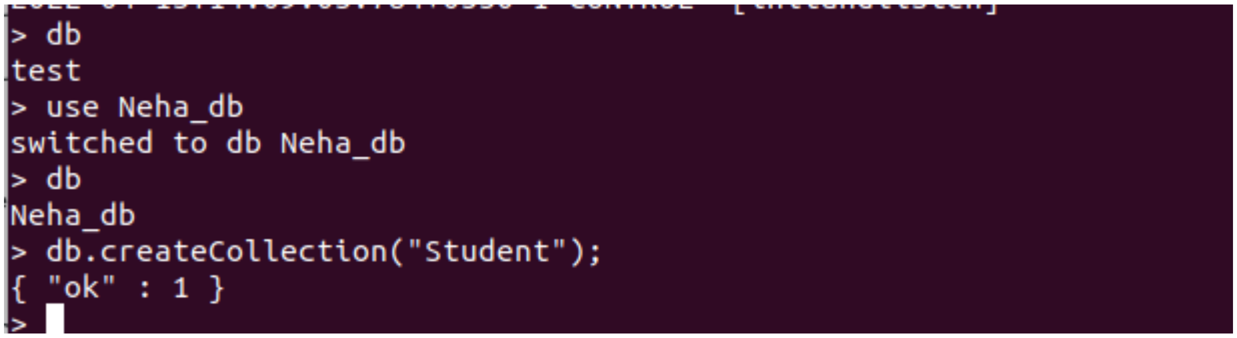
Nikhil Abraham

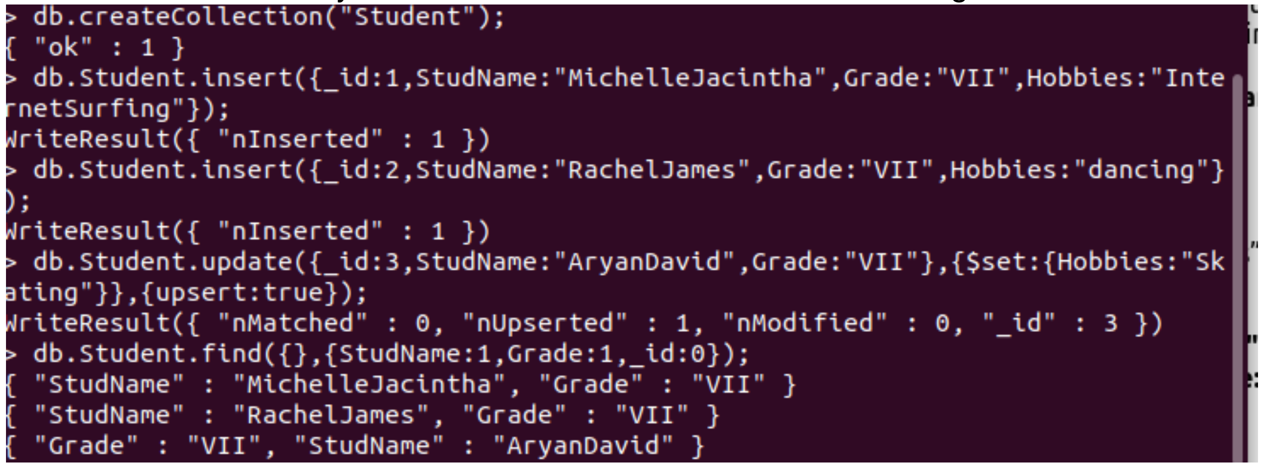
1BM19CS101

CREATE DATABASE IN MONGODB.



```
> db
test
> use Neha_db
switched to db Neha_db
> db
Neha_db
> db.createCollection("Student");
{ "ok" : 1 }
>
```

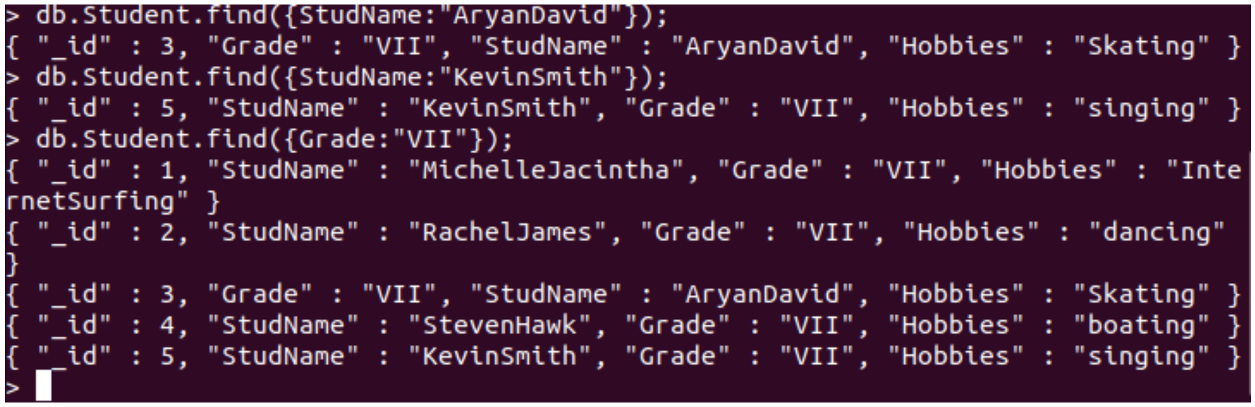
Create a collection by the name "Students" and store the following data in it.



```
> db.createCollection("Student");
{ "ok" : 1 }
> db.Student.insert({_id:1,StudName:"MichelleJacintha",Grade:"VII",Hobbies:"Inte
rnetSurfing"});
WriteResult({ "nInserted" : 1 })
> db.Student.insert({_id:2,StudName:"RachelJames",Grade:"VII",Hobbies:"dancing"}
);
WriteResult({ "nInserted" : 1 })
> db.Student.update({_id:3,StudName:"AryanDavid",Grade:"VII"},{$set:{Hobbies:"Sk
ating"}},{upsert:true});
WriteResult({ "nMatched" : 0, "nUpserted" : 1, "nModified" : 0, "_id" : 3 })
> db.Student.find({},{StudName:1,Grade:1,_id:0});
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
{ "StudName" : "RachelJames", "Grade" : "VII" }
{ "Grade" : "VII", "StudName" : "AryanDavid" }
```

