

NAME : ASHWINI

USN: 1BM20CS402

```
chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("8518712-1041-4807-b3de-1c8f83b1090") }
MongoDB server version: 3.6.0
Server has startup warnings:
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten]
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/production-file-system
2022-04-20T19:40:03.223+0530 I CONTROL [initandlisten]
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten]
> use chethana
switched to db chethana
> db.createCollection("Student");
{ "ok" : 1 }
>
>
chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("2af7a74-2022-4474-995d-c09c1f0e400") }
MongoDB server version: 3.6.0
Server has startup warnings:
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten]
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:36:04.241+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/production-file-system
2022-04-20T19:40:03.223+0530 I CONTROL [initandlisten]
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten]
> db.Student.insert({StudentName:"Chethana",USN:"1BM20CS402",Semester:"VII",DeptName:"CSE",CGPA:8.4,Hobbies:"Dance"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({StudentName:"Lalitha",USN:"1BM20EC105",Semester:"VII",DeptName:"ECE",CGPA:8,Hobbies:"Sleep"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({StudentName:"Hayer",USN:"1BM20ME200",Semester:"VI",DeptName:"ME",CGPA:9,Hobbies:"Coding"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({StudentName:"Archana",USN:"1BM20CV300",Semester:"VI",DeptName:"CV",CGPA:8.7,Hobbies:"Cooking"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({StudentName:"John",USN:"1BM20ME00",Semester:"I",DeptName:"IN",CGPA:7,Hobbies:"Swimming"});
WriteResult({ "nInserted" : 1 })
> db.Student.find({});
{ "_id" : ObjectId("620018c6e7d28378841ed2af"), "StudentName" : "Chethana", "USN" : "1BM20CS402", "Semester" : "VII", "DeptName" : "CSE", "CGPA" : 8.4, "Hobbies" : "Dance" }
{ "_id" : ObjectId("620019d9e7d28378841ed2af"), "StudentName" : "Lalitha", "USN" : "1BM20EC105", "Semester" : "VII", "DeptName" : "ECE", "CGPA" : 8, "Hobbies" : "Sleep" }
{ "_id" : ObjectId("620019f2e7d28378841ed2af"), "StudentName" : "Hayer", "USN" : "1BM20ME200", "Semester" : "VI", "DeptName" : "ME", "CGPA" : 9, "Hobbies" : "Coding" }
{ "_id" : ObjectId("620019fc7d28378841ed2af"), "StudentName" : "Archana", "USN" : "1BM20CV300", "Semester" : "VI", "DeptName" : "CV", "CGPA" : 8.7, "Hobbies" : "Cooking" }
{ "_id" : ObjectId("62001a0e7d28378841ed2af"), "StudentName" : "John", "USN" : "1BM20ME00", "Semester" : "I", "DeptName" : "IN", "CGPA" : 7, "Hobbies" : "Swimming" }
```

```
chethana@chethana04-VirtualBox:~$ mongoexport --host localhost --C
chethana@chethana04-VirtualBox:~$ mongoexport --host localhost --db chethana --collection Student --type-csv --out /home/chethana04/Desktop/output.txt
2022-04-20T21:15:00.315+0530 Failed: CSV node requires a field list
chethana@chethana04-VirtualBox:~$ mongoexport --host localhost --db chethana --collection Student --type-csv --out /home/chethana04/Desktop/output.txt --fields "_id","StudentName","USN","Semester","Dept
ame","CGPA","Hobbies"
2022-04-20T21:36:03.636+0530 connected to localhost
2022-04-20T21:36:03.658+0530 exported 6 records
chethana@chethana04-VirtualBox:~$ sudo mongoexport --host localhost --db chethana --collection Student --type-csv --out /home/chethana04/Desktop/output.txt --fields "_id","StudentName","USN","Semester","
eptName","CGPA","Hobbies"
[sudo] password for chethana04:
2022-04-20T21:38:15.985+0530 connected to localhost
2022-04-20T21:38:15.985+0530 exported 6 records
chethana@chethana04-VirtualBox:~$ mongoexport --host localhost --db chethana --collection Student --type-csv --out /home/chethana04/Desktop/output.txt --fields "_id","StudentName","USN","Semester","Dept
ame","CGPA","Hobbies"
2022-04-20T21:38:35.432+0530 connected to localhost
2022-04-20T21:38:35.433+0530 exported 6 records
chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.0
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("feb0471-d22a-401f-8300-1f67551fc1f1") }
MongoDB server version: 3.6.0
Server has startup warnings:
2022-04-20T21:23:00.061+0530 I STORAGE [initandlisten]
2022-04-20T21:23:00.061+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T21:23:00.061+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/production-file-system
2022-04-20T21:23:04.011+0530 I CONTROL [initandlisten]
2022-04-20T21:23:04.011+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T21:23:04.011+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T21:23:04.011+0530 I CONTROL [initandlisten]
> db.students.find({})
> db.createC
```

```
2022-04-18T14:35:57.967+0530 E QUERY [thread1] SyntaxError: illegal character @ (shell):1:43
> db.student.aggregate({$match:{DeptName:"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
2022-04-18T15:04:00.798+0530 E QUERY [thread1] SyntaxError: missing ; before statement @ (shell):1:14
> /home/hduser/Desktop/output.txt --fields "Year","Quarter"
```

```

> 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({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" : "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 '' :
@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,

