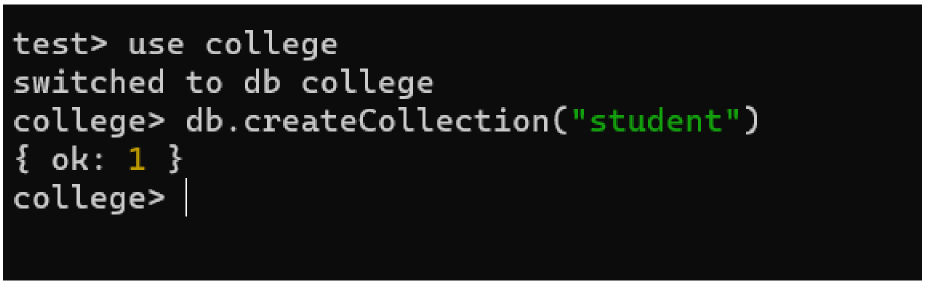
Assignment 2

Name-Harshayee Jadhav

PRN No-03

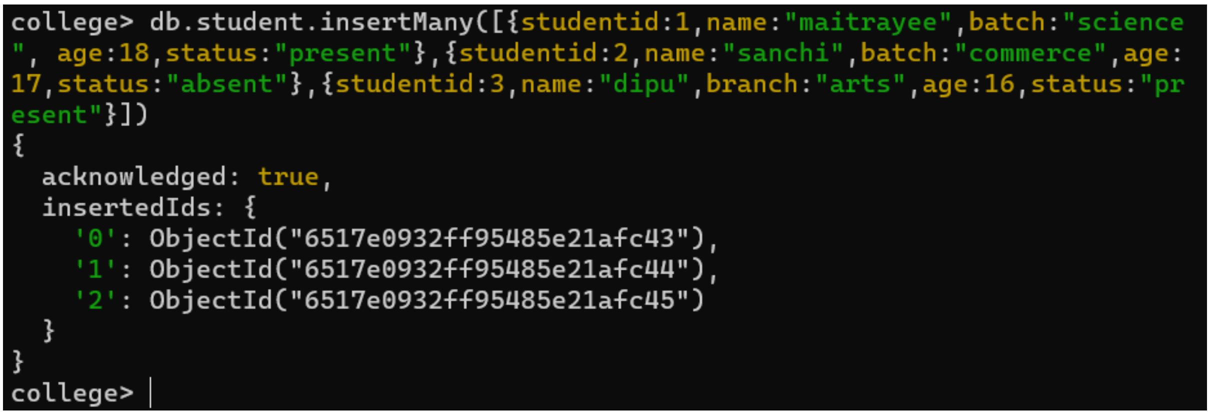
Subject-DBMS(Mongodb lab 2)

