

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY**  
“JnanaSangama”, Belgaum -590014, Karnataka.



**LAB REPORT**  
**on**  
**BIG DATA ANALYTICS**  
**(20CS6PEBDA)**

*Submitted by*

**NAME (1BM18CS111)**

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



**B.M.S. COLLEGE OF ENGINEERING**  
(Autonomous Institution under VTU)  
**BENGALURU-560019**  
**May-2022 to July-2022**

**B. M. S. College of Engineering,  
Bull Temple Road, Bangalore 560019**  
(Affiliated To Visvesvaraya Technological University, Belgaum)  
**Department of Computer Science and Engineering**



**CERTIFICATE**

This is to certify that the Lab work entitled “**BIG DATA ANALYTICS**” carried out by **MANJULA (1BM20CS408)**, who is bonafide student of **B. M. S. College of Engineering**. It is in partial fulfillment for the award of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belgaum during the year 2022. The Lab report has been approved as it satisfies the academic requirements in respect of a **Course Title - (Course code)** work prescribed for the said degree.

Ms. Antara Roy Choudhury  
Assistant Professor  
Department of CSE  
BMSCE, Bengaluru

**Dr. Jyothi S Nayak**  
Professor and Head  
Department of CSE  
BMSCE, Bengaluru

## **Index Sheet**

## Course Outcome

|     |  |
|-----|--|
| CO1 | Apply the concept of NoSQL, Hadoop or Spark for a given task             |
| CO2 | Analyze the Big Data and obtain insight using data analytics mechanisms. |
| CO3 | Design and implement Big data applications by applying NoSQL, Hadoop or  |

|  |       |
|--|-------|
|  | Spark |
|--|-------|

**BIG DATA  
ANALYTICS :  
WEEK1  
ASSIGNMENT**

**NAME:** MANJULA

**USN:** 1BM20CS408

```
bmsce@bmsce-OptiPlex-3060:~$ mongo
MongoDB shell version v4.0.28
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("c2e3109b-0341-483b-ba3a-f9fb3b1aed87") }
MongoDB server version: 4.0.28
Server has startup warnings:
2022-04-11T14:03:08.254+0530 I STORAGE  [initandlisten]
2022-04-11T14:03:08.254+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-11T14:03:08.254+0530 I STORAGE  [initandlisten] **             See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-11T14:03:10.024+0530 I CONTROL  [initandlisten]
2022-04-11T14:03:10.024+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-11T14:03:10.024+0530 I CONTROL  [initandlisten] **             Read and write access to data and configuration is unrestricted.
2022-04-11T14:03:10.024+0530 I CONTROL  [initandlisten]
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> db;
test
> use lab1DB;
switched to db lab1DB
> db;
lab1DB
```

```
> show dbs;
admin 0.000GB
config 0.000GB
local 0.000GB
myDB 0.000GB
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.drop();
true
> db.getCollectionNames()
[ ]
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1, StudName:"Jeevan", Grade:"VI", Hobbies:"InternetSurfing"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2, StudName:"Vamsi", Grade:"VI", Hobbies:["Watching Movies", "Reading Novels", "Drugs"]});
WriteResult({ "nInserted" : 1 })
> db.Student.find({});
{ "_id" : 1, "StudName" : "Jeevan", "Grade" : "VI", "Hobbies" : "InternetSurfing" }
{ "_id" : 2, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : [ "Watching Movies", "Reading Novels", "Drugs" ] }
```

```

> db.Student.find({}).pretty();
{
    "_id" : 1,
    "StudName" : "Jeevan",
    "Grade" : "VI",
    "Hobbies" : "InternetSurfing"
}
[
    "_id" : 2,
    "StudName" : "Vamsi",
    "Grade" : "VI",
    "Hobbies" : [
        "Watching Movies",
        "Reading Novels",
        "Drugs"
    ]
}
> db.Student.insert({_id:3, StudName:"Sharat", Grade:"V", Hobbies:"Reading"});
WriteResult({ "nInserted" : 1 })
> db.Student.find({});
{
    "_id" : 1, "StudName" : "Jeevan", "Grade" : "VI", "Hobbies" : "InternetSurfing" }
{
    "_id" : 2, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : [ "Watching Movies", "Reading Novels", "Drugs" ] }
{
    "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }

```

```

> db.Student.update({_id:5, StudName:"Kusum", Grade:"VI"}, {$set:{Hobbies: ["Golf", "Sea Shell Collection"]}}, {upsert:true});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 5 })
> db.Student.find({}).pretty();
{
    "_id" : 1,
    "StudName" : "Jeevan",
    "Grade" : "VI",
    "Hobbies" : "InternetSurfing"
}
[
    "_id" : 2,
    "StudName" : "Vamsi",
    "Grade" : "VI",
    "Hobbies" : [
        "Watching Movies",
        "Reading Novels",
        "Drugs"
    ]
}
{
    "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }
{
    "_id" : 5,
    "Grade" : "VI",
    "StudName" : "Kusum",
    "Hobbies" : [
        "Golf",
        "Sea Shell Collection"
    ]
}
> db.Student.find({StudName:"Kusum"});
{
    "_id" : 5, "Grade" : "VI", "StudName" : "Kusum", "Hobbies" : [ "Golf", "Sea Shell Collection" ] }