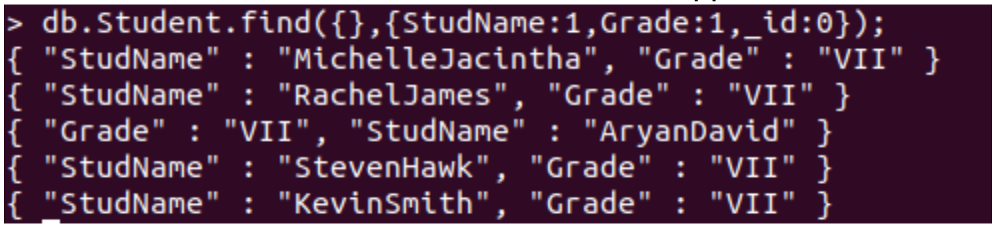
FIND METHOD

A. To search for documents from the "Students" collection based on certain searchcriteria.



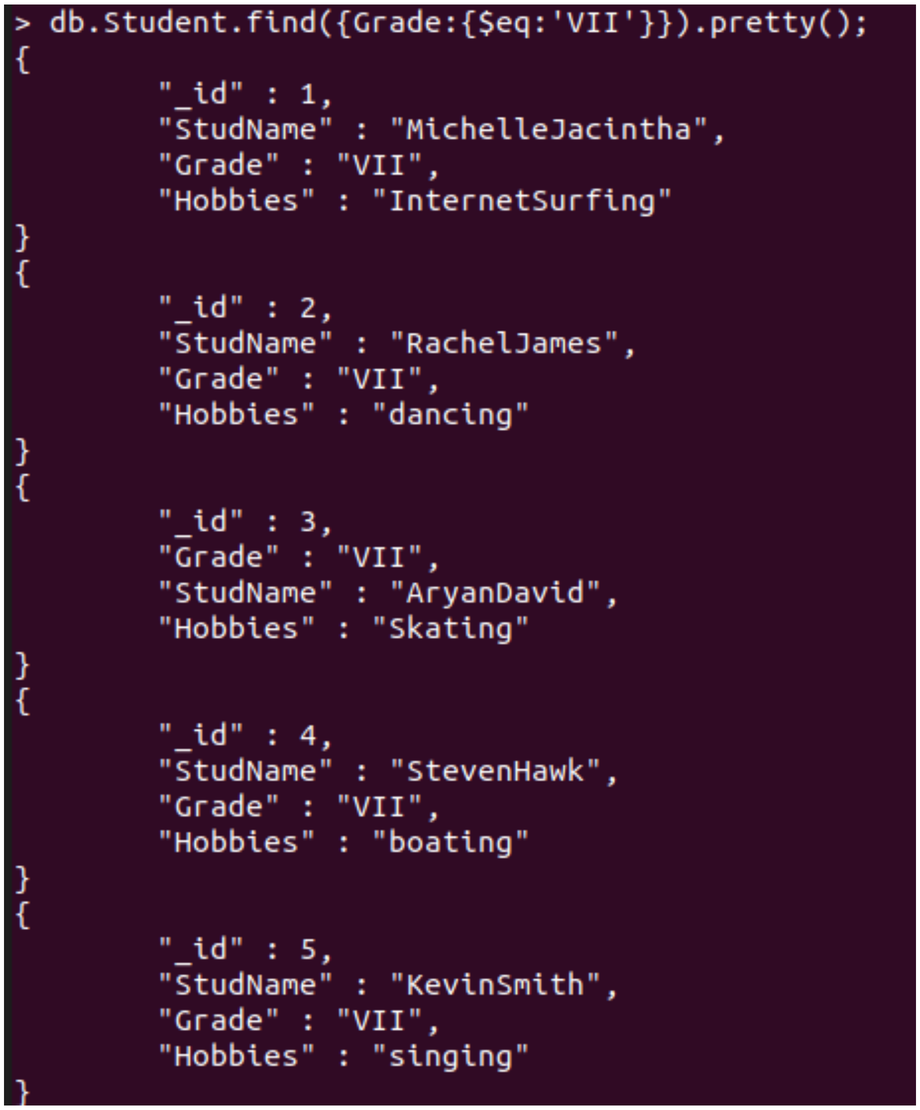
```
> db.Student.find({StudName:"AryanDavid"});
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
> db.Student.find({StudName:"KevinSmith"});
{ "_id" : 5, "StudName" : "KevinSmith", "Grade" : "VII", "Hobbies" : "singing" }
> db.Student.find({Grade:"VII"});
{ "_id" : 1, "StudName" : "MichelleJacintha", "Grade" : "VII", "Hobbies" : "Inte
rnetSurfing" }
{ "_id" : 2, "StudName" : "RachelJames", "Grade" : "VII", "Hobbies" : "dancing"
}
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
{ "_id" : 4, "StudName" : "StevenHawk", "Grade" : "VII", "Hobbies" : "boating" }
{ "_id" : 5, "StudName" : "KevinSmith", "Grade" : "VII", "Hobbies" : "singing" }
>
```

B. To display only the StudName and Grade from all the documents of the Studentscollection. The identifier_id should be suppressed and NOT displayed.



