

Manjula

1BM20CS408

```
Activities Terminal Apr 20 15:29
bmsce@bmsce-Precision-T1700:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
implicit session: session { "id" : UUID("a795136f-f70a-45dd-8676-610b3cdd0b2") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-20T19:31:41.433+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:31:41.433+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-20T19:31:44.487+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:31:44.487+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
> use test
bad use parameter
> use students
switched to db students
> db.createcollection("students");
2022-04-20T14:13:04.222+0530 E QUERY [thread1] TypeError: db.createcollection is not a function :
@(&shell):1:1
> db.createcollection("students");
{ "ok" : 1 }
> db.students.insert({_id:1,name:"manju",usn:"105",sen:"5",dept_name:"cse",cgpa:"8.1",hobbies:"singing",});
>
>
>
>
> db.students.insert({_id:1,name:"manju",usn:"105",sen:"5",dept_name:"cse",cgpa:"8.1",hobbies:"singing"});
WriteResult({
  "inserted" : 0,
  "writeError" : {
    "code" : 11000,
    "errmsg" : "E11000 duplicate key error collection: students.students index: _id_dup key: { : 1.0 }"
  }
})
> db.students.insert({_id:"2",name:"arfa",usn:"100",sen:"6",dept_name:"EC",cgpa:"8.4",hobbies:"dance"});
WriteResult({ "inserted" : 1 })
> db.students.insert({_id:"2",name:"arfa",usn:"100",sen:"6",dept_name:"EC",cgpa:"8.4",hobbies:"dance"});
WriteResult({ "inserted" : 1 })
> db.Student.insert({StudentName:"Rahul",USN:"1BM19CS105",Semester:"VI",Dept_name:"Computer Science",CGPA:8.6,Hobbies:["Sleep","eat"]});
WriteResult({ "inserted" : 1 })
> db.Student.insert({StudentName:"Radha",USN:"1BM19CS108",Semester:"VI",Dept_name:"Cse",CGPA:7.6,Hobbies:["dance"]});
WriteResult({ "inserted" : 1 })
> db.Student.insert({StudentName:"arfa",USN:"1BM19CS205",Semester:"2",Dept_name:"Cse",CGPA:9.6,Hobbies:["singing"]});
WriteResult({ "inserted" : 1 })
> db.Student.aggregate($match:{Dept_name:"Computer Science"},{$group:{_id:$Semester,AvgCGPA:{$avg:"$CGPA"}},{$match:{AvgCGPA:{$gt:7.5}}});
{ "_id" : "VI", "AvgCGPA" : 8.6 }
{ "_id" : "VIII", "AvgCGPA" : 8.6 }
2022-04-20T15:10:29.637+0530 E QUERY [thread1] SyntaxError: missing ; before statement @(&shell):1:8
{ "_id" : "VI", "AvgCGPA" : 8.633333333333333 }
2022-04-20T15:10:29.637+0530 E QUERY [thread1] SyntaxError: missing ; before statement @(&shell):1:8
```

```
switched to db Bank
> use Students
switched to db Students
> db.createcollection("Bank");
{ "ok" : 1 }
> db.Bank.insert({_Bank_id:01,BankName:"Canara",Location:"Aldur"})
WriteResult({ "inserted" : 1 })
> db.Bank.insert({_Bank_id:02,BankName:"Vijaya",Location:"Delhi"})
WriteResult({ "inserted" : 1 })
> db.Bank.insert({_Bank_id:03,BankName:"Alhabad",Location:"Hanumanthnagar"})
WriteResult({ "inserted" : 1 })
> db.Bank.update({_Bank_id:01},{$push:{Branch:{Mudigere,"BALEHANOORU"}}});
2022-04-20T15:08:12.974+0530 E QUERY [thread1] ReferenceError: Mudigere is not defined :
@(&shell):1:46
> db.Bank.update({_Bank_id:01},{$push:{Branch:{Mudigere,"BALEHANOORU"}}});
2022-04-20T15:08:55.080+0530 E QUERY [thread1] SyntaxError: missing ; after property id @(&shell):1:55
> db.Bank.update({_Bank_id:01},{$push:{Branch:{canara:"BALEHANOOR",Vijaya:"Gulanpet"}}});
WriteResult({ "matched" : 1, "upserted" : 0, "modified" : 1 })
> db.Bank.update({_Bank_id:01},{$pop:{Branch:{canara:"BALEHANOOR",Vijaya:"Gulanpet"}}});
WriteResult({
  "matched" : 0,
  "upserted" : 0,
  "modified" : 0,
  "writeError" : {
    "code" : 9,
    "errmsg" : "Expected a number in: Branch: { canara: \"BALEHANOOR\", Vijaya: \"Gulanpet\" }"
  }
})
> db.Bank.update({_Bank_id:01},{$pull:{Branch:{canara:"BALEHANOOR",Vijaya:"Gulanpet"}}});
WriteResult({ "matched" : 1, "upserted" : 0, "modified" : 1 })
> db.Bank.find()
{ "_id" : ObjectId("625fd1aaaf3e5ae2d1a075"), "_Bank_id" : 1, "BankName" : "Canara", "Location" : "Aldur", "Branch" : [ ] }
{ "_id" : ObjectId("625fd13aaf3e5ae2d1a076"), "_Bank_id" : 2, "BankName" : "Vijaya", "Location" : "Delhi" }
{ "_id" : ObjectId("625fd173aaf3e5ae2d1a077"), "_Bank_id" : 3, "BankName" : "Alhabad", "Location" : "Hanumanthnagar" }
> db.Bank.find().count()
2022-04-20T15:21:58.079+0530 E QUERY [thread1] SyntaxError: missing name after . operator @(&shell):1:15
> db.Bank.find().count()
{ "_id" : ObjectId("625fd1aaaf3e5ae2d1a075"),
  "_Bank_id" : 1,
  "BankName" : "Canara",
  "Location" : "Aldur",
  "Branch" : [ ] }
> use Students
switched to db Students
> db.Students.find({_id:1})
{ "_id" : 1, "Name" : "Arfa" }
{ "_id" : 2, "Name" : "Rafa" }
{ "_id" : 3, "Name" : "shifa" }
> db.Students.find({_id:1})
```

```

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: mongod://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 stora
ge 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:{$eq:9}},{pname:{$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 Chorki",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:{$gt:50000}}});
{ "_id" : "Associate Professor", "AverageSal" : 75000 }
{ "_id" : "Assistant Professor", "AverageSal" : 95000 }
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert({_id:1,pname:"keyboard",mdate:2001,price:1800,quantity:2});
WriteResult({ "nInserted" : 1 })
> db.product.insert({_id:2,pname:"mouse",mdate:2005,price:1500,quantity:5});
WriteResult({ "nInserted" : 1 })
> db.product.insert({_id:3,pname:"monitor",mdate:2015,price:10000,quantity:9});
WriteResult({ "nInserted" : 1 })
> db.product.insert({_id:4,pname:"motherboard",mdate:2021,price:15000,quantity:4});
WriteResult({ "nInserted" : 1 })
> db.product.find({},{pname:1,_id:0})
{ "pname" : "keyboard" }
{ "pname" : "mouse" }
{ "pname" : "monitor" }
{ "pname" : "motherboard" }
> db.product.find({_id:1},{pname:1,_id:0,mdate:1,quantity:1});
{ "pid" : 1, "mdate" : 2001, "quantity" : 2 }
> db.product.find({price:{$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" : "Pranod 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" : "Pranod 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
0(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:"Pranod 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" : "Pranod 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},{ $pull:{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" : "Pranod 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.createIndex({Name:1, Type:1},{name:});
uncaught exception: SyntaxError: expected expression, got ';'
0(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" : "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" : "Pranod 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,

