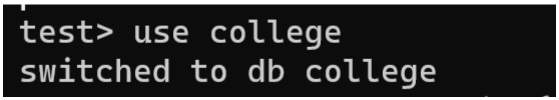
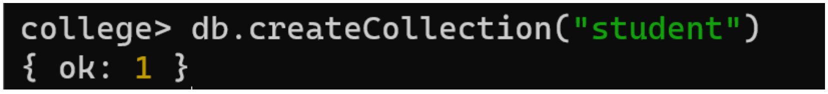
MongoDB – Lab 2

1. **a) Create a database named college and create a collection named student.**

```
test> use college
college> db.createCollection("student")
{ ok: 1 }
```



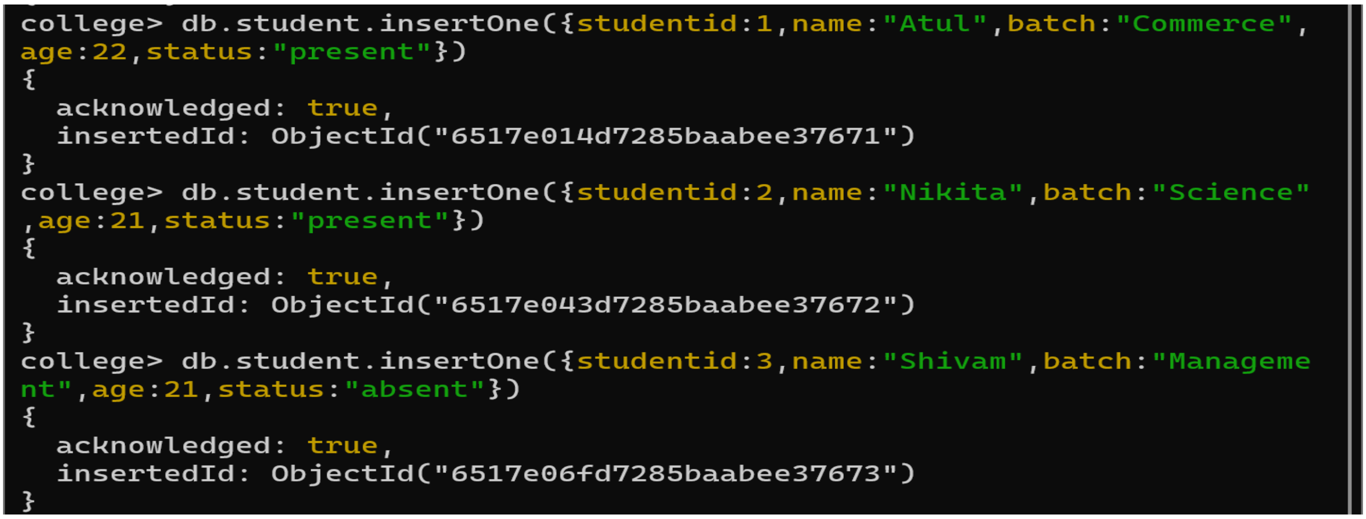
```
test> use college
switched to db college
```



```
college> db.createCollection("student")
{ ok: 1 }
```

b) Insert some documents to the collection with fields studentid, name, batch(Science ,Commerce etc), age, status(present/absent).

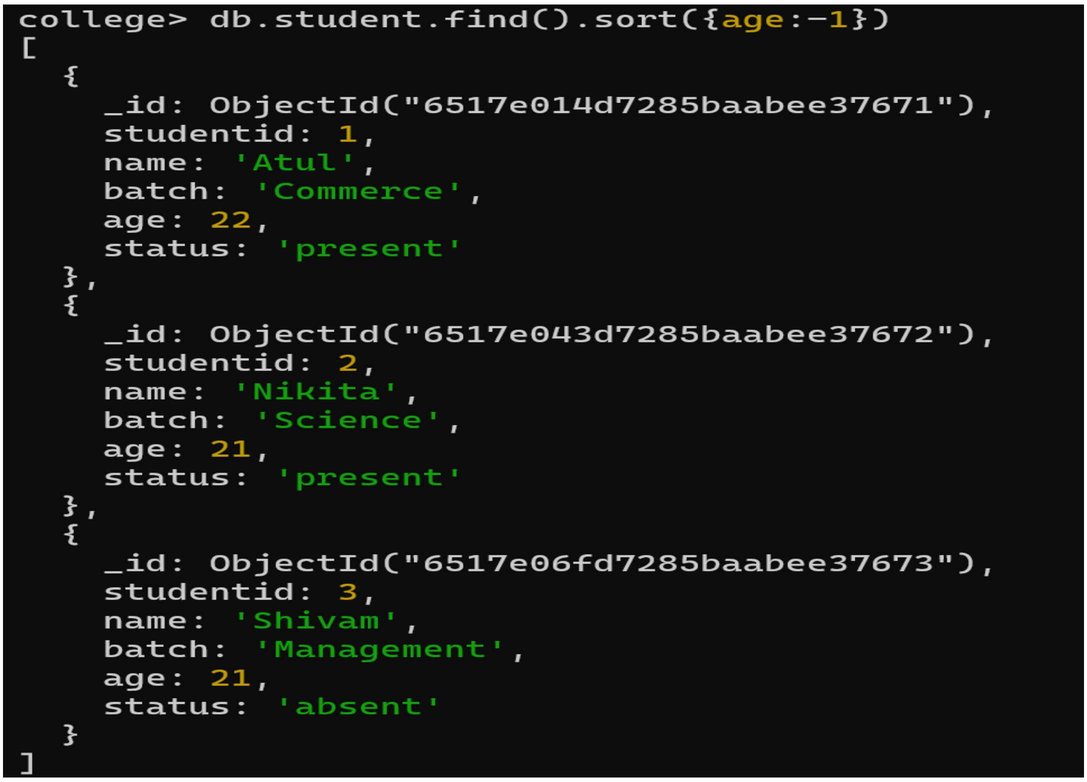
```
college>db.student.insertOne({studentid:1,name:"Atul",batch:"Commerce",age:22,s
tatus:"present"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e014d7285baabee37671")
}
college>db.student.insertOne({studentid:2,name:"Nikita",batch:"Science",age:21,sta
tus:"present"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e043d7285baabee37672")
}
college>db.student.insertOne({studentid:3,name:"Shivam",batch:"Management",ag
e:21,status:"absent"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e06fd7285baabee37673")
}
```



```
college> db.student.insertOne({studentid:1,name:"Atul",batch:"Commerce",
age:22,status:"present"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e014d7285baabee37671")
}
college> db.student.insertOne({studentid:2,name:"Nikita",batch:"Science"
,age:21,status:"present"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e043d7285baabee37672")
}
college> db.student.insertOne({studentid:3,name:"Shivam",batch:"Manageme
nt",age:21,status:"absent"})
{
  acknowledged: true,
  insertedId: ObjectId("6517e06fd7285baabee37673")
}
```

c) Display the students details in descending order based on their age.