```

```

> db.Student.find({StudName:"Kusum"}, {StudName:1, Grade:1}).pretty();
{ "_id" : 5, "Grade" : "VI", "StudName" : "Kusum" }
> db.Student.count();
4
> db.Student.find({}).sort({StudName:1});
{ "_id" : 1, "StudName" : "Jeevan", "Grade" : "VI", "Hobbies" : "InternetSurfing" }
{ "_id" : 5, "Grade" : "VI", "StudName" : "Kusum", "Hobbies" : [ "Golf", "Sea Shell Collection" ] }
{ "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }
{ "_id" : 2, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : [ "Watching Movies", "Reading Novels", "Drugs" ] }
> db.Student.find({}).sort({StudName:-1});
{ "_id" : 2, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : [ "Watching Movies", "Reading Novels", "Drugs" ] }
{ "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }
{ "_id" : 5, "Grade" : "VI", "StudName" : "Kusum", "Hobbies" : [ "Golf", "Sea Shell Collection" ] }
{ "_id" : 1, "StudName" : "Jeevan", "Grade" : "VI", "Hobbies" : "InternetSurfing" }
> db.Student.update({_id:5}, {$set:{Location:'NIHMANS'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find({StudName:"Kusum"}).pretty();
{
    "_id" : 5,
    "Grade" : "VI",
    "StudName" : "Kusum",
    "Hobbies" : [
        "Golf",
        "Sea Shell Collection"
    ],
    "Location" : "NIHMANS"
}

```

```

> db.Student.find({}).skip(2).pretty();
{ "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }
{
    "_id" : 5,
    "Grade" : "VI",
    "StudName" : "Kusum",
    "Hobbies" : [
        "Golf",
        "Sea Shell Collection"
    ],
    "Location" : "NIHMANS"
}
> db.Student.find().skip(2).pretty();
{ "_id" : 3, "StudName" : "Sharat", "Grade" : "V", "Hobbies" : "Reading" }
{
    "_id" : 5,
    "Grade" : "VI",
    "StudName" : "Kusum",
    "Hobbies" : [
        "Golf",
        "Sea Shell Collection"
    ],
    "Location" : "NIHMANS"
}
> db.createCollection("food")
{ "ok" : 1 }
> db.food.insert({_id:1,fruits:['avacado','dragon fruit']})
WriteResult({ "nInserted" : 1 })

```

```

> db.food.insert({_id:1,fruits:['avacado','dragon fruit']})
WriteResult({ "nInserted" : 1 })
> db.food.insert({_id:2,fruits:['strawberry','dragon fruit']})
WriteResult({ "nInserted" : 1 })
> db.food.find({'fruits.1':'avacado'}).pretty()
> db.food.find().pretty()
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
{ "_id" : 2, "fruits" : [ "strawberry", "dragon fruit" ] }
> db.food.find({'fruits.1':'avacado'}).pretty()
> db.food.find({'fruits.1':'avacado'})
> db.food.find({'fruits.0':'avacado'})
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
> db.food.find({'fruits.0':'avacado'}).pretty()
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
> db.food.find({'fruits.0':'avacado'}).pretty();
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
> db.food.find({'fruits.0':{$size:2}}).pretty();
> db.food.find({'fruits':{$size:2}})
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
{ "_id" : 2, "fruits" : [ "strawberry", "dragon fruit" ] }
> db.food.find({_id:2},{'fruits':{$slice:2}});
{ "_id" : 2, "fruits" : [ "strawberry", "dragon fruit" ] }
> db.food.find({_id:2},{'fruits':{$slice:1}});
{ "_id" : 2, "fruits" : [ "strawberry" ] }
> db.food.find({fruits:{$all:["avacado"]}})
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
> db.food.find({fruits:{$all:["avacado","dragon fruit"]}})
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
> db.food.find({fruits:{$all:["dragon fruit"]}})
{ "_id" : 1, "fruits" : [ "avacado", "dragon fruit" ] }
{ "_id" : 2, "fruits" : [ "strawberry", "dragon fruit" ] }

```

```

> show collections;
Student
customer
food
> db.customer.aggregate({$match:{AcctType:"FD"}},{$group:{_id:"$custID",TotalAccBal:{$sum:"$AcctBal"}}})
{ "_id" : 2, "TotalAccBal" : 20000000 }
{ "_id" : 1, "TotalAccBal" : 10000000 }
> db.customer.find()
{ "_id" : ObjectId("6253f945d7ce1043c6d5c8cc"), "custID" : 1, "AcctBal" : 10000000, "AcctType" : "FD" }
{ "_id" : ObjectId("6253f963d7ce1043c6d5c8cd"), "custID" : 2, "AcctBal" : 20000000, "AcctType" : "FD" }
{ "_id" : ObjectId("6253f973d7ce1043c6d5c8ce"), "custID" : 3, "AcctBal" : 30000000, "AcctType" : "RD" }
> db.customer.aggregate({$match:{AcctType:"FD"}},{$group:{_id:"$custID",TotalAccBal:{$sum:"$AcctBal"}},{$match:{TotAccBal:{$gt:10000000}}}})
> db.customer.aggregate({$match:{AcctType:"FD"}},{$group:{_id:"$custID",TotalAccBal:{$sum:"$AcctBal"}},{$match:{TotalAccBal:{$gt:10000000}}}});
{ "_id" : 2, "TotalAccBal" : 20000000 }
> quit()