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



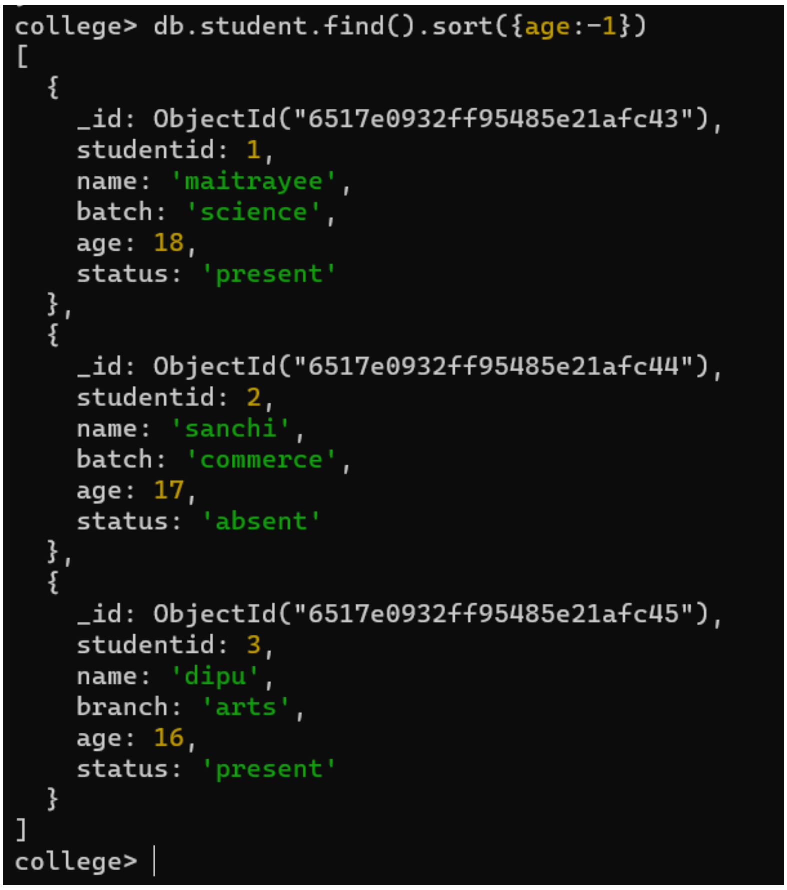
```
test> use college
switched to db college
college> db.createCollection("student")
{ ok: 1 }
college> 
```

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



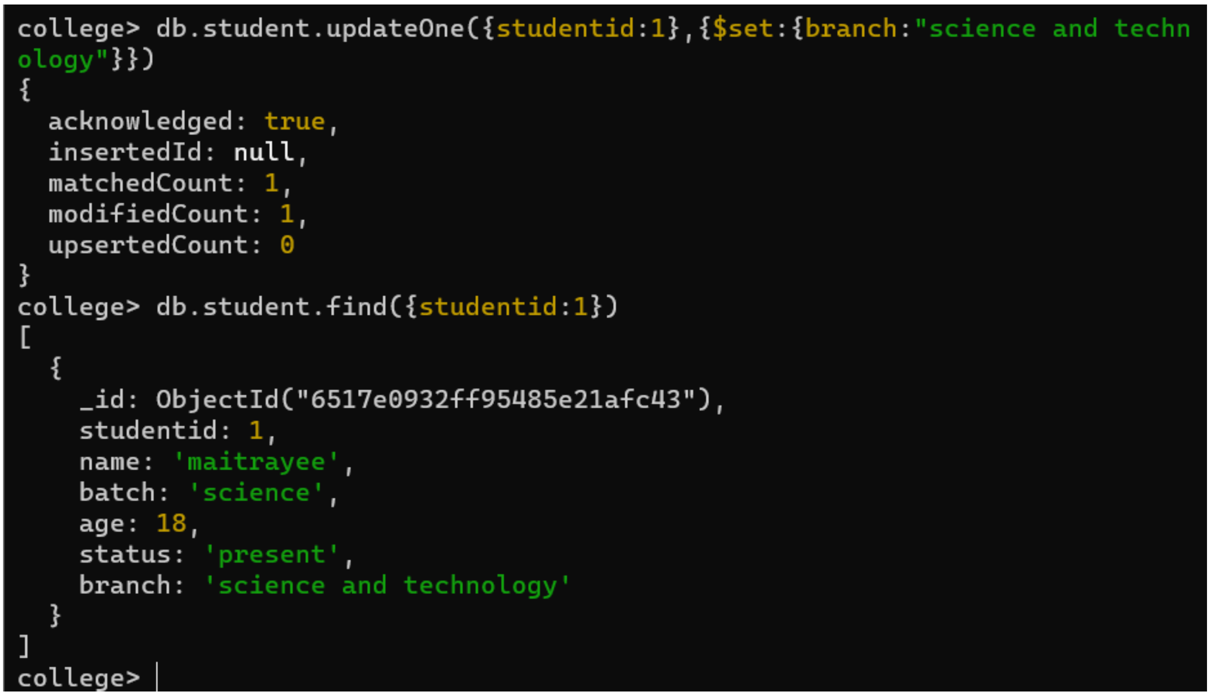
```
college> db.student.insertMany([{studentid:1,name:"maitrayee",batch:"science
", age:18,status:"present"},{studentid:2,name:"sanchi",batch:"commerce",age:
17,status:"absent"},{studentid:3,name:"dipu",branch:"arts",age:16,status:"pr
esent"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6517e0932ff95485e21afc43"),
    '1': ObjectId("6517e0932ff95485e21afc44"),
    '2': ObjectId("6517e0932ff95485e21afc45")
  }
}
college> 
```