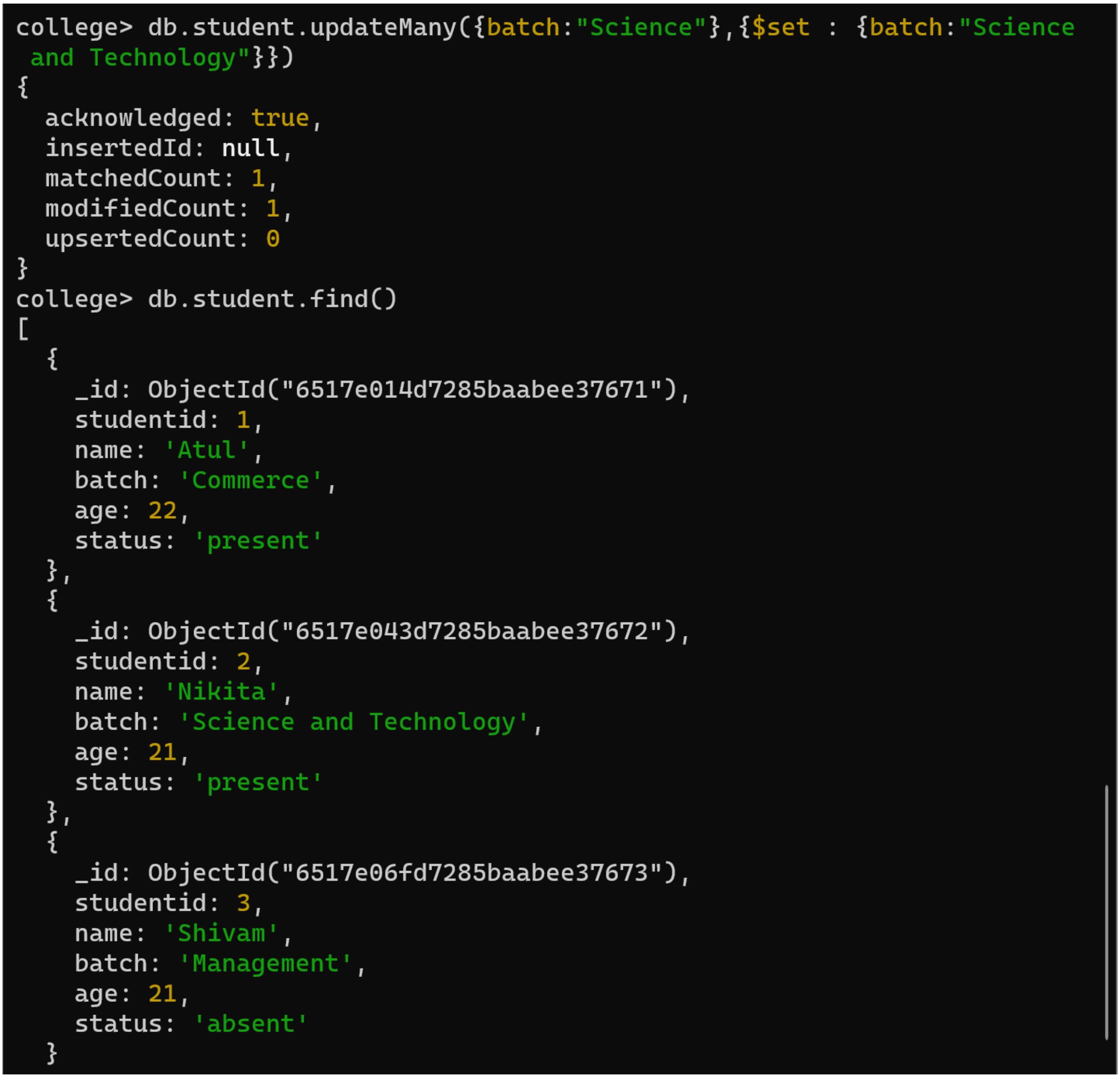
```
college> db.student.find().sort({age:-1})
[
  {
    _id: ObjectId("6517e014d7285baabee37671"),
    studentid: 1,
    name: 'Atul',
    batch: 'Commerce',
    age: 22,
    status: 'present'
  },
  {
    _id: ObjectId("6517e043d7285baabee37672"),
    studentid: 2,
    name: 'Nikita',
    batch: 'Science',
    age: 21,
    status: 'present'
  },
  {
    _id: ObjectId("6517e06fd7285baabee37673"),
    studentid: 3,
    name: 'Shivam',
    batch: 'Management',
    age: 21,
    status: 'absent'
  }
]
```



```
college> db.student.find().sort({age:-1})
[
  {
    _id: ObjectId("6517e014d7285baabee37671"),
    studentid: 1,
    name: 'Atul',
    batch: 'Commerce',
    age: 22,
    status: 'present'
  },
  {
    _id: ObjectId("6517e043d7285baabee37672"),
    studentid: 2,
    name: 'Nikita',
    batch: 'Science',
    age: 21,
    status: 'present'
  },
  {
    _id: ObjectId("6517e06fd7285baabee37673"),
    studentid: 3,
    name: 'Shivam',
    batch: 'Management',
    age: 21,
    status: 'absent'
  }
]
```

d) Update the batch-name science to science and technology

db.student.updateMany({batch:"Science"},{$set : {batch:"Science and Technology"}})

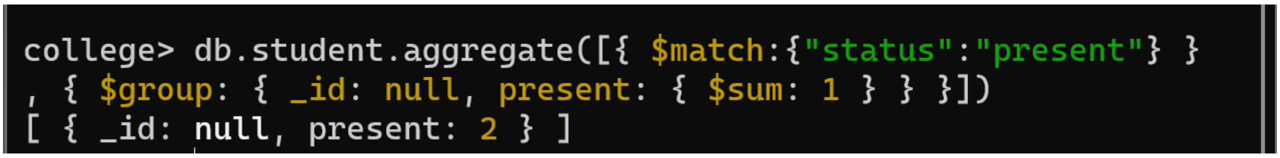


```
college> db.student.updateMany({batch:"Science"},{$set : {batch:"Science
 and Technology"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
college> db.student.find()
[
  {
    _id: ObjectId("6517e014d7285baabee37671"),
    studentid: 1,
    name: 'Atul',
    batch: 'Commerce',
    age: 22,
    status: 'present'
  },
  {
    _id: ObjectId("6517e043d7285baabee37672"),
    studentid: 2,
    name: 'Nikita',
    batch: 'Science and Technology',
    age: 21,
    status: 'present'
  },
  {
    _id: ObjectId("6517e06fd7285baabee37673"),
    studentid: 3,
    name: 'Shivam',
    batch: 'Management',
    age: 21,
    status: 'absent'
  }
```

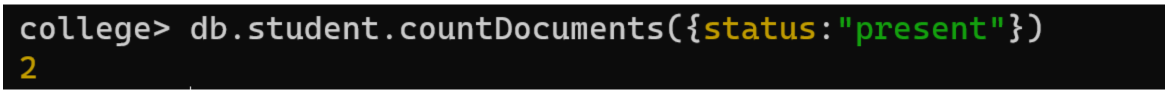
e) Count the number of students who are present.