```

```
Q bmsce@bmsce-Precision-T1700:~ F+ | ≡ | - | o | X
bye
bmsce@bmsce-Precision-T1700:~$ mongoexport --host localhost --db Karthik --collection Student --csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies";
2022-04-18T15:11:00.509+0530      csv flag is deprecated; please use --type=csv instead
2022-04-18T15:11:00.511+0530      connected to: localhost
2022-04-18T15:11:00.511+0530      exported 1 record
bmsce@bmsce-Precision-T1700:~$ pwd
/home/bmsce
bmsce@bmsce-Precision-T1700:~$ mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
2022-04-18T15:16:31.097+0530      connected to: localhost
2022-04-18T15:16:31.098+0530      exported 1 record
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
[sudo] password for bmsce:
2022-04-18T15:19:13.567+0530      connected to: localhost
2022-04-18T15:19:13.567+0530      exported 1 record
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("55fb0928-57d3-4119-a529-34f8a2619c30") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten]
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten] **           See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten]
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten] **           Read and write access to data and configuration is unrestricted.
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten]
> use Karthik
switched to db Karthik
> db.Student.find({});
{ "_id" : 1, "studName" : "John", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
> ^C
bye
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection student --type=csv --out /home/bmsce/Desktop/output
```

```
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("073d51e7-4bf1-4b13-88e8-b10f6ed9ee36") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten]
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] **           See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
> use Karthik
switched to db Karthik
> show dbs;
admin      0.000GB
config     0.000GB
firstDB    0.000GB
local      0.000GB
myDB       0.000GB
newDB      0.000GB
saffanDB   0.000GB
test       0.000GB
> db.createCollection("Student");
2022-04-18T14:06:18.487+0530 E QUERY    [thread1] SyntaxError: illegal character @shell):1:20
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"Internetsurfing"});
2022-04-18T14:08:15.694+0530 E QUERY    [thread1] SyntaxError: expected expression, got '&' @shell):1:34
> db.student.insert({_id:3,StudName:"Karthik",Usn:"1BM19CS070",Semester:"6thSem",Dept_Name:"CSE",CGPA:9,Hobbies:"YouTuber"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"Internetsurfing"});
[1]+  Stopped                  mongo
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
```

```
bmsce@bmsce-Precision-T1700:~$ 
2022-04-18T15:11:00.511+0530  exported 1 record
bmsce@bmsce-Precision-T1700:~$ pwd
/home/bmsce
bmsce@bmsce-Precision-T1700:~$ mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
2022-04-18T15:16:31.097+0530  connected to: localhost
2022-04-18T15:16:31.098+0530  exported 1 record
bmsce@bmsce-Precision-T1700:~$ 
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
[sudo] password for bmsce:
2022-04-18T15:19:13.567+0530  connected to: localhost
2022-04-18T15:19:13.567+0530  exported 1 record
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("55fb0928-57d3-4119-a529-34f8a2619c30") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten]
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] **           See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] **           Read and write access to data and configuration is unrestricted.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
> use Karthik
switched to db Karthik
> db.Student.find();
{ "_id" : 1, "studName" : "John", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
> ^C
bye
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
2022-04-18T15:21:30.067+0530  connected to: localhost
2022-04-18T15:21:30.067+0530  exported 4 records
bmsce@bmsce-Precision-T1700:~$ 
```

```

Q bmsce@bmsce-Precision-T1700:~ + - x
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("dd4034b4-efba-4547-a66c-b4c5cfa8c452") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten]
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T13:57:39.395+0530 I STORAGE [initandlisten] **          see http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-04-18T13:57:48.671+0530 I CONTROL [initandlisten]
> use Karthik
switched to db Karthik
> db.Student.insert({_id:1,studName:"John",Grade:"VII",Hobbies:"InternetSurfing"});
WriteResult({ "nInserted" : 1 })
> db.student.insert({_id:3,StudName:"Karthik",Usn:"1BM19CS070",Semester:"6thSem",Dept_Name:"CSE",CGPA:9,Hobbies:"YouTuber"});
WriteResult({
    "nInserted" : 0,
    "writeError" : {
        "code" : 11000,
        "errmsg" : "E11000 duplicate key error collection: Karthik.student index: _id_ dup key: { : 3.0 }"
    }
})
> db.student.insert({_id:3,StudName:"Karthik",Usn:"1BM19CS070",Semester:"6thSem",Dept_Name:"CSE",CGPA:9,Hobbies:"YouTuber"});
WriteResult({
    "nInserted" : 0,
    "writeError" : {
        "errmsg" : "db.CUSTOMERS.aggregate ( { $match:{AcctType:"S"}},{$group : { _id : "$custID",TotAccBal : ... {$sum:"$AccBal"} } }, {$match:{TotAccBal:[$gt:1200]} })"
    }
})
2022-04-18T14:35:57.967+0530 E QUERY    [thread1] SyntaxError: illegal character @shell):1:43
> db.student.aggregate([{$match:{Dept_Name:"CSE"}},{$group:{_id:"$Semester",AvgCGPA:{$avg:"$CGPA"} }},{$match:{AvgCGPA:[$gt:7.5]} }]);
{ "_id" : "5thSem", "AvgCGPA" : 9.12 }
{ "_id" : "6thSem", "AvgCGPA" : 9 }
> mongoexport --host localhost --db Student --collection airlines --csv --out /home/hduser/Desktop/outout.txt --fields "Year","Quarter"
2022-04-18T15:04:00.798+0530 E QUERY    [thread1] SyntaxError: missing ; before statement @shell):1:14
> /home/hduser/Desktop/outout.txt -fields "Year","Quarter"

{
  "pname" : "motherboard"
}
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
> db.product.find({$and:[{quantity:{$eq:9}}, {pname:{$eq:"monitor"} }]}),{pname:1,_id:0})
{ "pname" : "monitor" }
> db.product.find({pname:/d$/},{pname:1,quantity:1,_id:0})
{ "pname" : "keyboard", "quantity" : 2 }
{ "pname" : "motherboard", "quantity" : 4 }
> db.createCollection("hospital");
{ "ok" : 1 }
> db.hospital.insert({_id:1, Name: "Anshuman Agarwal", age:23, diseases:["fever", "diarrhoea", "wheezing", "gastritis"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:2, Name: "Pinky Chaubey", age:35, diseases:["fever", "nausea", "food infection", "indigestion", "kidney stones"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.insert({_id:3, Name: "Amresh Chowpati", age:63, diseases:["hyperglycemia", "diabetes mellitus", "food poisoning", "cold"]});
WriteResult({ "nInserted" : 1 })
> db.hospital.updateMany([],{$pull:{diseases:"fever"}});
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 2 }
> db.hospital.updateOne({_id:1},{$pop:{diseases:-1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.hospital.find({"diseases.2":"nausea"});
> db.hospital.find({"diseases.1":"nausea"});
> d.hospital.find();
uncaught exception: ReferenceError: d is not defined :
@shell):1:1
> db.hospital.find();
{ "_id" : 1, "Name" : "Anshuman Agarwal", "age" : 23, "diseases" : [ "wheezing", "gastritis" ] }
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] }
{ "_id" : 3, "Name" : "Amresh Chowpati", "age" : 63, "diseases" : [ "hyperglycemia", "diabetes mellitus", "food poisoning", "cold" ] }
> db.hospital.find({"diseases.0":"nausea"});
{ "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] }
> db.hospital.update({_id:3},{$set:{'diseases.1':'sarscov'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> 

