027.0
3	3.0	Treasure Hunt	Fiction	Nam	CSE	-2026.0
4	4.0	Singing	Cultural	Shelly	ETE	-2026.0

LAB 2

Create **Company database** with following collections

1. Employee

2. Department

Perform following MongoDB Operations

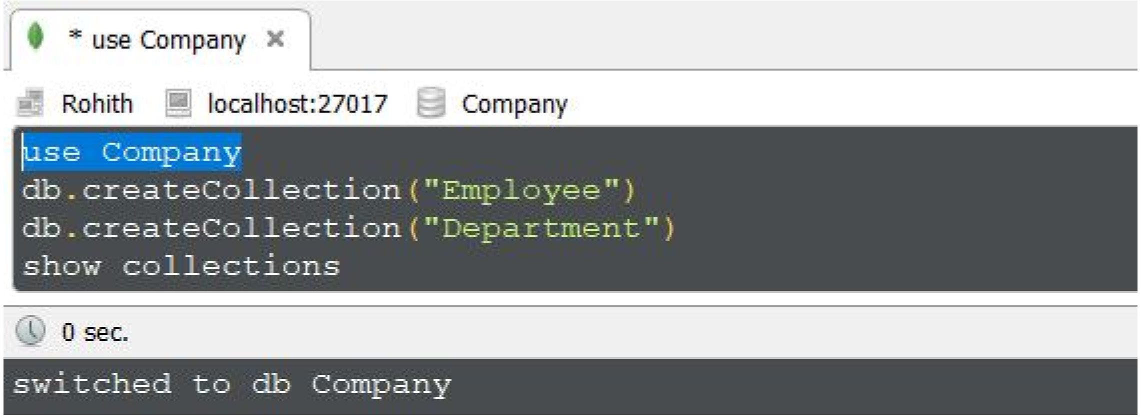
1. Insert at least five documents in each collection using all three methods of insertion.{ Show update method using upsert set to true and set to false}
2. Update &quot;Employee&quot; collection to add a new field to an existing document .
3. Remove a field from an existing document
4. Select all documents from both collections.
5. Select only employee name and department number whose department number falls between
1001 to 1005
6. Select employee documents whose name begins with &#39;A&#39;
7. Select employee document s whose age is greater than 30

Create **Company database** with following collections

1. Employee

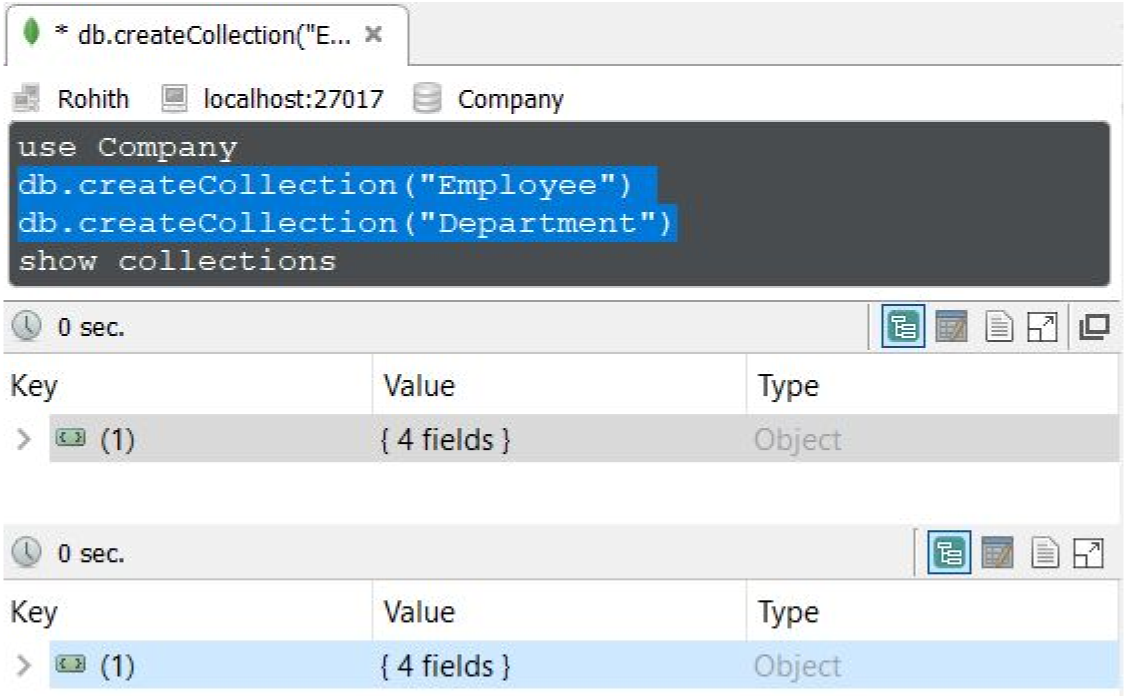
2. Department

use Company
db.createCollection("Employee")
db.createCollection("Department")
show collections



```
use Company
db.createCollection("Employee")
db.createCollection("Department")
show collections
```

```
switched to db Company
```



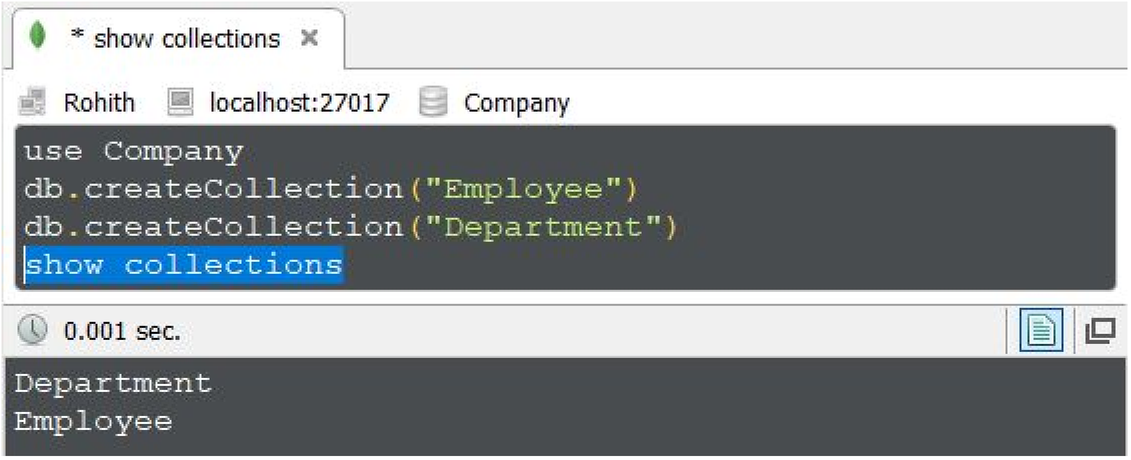
```
use Company
db.createCollection("Employee")
db.createCollection("Department")
show collections
```

0 sec.

Key	Value	Type
(1)	{ 4 fields }	Object

0 sec.

Key	Value	Type
(1)	{ 4 fields }	Object



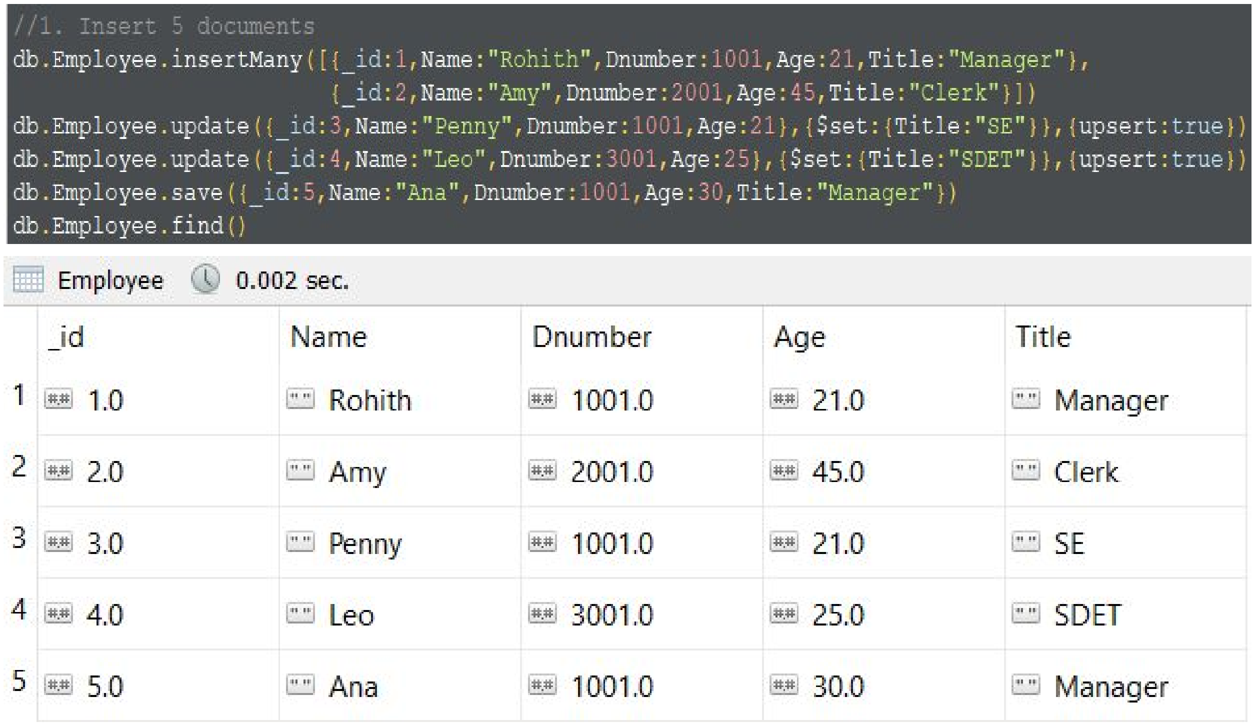
```
use Company
db.createCollection("Employee")
db.createCollection("Department")
show collections
```

0.001 sec.

```
Department
Employee
```

1. Insert at least five documents in each collection using all three methods of insertion.{ Show update method using upsert set to true and set to false}

```
//1. Insert 5 documents
db.Employee.insertMany([{_id:1,Name:"Rohith",Dnumber:1001,Age:21,Title:"Manager"},
                        {_id:2,Name:"Amy",Dnumber:2001,Age:45,Title:"Clerk"}])
db.Employee.update({_id:3,Name:"Penny",Dnumber:1001,Age:21},{$set:{Title:"SE"}},{upsert:true})
db.Employee.update({_id:4,Name:"Leo",Dnumber:3001,Age:25},{$set:{Title:"SDET"}},{upsert:true})
db.Employee.save({_id:5,Name:"Ana",Dnumber:1001,Age:30,Title:"Manager"})
db.Employee.find()
```

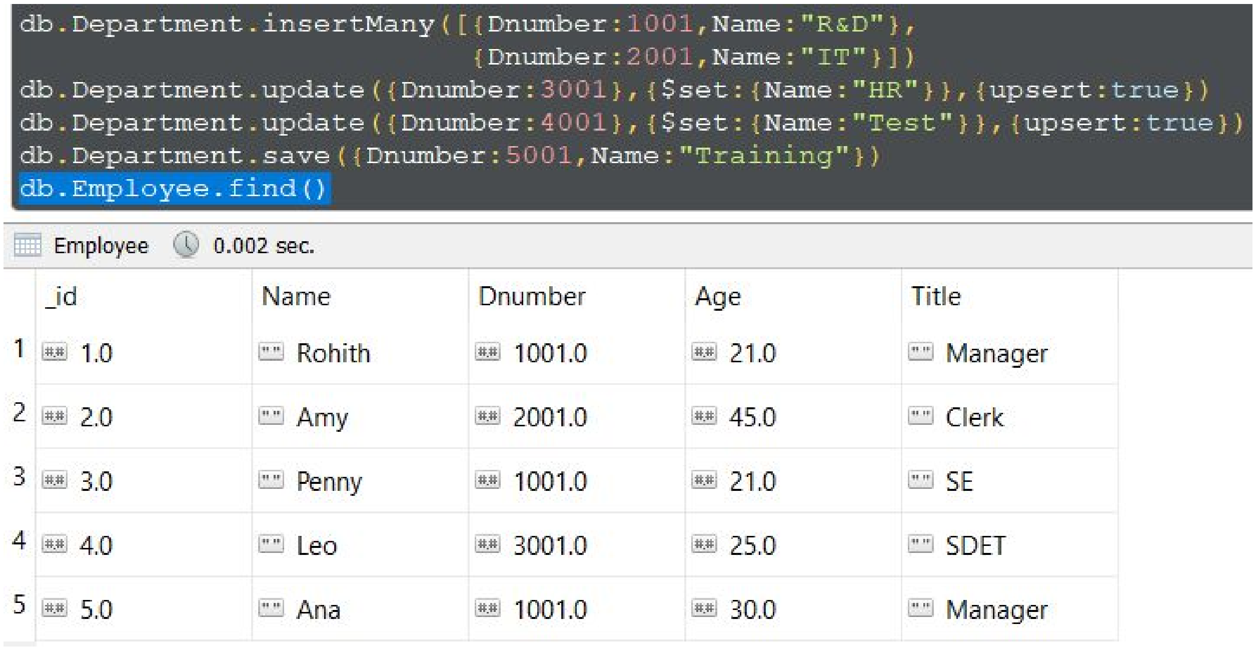


```
//1. Insert 5 documents
db.Employee.insertMany([{_id:1,Name:"Rohith",Dnumber:1001,Age:21,Title:"Manager"},
                        {_id:2,Name:"Amy",Dnumber:2001,Age:45,Title:"Clerk"}])
db.Employee.update({_id:3,Name:"Penny",Dnumber:1001,Age:21},{$set:{Title:"SE"}},{upsert:true})
db.Employee.update({_id:4,Name:"Leo",Dnumber:3001,Age:25},{$set:{Title:"SDET"}},{upsert:true})
db.Employee.save({_id:5,Name:"Ana",Dnumber:1001,Age:30,Title:"Manager"})
db.Employee.find()
```

Employee 0.002 sec.

	_id	Name	Dnumber	Age	Title
1	1.0	Rohith	1001.0	21.0	Manager
2	2.0	Amy	2001.0	45.0	Clerk
3	3.0	Penny	1001.0	21.0	SE
4	4.0	Leo	3001.0	25.0	SDET
5	5.0	Ana	1001.0	30.0	Manager

```
db.Department.insertMany([{Dnumber:1001,Name:"R&D"},{Dnumber:2001,Name:"IT"}])
db.Department.update({Dnumber:3001},{$set:{Name:"HR"}},{upsert:true})
db.Department.update({Dnumber:4001},{$set:{Name:"Test"}},{upsert:true})
db.Department.save({Dnumber:5001,Name:"Training"})
db.Employee.find()
```



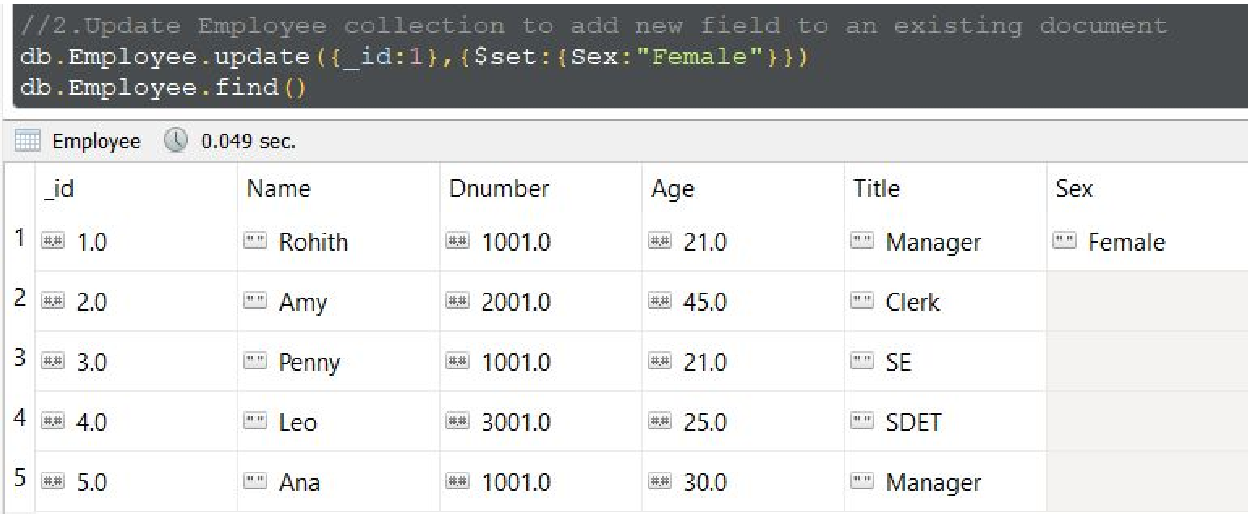
```
db.Department.insertMany([{Dnumber:1001,Name:"R&D"},
                          {Dnumber:2001,Name:"IT"}])
db.Department.update({Dnumber:3001},{$set:{Name:"HR"}},{upsert:true})
db.Department.update({Dnumber:4001},{$set:{Name:"Test"}},{upsert:true})
db.Department.save({Dnumber:5001,Name:"Training"})
db.Employee.find()
```

Employee 0.002 sec.

	_id	Name	Dnumber	Age	Title
1	1.0	Rohith	1001.0	21.0	Manager
2	2.0	Amy	2001.0	45.0	Clerk
3	3.0	Penny	1001.0	21.0	SE
4	4.0	Leo	3001.0	25.0	SDET
5	5.0	Ana	1001.0	30.0	Manager

2. Update &quot;Employee&quot; collection to add a new field to an existing document .

```
//2.Update Employee collection to add new field to an existing document
db.Employee.update({_id:1},{$set:{Sex:"Female"}})
```



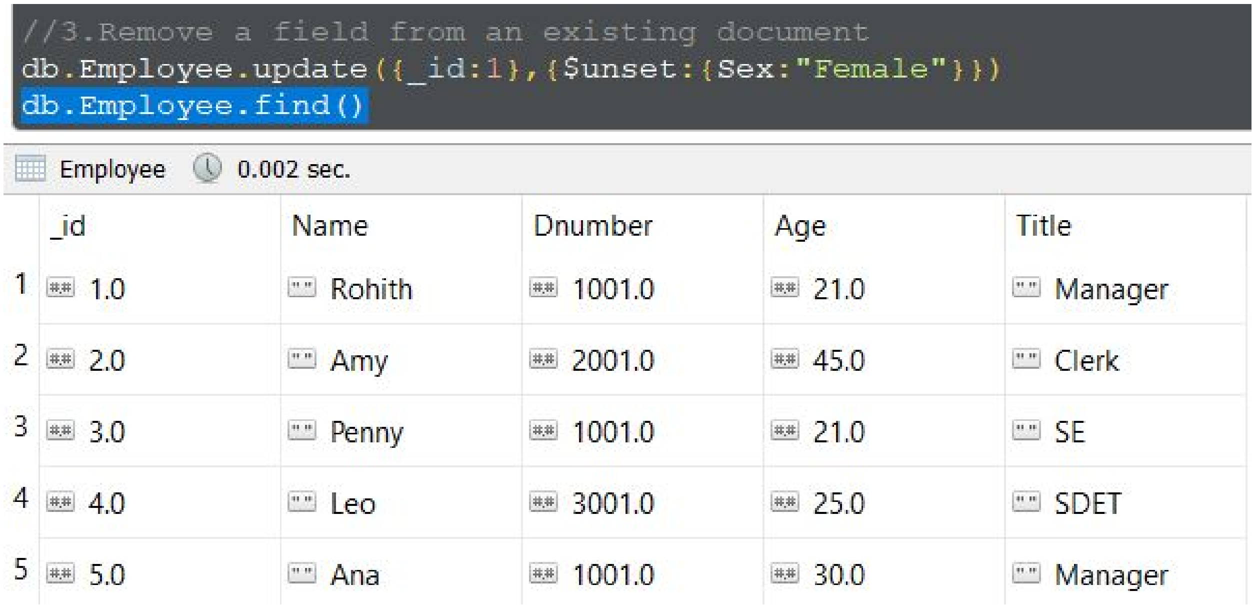
```
//2.Update Employee collection to add new field to an existing document
db.Employee.update({_id:1},{$set:{Sex:"Female"}})
db.Employee.find()
```

Employee 0.049 sec.