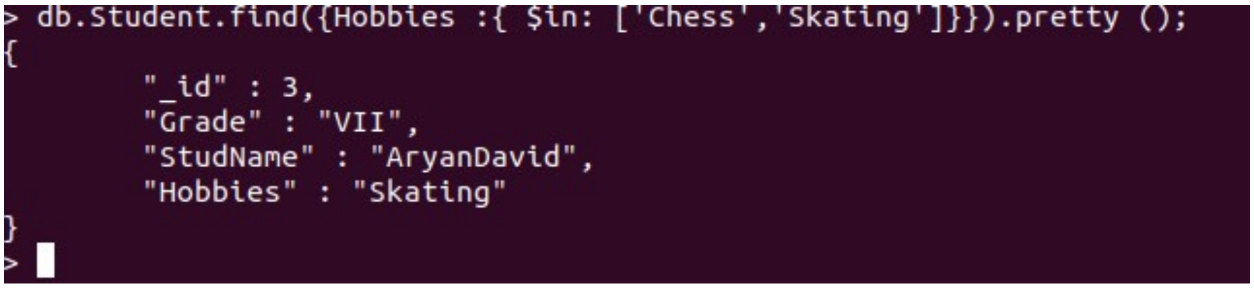
```
> db.Student.find({},{StudName:1,Grade:1,_id:0});
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
{ "StudName" : "RachelJames", "Grade" : "VII" }
{ "Grade" : "VII", "StudName" : "AryanDavid" }
{ "StudName" : "StevenHawk", "Grade" : "VII" }
{ "StudName" : "KevinSmith", "Grade" : "VII" }
```

C. To find those documents where the Grade is set to ‘VII’



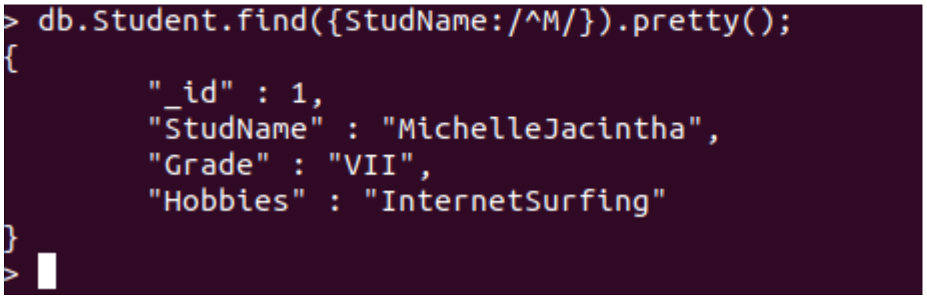
```
> db.Student.find({Grade:{$eq:'VII'}}).pretty();
{
        "_id" : 1,
        "StudName" : "MichelleJacintha",
        "Grade" : "VII",
        "Hobbies" : "InternetSurfing"
}
{
        "_id" : 2,
        "StudName" : "RachelJames",
        "Grade" : "VII",
        "Hobbies" : "dancing"
}
{
        "_id" : 3,
        "Grade" : "VII",
        "StudName" : "AryanDavid",
        "Hobbies" : "Skating"
}
{
        "_id" : 4,
        "StudName" : "StevenHawk",
        "Grade" : "VII",
        "Hobbies" : "boating"
}
{
        "_id" : 5,
        "StudName" : "KevinSmith",
        "Grade" : "VII",
        "Hobbies" : "singing"
}
```

D. To find those documents from the Students collection where the Hobbies is set to either ‘Chess’ or is set to ‘Skating’.



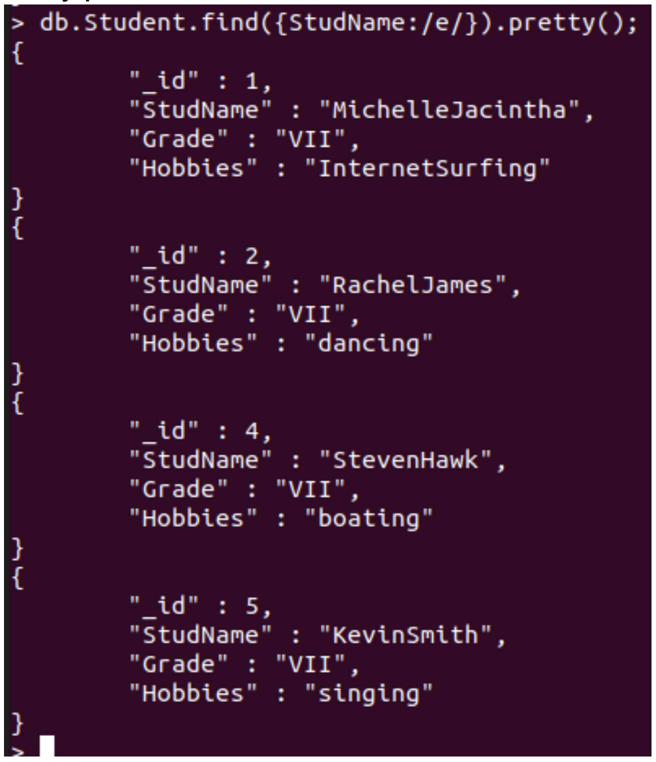
```
> db.Student.find({Hobbies :{ $in: ['Chess','Skating']}}).pretty ();
{
        "_id" : 3,
        "Grade" : "VII",
        "StudName" : "AryanDavid",
        "Hobbies" : "Skating"
}
> 
```