```

```

}
> db.createCollection("faculty");
{ "ok" : 1 }
> db.faculty.insert({_id:1,name:"Dr. Balaraman Ravindran",designation:"Professor",department:"CSE",age:45,salary:100000,specialization:['python','mysql','sklearn', 'tensorflow']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:2,name:"Dr. Mahadev Ghorki",designation:"Assistant Professor",department:"CSE",age:35,salary:80000,specialization:['python','numpy','sklearn', 'tensorflow', 'java']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:3,name:"Dr. Praveen Borade",designation:"Associate Professor",department:"ME",age:40,salary:75000,specialization:['autocad', 'aerodynamics', 'thermal physics']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:4,name:"Dr. Madhav Nayak",designation:"Assistant Professor",department:"ME",age:37,salary:95000,specialization:['autocad', 'flight-dynamics', 'Finite Element Analysis']});
WriteResult({ "nInserted" : 1 })
> db.faculty.aggregate ( {$match:{department:"ME"}}, { $group : {_id : "$designation", AverageSal :{$avg:"$salary"} } }, { $match:{AverageSal:{$gt:50000}} });
{ "_id" : "Associate Professor", "AverageSal" : 75000 }
{ "_id" : "Assistant Professor", "AverageSal" : 95000 }
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert({pid:1,pname:"keyboard",mdate:2001,price:1800,quantity:2});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:2,pname:"mouse",mdate:2005,price:1500,quantity:5});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:3,pname:"monitor",mdate:2015,price:10000,quantity:9});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:4,pname:"motherboard",mdate:2021,price:15000,quantity:4});
WriteResult({ "nInserted" : 1 })
> db.product.find({}, {pname:1,_id:0})
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
{ "pname" : "motherboard" }
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$ne:15000}}, {pname:1,_id:0});
{ "pname" : "keyboard" }

```

```

@shell:1:20
> db.Bank.update({_id:625d78659329139694f188a6}, {$set: {CustID:5}}, {upsert:true});
uncaught exception: SyntaxError: identifier starts immediately after numeric literal :
@shell:1:20
> db.Bank.update({_id:"625d78659329139694f188a6"}, {$set: {CustID:5}}, {upsert:true});
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : "625d78659329139694f188a6"
})
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] }
{ "_id" : "625d78659329139694f188a6", "CustID" : 5 }
> db.Bank.update({_id:"625d78659329139694f188a6", CustID:5}, {$set: {Name:"Sumantha K S", Type:"Savings", Contact:["9856321478", "011-65897458"]}}, {upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.find();
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] }
{ "_id" : "625d78659329139694f188a6", "CustID" : 5, "Contact" : [ "9856321478", "011-65897458" ], "Name" : "Sumantha K S", "Type" : "Savings" }