	_id	Name	Dnumber	Age	Title	Sex
1	1.0	Rohith	1001.0	21.0	Manager	Female
2	2.0	Amy	2001.0	45.0	Clerk	
3	3.0	Penny	1001.0	21.0	SE	
4	4.0	Leo	3001.0	25.0	SDET	
5	5.0	Ana	1001.0	30.0	Manager	

3. Remove a field from an existing document.

//3.Remove a field from an existing document
db.Employee.update({_id:1},{$unset:{Sex:"Female"}})
db.Employee.find()



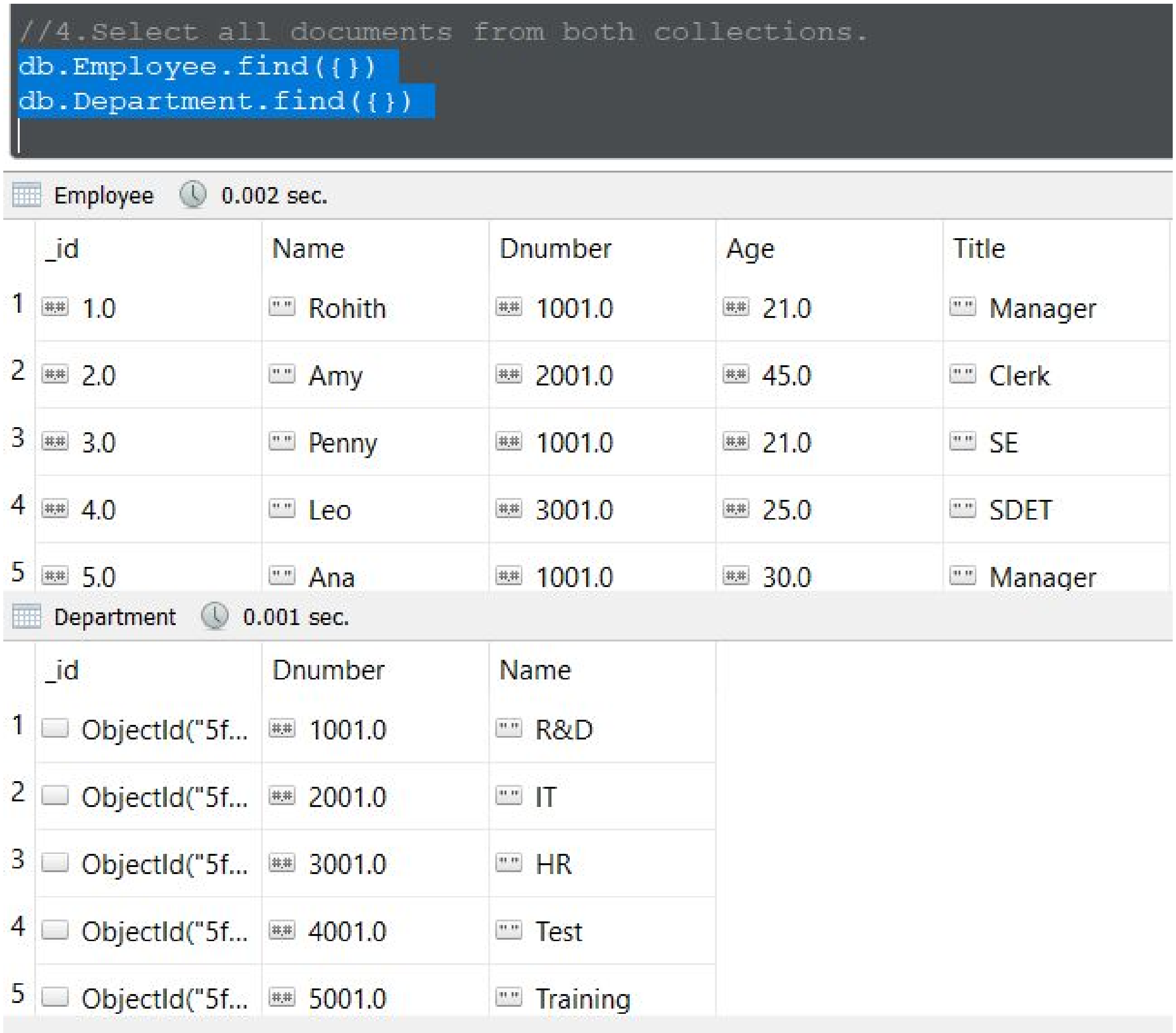
```
//3.Remove a field from an existing document
db.Employee.update({_id:1},{$unset:{Sex:"Female"}})
db.Employee.find()
```

Employee 0.002 sec.

	_id	Name	Dnumber	Age	Title
1	1.0	Rohith	1001.0	21.0	Manager
2	2.0	Amy	2001.0	45.0	Clerk
3	3.0	Penny	1001.0	21.0	SE
4	4.0	Leo	3001.0	25.0	SDET
5	5.0	Ana	1001.0	30.0	Manager

4. Select all documents from both collections.

```
//4.Select all documents from both collections.
db.Employee.find({})
db.Department.find({})
```



```
//4.Select all documents from both collections.
db.Employee.find({})
db.Department.find({})
```

Employee 0.002 sec.

	_id	Name	Dnumber	Age	Title
1	1.0	Rohith	1001.0	21.0	Manager
2	2.0	Amy	2001.0	45.0	Clerk
3	3.0	Penny	1001.0	21.0	SE
4	4.0	Leo	3001.0	25.0	SDET
5	5.0	Ana	1001.0	30.0	Manager

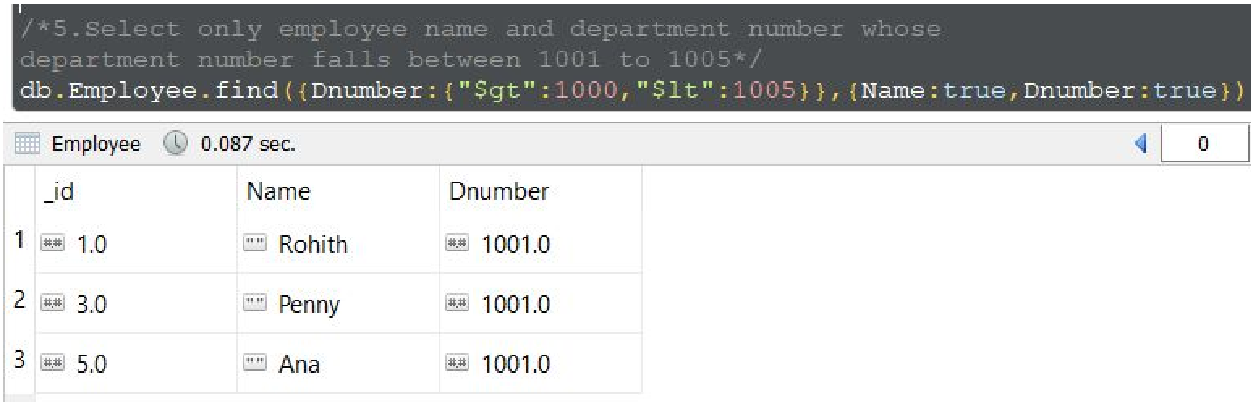
Department 0.001 sec.

	_id	Dnumber	Name
1	ObjectId("5f...	1001.0	R&D
2	ObjectId("5f...	2001.0	IT
3	ObjectId("5f...	3001.0	HR
4	ObjectId("5f...	4001.0	Test
5	ObjectId("5f...	5001.0	Training

5. Select only employee name and department number whose department number falls between 1001 to 1005

/*5.Select only employee name and department number whose department number falls between 1001 to 1005*/

db.Employee.find({Dnumber:{"$gt":1000,"$lt":1005}},{Name:true,Dnumber:true })



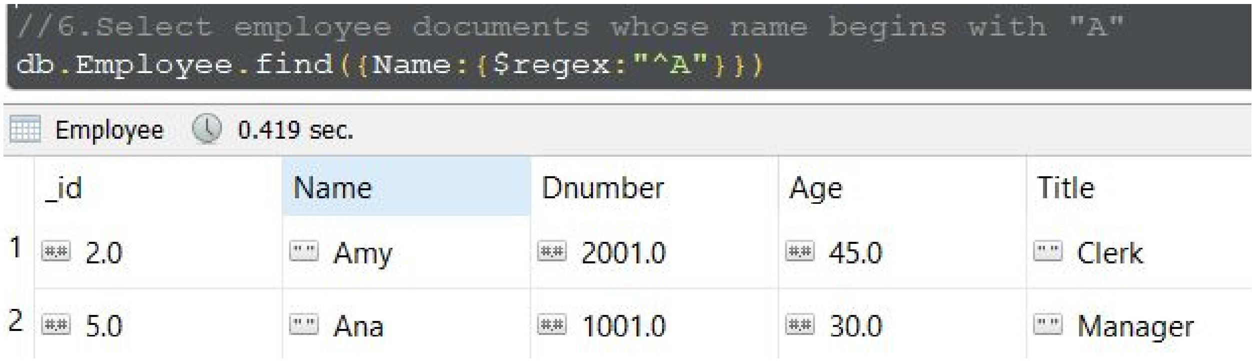
```
/*5.Select only employee name and department number whose
department number falls between 1001 to 1005*/
db.Employee.find({Dnumber:{"$gt":1000,"$lt":1005}},{Name:true,Dnumber:true})
```

	_id	Name	Dnumber
1	1.0	Rohith	1001.0
2	3.0	Penny	1001.0
3	5.0	Ana	1001.0

6. Select employee documents whose name begins with &#39;A&#39;

//6.Select employee documents whose name begins with "A"

db.Employee.find({Name:{$regex:"^A"}})

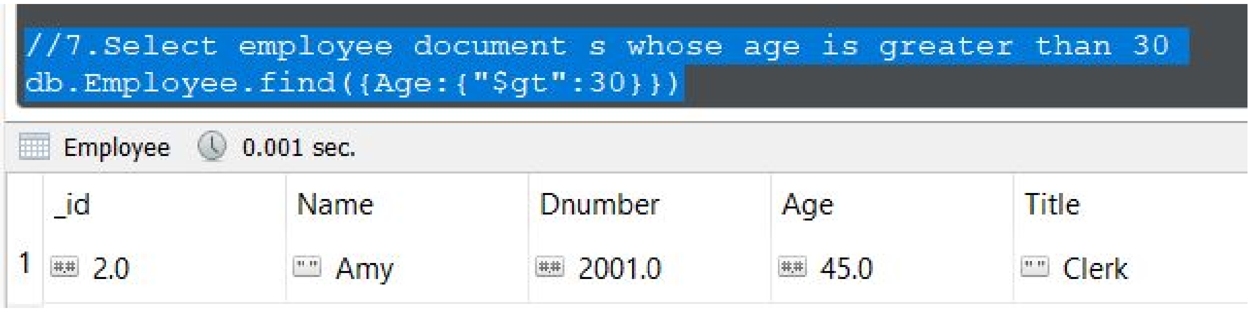


```
//6.Select employee documents whose name begins with "A"
db.Employee.find({Name:{$regex:"^A"}})
```

	_id	Name	Dnumber	Age	Title
1	2.0	Amy	2001.0	45.0	Clerk
2	5.0	Ana	1001.0	30.0	Manager

7. Select employee document s whose age is greater than 30

```
//7.Select employee document s whose age is greater than 30
db.Employee.find({Age:{"$gt":30}})
```



```
//7.Select employee document s whose age is greater than 30
db.Employee.find({Age:{"$gt":30}})
```

Employee 0.001 sec.

	_id	Name	Dnumber	Age	Title
1	2.0	Amy	2001.0	45.0	Clerk

LAB - 3

PART 1

Perform the following DB operations using MongoDB.

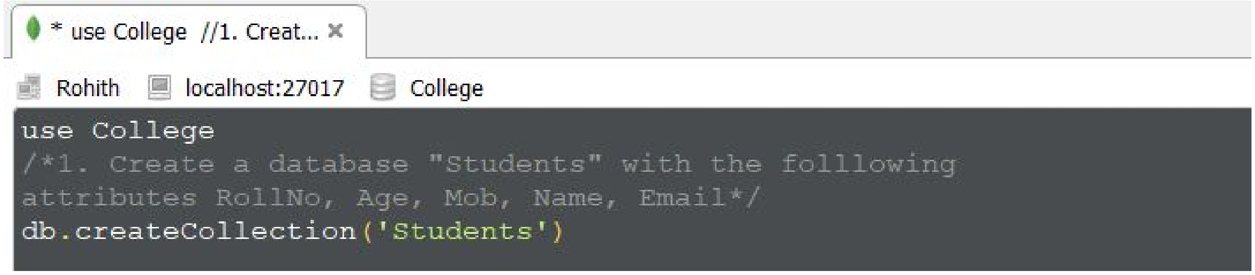
1. Create a database “Students” with the following attributes RollNo, Age, ContactNo, Email.
2. Insert appropriate values
3. Write a query to update Email of a student with RollNo 10.
4. Replace the student name from “ABC” to “FEM” of RollNo 11.
5. Export the created table into local file system
6. Drop the table
7. Import a given csv dataset from the local file system into mongodb collection.

Perform the following DB operations using MongoDB.

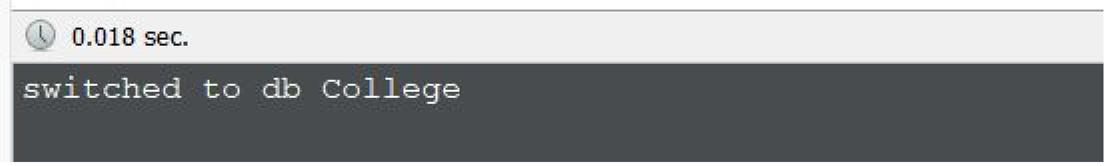
1. Create a database “Students” with the following attributes RollNo, Age, Mob, Email.

use College

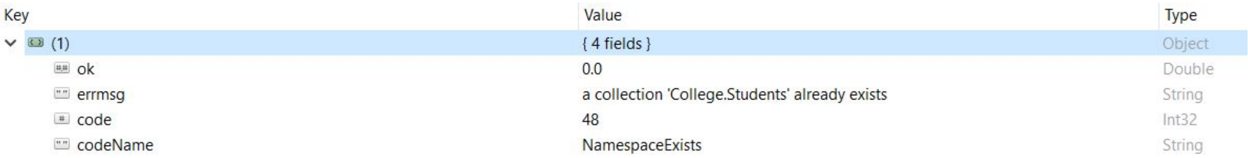
//1. Create a database "Students" with the following attributes RollNo, Age, Mob, Name, Email
db.createCollection('Students')



```
use College
/*1. Create a database "Students" with the folllowing
attributes RollNo, Age, Mob, Name, Email*/
db.createCollection('Students')
```



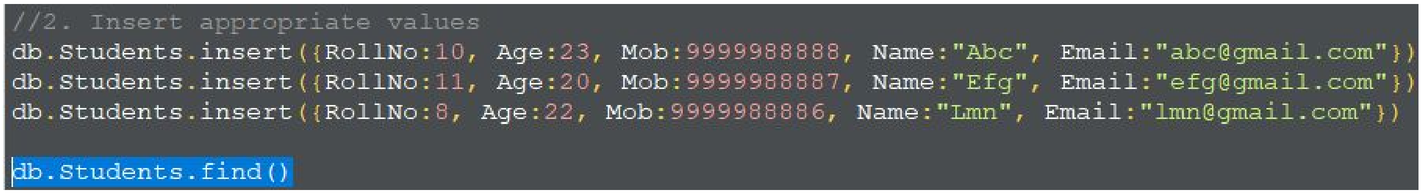
```
switched to db College
```



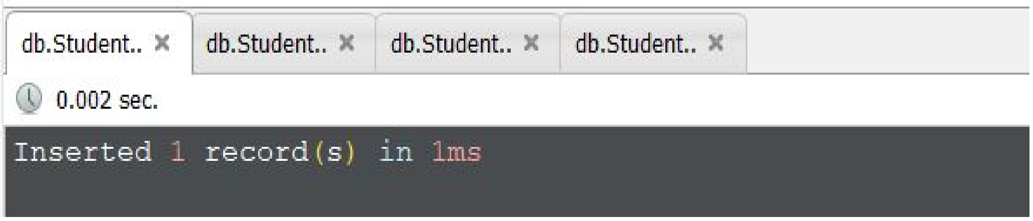
Key	Value	Type
(1)	{ 4 fields }	Object
ok	0.0	Double
errmsg	a collection 'College.Students' already exists	String
code	48	Int32
codeName	NamespaceExists	String

2. Insert appropriate values

```
//2. Insert appropriate values
db.Students.insert({RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})
```

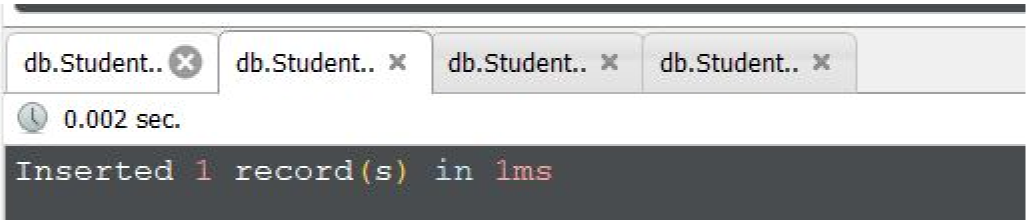


```
//2. Insert appropriate values
db.Students.insert({RollNo:10, Age:23, Mob:9999988888, Name:"Abc", Email:"abc@gmail.com"})
db.Students.insert({RollNo:11, Age:20, Mob:9999988887, Name:"Efg", Email:"efg@gmail.com"})
db.Students.insert({RollNo:8, Age:22, Mob:9999988886, Name:"Lmn", Email:"lmn@gmail.com"})

db.Students.find()
```



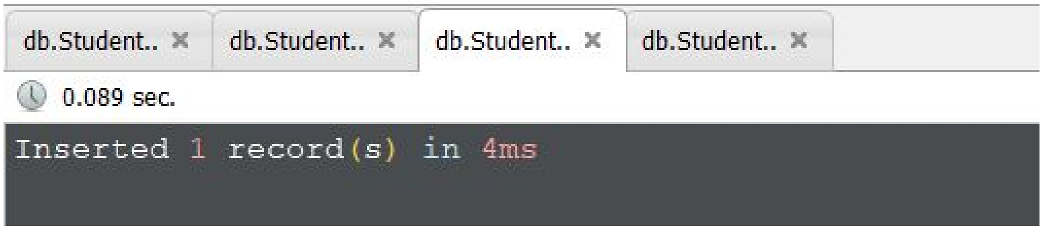
0.002 sec.

```
Inserted 1 record(s) in 1ms
```



0.002 sec.

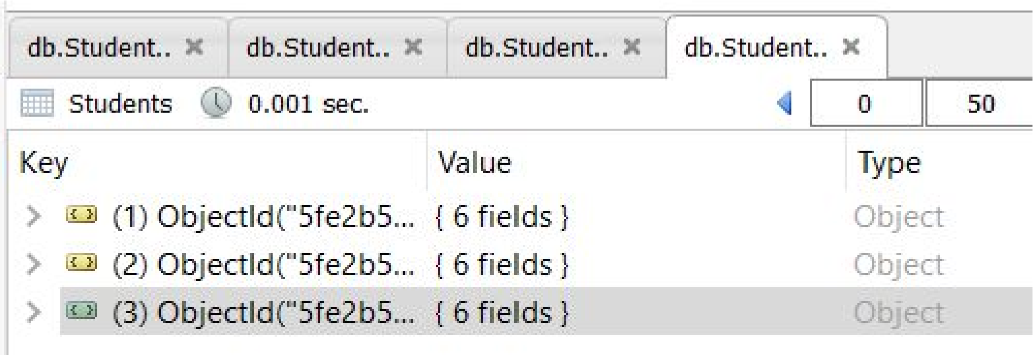
```
Inserted 1 record(s) in 1ms
```



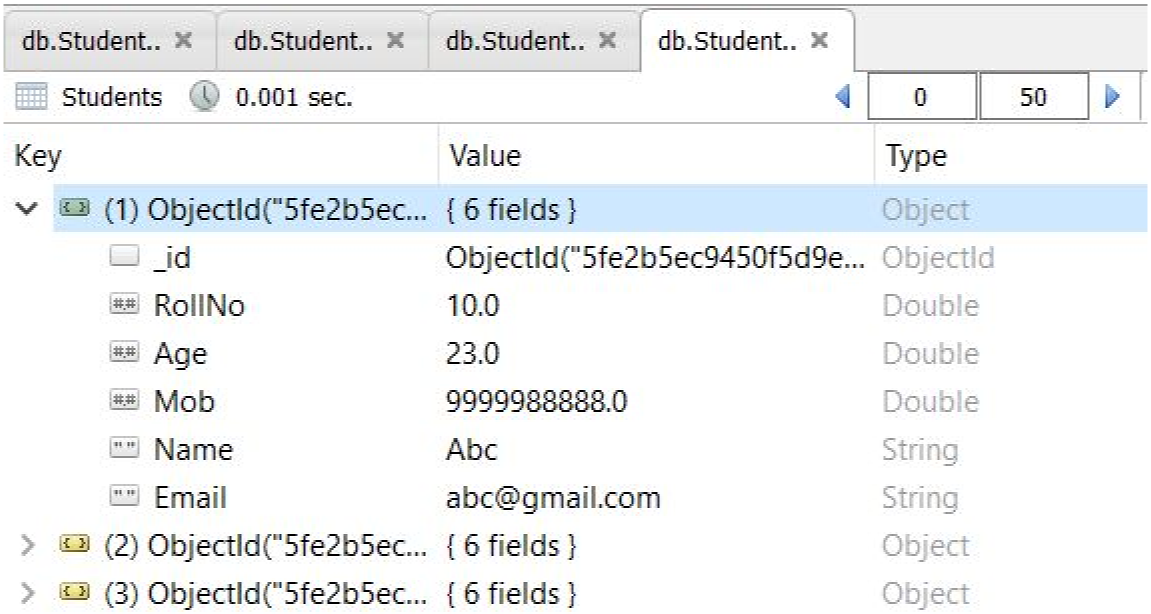
0.089 sec.

```
Inserted 1 record(s) in 4ms
```