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



```
college> db.student.find().sort({age:-1})
[
  {
    _id: ObjectId("6517e0932ff95485e21afc43"),
    studentid: 1,
    name: 'maitrayee',
    batch: 'science',
    age: 18,
    status: 'present'
  },
  {
    _id: ObjectId("6517e0932ff95485e21afc44"),
    studentid: 2,
    name: 'sanchi',
    batch: 'commerce',
    age: 17,
    status: 'absent'
  },
  {
    _id: ObjectId("6517e0932ff95485e21afc45"),
    studentid: 3,
    name: 'dipu',
    branch: 'arts',
    age: 16,
    status: 'present'
  }
]
college> 
```

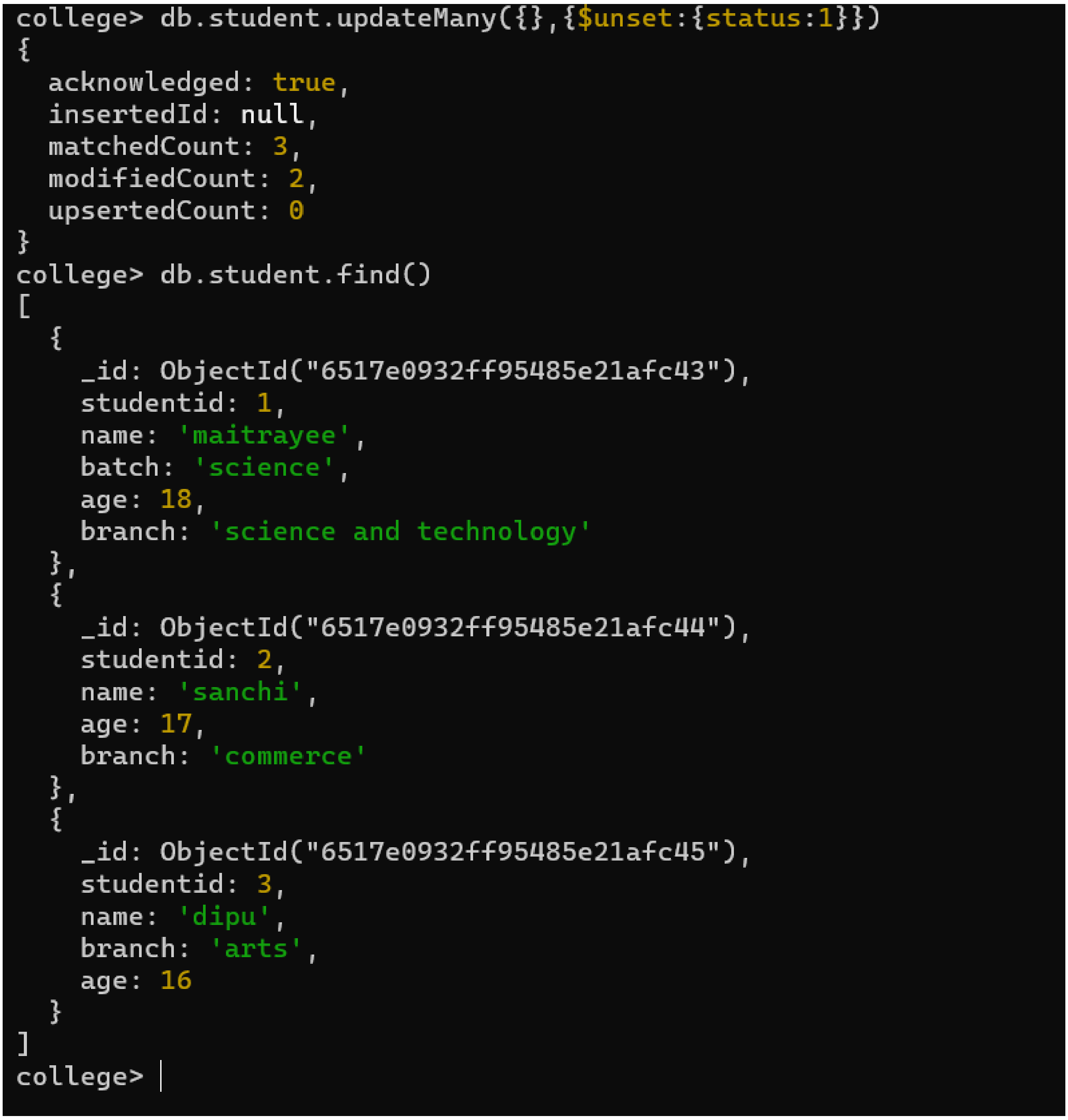
d) Update the batch-name science to science and technology



```
college> db.student.updateOne({studentid:1},{$set:{branch:"science and techn
ology"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
college> db.student.find({studentid:1})
[
  {
    _id: ObjectId("6517e0932ff95485e21afc43"),
    studentid: 1,
    name: 'maitrayee',
    batch: 'science',
    age: 18,
    status: 'present',
    branch: 'science and technology'
  }
]
college> 
```