```

```

> db.createCollection("Bank");
{ "ok" : 1 }
> db.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
uncaught exception: TypeError: db.insert is not a function
@shell):1:1
> db.Bank.insert({CustID:1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:2, Name:"Vishvesh Bhat", Type:"Savings", Contact:["6325985615", "080-23651452"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:3, Name:"Vaishak Bhat", Type:"Savings", Contact:["8971456321", "080-33529458"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Pramod P Parande", Type:"Current", Contact:["9745236589", "080-56324587"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({CustID:4, Name:"Shreyas R S", Type:"Current", Contact:["9445678321", "044-65611729", "080-25639856"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231", "080-22364587" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.updateMany({CustID:1},{$pop:{Contact:1}});
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.updateMany({CustID:4},{$pull:{Contact:"080-25639856"}});
{ "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 1 }
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.createIndex({Name:1, Type:1},{name:"Find current account holders"});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] }
> db.Bank.getIndexes()
[ {
  "v" : 2,
  "key": {
    "Name": 1,
    "Type": 1
  },
  "ns": "Bank"
} ]

```

# Manjula

1BM20CS408

```
switched to db Bank
use Students
switched to db Students
> db.createCollection("Bank");
> db.Bank.insert({_Bank_Id:01,BankName:"Canara",Location:"Aldur"})
WriteResult({"_id": 1 })
> db.Bank.insert({_Bank_Id:02,BankName:"Vljaya",Location:"Delhi"})
WriteResult({"_id": 2 })
> db.Bank.insert({_Bank_Id:03,BankName:"Alhabad",Location:"Hanumanthnagar"})
WriteResult({"_id": 3 })
> db.Bank.update({_Bank_Id:01},{$push:{Branch:(Mudigere,BALEHAANORU)})}
2022-04-20T15:06:12.976+0530 E QUERY [threadId:1] ReferenceError: Mudigere is not defined :
> db.Bank.update({_Bank_Id:01},{$push:{Branch:(Mudigere,"BALEHAANORU")}})
2022-04-20T15:06:12.976+0530 E QUERY [threadId:1] syntaxError: missing : after property id @($shell):1:55
> db.Bank.update({_Bank_Id:01},{$push:{Branch:(canara,"BALEHAANORU")}})
WriteResult({"_id": 1,"$pushed": 1,"$modified": 1 })
> db.Bank.update({_Bank_Id:01},{$pop:{Branch:(canara,"BALEHAANORU"),vljaya:"Gulanpet"}})
WriteResult({"_id": 1,"$popped": 1,"$modified": 1 })
{
  "_id": 1,
  "Branch": [
    {
      "Branch": "(canara, \"BALEHAANORU\")",
      "Name": "Vljaya"
    },
    {
      "Branch": "(canara, \"BALEHAANORU\")",
      "Name": "Gulanpet"
    }
  ],
  "Location": "Delhi",
  "Name": "Vljaya"
}
> db.Bank.update({_Bank_Id:01},{$pull:{Branch:(canara,"BALEHAANORU")}})
WriteResult({"_id": 1,"$pulled": 1,"$modified": 1 })
> db.Bank.find()
[{"_id": ObjectId("625fd1aaaf3e3ae2dia8767"), "_Bank_Id": 1, "BankName": "Canara", "Location": "Aldur", "Branch": []},
 {"_id": ObjectId("625fd1aaaf3e3ae2dia8768"), "_Bank_Id": 2, "BankName": "Vljaya", "Location": "Delhi", "Branch": []},
 {"_id": ObjectId("625fd1aaaf3e3ae2dia877"), "_Bank_Id": 3, "BankName": "Alhabad", "Location": "Hanumanthnagar"}]
db.Bank.find();
[{"_id": ObjectId("625fd1aaaf3e3ae2dia875"),
  "_Bank_Id": 1,
  "BankName": "Canara",
  "Location": "Aldur",
  "Branch": []}
]
use Students
switched to db Students
> db.students.find({_id:1})
[{"_id": 1, "Name": "Rafa", "Age": 20, "Address": "Kashmir"}, {"_id": 2, "Name": "Kafa", "Age": 21, "Address": "Kashmir"}, {"_id": 3, "Name": "Rafa", "Age": 22, "Address": "Kashmir"}]
> db.students.find({_id:1})
[{"_id": 1, "Name": "Rafa", "Age": 20, "Address": "Kashmir"}, {"_id": 2, "Name": "Kafa", "Age": 21, "Address": "Kashmir"}, {"_id": 3, "Name": "Rafa", "Age": 22, "Address": "Kashmir"}]