E. To find documents from the Students collection where the StudName begins with“M”.



```
> db.Student.find({StudName:/^M/}).pretty();
{
        "_id" : 1,
        "StudName" : "MichelleJacintha",
        "Grade" : "VII",
        "Hobbies" : "InternetSurfing"
}
> 
```

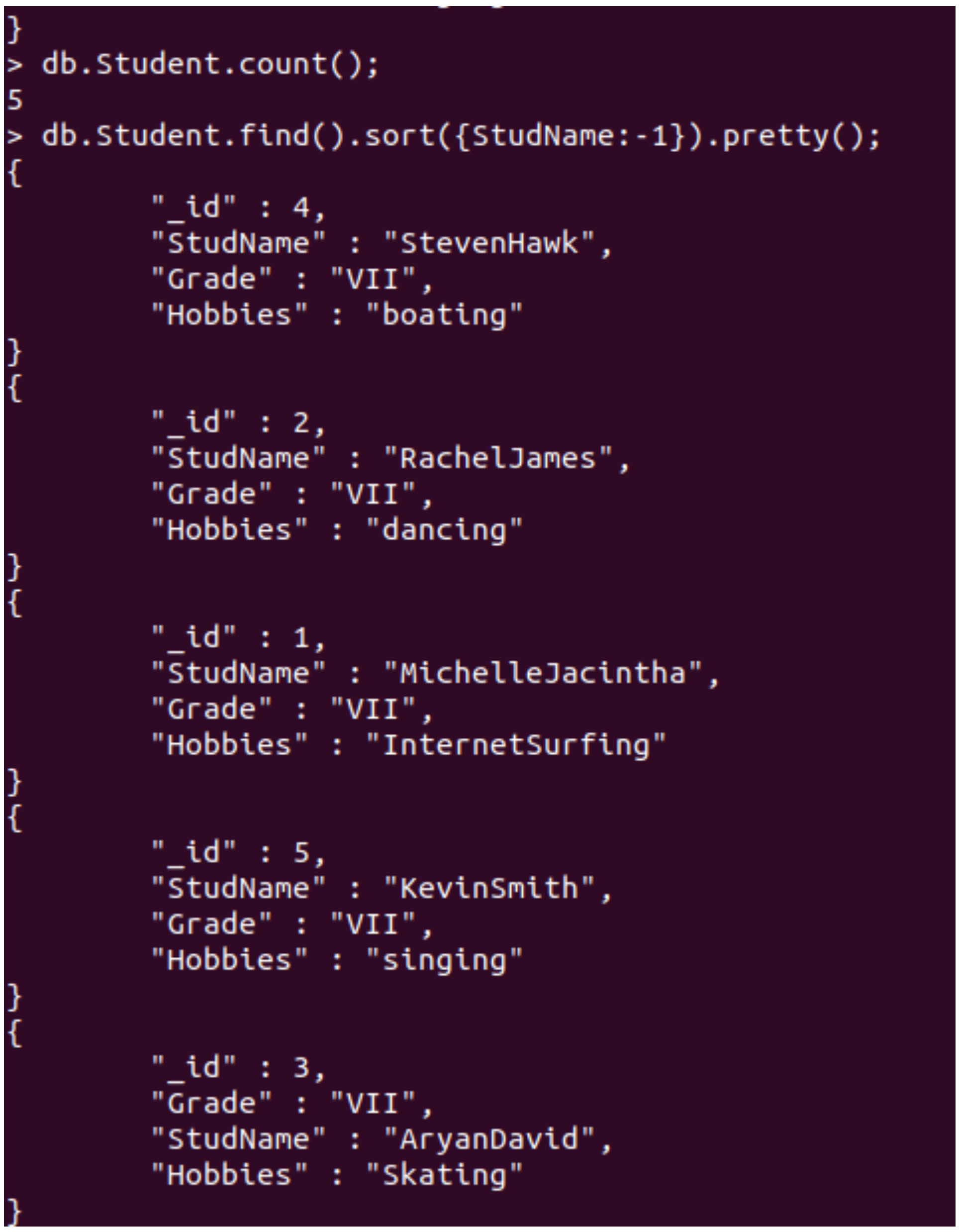
F. To find documents from the Students collection where the StudName has an “e” inany position.



```
> db.Student.find({StudName:/e/}).pretty();
{
        "_id" : 1,
        "StudName" : "MichelleJacintha",
        "Grade" : "VII",
        "Hobbies" : "InternetSurfing"
}
{
        "_id" : 2,
        "StudName" : "RachelJames",
        "Grade" : "VII",
        "Hobbies" : "dancing"
}
{
        "_id" : 4,
        "StudName" : "StevenHawk",
        "Grade" : "VII",
        "Hobbies" : "boating"
}
{
        "_id" : 5,
        "StudName" : "KevinSmith",
        "Grade" : "VII",
        "Hobbies" : "singing"
}
> 
```