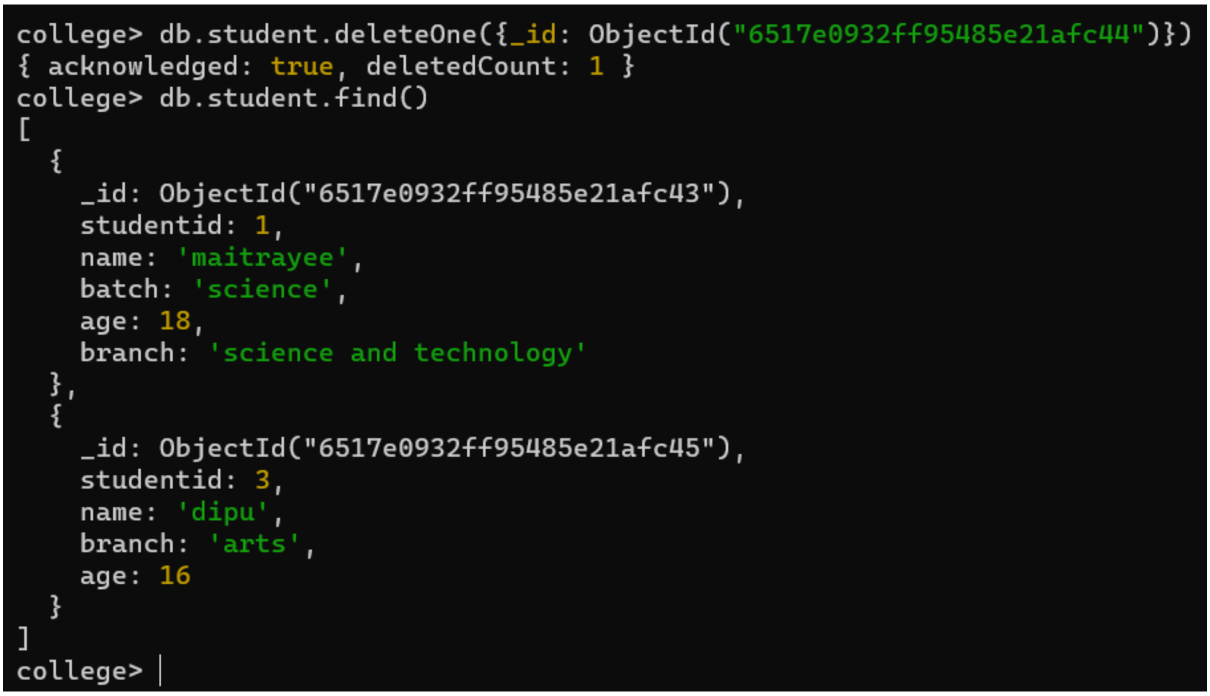
e) Count the number of students who are present.