college> db.student.aggregate([{ $match:{"status":"present"} } , { $group: { _id: null, present: { $sum: 1 } } }])

[{ _id: null, present: 2 }]



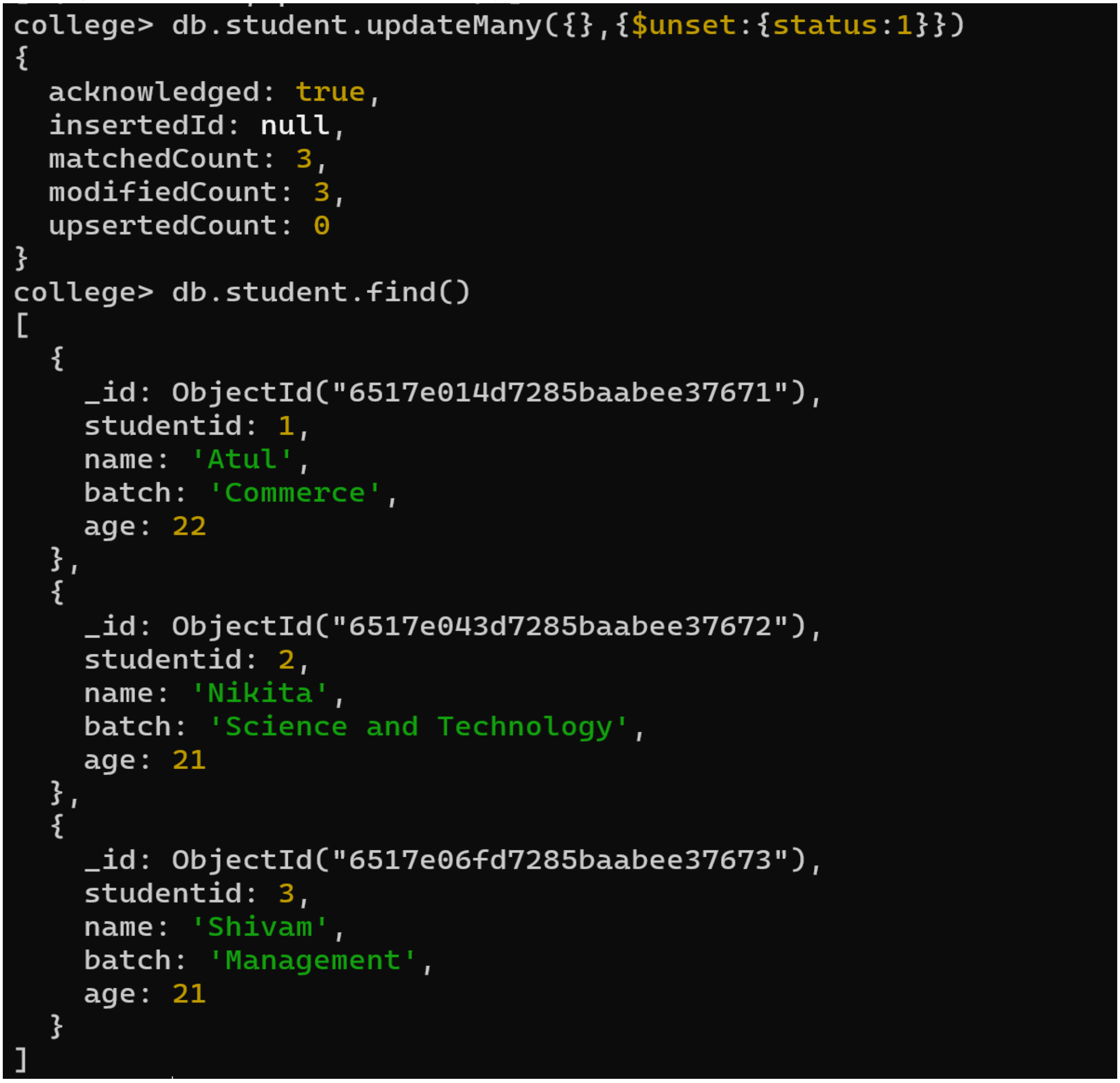
```
college> db.student.aggregate([{ $match:{"status":"present"} }
, { $group: { _id: null, present: { $sum: 1 } } }])
[ { _id: null, present: 2 } ]
```



```
college> db.student.countDocuments({status:"present"})
2
```

f) Remove the status field.

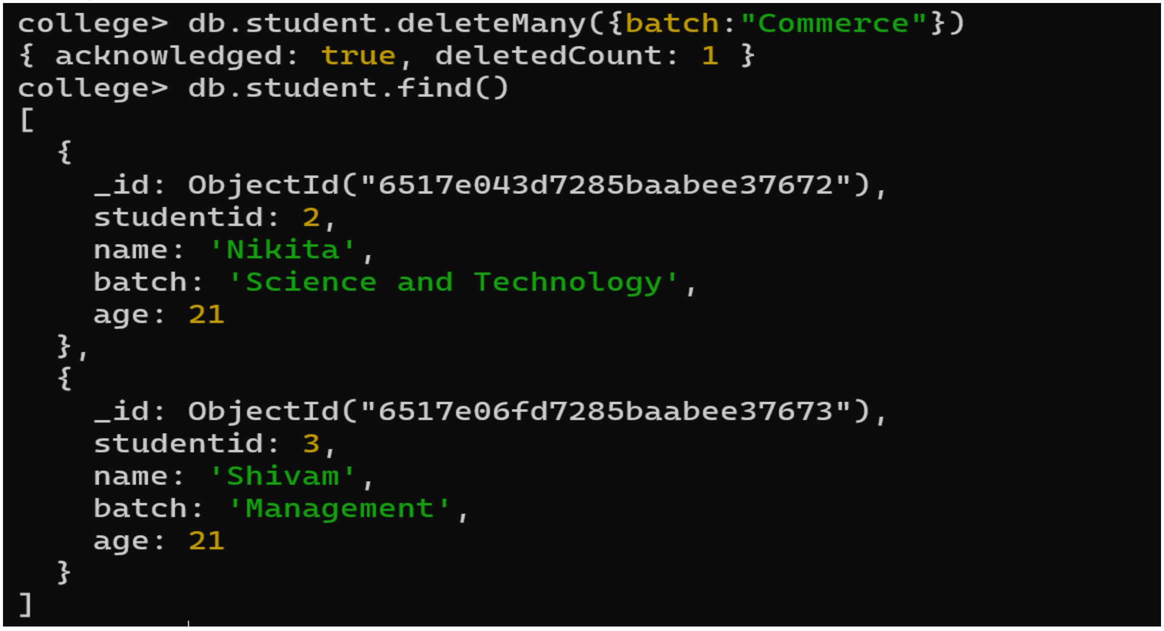
college> db.student.updateMany({},{$unset:{status:1}})



```
college> db.student.updateMany({},{$unset:{status:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
college> db.student.find()
[
  {
    _id: ObjectId("6517e014d7285baabee37671"),
    studentid: 1,
    name: 'Atul',
    batch: 'Commerce',
    age: 22
  },
  {
    _id: ObjectId("6517e043d7285baabee37672"),
    studentid: 2,
    name: 'Nikita',
    batch: 'Science and Technology',
    age: 21
  },
  {
    _id: ObjectId("6517e06fd7285baabee37673"),
    studentid: 3,
    name: 'Shivam',
    batch: 'Management',
    age: 21
  }
]
```

g) Remove all students from commerce batch.