> db.student.find({_id:1})
```

```

Q bmsce@bmsce-Precision-T1700:~ 
2022-04-18T15:11:00.511+0530    exported 1 record
bmsce@bmsce-Precision-T1700:~$ pwd
/home/bmsce
bmsce@bmsce-Precision-T1700:~$ mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
2022-04-18T15:16:31.097+0530    connected to: localhost
2022-04-18T15:16:31.098+0530    exported 1 record
bmsce@bmsce-Precision-T1700:~$ 
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection Student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
[sudo] password for bmsce:
2022-04-18T15:19:13.567+0530    connected to: localhost
2022-04-18T15:19:13.567+0530    exported 1 record
bmsce@bmsce-Precision-T1700:~$ Mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("55fb0928-57d3-4119-a529-34f8a2619c30") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten]
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-18T13:57:39.395+0530 I STORAGE  [initandlisten] **          See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten]
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2022-04-18T13:57:48.671+0530 I CONTROL  [initandlisten]
> use Karthik
switched to db Karthik
> db.Student.find({}); 
{ "_id" : 1, "studName" : "John", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
> ^C
bye
bmsce@bmsce-Precision-T1700:~$ sudo mongoexport --host localhost --db Karthik --collection student --type=csv --out /home/bmsce/Desktop/output.txt --fields "_id","StudName","Usn","Semester","Dept_Name","CGPA","Hobbies"
2022-04-18T15:21:30.067+0530    connected to: localhost
2022-04-18T15:21:30.067+0530    exported 4 records
bmsce@bmsce-Precision-T1700:~$ 

[{"pname": "motherboard"}, 
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1}); 
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 } 
> db.product.find({price:{$ne:15000}},{pname:1,_id:0}); 
{ "pname": "keyboard" } 
{ "pname": "mouse" } 
{ "pname": "monitor" } 
> db.product.find({$and:[{quantity:{Seq:9}},{pname:{Seq:"monitor"}]}]}, {_id:0}); 
{ "pname" : "monitor" } 
> db.product.find({pname:/ds/},{pname:1,quantity:1,_id:0}); 
{ "pname" : "keyboard", "quantity" : 2 } 
{ "pname" : "motherboard", "quantity" : 4 } 
> db.createCollection("hospital"); 
{ "ok" : 1 } 
> db.hospital.insert({_id:1, Name: "Anshuman Agarwal", age:23, diseases:["fever", "diarrhoea", "wheezing", "gastritis"]}); 
WriteResult({ "nInserted" : 1 }) 
> db.hospital.insert({_id:2, Name: "Pinky Choubey", age:35, diseases:["fever","nausea", "food infection", "indigestion", "kidney stones"]}); 
WriteResult({ "nInserted" : 1 }) 
> db.hospital.insert({_id:3, Name: "Amresh Chowpati", age:63, diseases:["hyperglycemia", "diabetes mellitus", "food poisoning", "cold"]}); 
WriteResult({ "nInserted" : 1 }) 
> db.hospital.updateMany({}, {$pull:{diseases:"fever"}});
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 2 } 
> db.hospital.updateOne({_id:1},{pop:{diseases:-1}}); 
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 } 
> db.hospital.find({"diseases.2":"nausea"}); 
> db.hospital.find({"diseases.1":"nausea"}); 
> d.hospital.find(); 
uncaught exception: ReferenceError: d is not defined : 
@shell):1:1 
> db.hospital.find(); 
{ "_id" : 1, "Name" : "Anshuman Agarwal", "age" : 23, "diseases" : [ "wheezing", "gastritis" ] } 
{ "_id" : 2, "Name" : "Pinky Choubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] } 
{ "_id" : 3, "Name" : "Amresh Chowpati", "age" : 63, "diseases" : [ "hyperglycemia", "diabetes mellitus", "food poisoning", "cold" ] } 
> db.hospital.find({"diseases.0":"nausea"}); 
{ "_id" : 2, "Name" : "Pinky Choubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] } 
> db.hospital.update({_id:3},{$set:{'diseases.1':'sarscov'}}); 
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 }) 
> 