G. To find the number of documents in the Students collection.

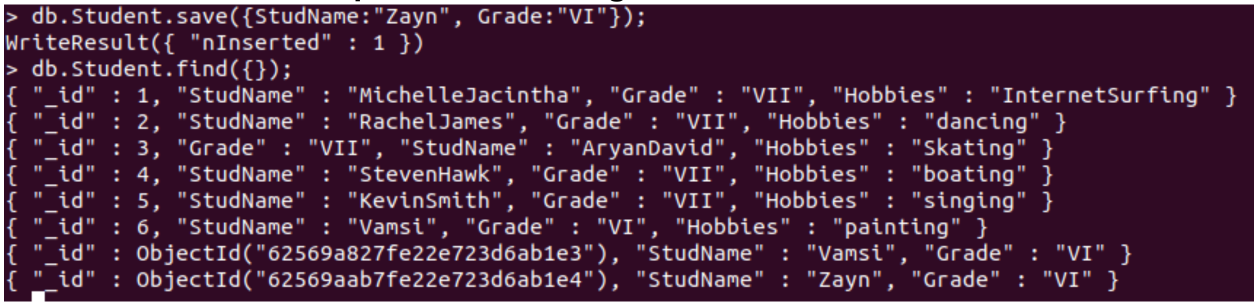
H. To sort the documents from the Students collection in the descending order of StudName.



```
}
> db.Student.count();
5
> db.Student.find().sort({StudName:-1}).pretty();
{
        "_id" : 4,
        "StudName" : "StevenHawk",
        "Grade" : "VII",
        "Hobbies" : "boating"
}
{
        "_id" : 2,
        "StudName" : "RachelJames",
        "Grade" : "VII",
        "Hobbies" : "dancing"
}
{
        "_id" : 1,
        "StudName" : "MichelleJacintha",
        "Grade" : "VII",
        "Hobbies" : "InternetSurfing"
}
{
        "_id" : 5,
        "StudName" : "KevinSmith",
        "Grade" : "VII",
        "Hobbies" : "singing"
}
{
        "_id" : 3,
        "Grade" : "VII",
        "StudName" : "AryanDavid",
        "Hobbies" : "Skating"
}
```

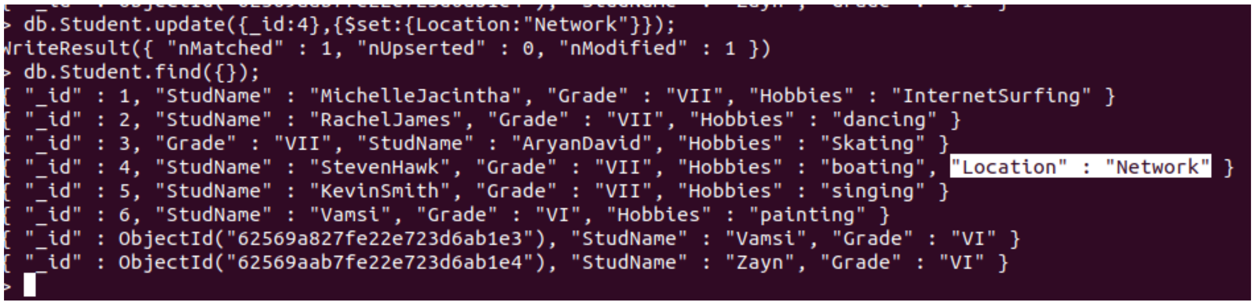
I. Save Method :

Save() method will insert a new document, if the document with the _id does not exist. If it exists it will replace the exisiting document.



```
> db.Student.save({StudName:"Zayn", Grade:"VI"});
WriteResult({ "nInserted" : 1 })
> db.Student.find({});
{ "_id" : 1, "StudName" : "MichelleJacintha", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
{ "_id" : 2, "StudName" : "RachelJames", "Grade" : "VII", "Hobbies" : "dancing" }
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
{ "_id" : 4, "StudName" : "StevenHawk", "Grade" : "VII", "Hobbies" : "boating" }
{ "_id" : 5, "StudName" : "KevinSmith", "Grade" : "VII", "Hobbies" : "singing" }
{ "_id" : 6, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : "painting" }
{ "_id" : ObjectId("62569a827fe22e723d6ab1e3"), "StudName" : "Vamsi", "Grade" : "VI" }
{ "_id" : ObjectId("62569aab7fe22e723d6ab1e4"), "StudName" : "Zayn", "Grade" : "VI" }
```