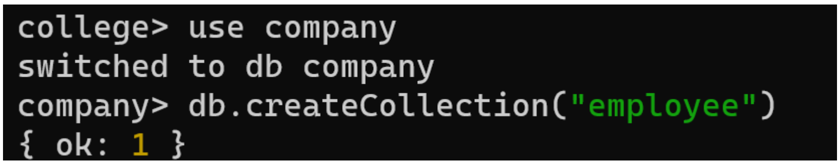
college> db.student.deleteMany({batch:"Commerce"})



```
college> db.student.deleteMany({batch:"Commerce"})
{ acknowledged: true, deletedCount: 1 }
college> db.student.find()
[
  {
    _id: ObjectId("6517e043d7285baabee37672"),
    studentid: 2,
    name: 'Nikita',
    batch: 'Science and Technology',
    age: 21
  },
  {
    _id: ObjectId("6517e06fd7285baabee37673"),
    studentid: 3,
    name: 'Shivam',
    batch: 'Management',
    age: 21
  }
]
```

2. **a) Create database named company and create a collection named employee.**

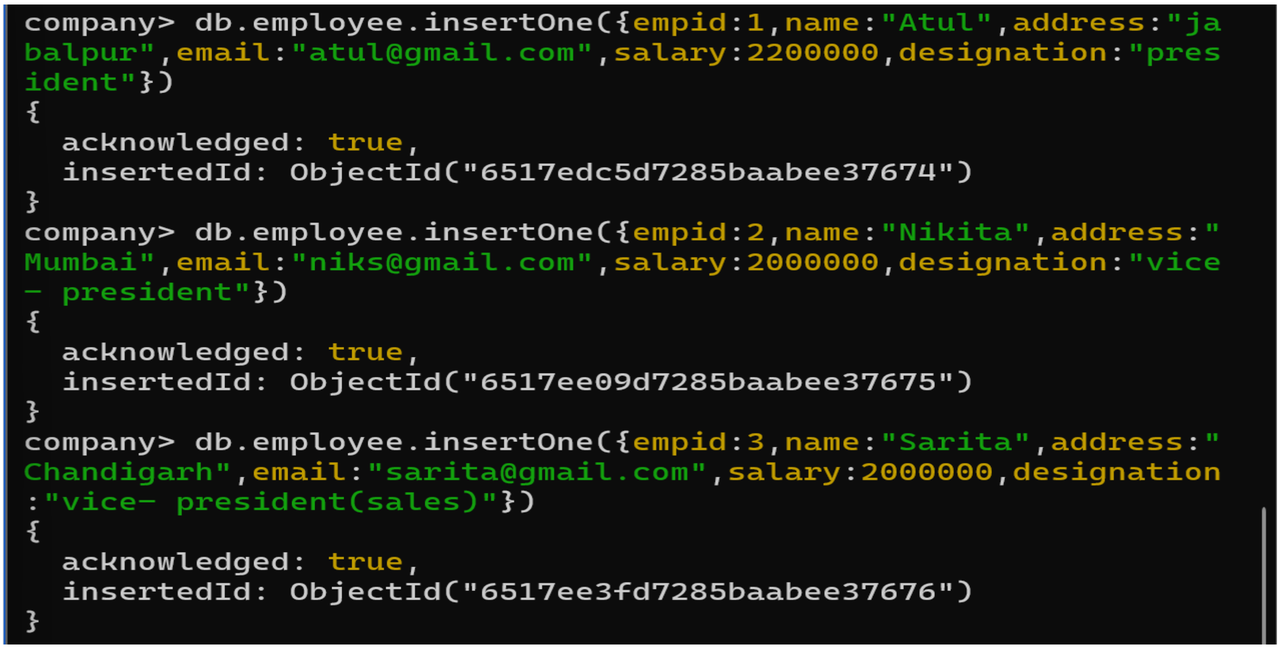
college> use company
switched to db company
company> db.createCollection("employee")
{ ok: 1 }



```
college> use company
switched to db company
company> db.createCollection("employee")
{ ok: 1 }
```

b) Insert some documents to the collection with fields empid, name, address, email, salary and designation.

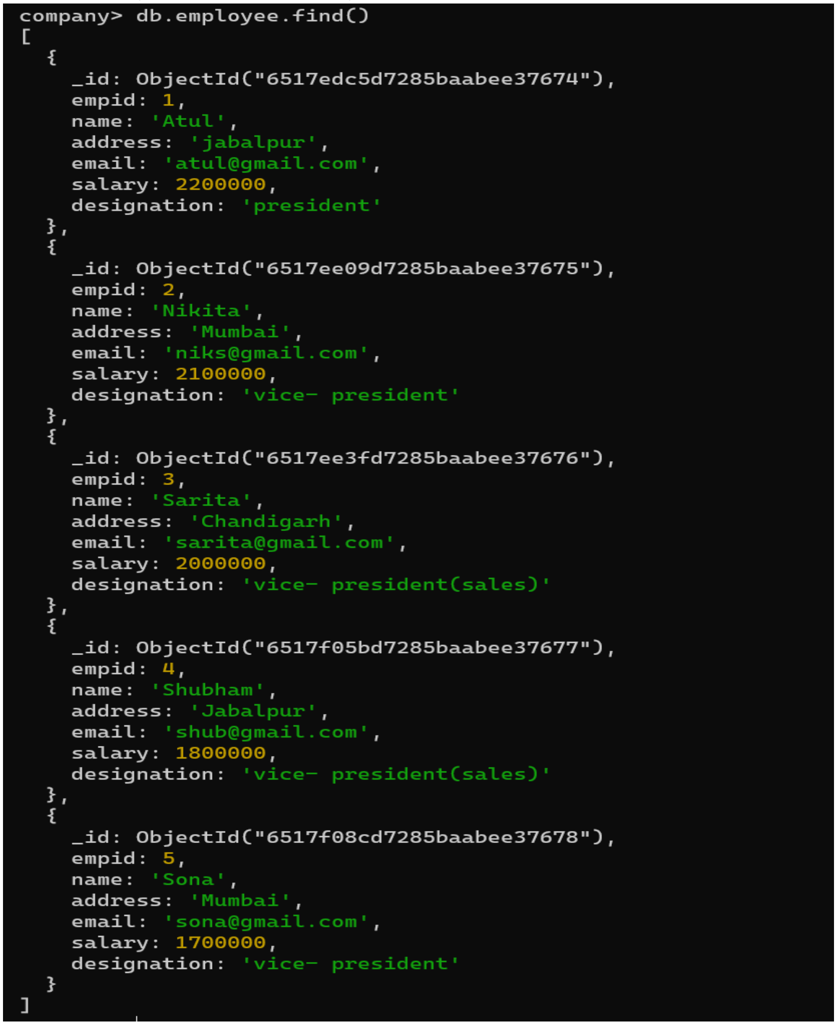
```
company>db.employee.insertOne({empid:1,name:"Atul",address:"jabalpur",email:"atul@gmail.com",salary:2200000,designation:"president"})
{
 acknowledged: true,
 insertedId: ObjectId("6517edc5d7285baabee37674")
}
company>db.employee.insertOne({empid:2,name:"Nikita",address:"Mumbai",email:"niks@gmail.com",salary:2000000,designation:"vice- president"})
{
 acknowledged: true,
 insertedId: ObjectId("6517ee09d7285baabee37675")
}
company>db.employee.insertOne({empid:3,name:"Sarita",address:"Chandigarh",email:"sarita@gmail.com",salary:2000000,designation:"vice- president(sales)"})
{
 acknowledged: true,
 insertedId: ObjectId("6517ee3fd7285baabee37676")
}
```



```
company> db.employee.insertOne({empid:1,name:"Atul",address:"ja
balpur",email:"atul@gmail.com",salary:2200000,designation:"pres
ident"})
{
  acknowledged: true,
  insertedId: ObjectId("6517edc5d7285baabee37674")
}
company> db.employee.insertOne({empid:2,name:"Nikita",address:"
Mumbai",email:"niks@gmail.com",salary:2000000,designation:"vice
- president"})
{
  acknowledged: true,
  insertedId: ObjectId("6517ee09d7285baabee37675")
}
company> db.employee.insertOne({empid:3,name:"Sarita",address:"
Chandigarh",email:"sarita@gmail.com",salary:2000000,designation
:"vice- president(sales)"})
{
  acknowledged: true,
  insertedId: ObjectId("6517ee3fd7285baabee37676")
}
```