f) Remove the status field.



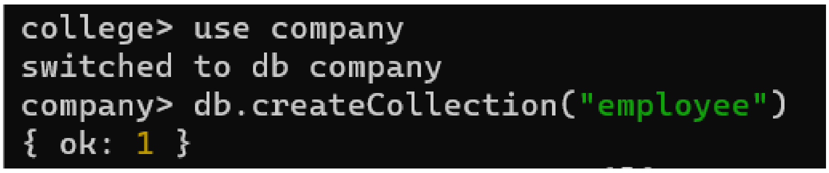
```
college> db.student.updateMany({},{$unset:{status:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 2,
  upsertedCount: 0
}
college> db.student.find()
[
  {
    _id: ObjectId("6517e0932ff95485e21afc43"),
    studentid: 1,
    name: 'maitrayee',
    batch: 'science',
    age: 18,
    branch: 'science and technology'
  },
  {
    _id: ObjectId("6517e0932ff95485e21afc44"),
    studentid: 2,
    name: 'sanchi',
    age: 17,
    branch: 'commerce'
  },
  {
    _id: ObjectId("6517e0932ff95485e21afc45"),
    studentid: 3,
    name: 'dipu',
    branch: 'arts',
    age: 16
  }
]
college> 
```

g) Remove all students from commerce batch.



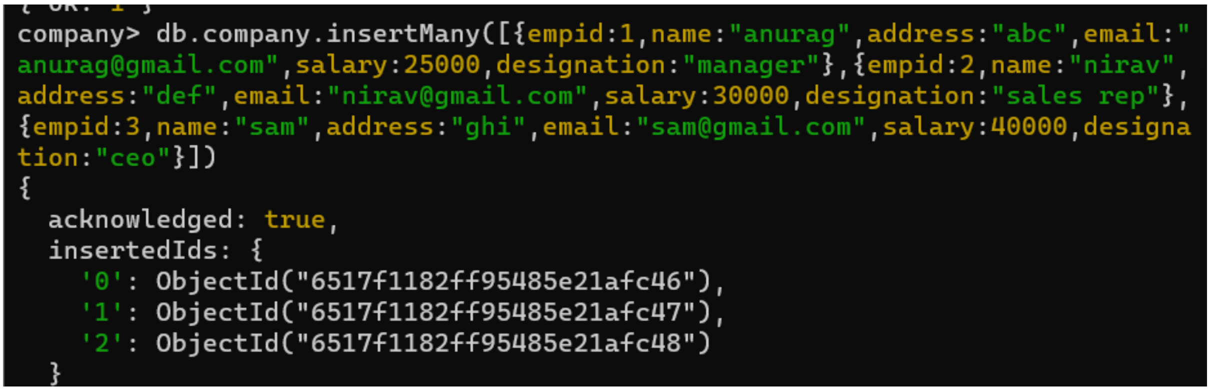
```
college> db.student.deleteOne({_id: ObjectId("6517e0932ff95485e21afc44")})
{ acknowledged: true, deletedCount: 1 }
college> db.student.find()
[
  {
    _id: ObjectId("6517e0932ff95485e21afc43"),
    studentid: 1,
    name: 'maitrayee',
    batch: 'science',
    age: 18,
    branch: 'science and technology'
  },
  {
    _id: ObjectId("6517e0932ff95485e21afc45"),
    studentid: 3,
    name: 'dipu',
    branch: 'arts',
    age: 16
  }
]
college> |
```