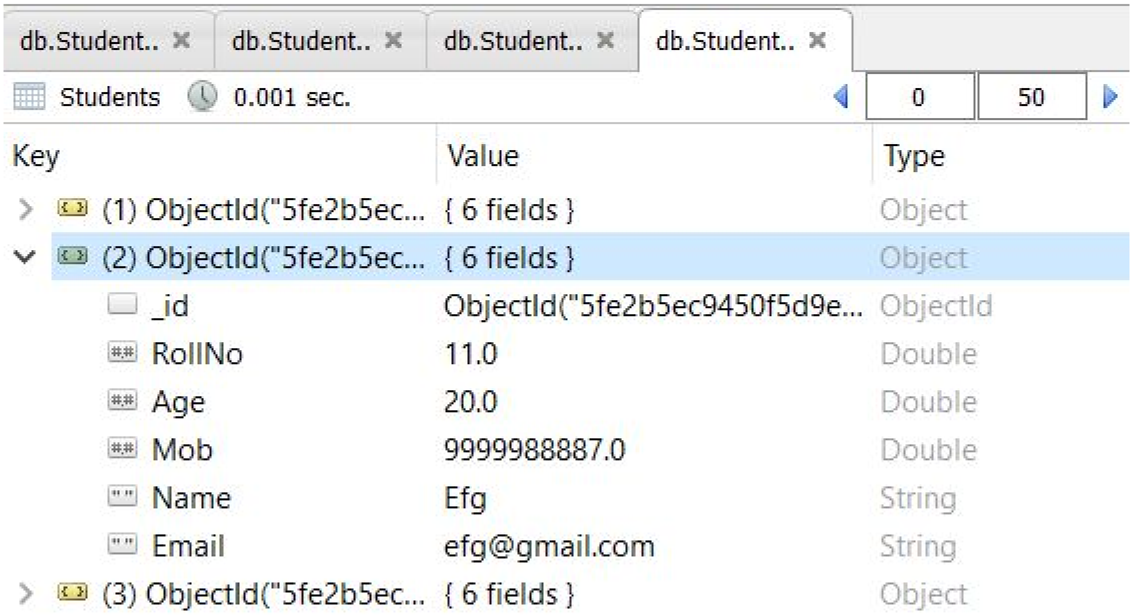
db.Students.find()



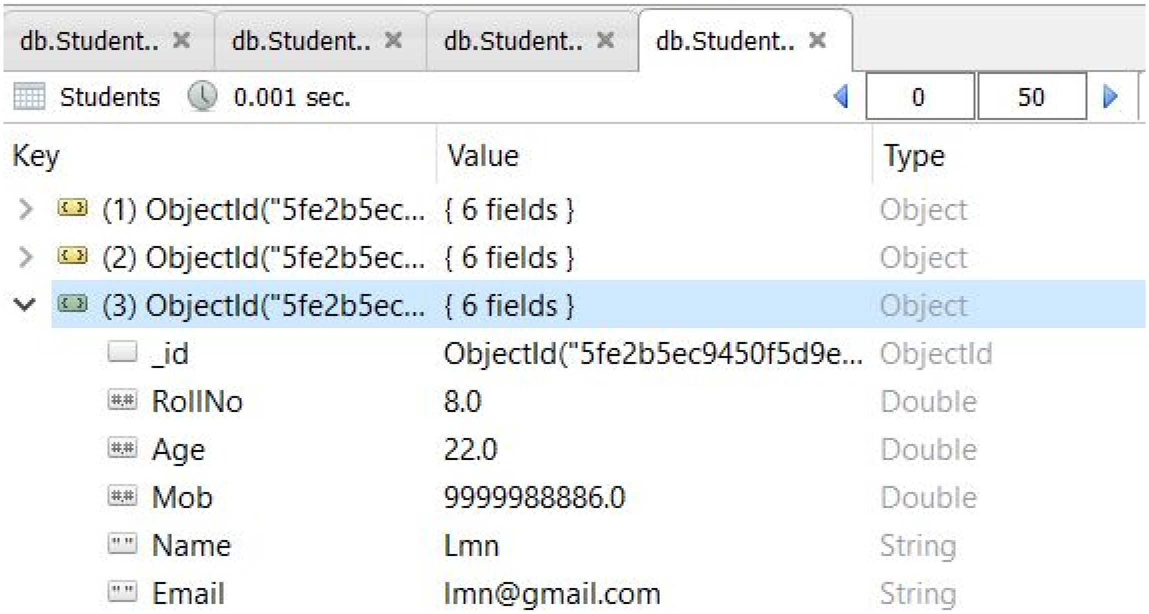
Key	Value	Type
(1) ObjectId("5fe2b5...	{ 6 fields }	Object
(2) ObjectId("5fe2b5...	{ 6 fields }	Object
(3) ObjectId("5fe2b5...	{ 6 fields }	Object



Key	Value	Type
(1) ObjectId("5fe2b5ec...	{ 6 fields }	Object
_id	ObjectId("5fe2b5ec9450f5d9e...	ObjectId
RollNo	10.0	Double
Age	23.0	Double
Mob	9999988888.0	Double
Name	Abc	String
Email	abc@gmail.com	String
(2) ObjectId("5fe2b5ec...	{ 6 fields }	Object
(3) ObjectId("5fe2b5ec...	{ 6 fields }	Object



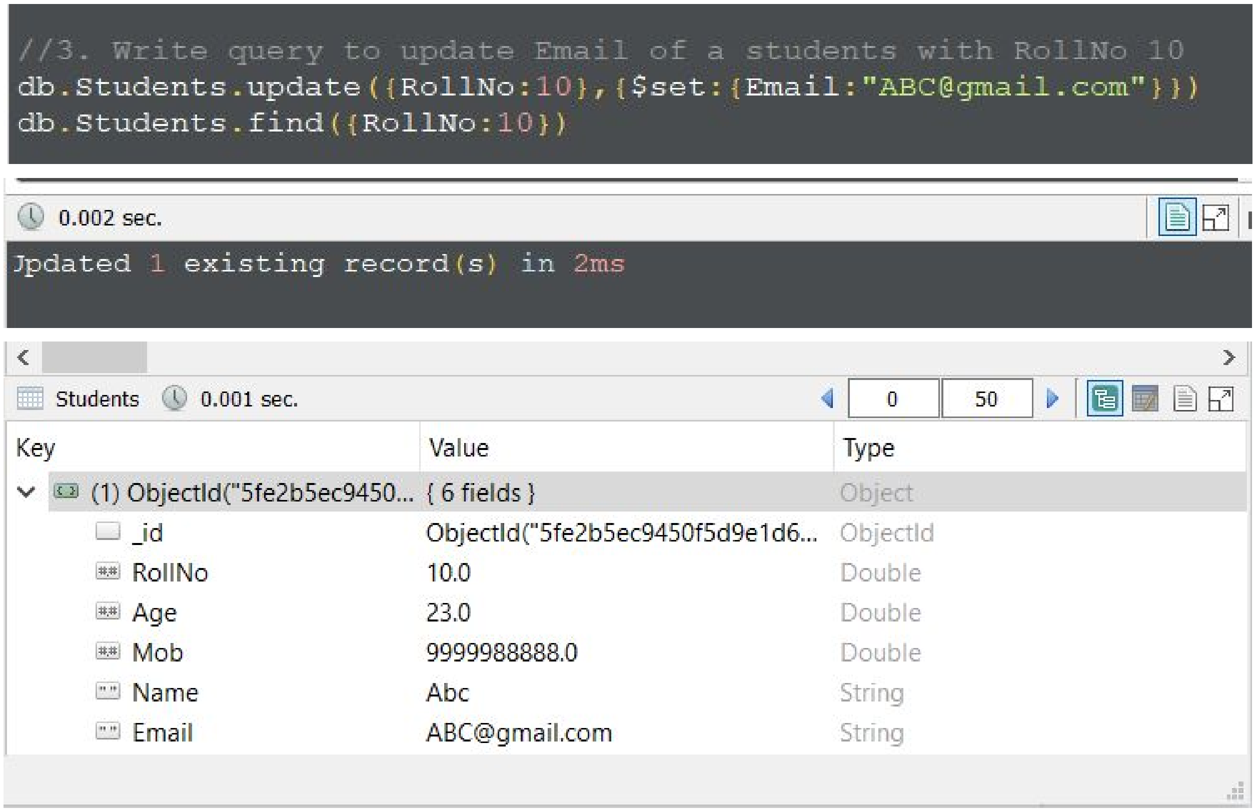
Key	Value	Type
(1) ObjectId("5fe2b5ec...	{ 6 fields }	Object
(2) ObjectId("5fe2b5ec...	{ 6 fields }	Object
_id	ObjectId("5fe2b5ec9450f5d9e...	ObjectId
RollNo	11.0	Double
Age	20.0	Double
Mob	9999988887.0	Double
Name	Efg	String
Email	efg@gmail.com	String
(3) ObjectId("5fe2b5ec...	{ 6 fields }	Object



Key	Value	Type
(1) ObjectId("5fe2b5ec...	{ 6 fields }	Object
(2) ObjectId("5fe2b5ec...	{ 6 fields }	Object
(3) ObjectId("5fe2b5ec...	{ 6 fields }	Object
_id	ObjectId("5fe2b5ec9450f5d9e...	ObjectId
RollNo	8.0	Double
Age	22.0	Double
Mob	9999988886.0	Double
Name	Lmn	String
Email	lmn@gmail.com	String

3. Write a query to update Email of a student with RollNo 10.

//3. Write query to update Email of a students with RollNo 10
db.Students.update({RollNo:10},{$set:{Email:"ABC@gmail.com"}})
db.Students.find({RollNo:10})



```
//3. Write query to update Email of a students with RollNo 10
db.Students.update({RollNo:10},{$set:{Email:"ABC@gmail.com"}})
db.Students.find({RollNo:10})
```

0.002 sec.

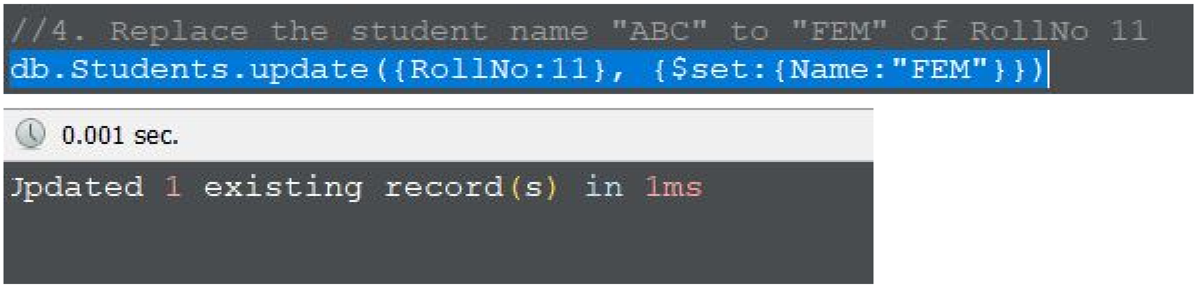
```
Jpdated 1 existing record(s) in 2ms
```

Students 0.001 sec. 0 50

Key	Value	Type
(1) ObjectId("5fe2b5ec9450...	{ 6 fields }	Object
_id	ObjectId("5fe2b5ec9450f5d9e1d6...	ObjectId
RollNo	10.0	Double
Age	23.0	Double
Mob	9999988888.0	Double
Name	Abc	String
Email	ABC@gmail.com	String

4. Replace the student name from “ABC” to “FEM” of RollNo 11.

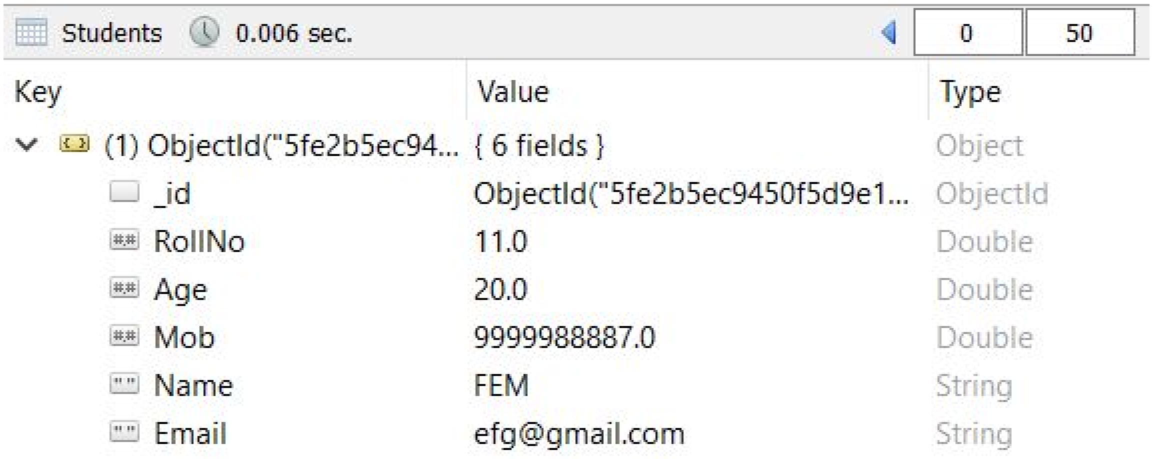
//4. Replace the student name "ABC" to "FEM" of RollNo 11
db.Students.update({RollNo:11}, {$set:{Name:"FEM"}})
db.Students.find({RollNo:11})



```
//4. Replace the student name "ABC" to "FEM" of RollNo 11
db.Students.update({RollNo:11}, {$set:{Name:"FEM"}})
```

0.001 sec.

```
Jpdated 1 existing record(s) in 1ms
```

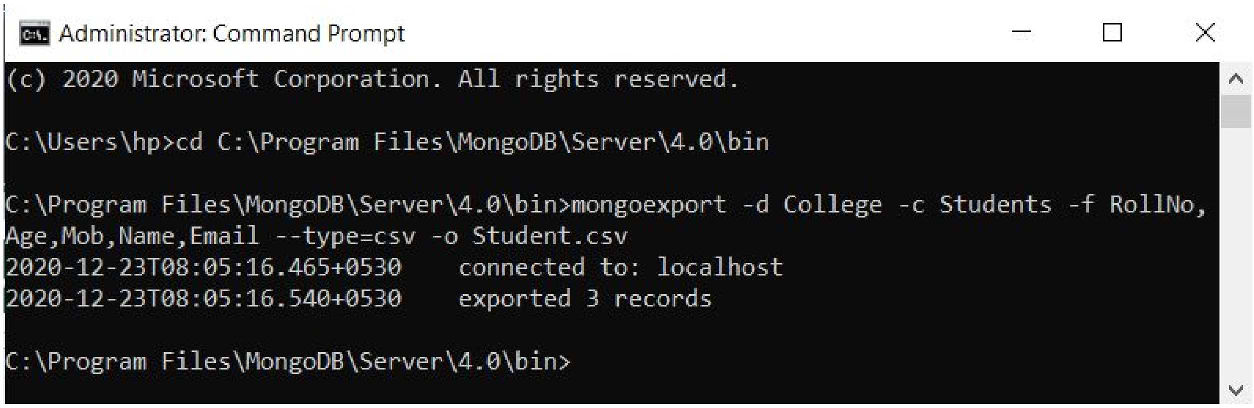


Key	Value	Type
(1) ObjectId("5fe2b5ec94...	{ 6 fields }	Object
_id	ObjectId("5fe2b5ec9450f5d9e1...	ObjectId
RollNo	11.0	Double
Age	20.0	Double
Mob	9999988887.0	Double
Name	FEM	String
Email	efg@gmail.com	String

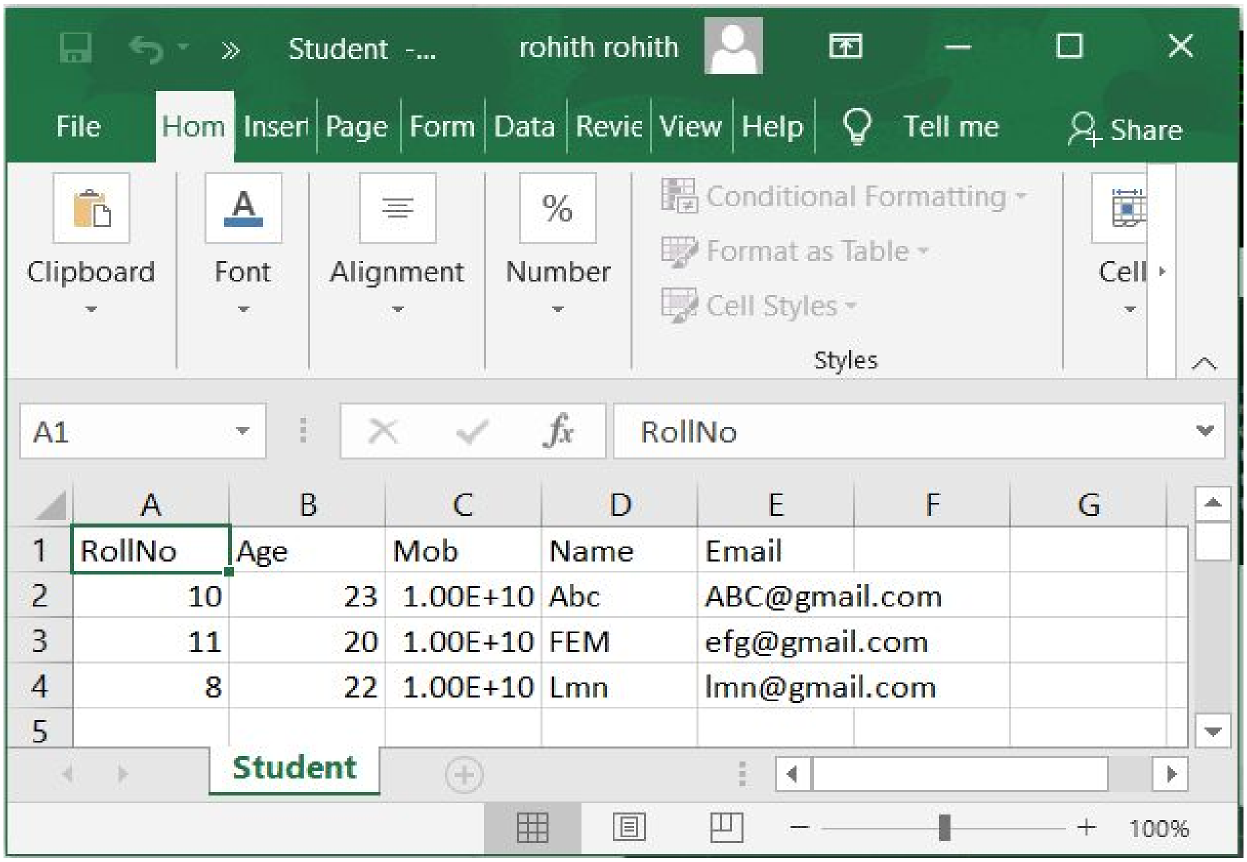
5. Export the created table into local file system

Open command prompt (run as administrator)

C:\Users\hp>cd C:\Program Files\MongoDB\Server\4.0\bin
C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d College -c Students -f RollNo,Age,Mob,Name,Email --type=csv -o Student.csv
2020-12-23T08:05:16.465+0530   connected to: localhost
2020-12-23T08:05:16.540+0530   exported 3 records



```
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\hp>cd C:\Program Files\MongoDB\Server\4.0\bin

C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d College -c Students -f RollNo,
Age,Mob,Name,Email --type=csv -o Student.csv
2020-12-23T08:05:16.465+0530    connected to: localhost
2020-12-23T08:05:16.540+0530    exported 3 records

C:\Program Files\MongoDB\Server\4.0\bin>
```

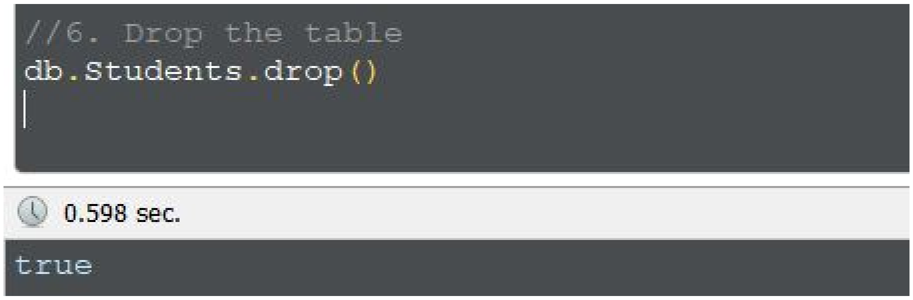


RollNo

	A	B	C	D	E	F	G
1	RollNo	Age	Mob	Name	Email		
2	10	23	1.00E+10	Abc	ABC@gmail.com		
3	11	20	1.00E+10	FEM	efg@gmail.com		
4	8	22	1.00E+10	Lmn	lmn@gmail.com		
5							

6. Drop the table

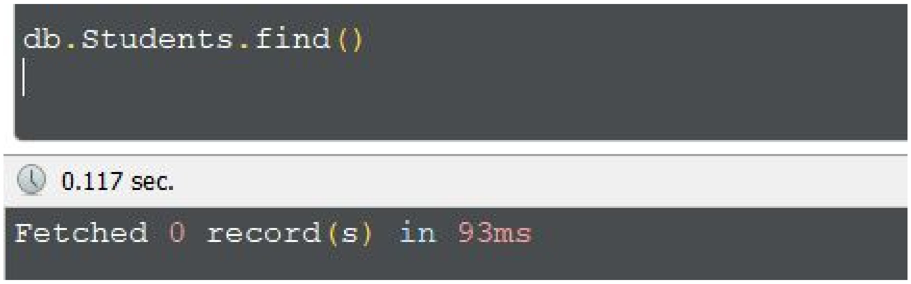
```
//6. Drop the table
  db.Students.drop()
```



```
//6. Drop the table
db.Students.drop()
```