```

```

}
> db.createCollection("faculty");
{ "ok" : 1 }
> db.faculty.insert({_id:1,name:"Dr. Balaraman Ravindran",designation:"Professor",department:"CSE",age:45,salary:100000,specialization:['python','mysql','sklearn','tensorflow']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:2,name:"Dr. Mahadev Ghorki",designation:"Assistant Professor",department:"CSE",age:35,salary:80000,specialization:['python','numpy','sklearn','tensorflow','java']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:3,name:"Dr. Praveen Borade",designation:"Associate Professor",department:"ME",age:40,salary:75000,specialization:['autocad','aerodynamics','thermal physics']});
WriteResult({ "nInserted" : 1 })
> db.faculty.insert({_id:4,name:"Dr. Madhav Nayak",designation:"Assistant Professor",department:"ME",age:37,salary:95000,specialization:['autocad','flight-dynamics','Finite Element Analysis']});
WriteResult({ "nInserted" : 1 })
> db.faculty.aggregate ( { $match:{department:"ME"}}, { $group : { _id : "$designation", AverageSal :{$avg:"$salary"} } }, { $match:{AverageSal:{$gt:50000}} });
{ "_id" : "Associate Professor", "AverageSal" : 75000 }
{ "_id" : "Assistant Professor", "AverageSal" : 95000 }
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert({pid:1,pname:"keyboard",mdate:2001,price:1800,quantity:2});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:2,pname:"mouse",mdate:2005,price:1500,quantity:5});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:3,pname:"monitor",mdate:2015,price:10000,quantity:9});
WriteResult({ "nInserted" : 1 })
> db.product.insert({pid:4,pname:"motherboard",mdate:2021,price:15000,quantity:4});
WriteResult({ "nInserted" : 1 })
> db.product.find({}, {pname:1,_id:0})
[{"pname" : "keyboard"}
 {"pname" : "mouse"}
 {"pname" : "monitor"}
 {"pname" : "motherboard"}]
> db.product.find({pid:1},{pid:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.flnd({price:{$ne:15000}},{pname:1,_id:0});
{ "pname" : "Keyboard" }
{ "pname" : "Mouse" }

@(shell):1:20
> db.Bank.update({_id:625d78659329139694f188a6}, {$set: {CustID:5}}, {upsert:true});
uncaught exception: SyntaxError: identifier starts immediately after numeric literal :
@(shell):1:20
> db.Bank.update({_id:"625d78659329139694f188a6"}, {$set: {CustID:5}}, {upsert:true});
WriteResult({
  "nMatched" : 0,
  "nUpserted" : 1,
  "nModified" : 0,
  "_id" : "625d78659329139694f188a6"
})
> db.Bank.find();
[ { "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] },
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] },
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] },
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] },
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] },
{ "_id" : "625d78659329139694f188a6", "CustID" : 5 }
> db.Bank.update({ id:"625d78659329139694f188a6", CustID:5}, {$set: {Name:"Sumantha K S", Type:"Savings", Contact:["9856321478","011-65897458"]}}, {upsert:true});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Bank.flnd();
[ { "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] },
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] },
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] },
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] },
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ] },
{ "_id" : "625d78659329139694f188a6", "CustID" : 5, "Contact" : [ "9856321478", "011-65897458" ], "Name" : "Sumantha K S", "Type" : "Savings" }
]
> █

```

```

> db.createCollection("Bank");
{
  "ok" : 1
}
> db.insert([{"custID":1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]});
uncought exception: TypeErroe: db.insert is not a function
@shell:1:1
> db.Bank.insert([{"custID":1, Name:"Trivikram Hegde", Type:"Savings", Contact:["9945678231", "080-22364587"]]);
WriteResult({ "nInserted" : 1 })
> db.Bank.insert([{"custID":2, Name:"Vlshvsh Bhat", Type:"Savings", Contact:["6325985615", "080-23651452"]});
WriteResult({ "nInserted" : 1 })
> db.Bank.insert([{"custID":3, Name:"Valshak Bhat", Type:"Savings", Contact:["8971456321", "080-33529458"]]);
WriteResult({ "nInserted" : 1 })
> db.Bank.insert([{"custID":4, Name:"Pramod P Parande", Type:"Current", Contact:["9745236589", "080-56324587"]]);
WriteResult({ "nInserted" : 1 })
> db.Bank.insert([{"custID":4, Name:"Shreyas R S", Type:"Current", Contact:["9445678321", "044-65611729", "080-25639856"]]);
WriteResult({ "nInserted" : 1 })
> db.Bank.find({})
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231", "080-22364587" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vlshvsh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vlshvsh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a2"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
db.Bank.updateMany({}, { $pull:{Contact:"080-25639856"} });
{
  "acknowledged" : true, "matchedCount" : 5, "modifiedCount" : 1
}
> db.Bank.find({})
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vlshvsh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ]
}
{
  "err" : {
    "code" : 11000,
    "errmsg" : "createIndex([Name:1, Type:1],{name:1}): syntax error near 'name:1', 'name' is not a valid identifier"
  }
}
uncought exception: SyntaxError: expected expression, got '}' ;
@shell:1:43
> db.Bank.createIndex([Name:1, Type:1],{name:"Find current account holders"});
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.Bank.find({})
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vlshvsh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pramod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ]
}
> db.Bank.getIndexes()
[ {
  "v" : 2,
  "v" : 2,
}
]