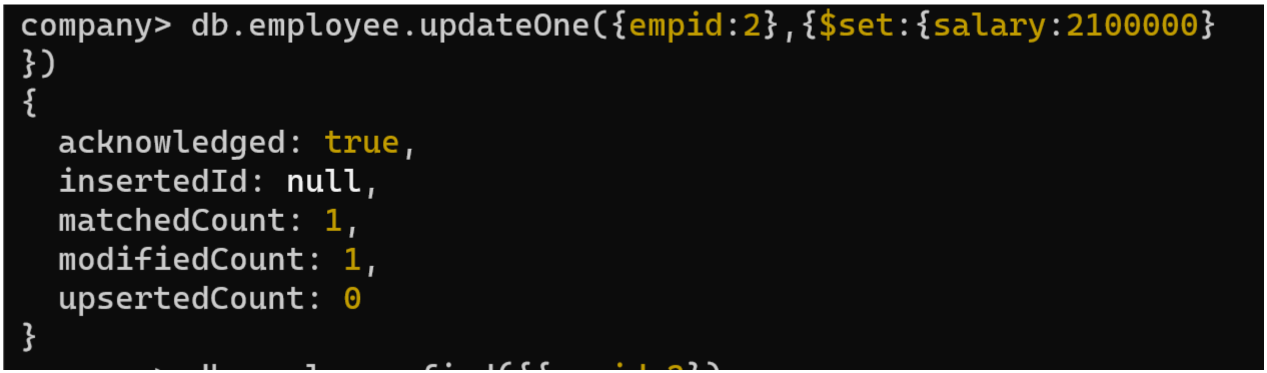
c) Display all the employee details.

company> db.employee.find()

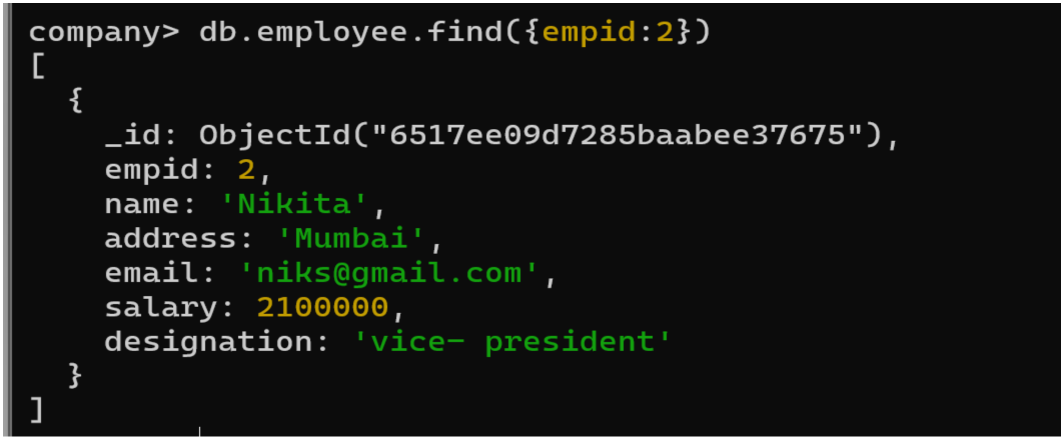


```
company> db.employee.find()
[
  {
    _id: ObjectId("6517edc5d7285baabee37674"),
    empid: 1,
    name: 'Atul',
    address: 'jabalpur',
    email: 'atul@gmail.com',
    salary: 2200000,
    designation: 'president'
  },
  {
    _id: ObjectId("6517ee09d7285baabee37675"),
    empid: 2,
    name: 'Nikita',
    address: 'Mumbai',
    email: 'niks@gmail.com',
    salary: 2100000,
    designation: 'vice- president'
  },
  {
    _id: ObjectId("6517ee3fd7285baabee37676"),
    empid: 3,
    name: 'Sarita',
    address: 'Chandigarh',
    email: 'sarita@gmail.com',
    salary: 2000000,
    designation: 'vice- president(sales)'
  },
  {
    _id: ObjectId("6517f05bd7285baabee37677"),
    empid: 4,
    name: 'Shubham',
    address: 'Jabalpur',
    email: 'shub@gmail.com',
    salary: 1800000,
    designation: 'vice- president(sales)'
  },
  {
    _id: ObjectId("6517f08cd7285baabee37678"),
    empid: 5,
    name: 'Sona',
    address: 'Mumbai',
    email: 'sona@gmail.com',
    salary: 1700000,
    designation: 'vice- president'
  }
]
```

d) Update salary of a particular employee.