II. Add a new field to existing Document:

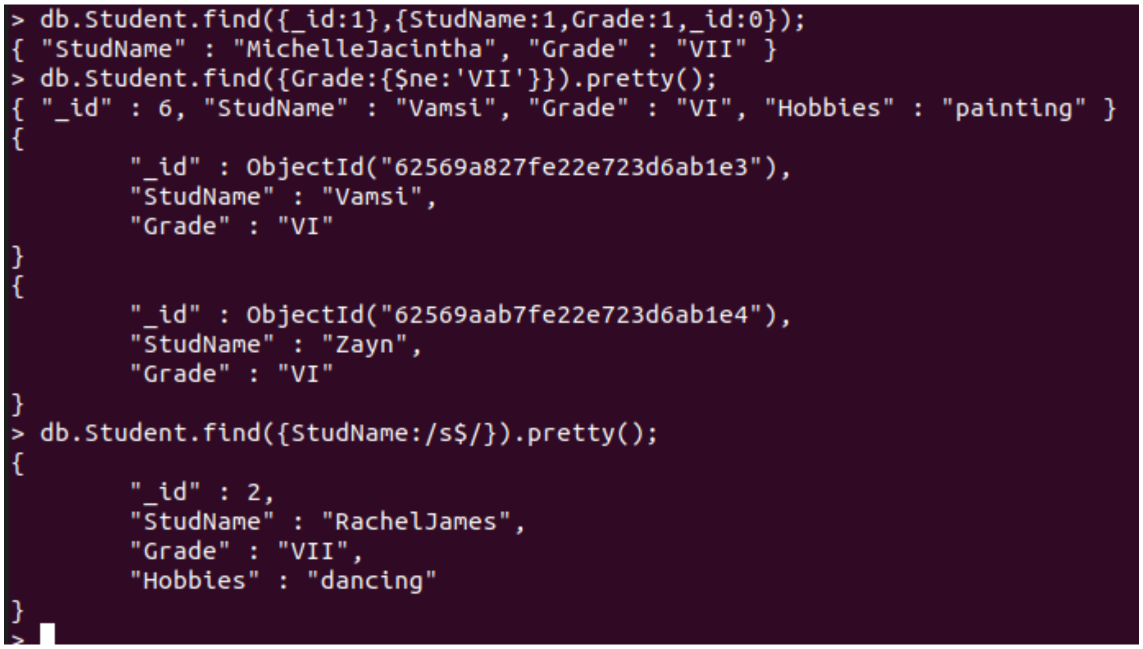


```
> db.Student.update({_id:4},{$set:{Location:"Network"}});
riteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find({});
 "_id" : 1, "StudName" : "MichelleJacintha", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
 "_id" : 2, "StudName" : "RachelJames", "Grade" : "VII", "Hobbies" : "dancing" }
 "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : "Skating" }
 "_id" : 4, "StudName" : "StevenHawk", "Grade" : "VII", "Hobbies" : "boating", "Location" : "Network" }
 "_id" : 5, "StudName" : "KevinSmith", "Grade" : "VII", "Hobbies" : "singing" }
 "_id" : 6, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : "painting" }
 "_id" : ObjectId("62569a827fe22e723d6ab1e3"), "StudName" : "Vamsi", "Grade" : "VI" }
 "_id" : ObjectId("62569aab7fe22e723d6ab1e4"), "StudName" : "Zayn", "Grade" : "VI" }
>
```

III. Finding Document based on search criteria suppressing few fields

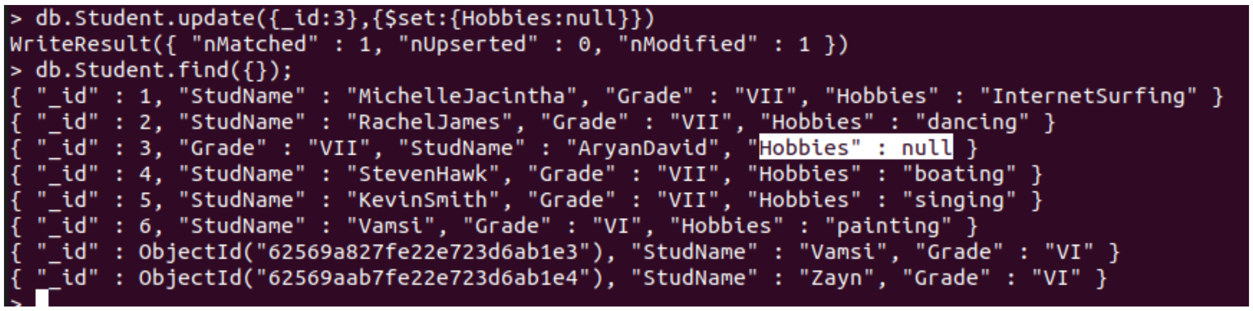
To find those documents where the Grade is not set to 'VII'

To find documents from the Students collection where the StudName ends with s.



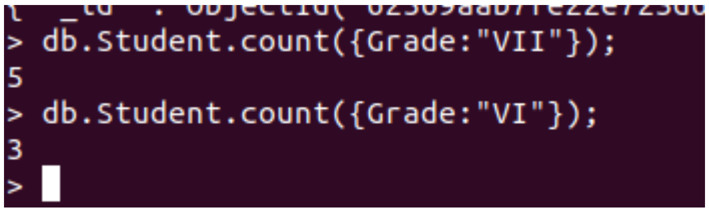
```
> db.Student.find({_id:1},{StudName:1,Grade:1,_id:0});
{ "StudName" : "MichelleJacintha", "Grade" : "VII" }
> db.Student.find({Grade:{$ne:'VII'}}).pretty();
{ "_id" : 6, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : "painting" }
{
        "_id" : ObjectId("62569a827fe22e723d6ab1e3"),
        "StudName" : "Vamsi",
        "Grade" : "VI"
}
{
        "_id" : ObjectId("62569aab7fe22e723d6ab1e4"),
        "StudName" : "Zayn",
        "Grade" : "VI"
}
> db.Student.find({StudName:/s$/}).pretty();
{
        "_id" : 2,
        "StudName" : "RachelJames",
        "Grade" : "VII",
        "Hobbies" : "dancing"
}
>
```