```

```

> db.createCollection("Bank");
{
  "ok" : 1
}
> db.insert([{"custID":1, "Name":"Trivikram Hegde", "Type":"Savings", "Contact":["9945678231", "080-22364587"]});
uncaught exception: TypeError: db.insert is not a function
@db.Bank.insert([{"custID":1, "Name":"Trivikram Hegde", "Type":"Savings", "Contact":["9945678231", "080-22364587"]]);
WriteResult({ "nInserted" : 1 })
{
  "ok" : 1
}
> db.Bank.insert([{"custID":2, "Name":"Vishvesh Bhat", "Type":"Savings", "Contact":["6325985615", "080-23651452"]});
WriteResult({ "nInserted" : 1 })
{
  "ok" : 1
}
> db.Bank.insert([{"custID":3, "Name":"Valshak Bhat", "Type":"Savings", "Contact":["8971456321", "080-33529458"]]);
WriteResult({ "nInserted" : 1 })
{
  "ok" : 1
}
> db.Bank.insert([{"custID":4, "Name":"Pranod P Parande", "Type":"Current", "Contact":["9745236589", "080-56324587"]]);
WriteResult({ "nInserted" : 1 })
{
  "ok" : 1
}
> db.Bank.insert([{"custID":5, "Name":"Shreyas R S", "Type":"Current", "Contact":["9445678321", "044-65611729", "080-25639856"]]);
WriteResult({ "nInserted" : 1 })
{
  "ok" : 1
}
> db.Bank.find({});
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231", "080-22364587" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a5"), "CustID" : 4, "Name" : "Pranod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 5, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.find({}).find();
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a5"), "CustID" : 4, "Name" : "Pranod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 5, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.find({}).find({}).find();
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.find({}).find({}).find({}).find();
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.find({}).find({}).find({}).find({}).find();
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.find({}).find({}).find({}).find({}).find({}).find();
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ]
}
{
  "ok" : 1
}
> db.Bank.createIndex([{"Name":1, "Type":1}, {"name":1}], { "spull":1 });
uncaught exception: SyntaxError: expected expression, got ':'
@shell:1:43
> db.Bank.createIndex([{"Name":1, "Type":1}, {"name":1}], {"Find current account holders"});
{
  "ok" : 1
}
{
  "createdCollectionAutomatically" : false,
  "numIndexesBefore" : 1,
  "numIndexesAfter" : 2,
  "ok" : 1
}
> db.Bank.find({});
{
  "ok" : 1
}
{
  "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ]
}
{
  "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ]
}
{
  "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Valshak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ]
}
{
  "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pranod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ]
}
{
  "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 5, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729" ]
}
> db.Bank.getIndexes()
{
  {
    "n" : 2,
    "v" : 2
  }
}

```

```

CREATE KEYSPACE library_info WITH REPLICATION
= {'class': 'SimpleStrategy', 'replication_factor': 1};

create table library_info.library_info(stud_id int, counter_value counter,
stud_name text, book_name text, date_of_issue timestamp, book_id int, PRIMARY
KEY(stud_id, stud_name, book_name, date_of_issue, book_id));

cqlsh> update library_info.library_info set counter_value = counter_value+1 where
stud_id=111 and stud_name = 'Karthik' and book_name='ML' and date_of_issue='2020-
10-11' and book_id=200;
cqlsh> update library_info.library_info set counter_value = counter_value+1 where
stud_id=112 and stud_name = 'Kiran' and book_name='BDA' and date_of_issue='2020-09-
21' and book_id=300;
cqlsh> update library_info.library_info set counter_value = counter_value+1 where
stud_id=113 and stud_name = 'Karana' and book_name='OODM' and date_of_issue='2020-
09-31' and book_id=400;

cqlsh> select * from library_info.library_info;

stud_id | stud_name | book_name | date_of_issue | book_id |
counter_value
-----+-----+-----+-----+-----+
3      111 |   Karthik |       ML | 2020-10-10 18:30:00.000000+0000 |     200 |
1      113 |   Karana |    OODM | 2020-09-21 18:30:00.000000+0000 |     400 |
1      112 |    Kiran |      BDA | 2020-09-20 18:30:00.000000+0000 |     300 |

cqlsh> update library_info.library_info set counter_value = counter_value+1 where
stud_id=112 and stud_name='Kiran' and book_name = 'BDA' and date_of_issue ='2020-
09-20' and book_id=300;
cqlsh> select * from library_info.library_info where stud_id=112;

stud_id | stud_name | book_name | date_of_issue | book_id |
counter_value
-----+-----+-----+-----+-----+
1      112 |    Kiran |      BDA | 2020-09-19 18:30:00.000000+0000 |     300 |
1      112 |    Kiran |      BDA | 2020-09-20 18:30:00.000000+0000 |     300 |

cqlsh> copy library_info.library_info
(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) to 'e:\
library_info.csv';
Using 11 child processes

Starting copy of library_info.library_info with columns [stud_id, stud_name,
book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate: 34 rows/s; Avg. rate: 34 rows/s
4 rows exported to 1 files in 0.124 seconds.

cqlsh> copy library_info.library_info
(stud_id,stud_name,book_name,book_id,date_of_issue,counter_value) from 'e:\\
library_info.csv'
...
Using 11 child processes

```

```
Starting copy of library_info.library_info with columns [stud_id, stud_name,
book_name, book_id, date_of_issue, counter_value].
Processed: 4 rows; Rate:      7 rows/s; Avg. rate:      11 rows/s
4 rows imported from 1 files in 0.377 seconds (0 skipped).
cqlsh> select * from library_info.library_info;
```

|   | stud_id       | stud_name | book_name | date_of_issue                   | book_id |  |
|---|---------------|-----------|-----------|---------------------------------|---------|--|
|   | counter_value |           |           |                                 |         |  |
| 6 | 111           | Karthik   | ML        | 2020-10-10 18:30:00.000000+0000 | 200     |  |
| 2 | 113           | Karana    | OOMD      | 2020-09-21 18:30:00.000000+0000 | 400     |  |
| 2 | 112           | Kiran     | BDA       | 2020-09-19 18:30:00.000000+0000 | 300     |  |
| 2 | 112           | Kiran     | BDA       | 2020-09-20 18:30:00.000000+0000 | 300     |  |