```

```

@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" }
>

```



```

2022-04-20T20:09:26.789+0530 Failed: CSV mode requires a field list
chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("931666de-fef7-499f-9022-435407d46f59") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten]
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-20T19:40:03.223+0530 I CONTROL [initandlisten]
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten]
> db.createCollection("Bank");
{ "ok" : 1 }
> db.Bank.insert({_id:1,name:"Radha",state:"Goa",country:"India",language:["marathi"]})
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({_id:1,name:"Sunil",state:"Andhra Pradesh",country:"India",language:["telugu,kannada"]})
WriteResult({ "nInserted" : 1 })
> db.Bank.insert({_id:1,name:"Vish",state:"Delhi",country:"India",language:["hindi"]})
WriteResult({ "nInserted" : 1 })
> db.Bank.find(
... )
{ "_id" : ObjectId("62681d131bdbccecc5a25af40"), "id" : 1, "name" : "Radha", "state" : "Goa", "country" : "India", "language" : [ "marathi" ] }
{ "_id" : ObjectId("62681d4e1bdbccecc5a25af41"), "id" : 1, "name" : "Sunil", "state" : "Andhra Pradesh", "country" : "India", "language" : [ "telugu,kannada" ] }
{ "_id" : ObjectId("62681d7d1bdbccecc5a25af42"), "id" : 1, "name" : "Vish", "state" : "Delhi", "country" : "India", "language" : [ "hindi" ] }
>
bye

```

```

chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("15eb19f9-be3f-4526-a0f1-9a30fd109f23") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten]
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-20T19:40:03.223+0530 I CONTROL [initandlisten]
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten]
>
>
bye
chethana@chethana04-VirtualBox:~$ pwd
/home/chethana
chethana@chethana04-VirtualBox:~$ mongo
MongoDB shell version v3.6.8
connecting to: mongodb://127.0.0.1:27017
Implicit session: session { "id" : UUID("6463978f-d263-4e07-8b45-98ae028af4b") }
MongoDB server version: 3.6.8
Server has startup warnings:
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten]
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** WARNING: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine
2022-04-20T19:39:54.241+0530 I STORAGE [initandlisten] ** See http://dochub.mongodb.org/core/prodnotes-filesystem
2022-04-20T19:40:03.223+0530 I CONTROL [initandlisten]
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten] ** Read and write access to data and configuration is unrestricted.
2022-04-20T19:40:03.224+0530 I CONTROL [initandlisten]
> db.Bank.update({_id:1},{push:{language:"marathi"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
> db.Bank.update({_id:2},{pull:{language:"english"}})
WriteResult({ "nMatched" : 0, "nUpserted" : 0, "nModified" : 0 })
>

```

```

)
> db.createCollection("faculty");
{ "ok" : 1 }
> db.faculty.insert([_id:1,name:"Dr. Balaranan 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:{$gt:50000}}});
[ { "_id" : "Associate Professor", "AverageSal" : 75000 }
  { "_id" : "Assistant Professor", "AverageSal" : 95000 } ]
> db.createCollection("product");
{ "ok" : 1 }
> db.product.insert([_id:1,pname:"keyboard",ndate:2001,price:1000,quantity:2]);
WriteResult({ "nInserted" : 1 })
> db.product.insert([_id:2,pname:"mouse",ndate:2005,price:1500,quantity:5]);
WriteResult({ "nInserted" : 1 })
> db.product.insert([_id:3,pname:"monitor",ndate:2015,price:10000,quantity:9]);
WriteResult({ "nInserted" : 1 })
> db.product.insert([_id:4,pname:"motherboard",ndate:2021,price:15000,quantity:4]);
WriteResult({ "nInserted" : 1 })
> db.product.find({},{pname:1,_id:0})
[ { "pname" : "keyboard" }
  { "pname" : "mouse" }
  { "pname" : "monitor" }
  { "pname" : "motherboard" } ]
> db.product.find([_id:1],[_id:0,ndate:1,quantity:1]);
[ { "pid" : 1, "ndate" : 2001, "quantity" : 2 } ]
> db.product.find([price:{$ne:15000}],{pname:1,_id:0});
[ { "pname" : "keyboard" } ]

```

```

[ { "pname" : "motherboard" } ]
> db.product.find([_id:1],[_id:0,ndate:1,quantity:1]);
[ { "pid" : 1, "ndate" : 2001, "quantity" : 2 } ]
> db.product.find([price:{$ne:15000}],{pname:1,_id:0});
[ { "pname" : "keyboard" }
  { "pname" : "mouse" }
  { "pname" : "monitor" } ]
> db.product.find([_id:1],[_id:0,ndate:1,quantity:1]);
[ { "pid" : 1, "ndate" : 2001, "quantity" : 2 } ]
> db.product.find([price:{$ne:15000}],{pname:1,_id:0});
[ { "pname" : "keyboard" } ]
> 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: "Anresh 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"]);
[ { "_id" : 2, "Name" : "Pinky Chaubey", "age" : 35, "diseases" : [ "nausea", "food infection", "indigestion", "kidney stones" ] } ]
> db.hospital.find([diseases.1:"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 })

```