IV. to set a particular field value to NULL



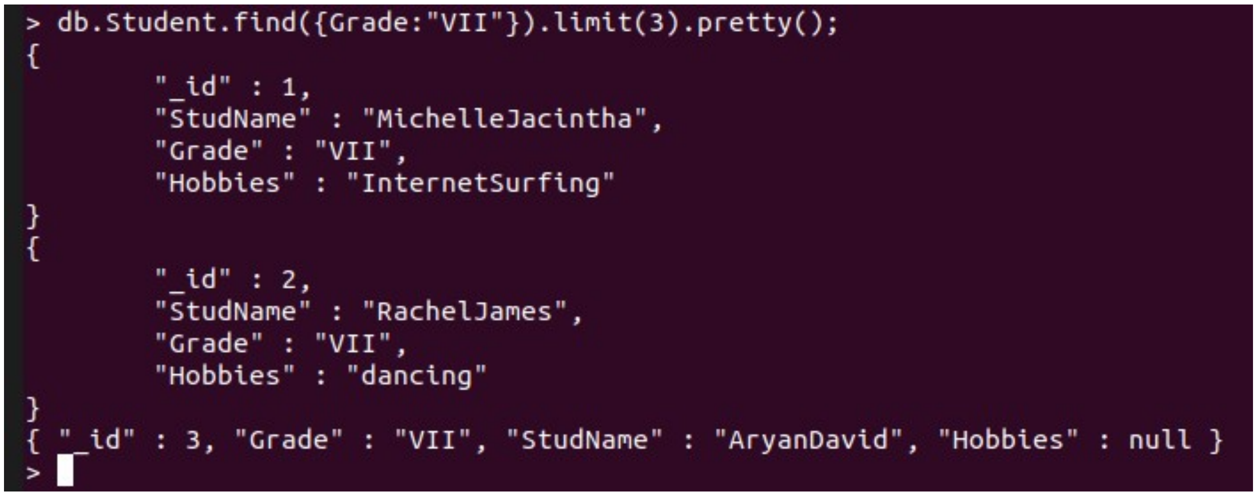
```
> db.Student.update({_id:3},{$set:{Hobbies:null}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.Student.find({});
{ "_id" : 1, "StudName" : "MichelleJacintha", "Grade" : "VII", "Hobbies" : "InternetSurfing" }
{ "_id" : 2, "StudName" : "RachelJames", "Grade" : "VII", "Hobbies" : "dancing" }
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : null }
{ "_id" : 4, "StudName" : "StevenHawk", "Grade" : "VII", "Hobbies" : "boating" }
{ "_id" : 5, "StudName" : "KevinSmith", "Grade" : "VII", "Hobbies" : "singing" }
{ "_id" : 6, "StudName" : "Vamsi", "Grade" : "VI", "Hobbies" : "painting" }
{ "_id" : ObjectId("62569a827fe22e723d6ab1e3"), "StudName" : "Vamsi", "Grade" : "VI" }
{ "_id" : ObjectId("62569aab7fe22e723d6ab1e4"), "StudName" : "Zayn", "Grade" : "VI" }
>
```

V. Count the number of documents in Student Collections with grade :VII



```
{ "_id" : ObjectId("62569aab7fe22e723d6
> db.Student.count({Grade:"VII"});
5
> db.Student.count({Grade:"VI"});
3
>
```

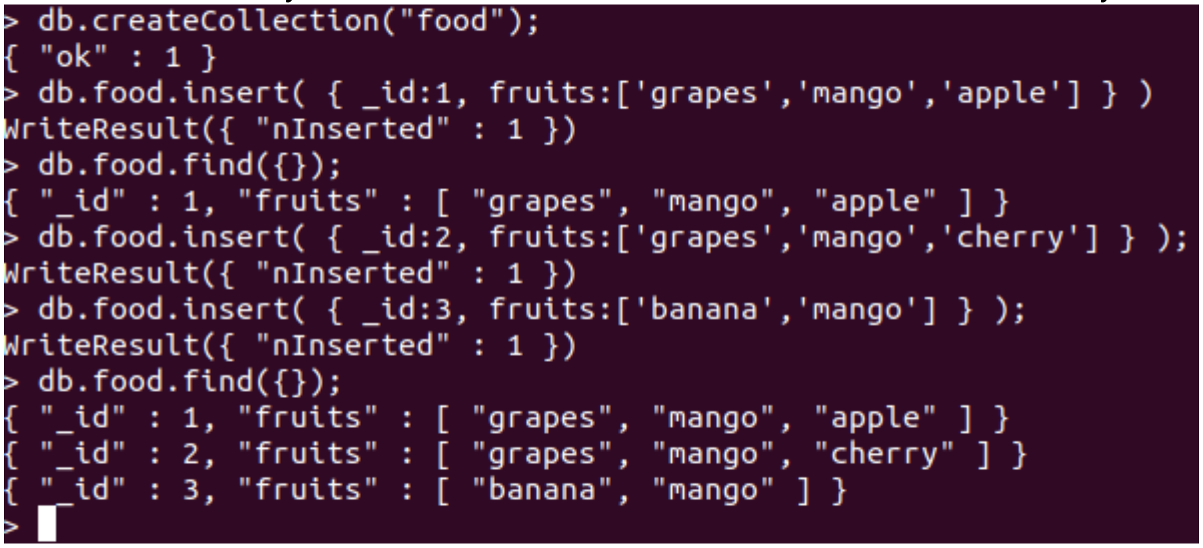
retrieve first 3 documents



```
> db.Student.find({Grade:"VII"}).limit(3).pretty();
{
        "_id" : 1,
        "StudName" : "MichelleJacintha",
        "Grade" : "VII",
        "Hobbies" : "InternetSurfing"
}
{
        "_id" : 2,
        "StudName" : "RachelJames",
        "Grade" : "VII",
        "Hobbies" : "dancing"
}
{ "_id" : 3, "Grade" : "VII", "StudName" : "AryanDavid", "Hobbies" : null }
>
```