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



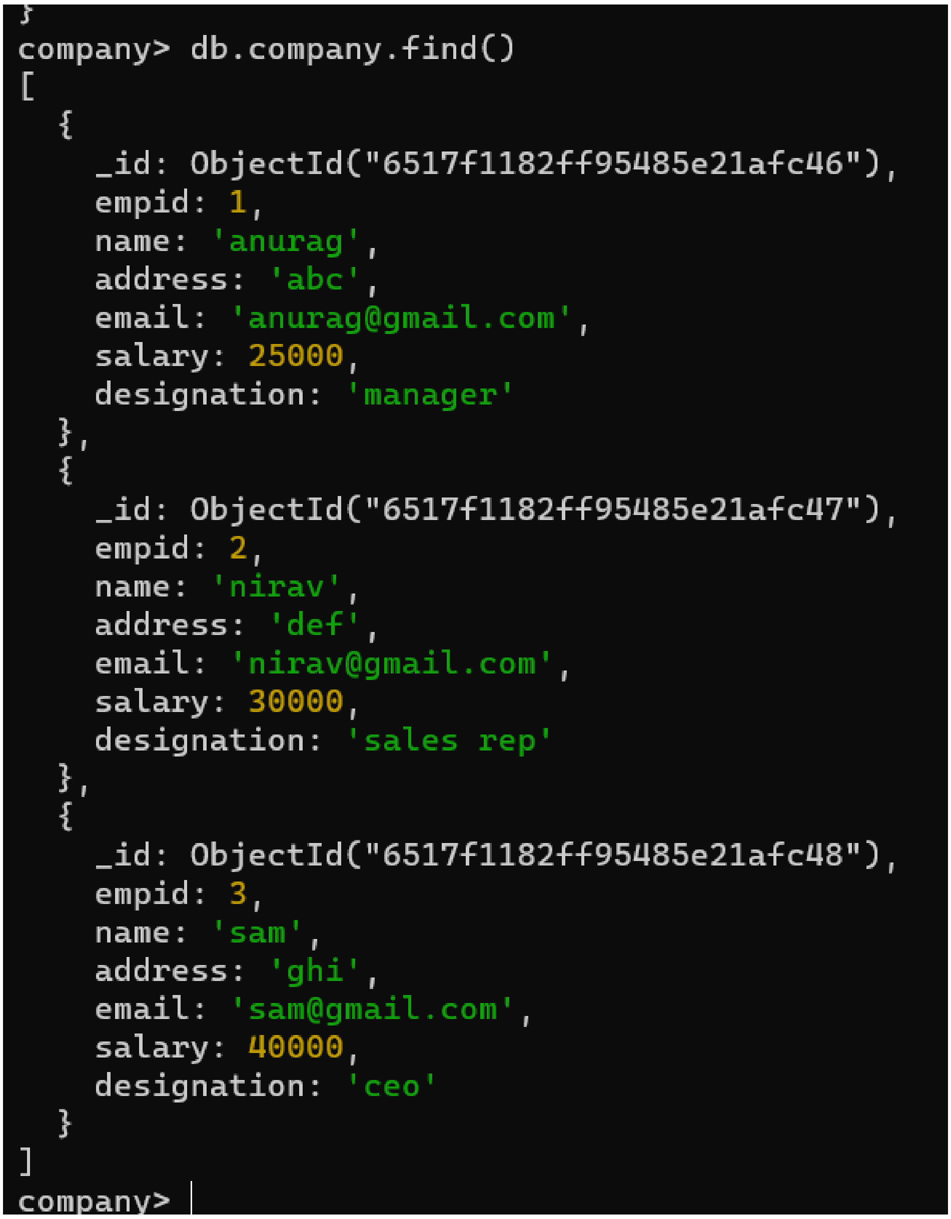
```
college> use company
switched to db company
company> db.createCollection("employee")
{ ok: 1 }
```