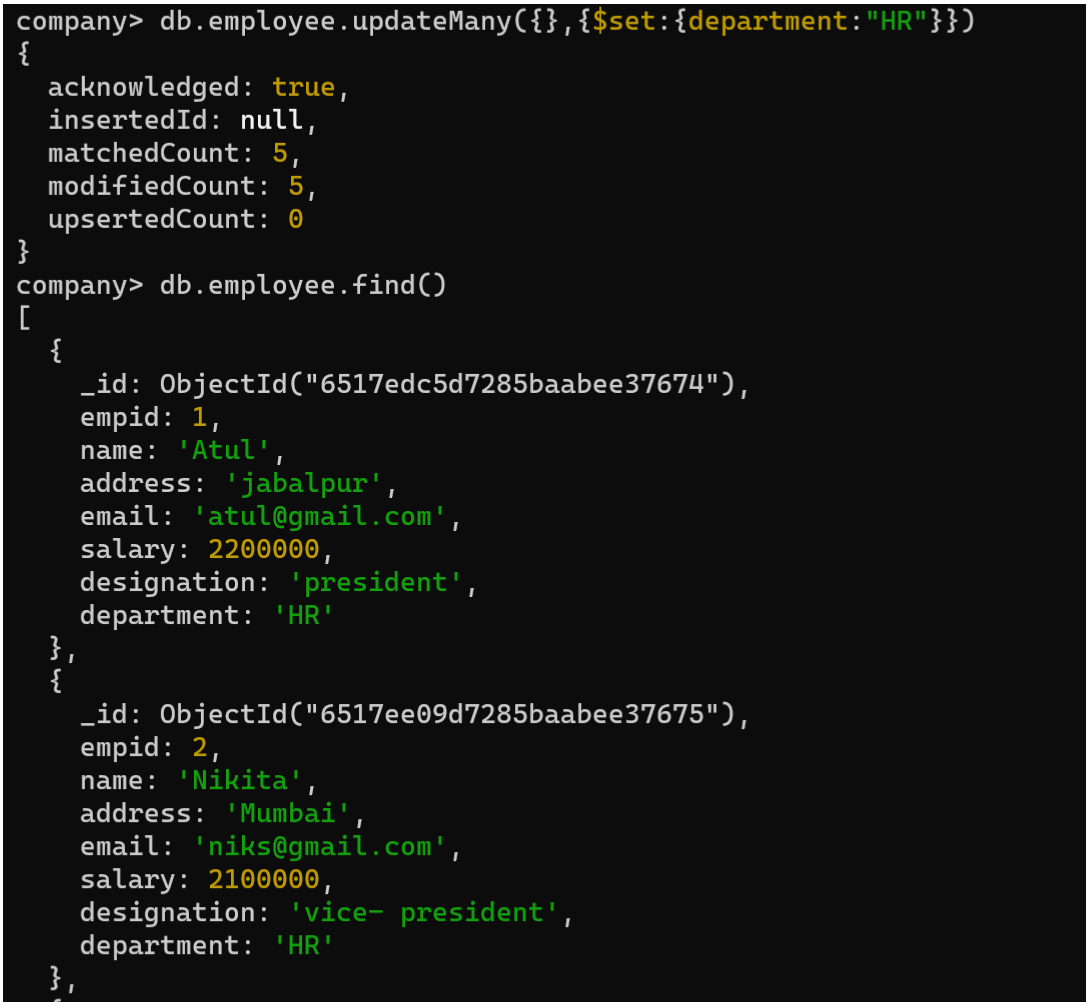
```
company> db.employee.updateOne({empid:2},{$set:{salary:2100000}
})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



```
company> db.employee.find({empid:2})
[
  {
    _id: ObjectId("6517ee09d7285baabee37675"),
    empid: 2,
    name: 'Nikita',
    address: 'Mumbai',
    email: 'niks@gmail.com',
    salary: 2100000,
    designation: 'vice- president'
  }
]
```

e) Add one more field department to the collection.

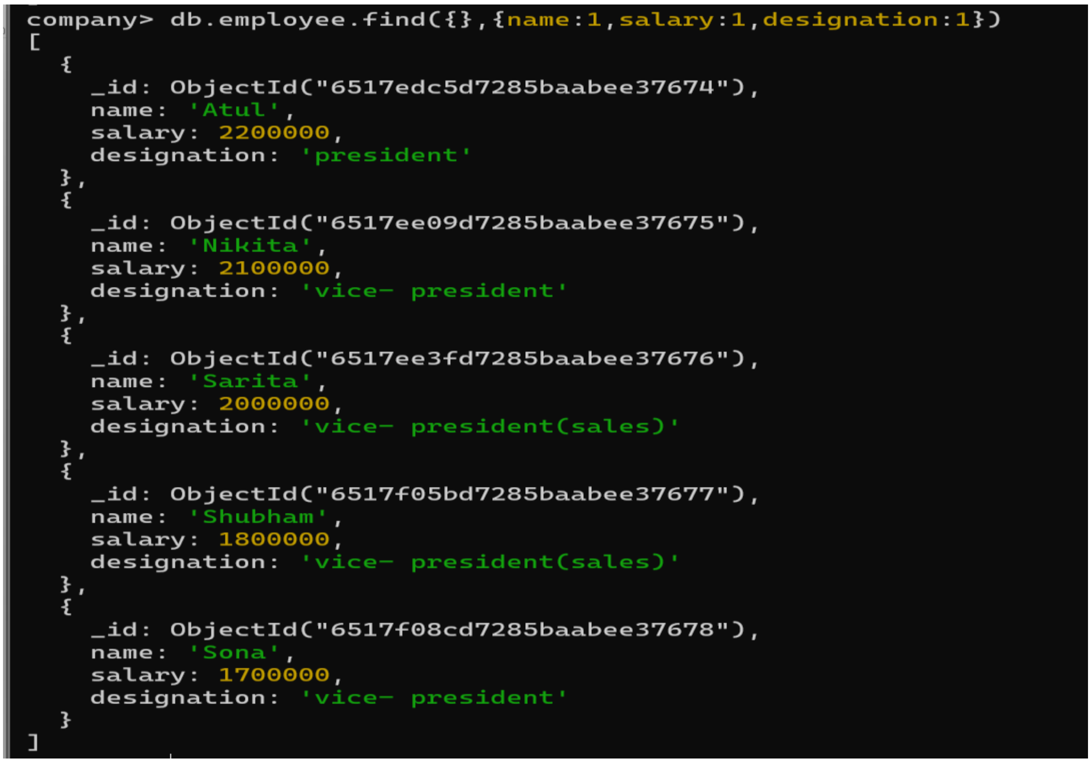
company> db.employee.updateMany({},{$set:{department:"HR"}})