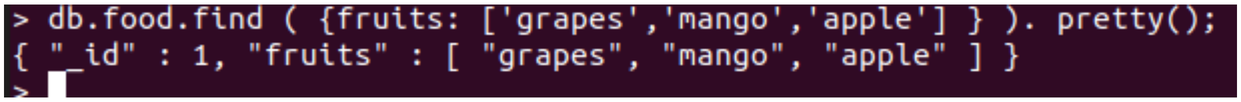
Food Collection

Create a collection by name “food” and add to each document add a “fruits” array



```
> db.createCollection("food");
{ "ok" : 1 }
> db.food.insert( { _id:1, fruits:['grapes','mango','apple'] } )
WriteResult({ "nInserted" : 1 })
> db.food.find({});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> db.food.insert( { _id:2, fruits:['grapes','mango','cherry'] } );
WriteResult({ "nInserted" : 1 })
> db.food.insert( { _id:3, fruits:['banana','mango'] } );
WriteResult({ "nInserted" : 1 })
> db.food.find({});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
> 
```

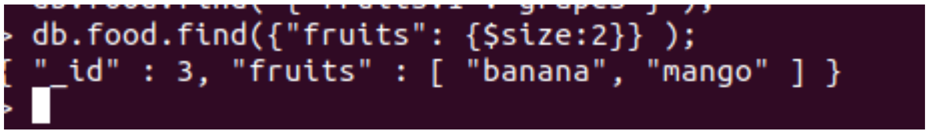
To find those documents from the “food” collection which has the “fruits array”constitute of “grapes”, “mango” and “apple”.



```
> db.food.find ( {fruits: ['grapes','mango','apple'] } ). pretty();
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
> 
```

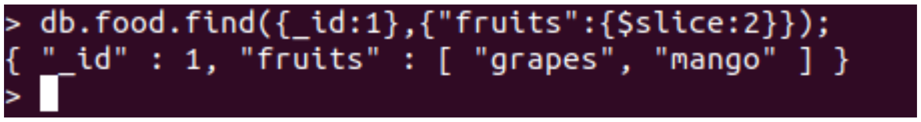
To find in “fruits” array having “mango” in the first index position.

To find those documents from the “food” collection where the size of the array is two.



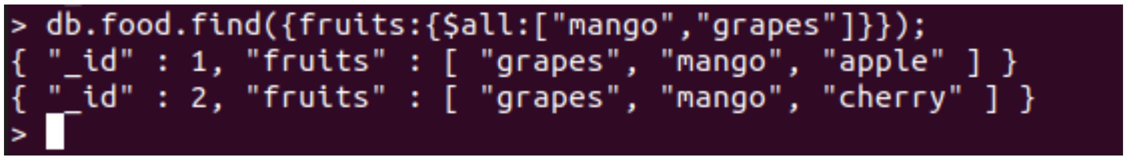
```
> db.food.find({"fruits": {$size:2}} );
{ "_id" : 3, "fruits" : [ "banana", "mango" ] }
> 
```

To find the document with a particular id and display the first two elements from the array “fruits”



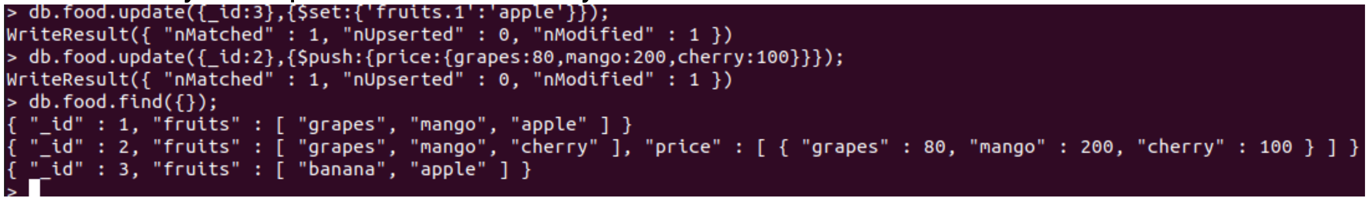
```
> db.food.find({_id:1},{"fruits":{$slice:2}});
{ "_id" : 1, "fruits" : [ "grapes", "mango" ] }
> 
```

To find all the documets from the food collection which have elements mango andgrapes in the array “fruits”



```
> db.food.find({fruits:{$all:["mango","grapes"]}});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ] }
> 
```

update on Array:
using particular id replace the element present in the 1st index position of thefruits array with apple
insert new key value pairs in the fruits array



```
> db.food.update({_id:3},{$set:{'fruits.1':'apple'}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.update({_id:2},{$push:{price:{grapes:80,mango:200,cherry:100}}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.food.find({});
{ "_id" : 1, "fruits" : [ "grapes", "mango", "apple" ] }
{ "_id" : 2, "fruits" : [ "grapes", "mango", "cherry" ], "price" : [ { "grapes" : 80, "mango" : 200, "cherry" : 100 } ] }
{ "_id" : 3, "fruits" : [ "banana", "apple" ] }
> 
```