0.598 sec.

```
true
```



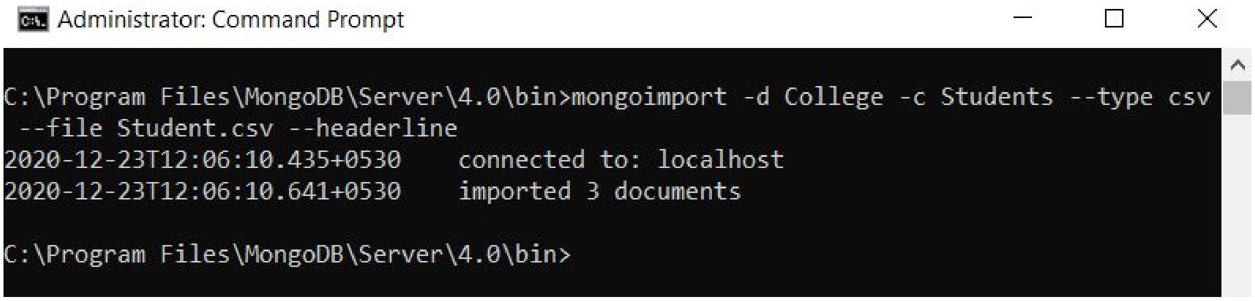
```
db.Students.find()
```

```
Fetched 0 record(s) in 93ms
```

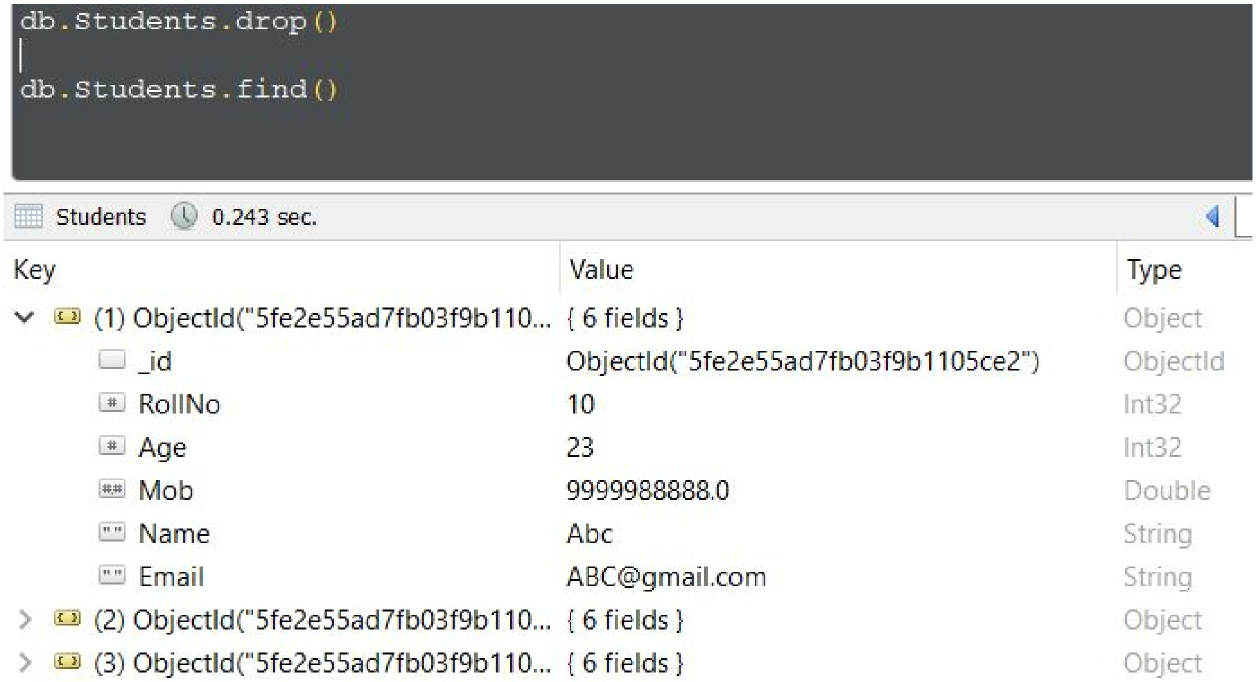
7. Import a given csv dataset from local file system into mongo dB collection.

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d College -c Students --type csv --file Student.csv --headerline

```
2020-12-23T12:06:10.435+0530    connected to: localhost
2020-12-23T12:06:10.641+0530    imported 3 documents
```



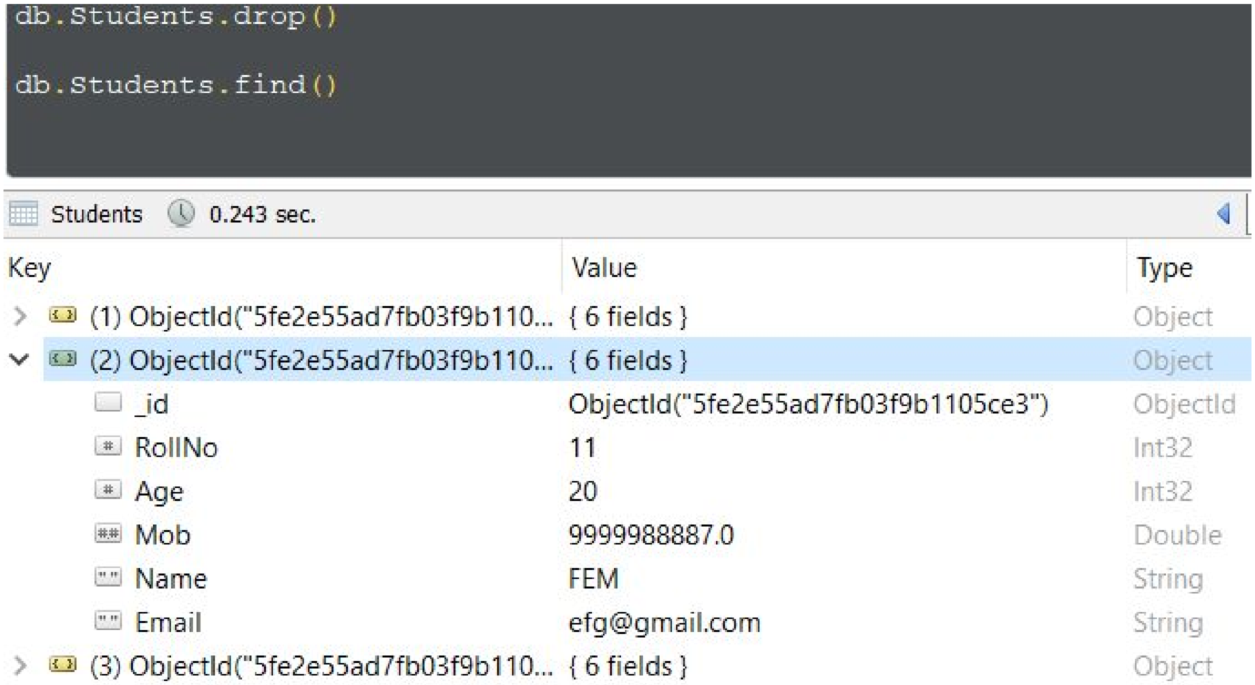
```
C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d College -c Students --type csv
 --file Student.csv --headerline
2020-12-23T12:06:10.435+0530    connected to: localhost
2020-12-23T12:06:10.641+0530    imported 3 documents

C:\Program Files\MongoDB\Server\4.0\bin>
```



```
db.Students.drop()

db.Students.find()
```

Students 0.243 sec.

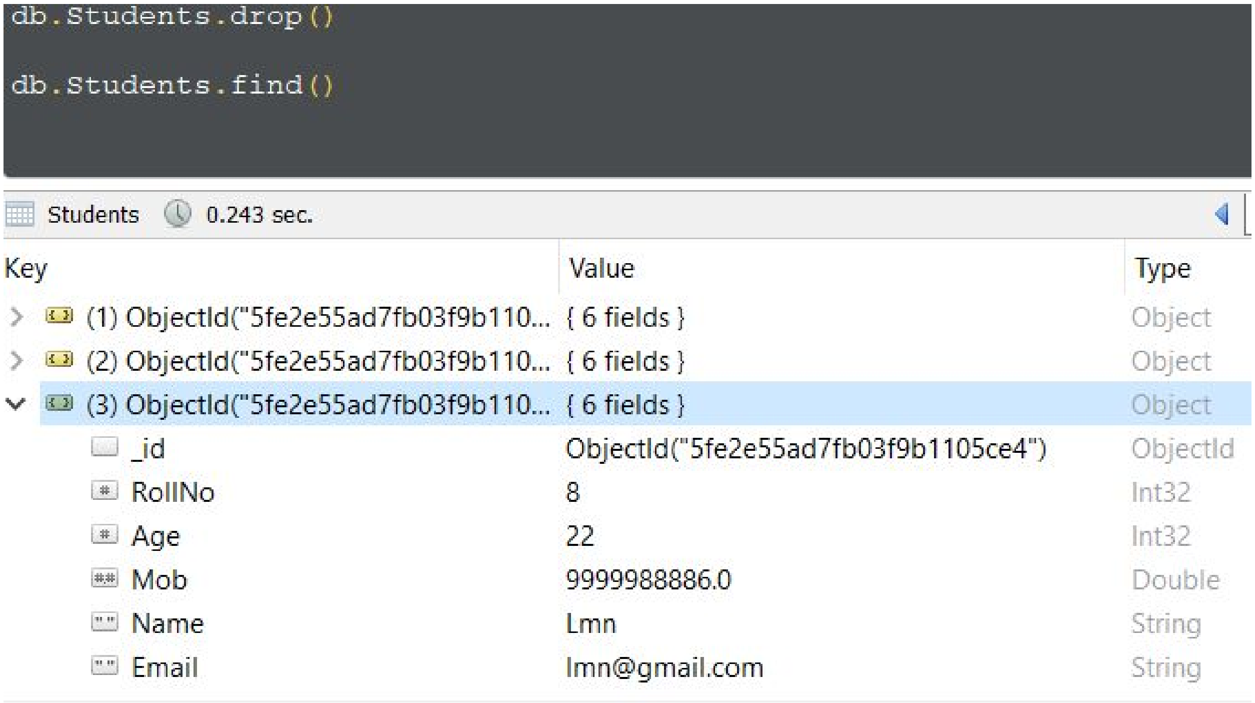
```
Key                                      Value                                  Type
(1) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
    _id                                  ObjectId("5fe2e55ad7fb03f9b1105ce2")   ObjectId
    RollNo                               10                                     Int32
    Age                                  23                                     Int32
    Mob                                  9999988888.0                           Double
    Name                                 Abc                                    String
    Email                                ABC@gmail.com                          String
(2) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
(3) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
```



```
db.Students.drop()

db.Students.find()
```

Students 0.243 sec.

```
Key                                      Value                                  Type
(1) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
(2) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
    _id                                  ObjectId("5fe2e55ad7fb03f9b1105ce3")   ObjectId
    RollNo                               11                                     Int32
    Age                                  20                                     Int32
    Mob                                  9999988887.0                           Double
    Name                                 FEM                                    String
    Email                                efg@gmail.com                          String
(3) ObjectId("5fe2e55ad7fb03f9b110...    { 6 fields }                           Object
```



```
db.Students.drop()

db.Students.find()
```

Students 0.243 sec.

Key	Value	Type
(1) ObjectId("5fe2e55ad7fb03f9b110...	{ 6 fields }	Object
(2) ObjectId("5fe2e55ad7fb03f9b110...	{ 6 fields }	Object
(3) ObjectId("5fe2e55ad7fb03f9b110...	{ 6 fields }	Object
_id	ObjectId("5fe2e55ad7fb03f9b1105ce4")	ObjectId
RollNo	8	Int32
Age	22	Int32
Mob	9999988886.0	Double
Name	Lmn	String
Email	lmn@gmail.com	String

PART 2

Perform the following DB operations using MongoDB.
1. Create a collection by name Customers with the following attributes.
 Cust_id, Acc_Bal, Acc_Type
2. Insert at least 5 values into the table
3. Write a query to display those records whose total account balance is greater than
   1200 of account type 'Z' for each customer_id.
4. Determine Minimum and Maximum account balance for each customer_id.
5. Export the created collection into local file system
6. Drop the table
7. Import a given csv dataset from the local file system into mongodb collection.

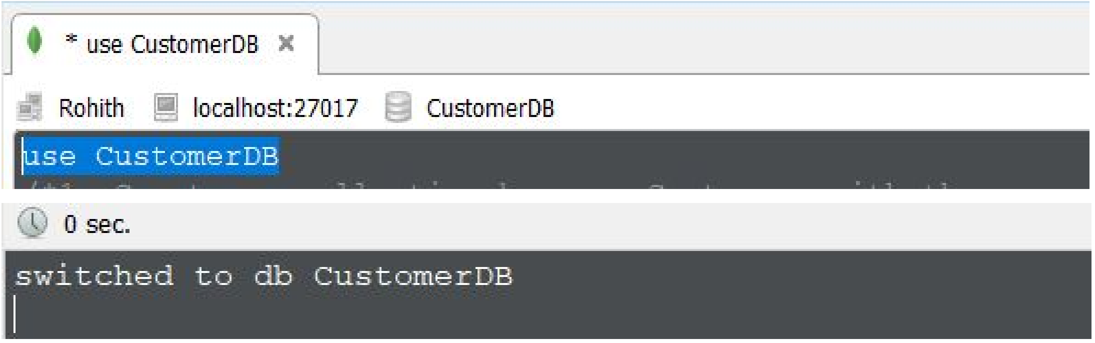
Perform the following DB operations using MongoDB.

1. **Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type**

   use CustomerDB

   //1. Create a collection by name Customers with the following attributes.Cust_id, Acc_Bal, Acc_Type

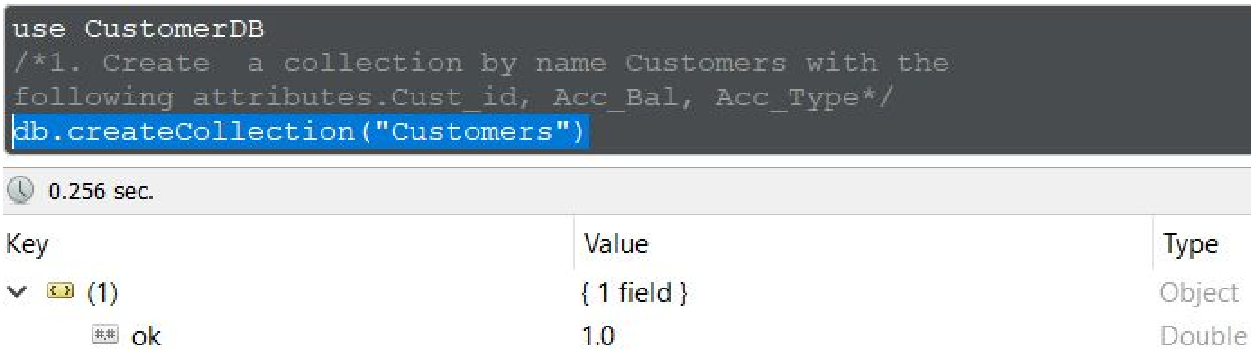
   db.createCollection("Customers")



```
use CustomerDB
```

0 sec.

```
switched to db CustomerDB
```



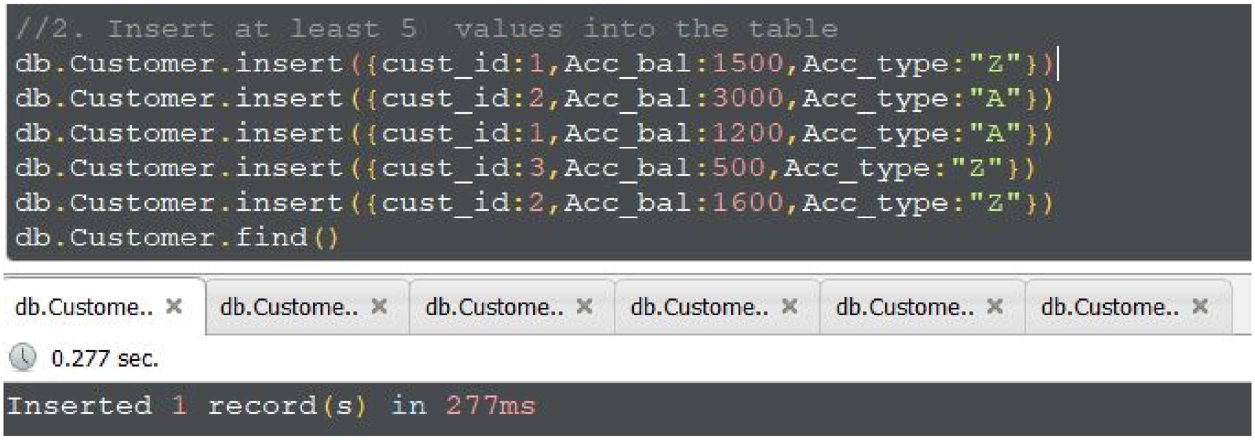
```
use CustomerDB
/*1. Create  a collection by name Customers with the
following attributes.Cust_id, Acc_Bal, Acc_Type*/
db.createCollection("Customers")
```

0.256 sec.

Key	Value	Type
(1)	{ 1 field }	Object
ok	1.0	Double

2. Insert at least 5 values into the table

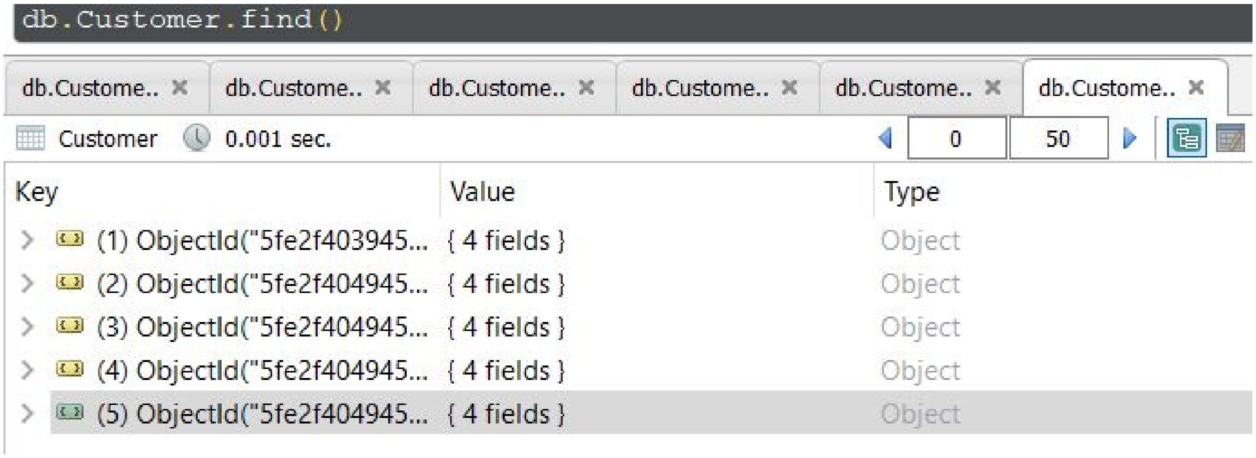
```
//2. Insert at least 5  values into the table
db.Customer.insert({cust_id:1,Acc_bal:1500,Acc_type:"Z"})
db.Customer.insert({cust_id:2,Acc_bal:3000,Acc_type:"A"})
db.Customer.insert({cust_id:1,Acc_bal:1200,Acc_type:"A"})
db.Customer.insert({cust_id:3,Acc_bal:500,Acc_type:"Z"})
db.Customer.insert({cust_id:2,Acc_bal:1600,Acc_type:"Z"})
db.Customer.find()
```



```
//2. Insert at least 5  values into the table
db.Customer.insert({cust_id:1,Acc_bal:1500,Acc_type:"Z"})
db.Customer.insert({cust_id:2,Acc_bal:3000,Acc_type:"A"})
db.Customer.insert({cust_id:1,Acc_bal:1200,Acc_type:"A"})
db.Customer.insert({cust_id:3,Acc_bal:500,Acc_type:"Z"})
db.Customer.insert({cust_id:2,Acc_bal:1600,Acc_type:"Z"})
db.Customer.find()
```

0.277 sec.