```

```

> 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
0(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:"Pranod 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" : "Pranod 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},{ $pop:{Contact:1} });
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
> db.Bank.find({});
{ "_id" : ObjectId("625d77809329139694f188a2"), "CustID" : 1, "Name" : "Trivikram Hegde", "Type" : "Savings", "Contact" : [ "9945678231" ] }
{ "_id" : ObjectId("625d77bd9329139694f188a3"), "CustID" : 2, "Name" : "Vishvesh Bhat", "Type" : "Savings", "Contact" : [ "6325985615", "080-23651452" ] }
{ "_id" : ObjectId("625d77e69329139694f188a4"), "CustID" : 3, "Name" : "Vaishak Bhat", "Type" : "Savings", "Contact" : [ "8971456321", "080-33529458" ] }
{ "_id" : ObjectId("625d78229329139694f188a5"), "CustID" : 4, "Name" : "Pranod P Parande", "Type" : "Current", "Contact" : [ "9745236589", "080-56324587" ] }
{ "_id" : ObjectId("625d78659329139694f188a6"), "CustID" : 4, "Name" : "Shreyas R S", "Type" : "Current", "Contact" : [ "9445678321", "044-65611729", "080-25639856" ] }
> db.Bank.updateMany({},{$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" : "Pranod 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.createIndex({Name:1, Type:1},{name:1});
Uncaught exception: SyntaxError: expected expression, got ','
0(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" : "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" : "Pranod 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,

```