```
company> db.employee.updateMany({},{$set:{department:"HR"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 5,
  modifiedCount: 5,
  upsertedCount: 0
}
company> db.employee.find()
[
  {
    _id: ObjectId("6517edc5d7285baabee37674"),
    empid: 1,
    name: 'Atul',
    address: 'jabalpur',
    email: 'atul@gmail.com',
    salary: 2200000,
    designation: 'president',
    department: 'HR'
  },
  {
    _id: ObjectId("6517ee09d7285baabee37675"),
    empid: 2,
    name: 'Nikita',
    address: 'Mumbai',
    email: 'niks@gmail.com',
    salary: 2100000,
    designation: 'vice- president',
    department: 'HR'
  },
```

f) Display the fields name, salary and designation for all the documents.

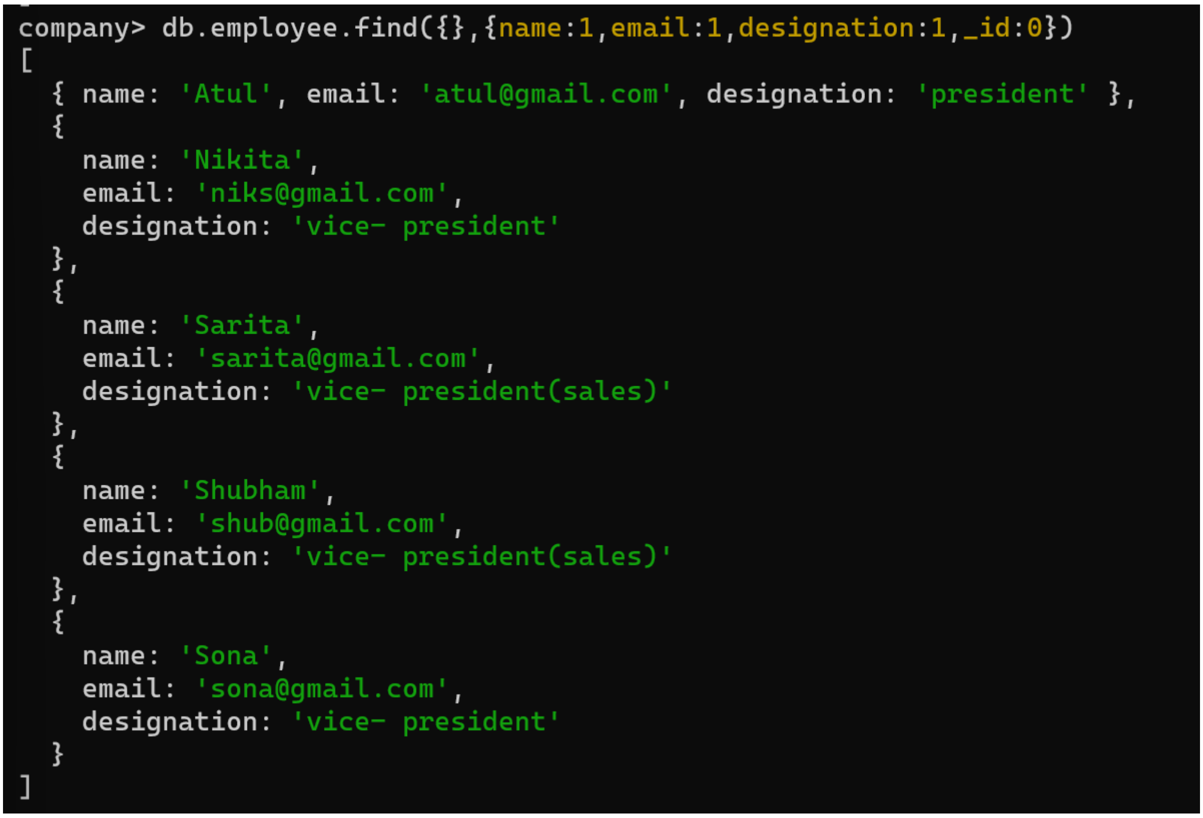
company> db.employee.find({},{name:1,salary:1,designation:1})



```
company> db.employee.find({},{name:1,salary:1,designation:1})
[
  {
    _id: ObjectId("6517edc5d7285baabee37674"),
    name: 'Atul',
    salary: 2200000,
    designation: 'president'
  },
  {
    _id: ObjectId("6517ee09d7285baabee37675"),
    name: 'Nikita',
    salary: 2100000,
    designation: 'vice- president'
  },
  {
    _id: ObjectId("6517ee3fd7285baabee37676"),
    name: 'Sarita',
    salary: 2000000,
    designation: 'vice- president(sales)'
  },
  {
    _id: ObjectId("6517f05bd7285baabee37677"),
    name: 'Shubham',
    salary: 1800000,
    designation: 'vice- president(sales)'
  },
  {
    _id: ObjectId("6517f08cd7285baabee37678"),
    name: 'Sona',
    salary: 1700000,
    designation: 'vice- president'
  }
]
```

g) Display the fields name, email and designation for all the documents but exclude the field _id.

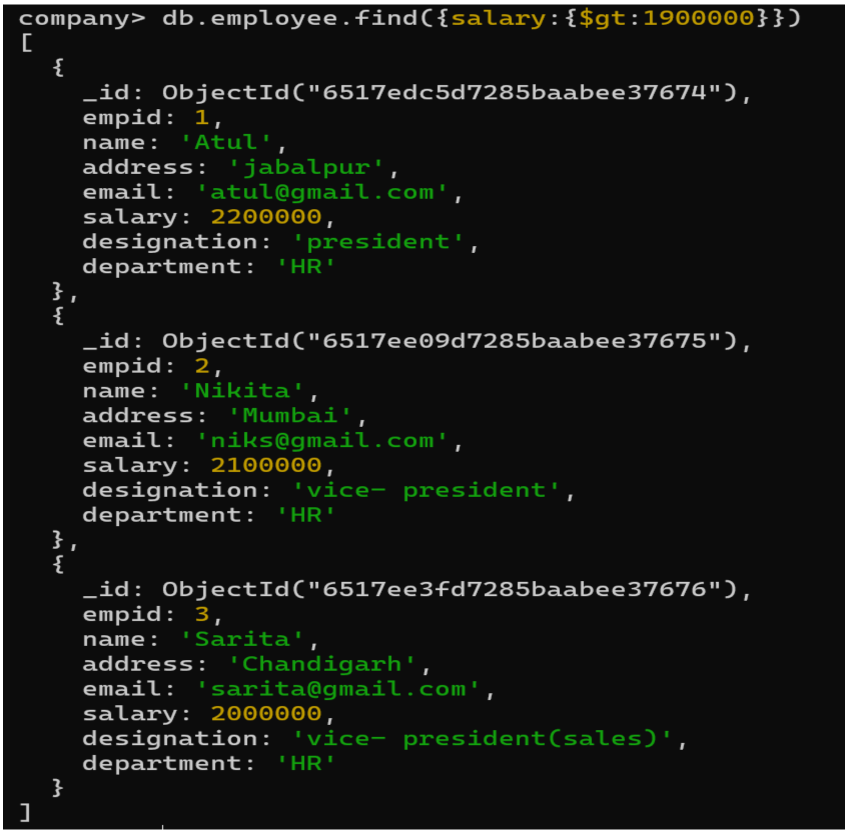
company> db.employee.find({},{name:1,email:1,designation:1,_id:0})



```
company> db.employee.find({},{name:1,email:1,designation:1,_id:0})
[
  { name: 'Atul', email: 'atul@gmail.com', designation: 'president' },
  {
    name: 'Nikita',
    email: 'niks@gmail.com',
    designation: 'vice- president'
  },
  {
    name: 'Sarita',
    email: 'sarita@gmail.com',
    designation: 'vice- president(sales)'
  },
  {
    name: 'Shubham',
    email: 'shub@gmail.com',
    designation: 'vice- president(sales)'
  },
  {
    name: 'Sona',
    email: 'sona@gmail.com',
    designation: 'vice- president'
  }
]
```

h) Display all employee details whose salary is greater than a specified value.

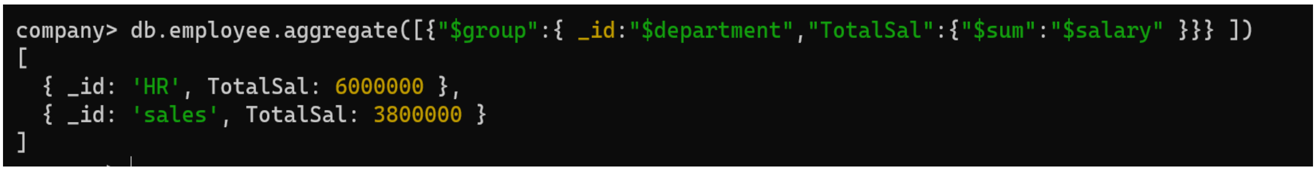
company> db.employee.find({salary:{$gt:1900000}})