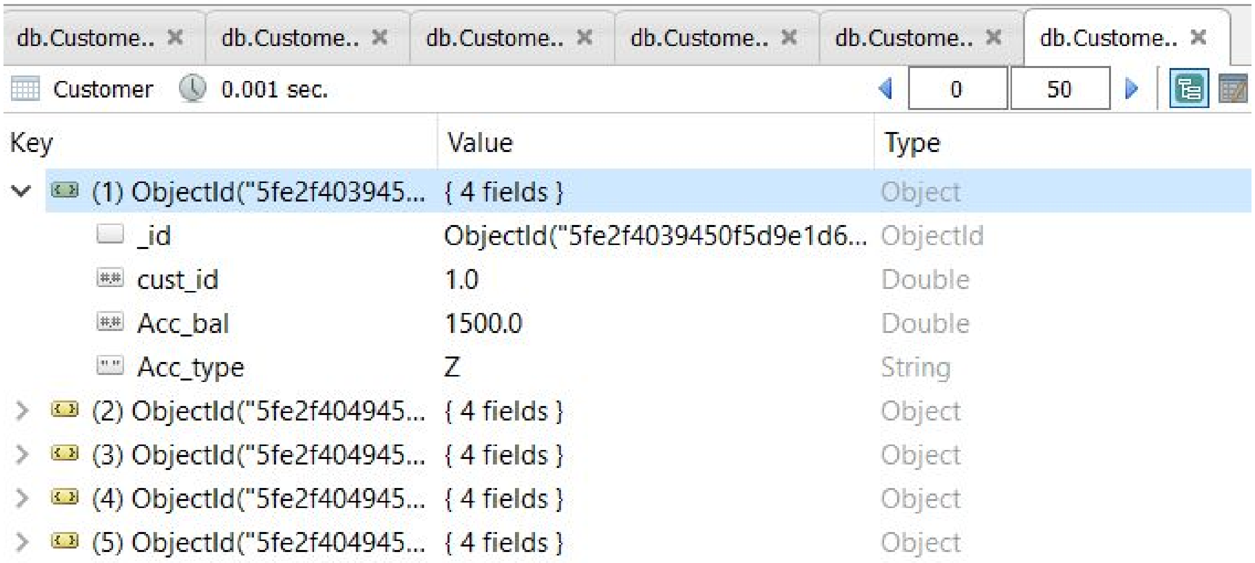
```
Inserted 1 record(s) in 277ms
```



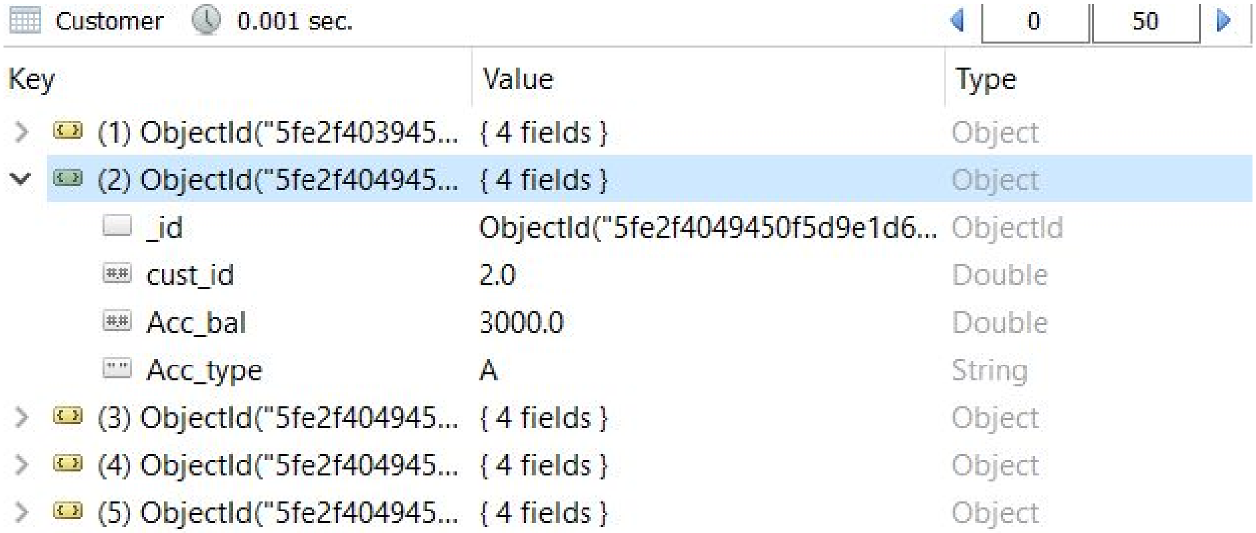
```
db.Customer.find()
```

Customer 0.001 sec.

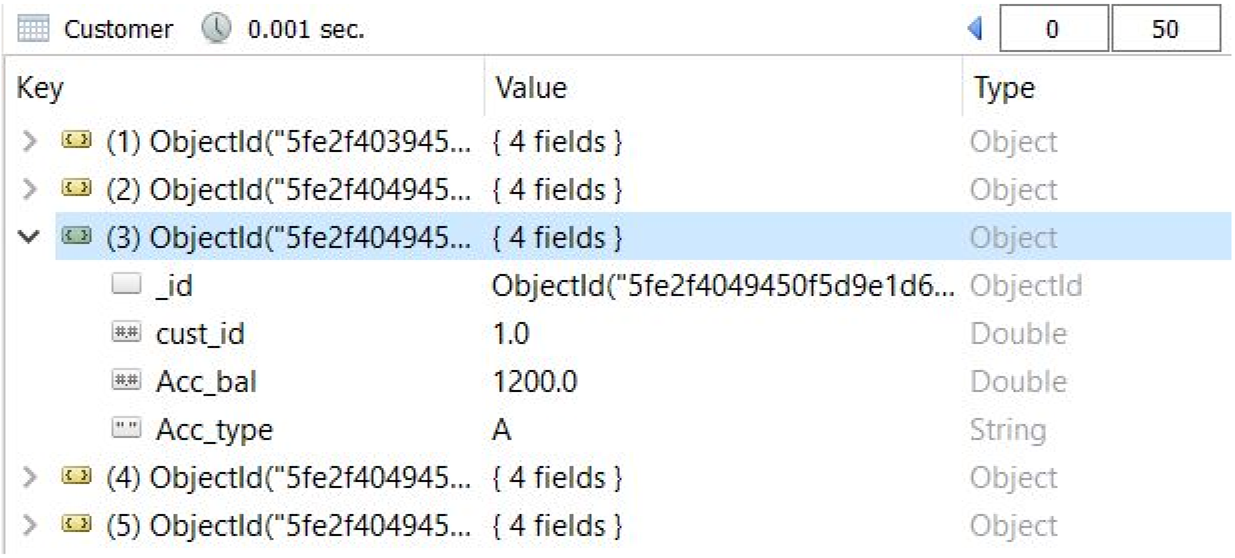
Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object



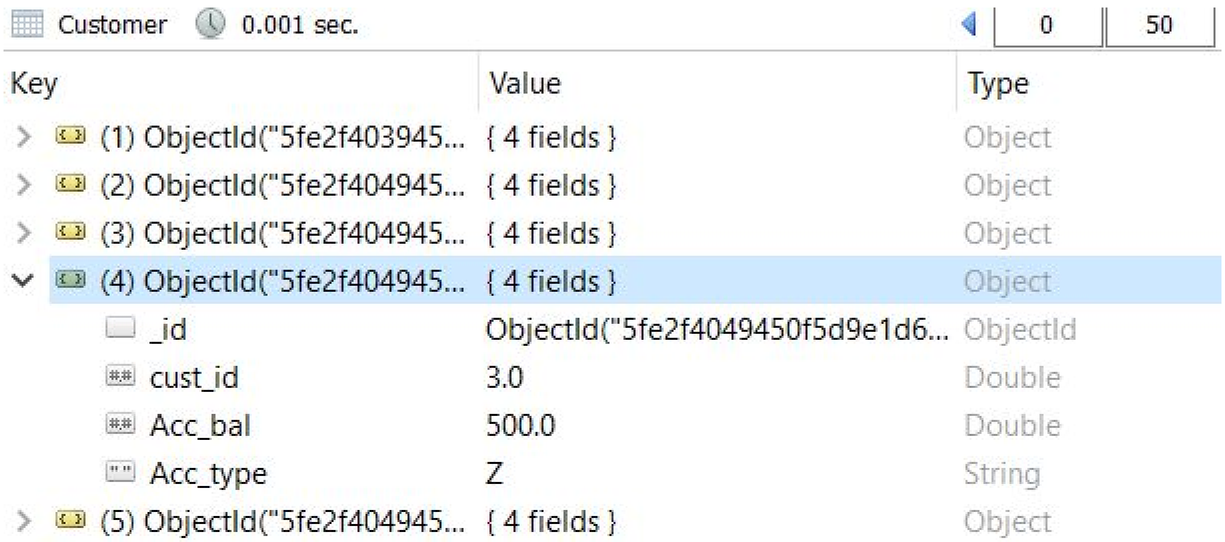
Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
_id	ObjectId("5fe2f4039450f5d9e1d6...	ObjectId
cust_id	1.0	Double
Acc_bal	1500.0	Double
Acc_type	Z	String
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object



Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
_id	ObjectId("5fe2f4049450f5d9e1d6...	ObjectId
cust_id	2.0	Double
Acc_bal	3000.0	Double
Acc_type	A	String
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object

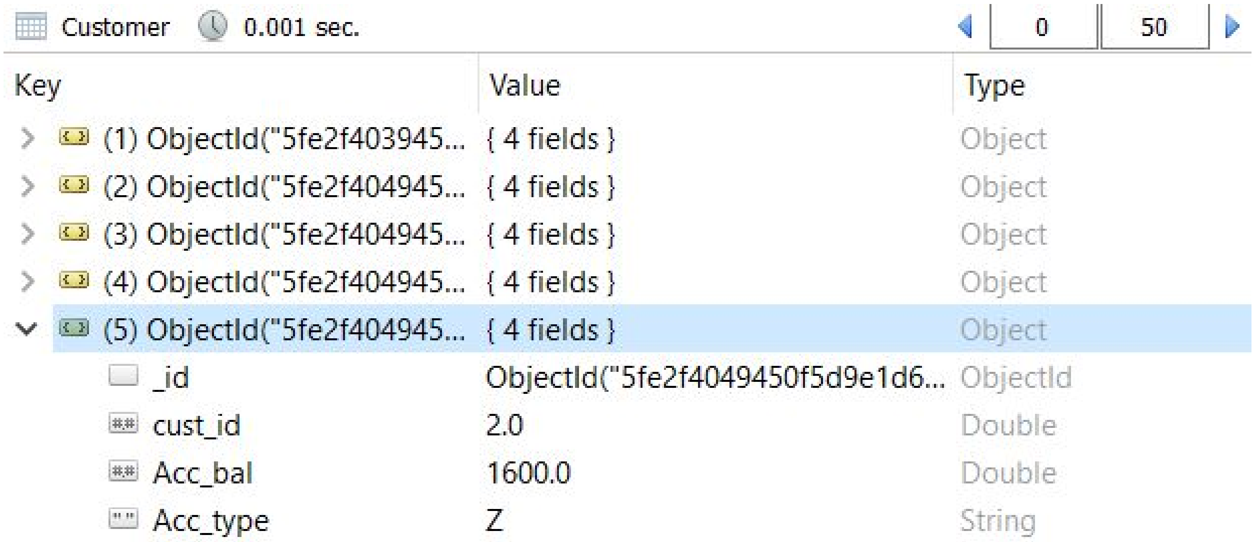


Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
_id	ObjectId("5fe2f4049450f5d9e1d6...	ObjectId
cust_id	1.0	Double
Acc_bal	1200.0	Double
Acc_type	A	String
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object



Customer 0.001 sec. 0 50

Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
_id	ObjectId("5fe2f4049450f5d9e1d6...	ObjectId
cust_id	3.0	Double
Acc_bal	500.0	Double
Acc_type	Z	String
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object



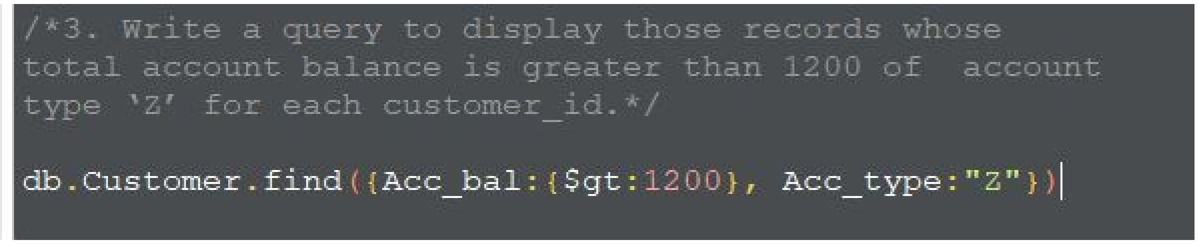
Customer 0.001 sec. 0 50

Key	Value	Type
(1) ObjectId("5fe2f403945...	{ 4 fields }	Object
(2) ObjectId("5fe2f404945...	{ 4 fields }	Object
(3) ObjectId("5fe2f404945...	{ 4 fields }	Object
(4) ObjectId("5fe2f404945...	{ 4 fields }	Object
(5) ObjectId("5fe2f404945...	{ 4 fields }	Object
_id	ObjectId("5fe2f4049450f5d9e1d6...	ObjectId
cust_id	2.0	Double
Acc_bal	1600.0	Double
Acc_type	Z	String

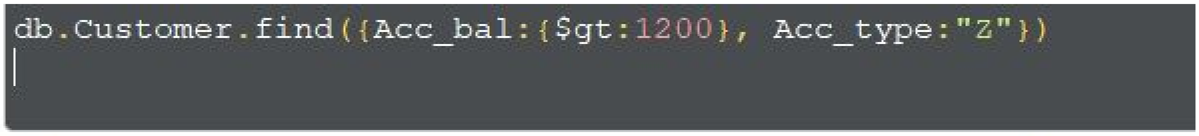
3. **Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.**

//3. Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.

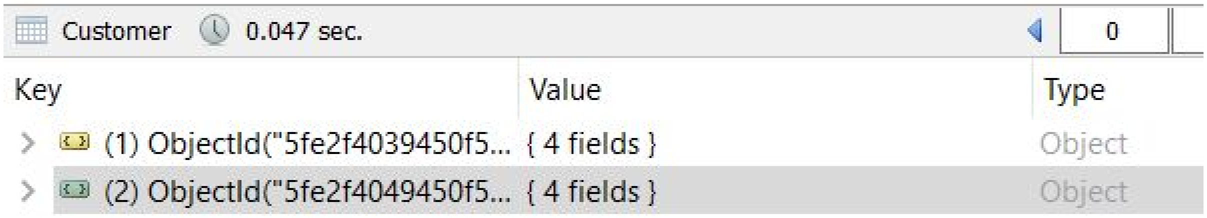
db.Customer.find({Acc_bal:{$gt:1200}, Acc_type:"Z"})



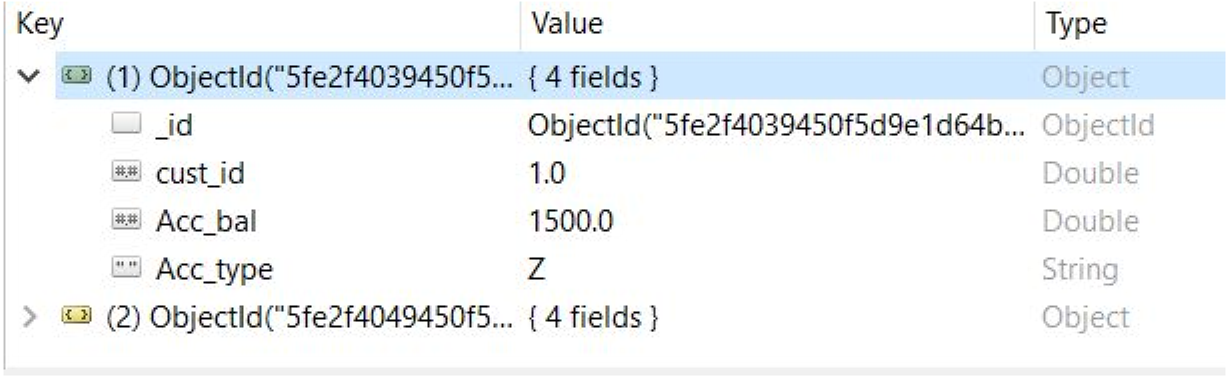
```
/*3. Write a query to display those records whose
total account balance is greater than 1200 of  account
type 'Z' for each customer_id.*/

db.Customer.find({Acc_bal:{$gt:1200}, Acc_type:"Z"})
```



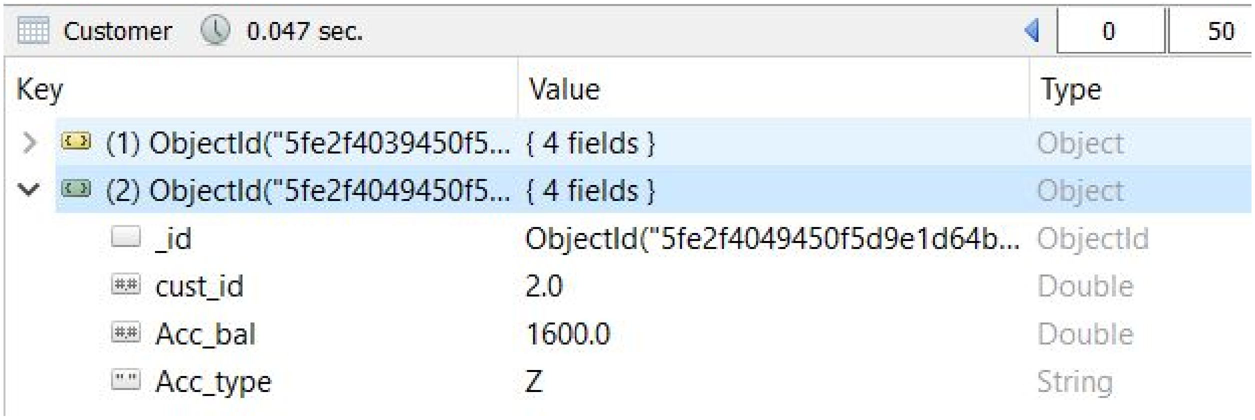
```
db.Customer.find({Acc_bal:{$gt:1200}, Acc_type:"Z"})
```



```
Customer   0.047 sec.                                  0
Key                                   Value            Type
> (1) ObjectId("5fe2f4039450f5...    { 4 fields }     Object
> (2) ObjectId("5fe2f4049450f5...    { 4 fields }     Object
```



```
Key                                   Value                              Type
v (1) ObjectId("5fe2f4039450f5...    { 4 fields }                       Object
      _id                             ObjectId("5fe2f4039450f5d9e1d64b...  ObjectId
      cust_id                         1.0                                Double
      Acc_bal                         1500.0                             Double
      Acc_type                        Z                                  String
> (2) ObjectId("5fe2f4049450f5...    { 4 fields }                       Object
```

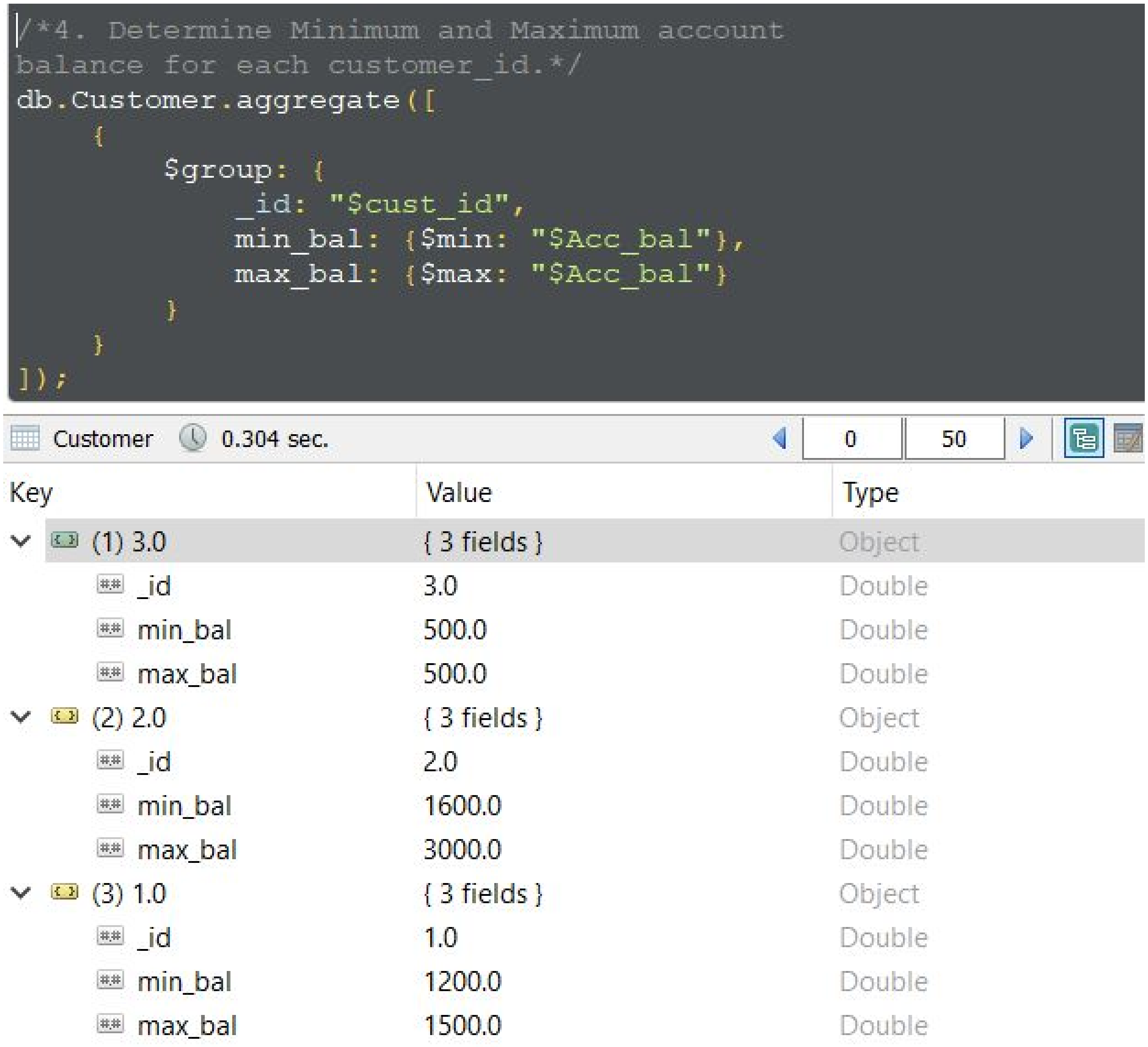


```
Customer   0.047 sec.                                  0      50
Key                                   Value                              Type
> (1) ObjectId("5fe2f4039450f5...    { 4 fields }                       Object
v (2) ObjectId("5fe2f4049450f5...    { 4 fields }                       Object
      _id                             ObjectId("5fe2f4049450f5d9e1d64b...  ObjectId
      cust_id                         2.0                                Double
      Acc_bal                         1600.0                             Double
      Acc_type                        Z                                  String
```

4. Determine Minimum and Maximum account balance for each customer_id.

//4. Determine Minimum and Maximum account balance for each customer_id.

```
db.Customer.aggregate([
   {
      $group: {
         _id: "$cust_id",
         min_bal: {$min: "$Acc_bal"},
         max_bal: {$max: "$Acc_bal"}
      }
   }
]);
```



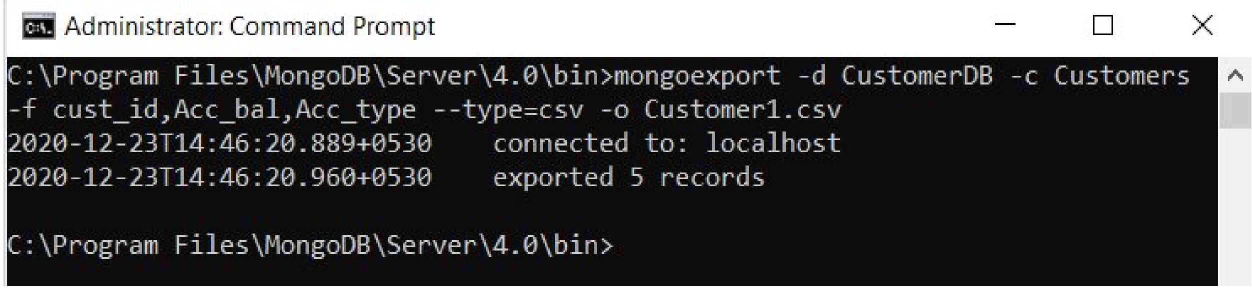
```
/*4. Determine Minimum and Maximum account
balance for each customer_id.*/
db.Customer.aggregate([
      {
            $group: {
                  _id: "$cust_id",
                  min_bal: {$min: "$Acc_bal"},
                  max_bal: {$max: "$Acc_bal"}
            }
      }
]);
```

```
Key                 Value           Type
(1) 3.0             { 3 fields }    Object
    _id             3.0             Double
    min_bal         500.0           Double
    max_bal         500.0           Double
(2) 2.0             { 3 fields }    Object
    _id             2.0             Double
    min_bal         1600.0          Double
    max_bal         3000.0          Double
(3) 1.0             { 3 fields }    Object
    _id             1.0             Double
    min_bal         1200.0          Double
    max_bal         1500.0          Double
```

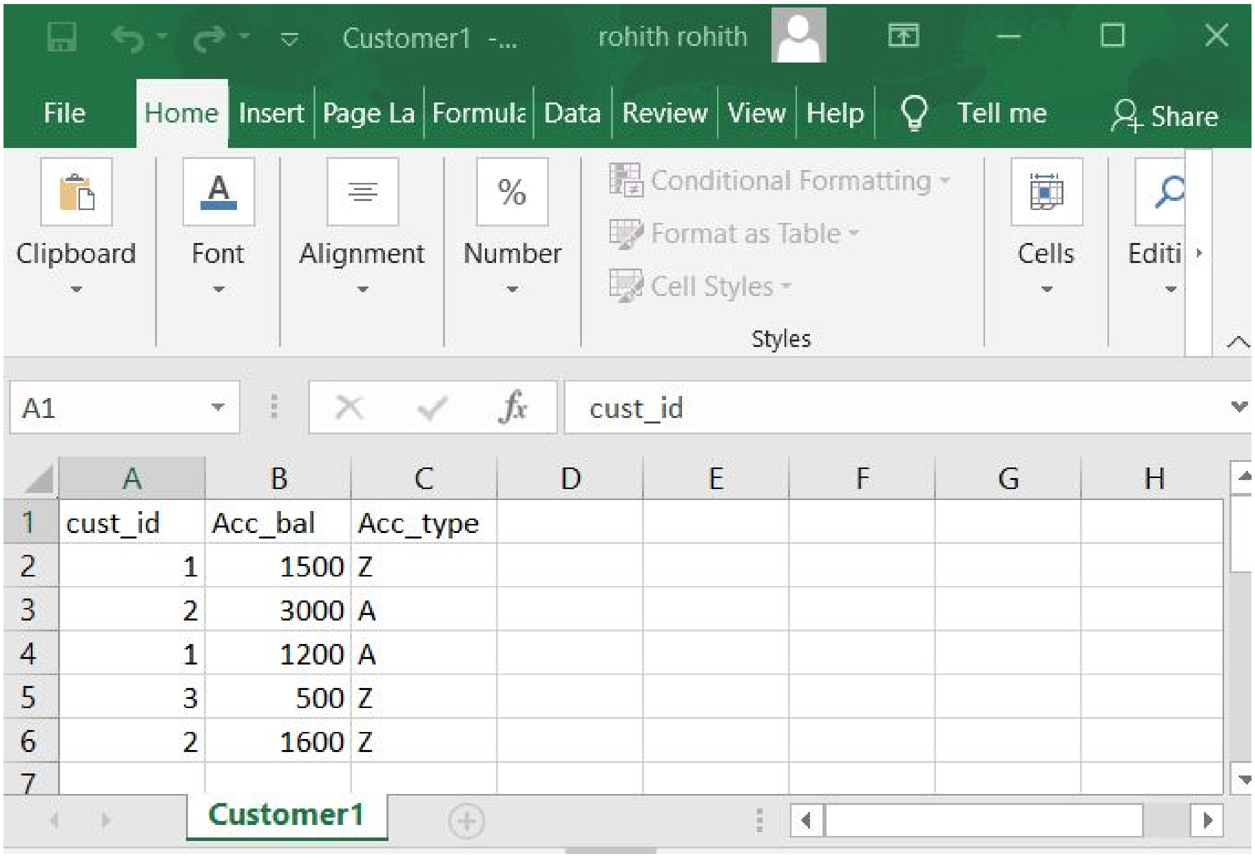
5. Export the created collection into local file system

//5. Export the created collection into local file system

C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d CustomerDB -c Customers -f cust_id,Acc_bal,Acc_type --type=csv -o Customer1.csv



```
C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d CustomerDB -c Customers
-f cust_id,Acc_bal,Acc_type --type=csv -o Customer1.csv
2020-12-23T14:46:20.889+0530    connected to: localhost
2020-12-23T14:46:20.960+0530    exported 5 records

C:\Program Files\MongoDB\Server\4.0\bin>
```

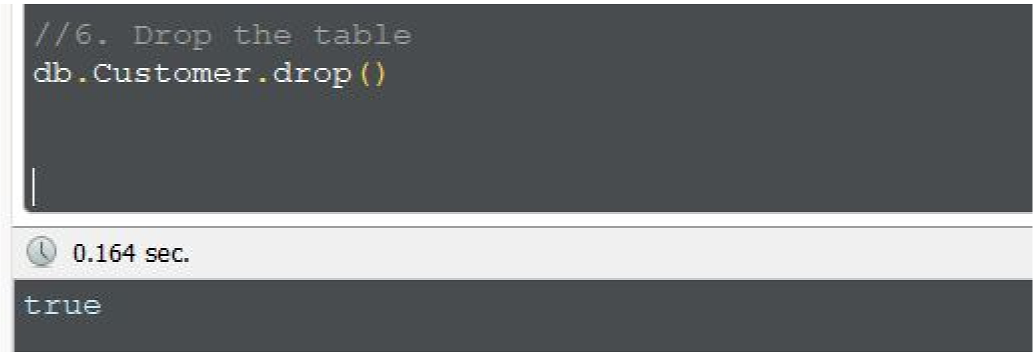


	A	B	C
1	cust_id	Acc_bal	Acc_type
2	1	1500	Z
3	2	3000	A
4	1	1200	A
5	3	500	Z
6	2	1600	Z
7			

6. Drop the table

//6. Drop the table

db.Customer.drop()



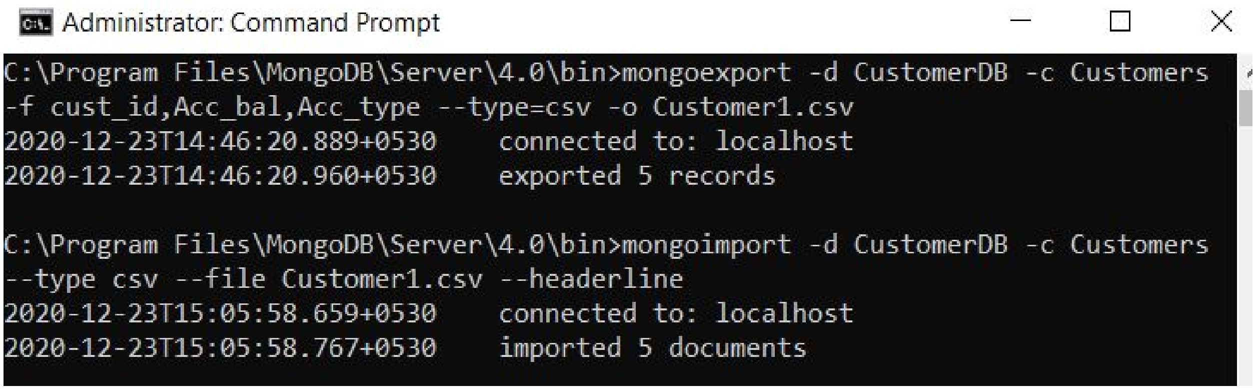
```
//6. Drop the table
db.Customer.drop()
```

```
true
```

7. Import a given csv dataset from the local file system into mongodb collection.

//7. Import a given csv dataset from local file system into mongodb collection

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d CustomerDB -c Customers --type csv --file Customer1.csv --headerline



```
C:\Program Files\MongoDB\Server\4.0\bin>mongoexport -d CustomerDB -c Customers
-f cust_id,Acc_bal,Acc_type --type=csv -o Customer1.csv
2020-12-23T14:46:20.889+0530    connected to: localhost
2020-12-23T14:46:20.960+0530    exported 5 records

C:\Program Files\MongoDB\Server\4.0\bin>mongoimport -d CustomerDB -c Customers
--type csv --file Customer1.csv --headerline
2020-12-23T15:05:58.659+0530    connected to: localhost
2020-12-23T15:05:58.767+0530    imported 5 documents
```

Perform the following DB operations using Cassandra.

LAB - 5

<u>Opening cassandra in command prompt(run as administrator)</u>

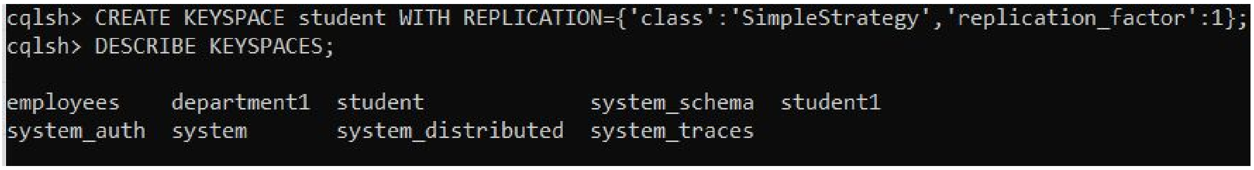
```
C:\Users\hp>cd C:\apache-cassandra-3.11.4\bin

C:\apache-cassandra-3.11.4\bin>cassandra.bat -f
```

<u>In another command prompt</u>

```
C:\Users\hp>cd C:\apache-cassandra-3.11.4\bin

C:\apache-cassandra-3.11.4\bin>cqlsh
```

1. **Create a keyspace by name <u>Student</u>**

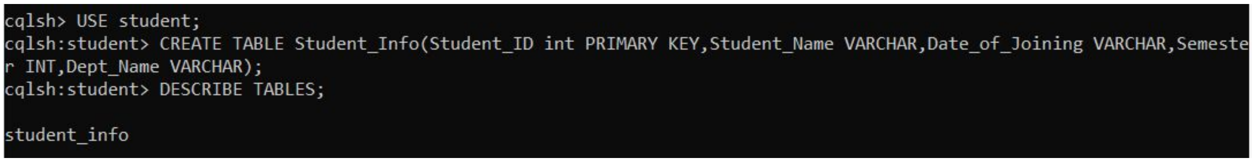
```
cqlsh> CREATE KEYSPACE student WITH
REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;
```



```
cqlsh> CREATE KEYSPACE student WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

employees    department1  student             system_schema  student1
system_auth  system       system_distributed  system_traces
```

2. **Create a column family by name Student-Info with attributes Student_Id Primary Key, Stude_Name, Date_of_Joining, Semester, Dept_Name**

```
cqlsh> USE student;

cqlsh:student> CREATE TABLE Student_Info(Student_ID int PRIMARY
KEY,Student_Name VARCHAR,Date_of_Joining VARCHAR,Semester
INT,Dept_Name VARCHAR);

cqlsh:student> DESCRIBE TABLES;
```

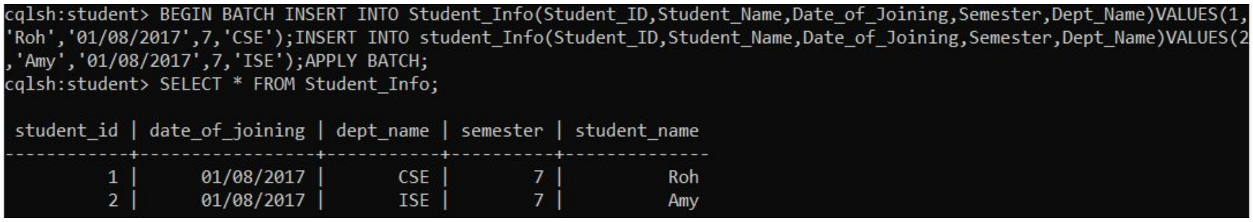


```
cqlsh> USE student;
cqlsh:student> CREATE TABLE Student_Info(Student_ID int PRIMARY KEY,Student_Name VARCHAR,Date_of_Joining VARCHAR,Semeste
r INT,Dept_Name VARCHAR);
cqlsh:student> DESCRIBE TABLES;

student_info
```

3. **Insert the values into the table in batch**

   cqlsh:student> BEGIN BATCH INSERT INTO Student_Info(Student_ID,Student_Name,Date_of_Joining,Semester,Dept_Name)VALUES(1,'Roh','01/08/2017',7,'CSE');INSERT INTO student_Info(Student_ID,Student_Name,Date_of_Joining,Semester,Dept_Name)VALUES(2,'Amy','01/08/2017',7,'ISE');APPLY BATCH;

   cqlsh:student> SELECT * FROM Student_Info;

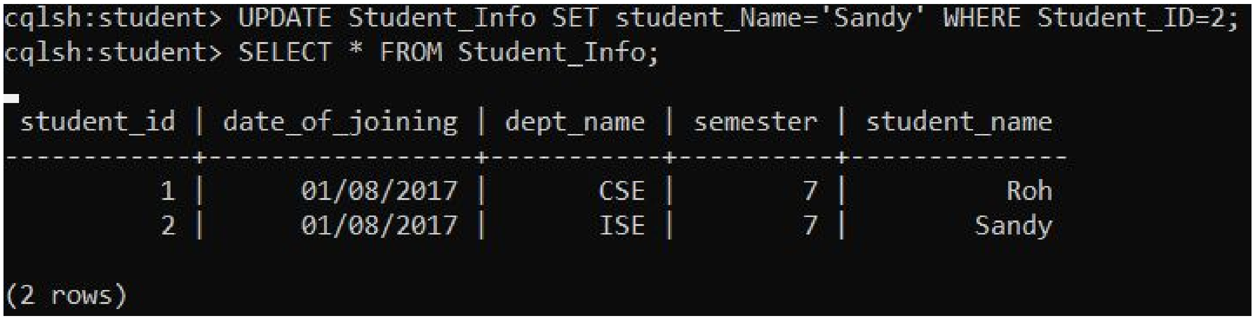


```
cqlsh:student> BEGIN BATCH INSERT INTO Student_Info(Student_ID,Student_Name,Date_of_Joining,Semester,Dept_Name)VALUES(1,
'Roh','01/08/2017',7,'CSE');INSERT INTO student_Info(Student_ID,Student_Name,Date_of_Joining,Semester,Dept_Name)VALUES(2
,'Amy','01/08/2017',7,'ISE');APPLY BATCH;
cqlsh:student> SELECT * FROM Student_Info;

 student_id | date_of_joining | dept_name | semester | student_name
------------+-----------------+-----------+----------+--------------
          1 |      01/08/2017 |       CSE |        7 |          Roh
          2 |      01/08/2017 |       ISE |        7 |          Amy
```

4. **Update Student name of Student_Id 2**

   cqlsh:student> UPDATE Student_Info SET student_Name='Sandy' WHERE Student_ID=2;

   cqlsh:student> SELECT * FROM Student_Info;

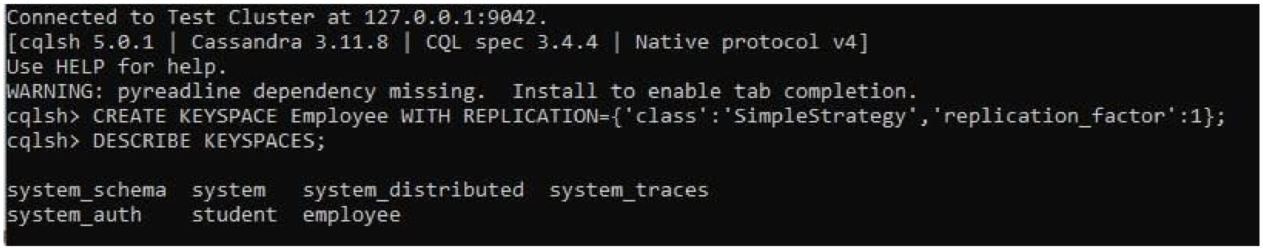


```
cqlsh:student> UPDATE Student_Info SET student_Name='Sandy' WHERE Student_ID=2;
cqlsh:student> SELECT * FROM Student_Info;

 student_id | date_of_joining | dept_name | semester | student_name
------------+-----------------+-----------+----------+--------------
          1 |      01/08/2017 |       CSE |        7 |          Roh
          2 |      01/08/2017 |       ISE |        7 |        Sandy

(2 rows)
```

LAB 6

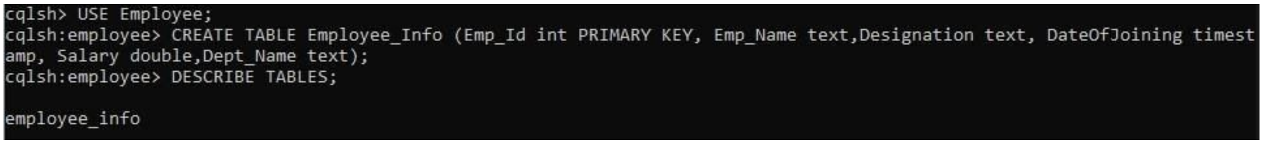
1. Perform the following DB operations using Cassandra.