```
company> db.employee.find({salary:{$gt:1900000}})
[
  {
    _id: ObjectId("6517edc5d7285baabee37674"),
    empid: 1,
    name: 'Atul',
    address: 'jabalpur',
    email: 'atul@gmail.com',
    salary: 2200000,
    designation: 'president',
    department: 'HR'
  },
  {
    _id: ObjectId("6517ee09d7285baabee37675"),
    empid: 2,
    name: 'Nikita',
    address: 'Mumbai',
    email: 'niks@gmail.com',
    salary: 2100000,
    designation: 'vice- president',
    department: 'HR'
  },
  {
    _id: ObjectId("6517ee3fd7285baabee37676"),
    empid: 3,
    name: 'Sarita',
    address: 'Chandigarh',
    email: 'sarita@gmail.com',
    salary: 2000000,
    designation: 'vice- president(sales)',
    department: 'HR'
  }
]
```

i) Find department wise total salary of employees.

db.employee.aggregate([{"$group":{_id:"$department","TotalSal":{"$sum":"$salary" }}}])

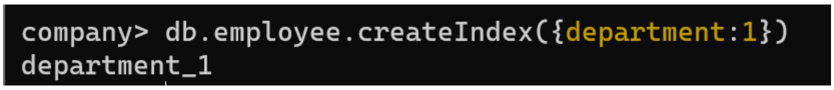


```
company> db.employee.aggregate([{"$group":{ _id:"$department","TotalSal":{"$sum":"$salary" }}} ])
[
  { _id: 'HR', TotalSal: 6000000 },
  { _id: 'sales', TotalSal: 3800000 }
]
```

j) Create an index for department field.

company> db.employee.createIndex({department:1})

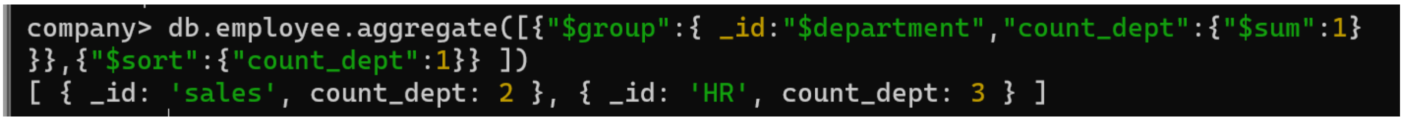
department_1



```
company> db.employee.createIndex({department:1})
department_1
```

k) Display the no: of employees belonging to each department sorted in ascending order.

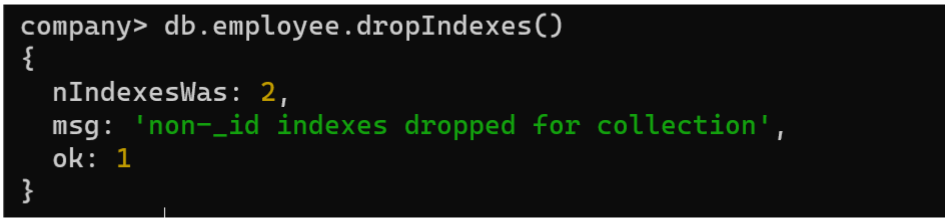
company> db.employee.aggregate([{"$group":{ _id:"$department","count_dept":{"$sum":1}}},{"$sort":{"count_dept":1}}])



```
company> db.employee.aggregate([{"$group":{ _id:"$department","count_dept":{"$sum":1}
}},{"$sort":{"count_dept":1}} ])
[ { _id: 'sales', count_dept: 2 }, { _id: 'HR', count_dept: 3 } ]
```

l) Remove all indexes from employee collection.

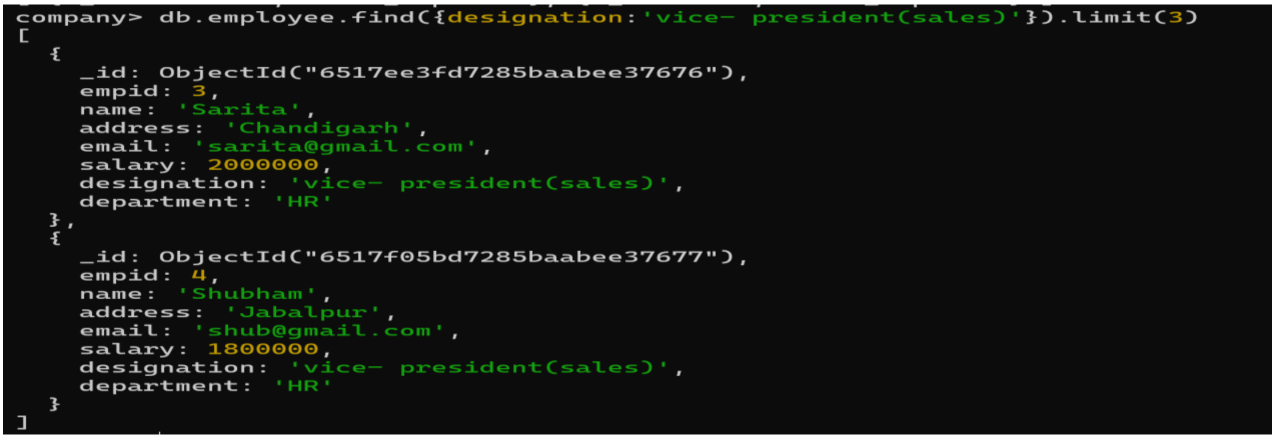
company> db.employee.dropIndexes()



```
company> db.employee.dropIndexes()
{
  nIndexesWas: 2,
  msg: 'non-_id indexes dropped for collection',
  ok: 1
}
```

m) Display only the first 3 employee details whose designation is given.

company> db.employee.find({designation:'vice- president(sales)'}).limit(3)



```
company> db.employee.find({designation:'vice- president(sales)'}).limit(3)
[
  {
    _id: ObjectId("6517ee3fd7285baabee37676"),
    empid: 3,
    name: 'Sarita',
    address: 'Chandigarh',
    email: 'sarita@gmail.com',
    salary: 2000000,
    designation: 'vice- president(sales)',
    department: 'HR'
  },
  {
    _id: ObjectId("6517f05bd7285baabee37677"),
    empid: 4,
    name: 'Shubham',
    address: 'Jabalpur',
    email: 'shub@gmail.com',
    salary: 1800000,
    designation: 'vice- president(sales)',
    department: 'HR'
  }
]
```