1. Create a keyspace by name **Employee**



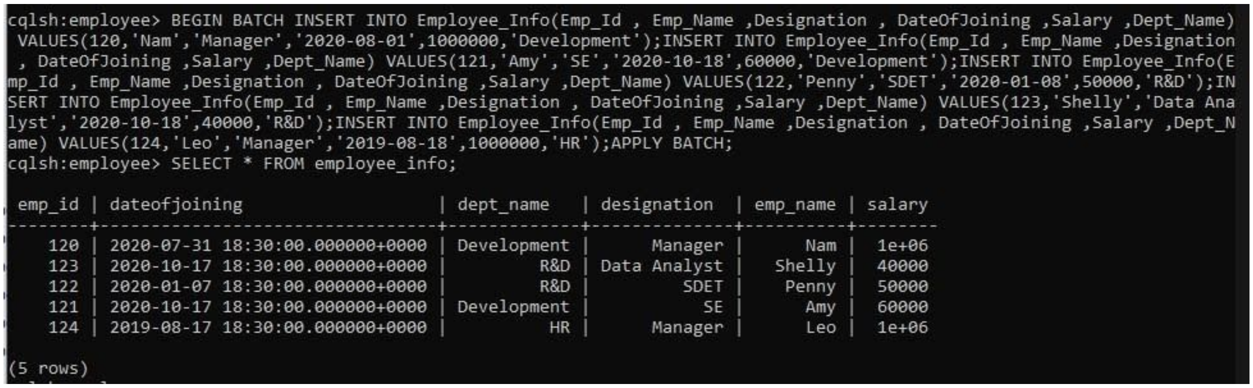
```
Connected to Test Cluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.8 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
WARNING: pyreadline dependency missing.  Install to enable tab completion.
cqlsh> CREATE KEYSPACE Employee WITH REPLICATION={'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

system_schema  system   system_distributed  system_traces
system_auth    student  employee
```

2. Create a column family by name Employee-Info with attributes Emp_Id Primary Key, Emp_Name, Designation, Date_of_Joining, Salary, Dept_Name



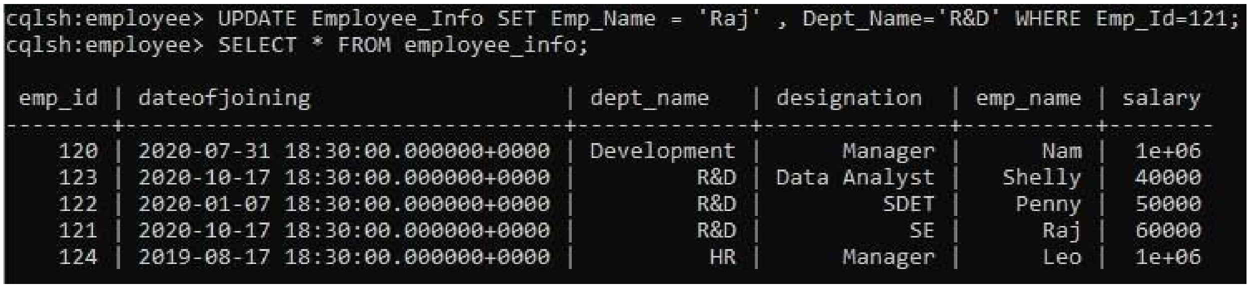
```
cqlsh> USE Employee;
cqlsh:employee> CREATE TABLE Employee_Info (Emp_Id int PRIMARY KEY, Emp_Name text,Designation text, DateOfJoining timest
amp, Salary double,Dept_Name text);
cqlsh:employee> DESCRIBE TABLES;

employee_info
```

3. Insert the values into the table in batch



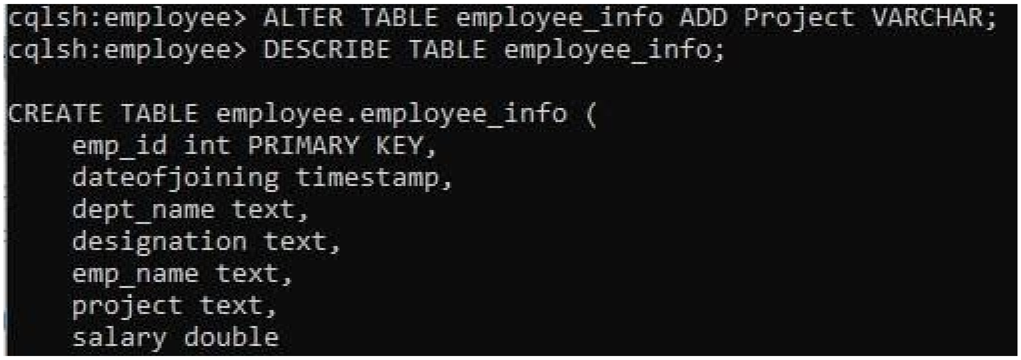
```
cqlsh:employee> BEGIN BATCH INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name)
 VALUES(120,'Nam','Manager','2020-08-01',1000000,'Development');INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation
 , DateOfJoining ,Salary ,Dept_Name) VALUES(121,'Amy','SE','2020-10-18',60000,'Development');INSERT INTO Employee_Info(E
mp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(122,'Penny','SDET','2020-01-08',50000,'R&D');IN
SERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(123,'Shelly','Data Ana
lyst','2020-10-18',40000,'R&D');INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_N
ame) VALUES(124,'Leo','Manager','2019-08-18',1000000,'HR');APPLY BATCH;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | salary
--------+---------------------------------+-------------+--------------+----------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny |  50000
    121 | 2020-10-17 18:30:00.000000+0000 | Development |           SE |      Amy |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |  1e+06

(5 rows)
```

4. Update Employee name and Department of Emp-Id 121



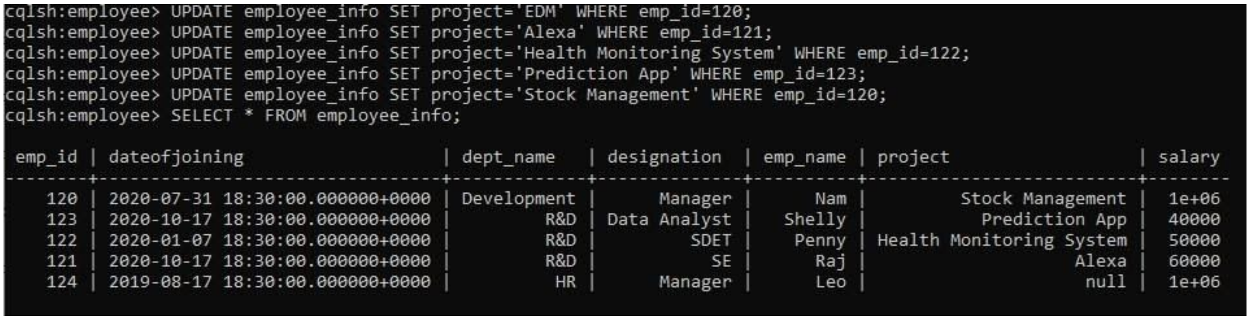
```
cqlsh:employee> UPDATE Employee_Info SET Emp_Name = 'Raj' , Dept_Name='R&D' WHERE Emp_Id=121;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | salary
--------+---------------------------------+-------------+--------------+----------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny |  50000
    121 | 2020-10-17 18:30:00.000000+0000 |         R&D |           SE |      Raj |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |  1e+06
```

5. Alter the schema of the table Employee_Info to add a column Projects which stores a set of Projects done by the corresponding Employee.



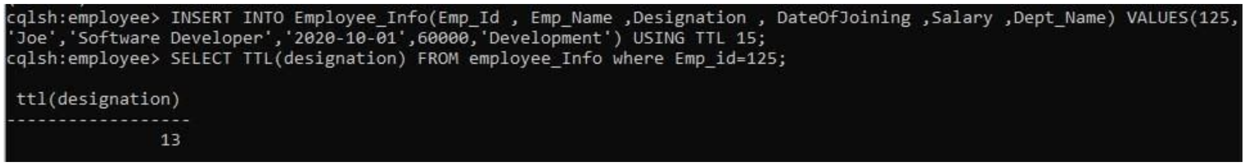
```
cqlsh:employee> ALTER TABLE employee_info ADD Project VARCHAR;
cqlsh:employee> DESCRIBE TABLE employee_info;

CREATE TABLE employee.employee_info (
    emp_id int PRIMARY KEY,
    dateofjoining timestamp,
    dept_name text,
    designation text,
    emp_name text,
    project text,
    salary double
```

6. Update the altered table to add project names.



```
cqlsh:employee> UPDATE employee_info SET project='EDM' WHERE emp_id=120;
cqlsh:employee> UPDATE employee_info SET project='Alexa' WHERE emp_id=121;
cqlsh:employee> UPDATE employee_info SET project='Health Monitoring System' WHERE emp_id=122;
cqlsh:employee> UPDATE employee_info SET project='Prediction App' WHERE emp_id=123;
cqlsh:employee> UPDATE employee_info SET project='Stock Management' WHERE emp_id=120;
cqlsh:employee> SELECT * FROM employee_info;

 emp_id | dateofjoining                   | dept_name   | designation  | emp_name | project                  | salary
--------+---------------------------------+-------------+--------------+----------+--------------------------+--------
    120 | 2020-07-31 18:30:00.000000+0000 | Development |      Manager |      Nam |         Stock Management |  1e+06
    123 | 2020-10-17 18:30:00.000000+0000 |         R&D | Data Analyst |   Shelly |           Prediction App |  40000
    122 | 2020-01-07 18:30:00.000000+0000 |         R&D |         SDET |    Penny | Health Monitoring System |  50000
    121 | 2020-10-17 18:30:00.000000+0000 |         R&D |           SE |      Raj |                    Alexa |  60000
    124 | 2019-08-17 18:30:00.000000+0000 |          HR |      Manager |      Leo |                     null |  1e+06
```

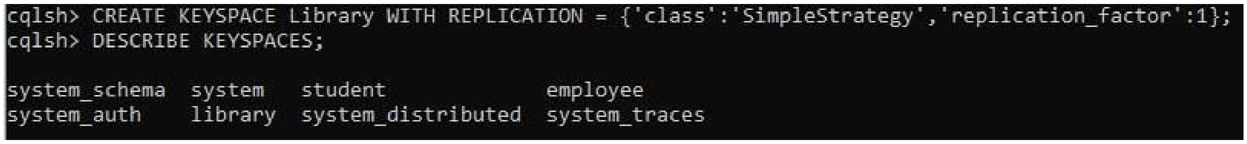
7.Create a TTL of 15 seconds to display the values of Employees.



```
cqlsh:employee> INSERT INTO Employee_Info(Emp_Id , Emp_Name ,Designation , DateOfJoining ,Salary ,Dept_Name) VALUES(125,
'Joe','Software Developer','2020-10-01',60000,'Development') USING TTL 15;
cqlsh:employee> SELECT TTL(designation) FROM employee_Info where Emp_id=125;

 ttl(designation)
------------------
               13
```

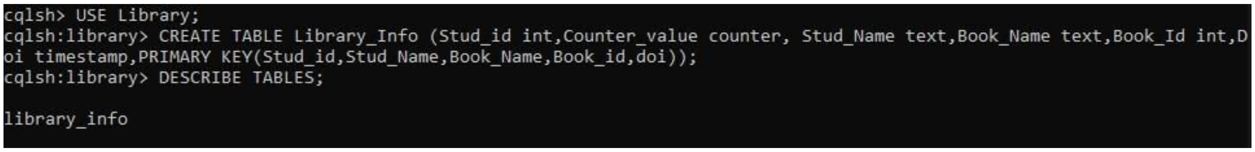
II. Perform the following DB operations using Cassandra..

1.Create a keyspace by name **Library**



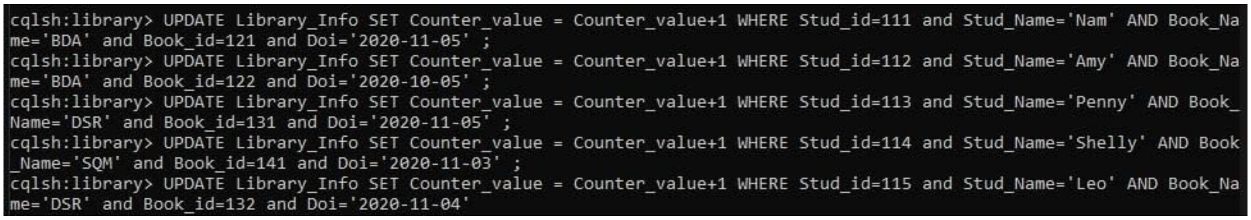
```
cqlsh> CREATE KEYSPACE Library WITH REPLICATION = {'class':'SimpleStrategy','replication_factor':1};
cqlsh> DESCRIBE KEYSPACES;

system_schema  system   student             employee
system_auth    library  system_distributed  system_traces
```

2.Create a column family by name Library-Info with attributes Stud_Id Primary Key, Counter_value of type Counter, Stud_Name, Book-Name, Book-Id, Date_of_issue



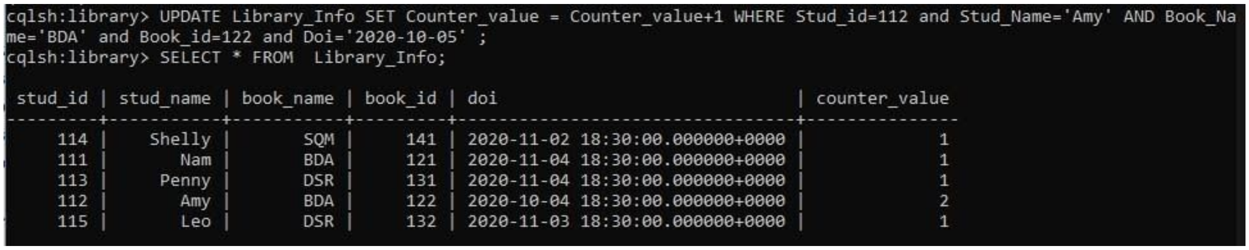
```
cqlsh> USE Library;
cqlsh:library> CREATE TABLE Library_Info (Stud_id int,Counter_value counter, Stud_Name text,Book_Name text,Book_Id int,D
oi timestamp,PRIMARY KEY(Stud_id,Stud_Name,Book_Name,Book_id,doi));
cqlsh:library> DESCRIBE TABLES;

library_info
```

3.Insert the values into the table in batch



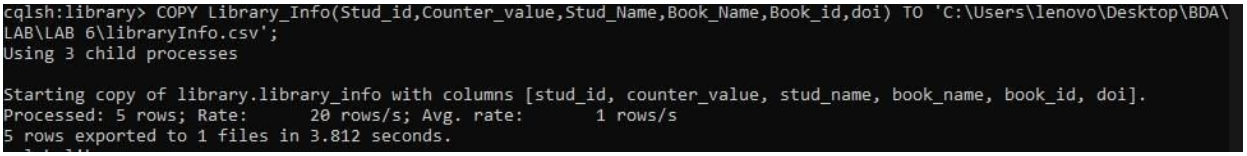
```
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=111 and Stud_Name='Nam' AND Book_Na
me='BDA' and Book_id=121 and Doi='2020-11-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=112 and Stud_Name='Amy' AND Book_Na
me='BDA' and Book_id=122 and Doi='2020-10-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=113 and Stud_Name='Penny' AND Book_
Name='DSR' and Book_id=131 and Doi='2020-11-05' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=114 and Stud_Name='Shelly' AND Book
_Name='SQM' and Book_id=141 and Doi='2020-11-03' ;
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=115 and Stud_Name='Leo' AND Book_Na
me='DSR' and Book_id=132 and Doi='2020-11-04'
```

4.Display the details of the table created and increase the value of the counter



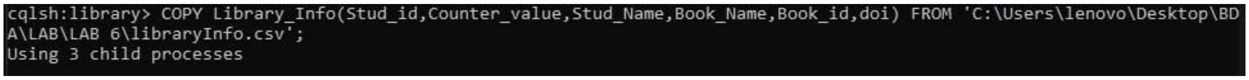
```
cqlsh:library> UPDATE Library_Info SET Counter_value = Counter_value+1 WHERE Stud_id=112 and Stud_Name='Amy' AND Book_Na
me='BDA' and Book_id=122 and Doi='2020-10-05' ;
cqlsh:library> SELECT * FROM  Library_Info;

 stud_id | stud_name | book_name | book_id | doi                             | counter_value
---------+-----------+-----------+---------+---------------------------------+---------------
     114 |    Shelly |       SQM |     141 | 2020-11-02 18:30:00.000000+0000 |             1
     111 |       Nam |       BDA |     121 | 2020-11-04 18:30:00.000000+0000 |             1
     113 |     Penny |       DSR |     131 | 2020-11-04 18:30:00.000000+0000 |             1
     112 |       Amy |       BDA |     122 | 2020-10-04 18:30:00.000000+0000 |             2
     115 |       Leo |       DSR |     132 | 2020-11-03 18:30:00.000000+0000 |             1
```

5.Export the created column to a csv file



```
cqlsh:library> COPY Library_Info(Stud_id,Counter_value,Stud_Name,Book_Name,Book_id,doi) TO 'C:\Users\lenovo\Desktop\BDA\
LAB\LAB 6\libraryInfo.csv';
Using 3 child processes

Starting copy of library.library_info with columns [stud_id, counter_value, stud_name, book_name, book_id, doi].
Processed: 5 rows; Rate:      20 rows/s; Avg. rate:       1 rows/s
5 rows exported to 1 files in 3.812 seconds.
```

6.Import a given csv dataset from local file system into Cassandra column family



```
cqlsh:library> COPY Library_Info(Stud_id,Counter_value,Stud_Name,Book_Name,Book_id,doi) FROM 'C:\Users\lenovo\Desktop\BD
A\LAB\LAB 6\libraryInfo.csv';
Using 3 child processes
```

Develop a MapReduce program to count the number of occurrences of words in a given file.

To start all the Hadoop deamons

```
$ ssh localhost
$ cd Hadoop/hadoop-3.2.1
$ sbin/start-dfs.sh
$ sbin/start-yarn.sh
```

To create a directory in hdfs

```
$ hadoop fs -mkdir /rgs1
```

To view all the directories in hdfs

```
$ hadoop fs -ls /
```

To copy a file from local system to hdfs directory

```
$ Hadoop fs -copyFromLocal  /home/lenavo/Desktop/file1.txt  /rgs1/test.txt
```

To view all files in /rgs1 hdfs directory

```
$ hadoop fs -ls /rgs1
```

To run a MapReduce program

```
$ hadoop jar /home/lenavo/Desktop/wordcount.jar WordCount /rgs1/test.txt /rgs1/output
```

To view the output text

```
$ hadoop fs -cat /rgs1/output/part-r-00000
```

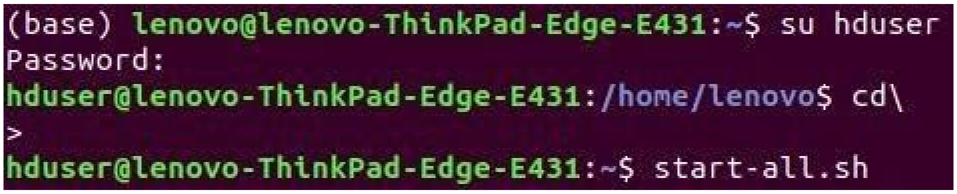
To stop all the Hadoop deamons

```
$ sbin/stop-yarn.sh
$ sbin/stop-dfs.sh
```

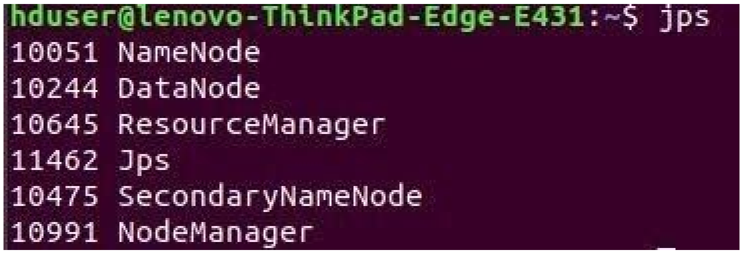
```
import java.io.IOException;
import java.util.StringTokenizer;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;
import org.apache.hadoop.mapreduce.Reducer;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.mapreduce.Job;
import org.apache.hadoop.mapreduce.lib.input.TextInputFormat;
import org.apache.hadoop.mapreduce.lib.output.TextOutputFormat;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;
import org.apache.hadoop.fs.Path;

public class WordCount{
    public static class Map extends
Mapper&lt;LongWritable,Text,Text,IntWritable&gt; {
        public void map(LongWritable key, Text value,Context context) throws
IOException,InterruptedException{
            String line = value.toString();
            StringTokenizer tokenizer = new StringTokenizer(line);
            while (tokenizer.hasMoreTokens()) {
                 value.set(tokenizer.nextToken());
                 context.write(value, new IntWritable(1));
            }
        }
    }
    public static class Reduce extends
Reducer&lt;Text,IntWritable,Text,IntWritable&gt; {
        public void reduce(Text key, Iterable&lt;IntWritable&gt; values,Context
context) throws IOException,InterruptedException {
           int sum=0;
           for(IntWritable x: values)
           {
               sum+=x.get();
           }
           context.write(key, new IntWritable(sum));
        }
    }
    public static void main(String[] args) throws Exception {
        Configuration conf= new Configuration();
        Job job = new Job(conf,"My Word Count Program");
        job.setJarByClass(WordCount.class);
        job.setMapperClass(Map.class);
        job.setReducerClass(Reduce.class);
        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(IntWritable.class);
        job.setInputFormatClass(TextInputFormat.class);
        job.setOutputFormatClass(TextOutputFormat.class);
        Path outputPath = new Path(args[1]);
        //Configuring the input/output path from the filesystem into the job
        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));
        //deleting the output path automatically from hdfs so that we don't
have to delete it explicitly
        outputPath.getFileSystem(conf).delete(outputPath);
        //exiting the job only if the flag value becomes false
        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

WORD COUNT PROGRAM

1. Starting Hadoop Cluster

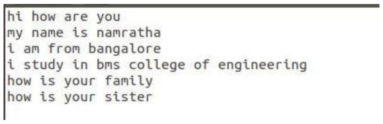


```
(base) lenovo@lenovo-ThinkPad-Edge-E431:~$ su hduser
Password:
hduser@lenovo-ThinkPad-Edge-E431:/home/lenovo$ cd\
>
hduser@lenovo-ThinkPad-Edge-E431:~$ start-all.sh
```



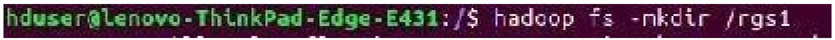
```
hduser@lenovo-ThinkPad-Edge-E431:~$ jps
10051 NameNode
10244 DataNode
10645 ResourceManager
11462 Jps
10475 SecondaryNameNode
10991 NodeManager
```

2. Creating a file to count words

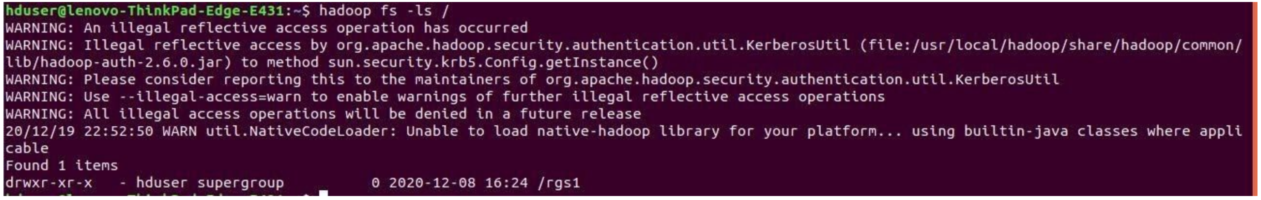


```
hi how are you
my name is namratha
i am from bangalore
i study in bms college of engineering
how is your family
how is your sister
```

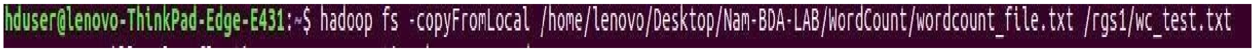
3.Moving file to Hadoop system



```
hduser@lenovo-ThinkPad-Edge-E431:/$ hadoop fs -mkdir /rgs1
```

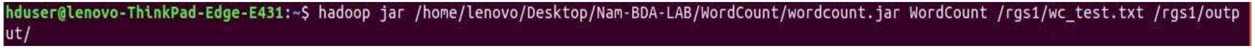


```
hduser@lenovo-ThinkPad-Edge-E431:~$ hadoop fs -ls /
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/
lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/12/19 22:52:50 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where appli
cable
Found 1 items
drwxr-xr-x   - hduser supergroup          0 2020-12-08 16:24 /rgs1
```



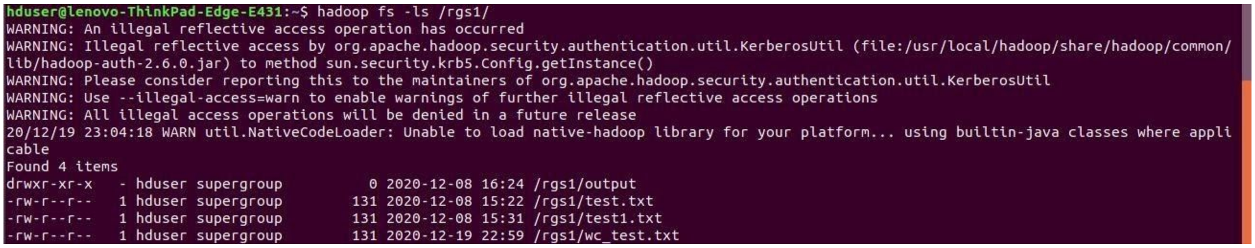
```
hduser@lenovo-ThinkPad-Edge-E431:~$ hadoop fs -copyFromLocal /home/lenovo/Desktop/Nam-BDA-LAB/WordCount/wordcount_file.txt /rgs1/wc_test.txt
```

4. Running the JAR file



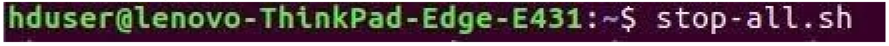
```
hduser@lenovo-ThinkPad-Edge-E431:~$ hadoop jar /home/lenovo/Desktop/Nam-BDA-LAB/WordCount/wordcount.jar WordCount /rgs1/wc_test.txt /rgs1/outp
ut/
```

5. Output



```
hduser@lenovo-ThinkPad-Edge-E431:~$ hadoop fs -ls /rgs1/
WARNING: An illegal reflective access operation has occurred
WARNING: Illegal reflective access by org.apache.hadoop.security.authentication.util.KerberosUtil (file:/usr/local/hadoop/share/hadoop/common/
lib/hadoop-auth-2.6.0.jar) to method sun.security.krb5.Config.getInstance()
WARNING: Please consider reporting this to the maintainers of org.apache.hadoop.security.authentication.util.KerberosUtil
WARNING: Use --illegal-access=warn to enable warnings of further illegal reflective access operations
WARNING: All illegal access operations will be denied in a future release
20/12/19 23:04:18 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where appli
cable
Found 4 items
drwxr-xr-x   - hduser supergroup          0 2020-12-08 16:24 /rgs1/output
-rw-r--r--   1 hduser supergroup        131 2020-12-08 15:22 /rgs1/test.txt
-rw-r--r--   1 hduser supergroup        131 2020-12-08 15:31 /rgs1/test1.txt
-rw-r--r--   1 hduser supergroup        131 2020-12-19 22:59 /rgs1/wc_test.txt
```

6. Stopping Hadoop



```
hduser@lenovo-ThinkPad-Edge-E431:~$ stop-all.sh
```

```
hi how are you

                  my name is rohith

                  i am from bangalore

                  i study in bms college of engineering

                  how is your family
```

For the given file, Create a Map Reduce program to Find the average temperature for each year from NCDC data set.

To create jar files using .java files

```
$ javac AverageReducer.java AverageDriver.java AverageMapper.java -cp $(hadoop classpath)
```

```
$ jar -cf Average.jar AverageReducer.class AverageDriver.class AverageMapper.class
```

To start all the Hadoop deamons

```
$ ssh localhost
```

```
$ cd Hadoop/hadoop-3.2.1
```

```
$ sbin/start-dfs.sh
```

```
$ sbin/start-yarn.sh
```

To create a directory in hdfs

```
$ hadoop fs -mkdir /rgs1
```

To view all the directories in hdfs

```
$ hadoop fs -ls /
```

To copy a file from local system to hdfs directory

```
$ Hadoop fs -copyFromLocal  /home/lenavo/Desktop/1901  /rgs1/AverageTest.txt
```

To view all files in /rgs1 hdfs directory

```
$ hadoop fs -ls /rgs1
```

To run a MapReduce program

```
$ hadoop jar /home/lenavo/Desktop/Average.jar AverageDriver  /rgs1/AverageTest.txt
/rgs1/AverageOutput
```

To view the output text

```
$ hadoop fs -cat /rgs1/output/part-r-00000
```

To stop all the Hadoop deamons

```
$ sbin/stop-yarn.sh

$ sbin/stop-dfs.sh
```

AverageDriver.java

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.fs.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class AverageDriver
{

        public static void main (String[] args) throws Exception
        {
                if (args.length != 2)
                {
                     System.err.println("Please Enter the input and output parameters");
                     System.exit(-1);
                }

               Job job = new Job();
               job.setJarByClass(AverageDriver.class);
               job.setJobName("Max temperature");

               FileInputFormat.addInputPath(job,new Path(args[0]));
               FileOutputFormat.setOutputPath(job,new Path (args[1]));

               job.setMapperClass(AverageMapper.class);
               job.setReducerClass(AverageReducer.class);

               job.setOutputKeyClass(Text.class);
               job.setOutputValueClass(IntWritable.class);

               System.exit(job.waitForCompletion(true)?0:1);
        }
}
```

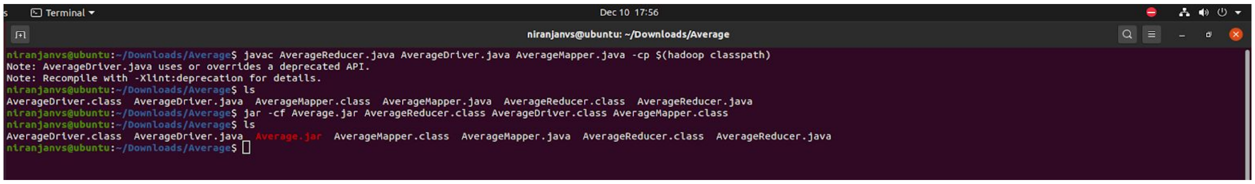
AverageMapper.java

```
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageMapper extends Mapper <LongWritable, Text, Text, IntWritable>
{

public static final int MISSING = 9999;

public void map(LongWritable key, Text value, Context context) throws
IOException,  InterruptedException
        {
                String line = value.toString();
                String year = line.substring(15,19);
                int temperature;
                if (line.charAt(87)=='+')
                        temperature = Integer.parseInt(line.substring(88, 92));
                else
                        temperature = Integer.parseInt(line.substring(87, 92));

                String quality = line.substring(92, 93);
                if(temperature != MISSING && quality.matches("[01459]"))
                context.write(new Text(year),new IntWritable(temperature));
        }
}
```

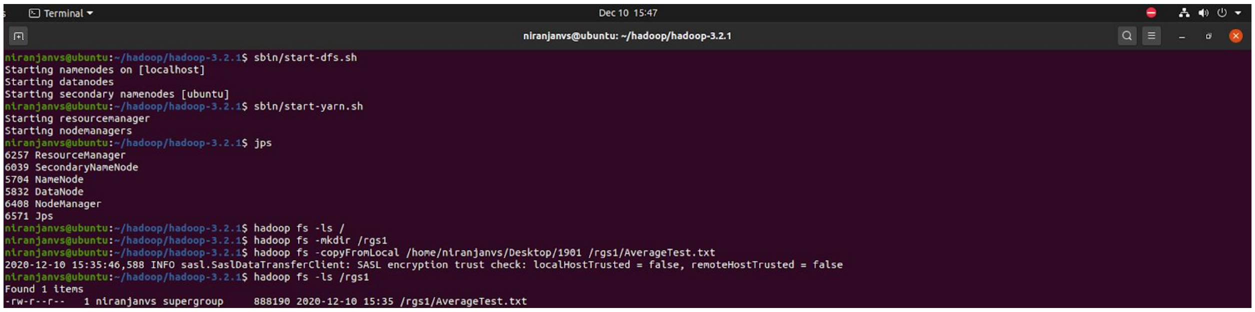
AverageReducer.java

```
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.*;
import java.io.IOException;

public class AverageReducer extends Reducer <Text, IntWritable,Text, IntWritable >
 {
public void reduce(Text key,  Iterable<IntWritable> values, Context context) throws IOException,
InterruptedException
        {
        int max_temp = 0;
        int count = 0;
        for (IntWritable value : values)
                {
                        max_temp += value.get();
                        count+=1;
```

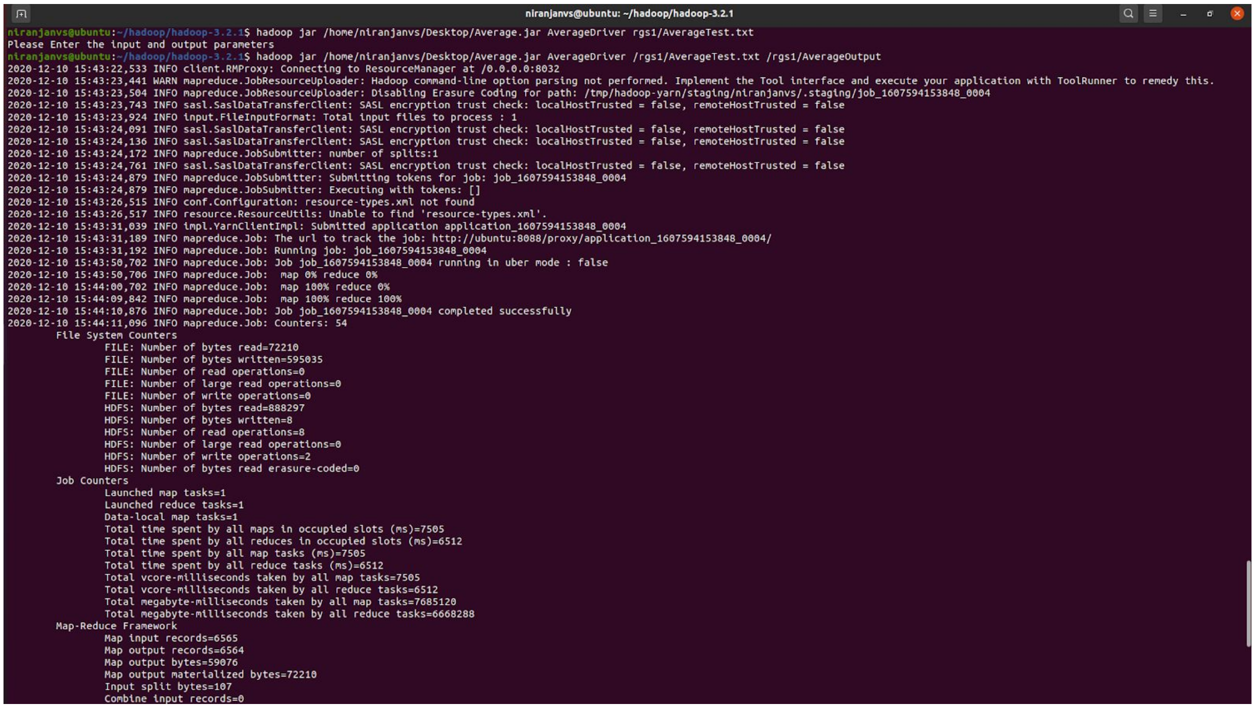
```
                    }
            context.write(key, new IntWritable(max_temp/count));
            }
    }
```



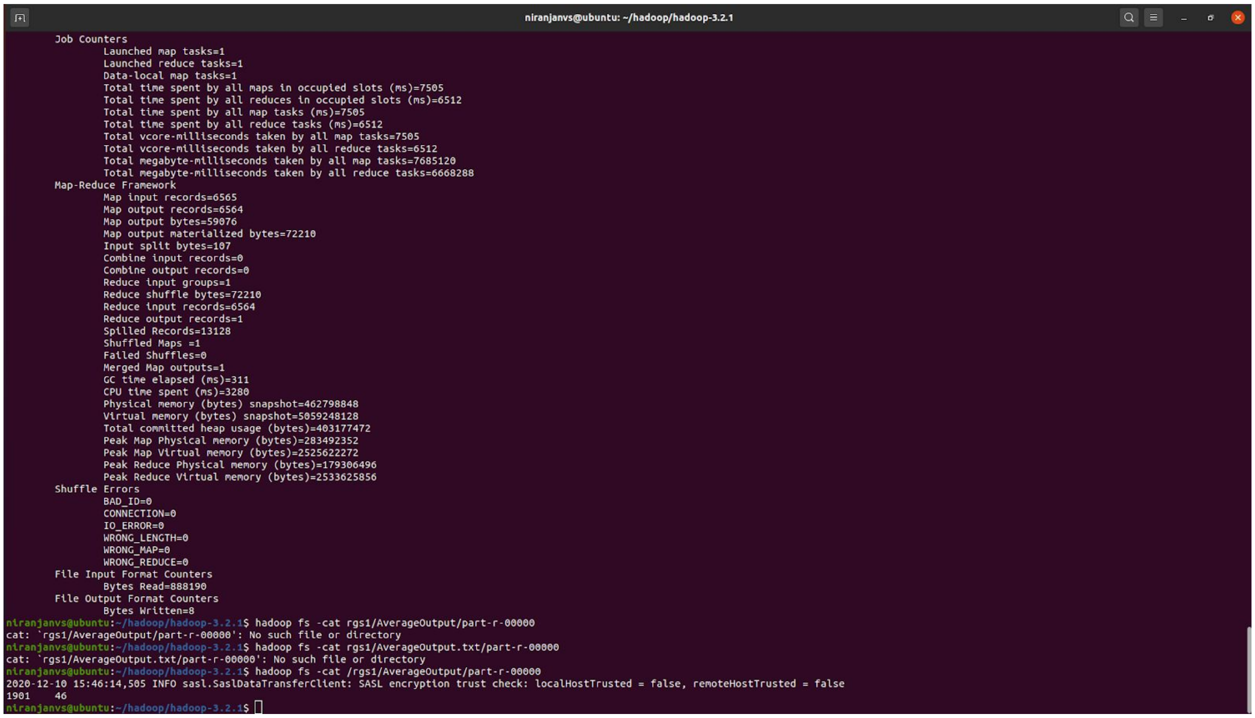
```
niranjanvs@ubuntu:~/Downloads/Average$ javac AverageReducer.java AverageDriver.java AverageMapper.java -cp $(hadoop classpath)
Note: AverageDriver.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
niranjanvs@ubuntu:~/Downloads/Average$ ls
AverageDriver.class  AverageDriver.java  AverageMapper.class  AverageMapper.java  AverageReducer.class  AverageReducer.java
niranjanvs@ubuntu:~/Downloads/Average$ jar -cf Average.jar AverageReducer.class AverageDriver.class AverageMapper.class
niranjanvs@ubuntu:~/Downloads/Average$ ls
AverageDriver.class  AverageDriver.java  Average.jar  AverageMapper.class  AverageMapper.java  AverageReducer.class  AverageReducer.java
niranjanvs@ubuntu:~/Downloads/Average$ 
```



```
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-dfs.sh
Starting namenodes on [localhost]
Starting datanodes
Starting secondary namenodes [ubuntu]
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ sbin/start-yarn.sh
Starting resourcemanager
Starting nodemanagers
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ jps
6257 ResourceManager
6039 SecondaryNameNode
5704 NameNode
5832 DataNode
6408 NodeManager
6571 Jps
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -mkdir /rgs1
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -copyFromLocal /home/niranjanvs/Desktop/1901 /rgs1/AverageTest.txt
2020-12-10 15:35:46,588 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -ls /rgs1
Found 1 items
-rw-r--r--   1 niranjanvs supergroup     888190 2020-12-10 15:35 /rgs1/AverageTest.txt
```



```
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop jar /home/niranjanvs/Desktop/Average.jar AverageDriver rgs1/AverageTest.txt
Please Enter the input and output parameters
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop jar /home/niranjanvs/Desktop/Average.jar AverageDriver /rgs1/AverageTest.txt /rgs1/AverageOutput
2020-12-10 15:43:22,533 INFO client.RMProxy: Connecting to ResourceManager at /0.0.0.0:8032
2020-12-10 15:43:23,441 WARN mapreduce.JobResourceUploader: Hadoop command-line option parsing not performed. Implement the Tool interface and execute your application with ToolRunner to remedy this.
2020-12-10 15:43:23,504 INFO mapreduce.JobResourceUploader: Disabling Erasure Coding for path: /tmp/hadoop-yarn/staging/niranjanvs/.staging/job_1607594153848_0004
2020-12-10 15:43:23,743 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:23,924 INFO input.FileInputFormat: Total input files to process : 1
2020-12-10 15:43:24,091 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,136 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,172 INFO mapreduce.JobSubmitter: number of splits:1
2020-12-10 15:43:24,761 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
2020-12-10 15:43:24,879 INFO mapreduce.JobSubmitter: Submitting tokens for job: job_1607594153848_0004
2020-12-10 15:43:24,879 INFO mapreduce.JobSubmitter: Executing with tokens: []
2020-12-10 15:43:26,515 INFO conf.Configuration: resource-types.xml not found
2020-12-10 15:43:26,517 INFO resource.ResourceUtils: Unable to find 'resource-types.xml'.
2020-12-10 15:43:31,039 INFO impl.YarnClientImpl: Submitted application application_1607594153848_0004
2020-12-10 15:43:31,189 INFO mapreduce.Job: The url to track the job: http://ubuntu:8088/proxy/application_1607594153848_0004/
2020-12-10 15:43:31,192 INFO mapreduce.Job: Running job: job_1607594153848_0004
2020-12-10 15:43:50,702 INFO mapreduce.Job: Job job_1607594153848_0004 running in uber mode : false
2020-12-10 15:43:50,706 INFO mapreduce.Job:  map 0% reduce 0%
2020-12-10 15:44:00,702 INFO mapreduce.Job:  map 100% reduce 0%
2020-12-10 15:44:09,842 INFO mapreduce.Job:  map 100% reduce 100%
2020-12-10 15:44:10,876 INFO mapreduce.Job: Job job_1607594153848_0004 completed successfully
2020-12-10 15:44:11,096 INFO mapreduce.Job: Counters: 54
        File System Counters
                FILE: Number of bytes read=72210
                FILE: Number of bytes written=595035
                FILE: Number of read operations=0
                FILE: Number of large read operations=0
                FILE: Number of write operations=0
                HDFS: Number of bytes read=888297
                HDFS: Number of bytes written=8
                HDFS: Number of read operations=8
                HDFS: Number of large read operations=0
                HDFS: Number of write operations=2
                HDFS: Number of bytes read erasure-coded=0
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7505
                Total time spent by all reduces in occupied slots (ms)=6512
                Total time spent by all map tasks (ms)=7505
                Total time spent by all reduce tasks (ms)=6512
                Total vcore-milliseconds taken by all map tasks=7505
                Total vcore-milliseconds taken by all reduce tasks=6512
                Total megabyte-milliseconds taken by all map tasks=7685120
                Total megabyte-milliseconds taken by all reduce tasks=6668288
        Map-Reduce Framework
                Map input records=6565
                Map output records=6564
                Map output bytes=59076
                Map output materialized bytes=72210
                Input split bytes=107
                Combine input records=0
```



```
        Job Counters
                Launched map tasks=1
                Launched reduce tasks=1
                Data-local map tasks=1
                Total time spent by all maps in occupied slots (ms)=7505
                Total time spent by all reduces in occupied slots (ms)=6512
                Total time spent by all map tasks (ms)=7505
                Total time spent by all reduce tasks (ms)=6512
                Total vcore-milliseconds taken by all map tasks=7505
                Total vcore-milliseconds taken by all reduce tasks=6512
                Total megabyte-milliseconds taken by all map tasks=7685120
                Total megabyte-milliseconds taken by all reduce tasks=6668288
        Map-Reduce Framework
                Map input records=6565
                Map output records=6564
                Map output bytes=59076
                Map output materialized bytes=72210
                Input split bytes=107
                Combine input records=0
                Combine output records=0
                Reduce input groups=1
                Reduce shuffle bytes=72210
                Reduce input records=6564
                Reduce output records=1
                Spilled Records=13128
                Shuffled Maps =1
                Failed Shuffles=0
                Merged Map outputs=1
                GC time elapsed (ms)=311
                CPU time spent (ms)=3280
                Physical memory (bytes) snapshot=462798848
                Virtual memory (bytes) snapshot=5059248128
                Total committed heap usage (bytes)=403177472
                Peak Map Physical memory (bytes)=283492352
                Peak Map Virtual memory (bytes)=2525622272
                Peak Reduce Physical memory (bytes)=179306496
                Peak Reduce Virtual memory (bytes)=2533625856
        Shuffle Errors
                BAD_ID=0
                CONNECTION=0
                IO_ERROR=0
                WRONG_LENGTH=0
                WRONG_MAP=0
                WRONG_REDUCE=0
        File Input Format Counters
                Bytes Read=888190
        File Output Format Counters
                Bytes Written=8
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat rgs1/AverageOutput/part-r-00000
cat: `rgs1/AverageOutput/part-r-00000': No such file or directory
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat rgs1/AverageOutput.txt/part-r-00000
cat: `rgs1/AverageOutput.txt/part-r-00000': No such file or directory
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ hadoop fs -cat /rgs1/AverageOutput/part-r-00000
2020-12-10 15:46:14,505 INFO sasl.SaslDataTransferClient: SASL encryption trust check: localHostTrusted = false, remoteHostTrusted = false
1901    46
niranjanvs@ubuntu:~/hadoop/hadoop-3.2.1$ 
```