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



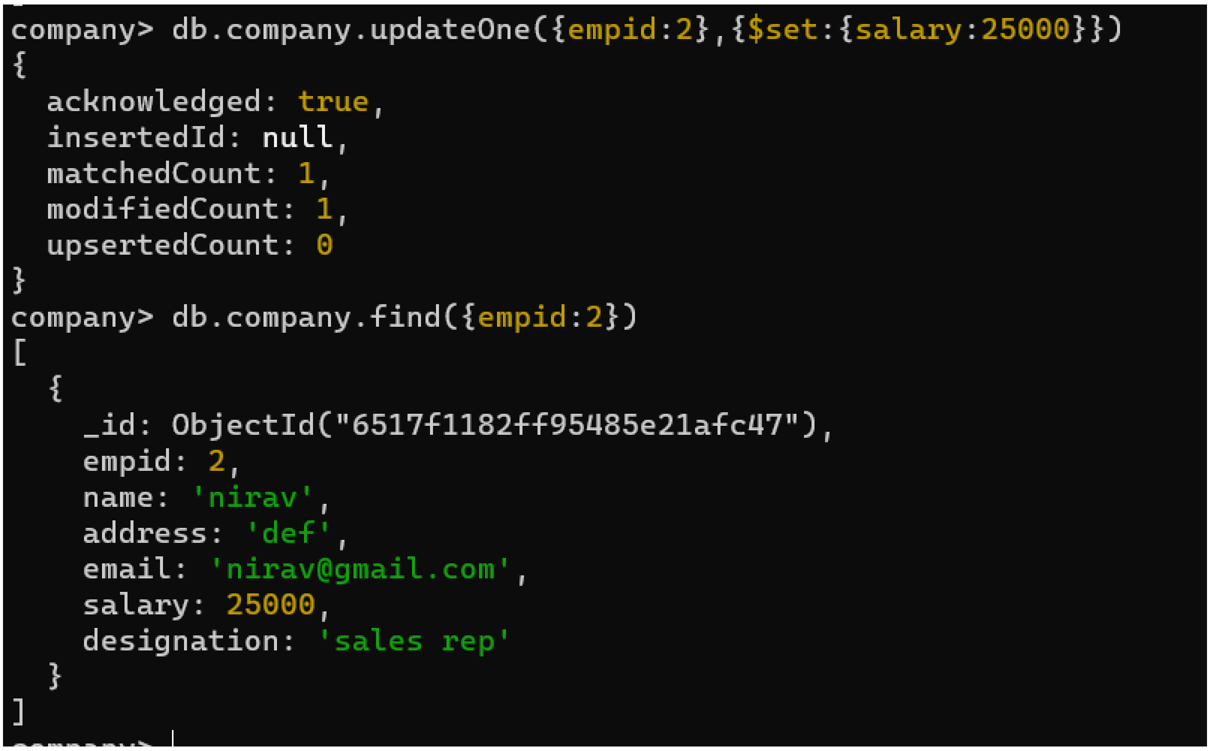
```
{ ok: 1 }
company> db.company.insertMany([{empid:1,name:"anurag",address:"abc",email:"
anurag@gmail.com",salary:25000,designation:"manager"},{empid:2,name:"nirav",
address:"def",email:"nirav@gmail.com",salary:30000,designation:"sales rep"},
{empid:3,name:"sam",address:"ghi",email:"sam@gmail.com",salary:40000,designa
tion:"ceo"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6517f1182ff95485e21afc46"),
    '1': ObjectId("6517f1182ff95485e21afc47"),
    '2': ObjectId("6517f1182ff95485e21afc48")
  }
```

c) Display all the employee details.



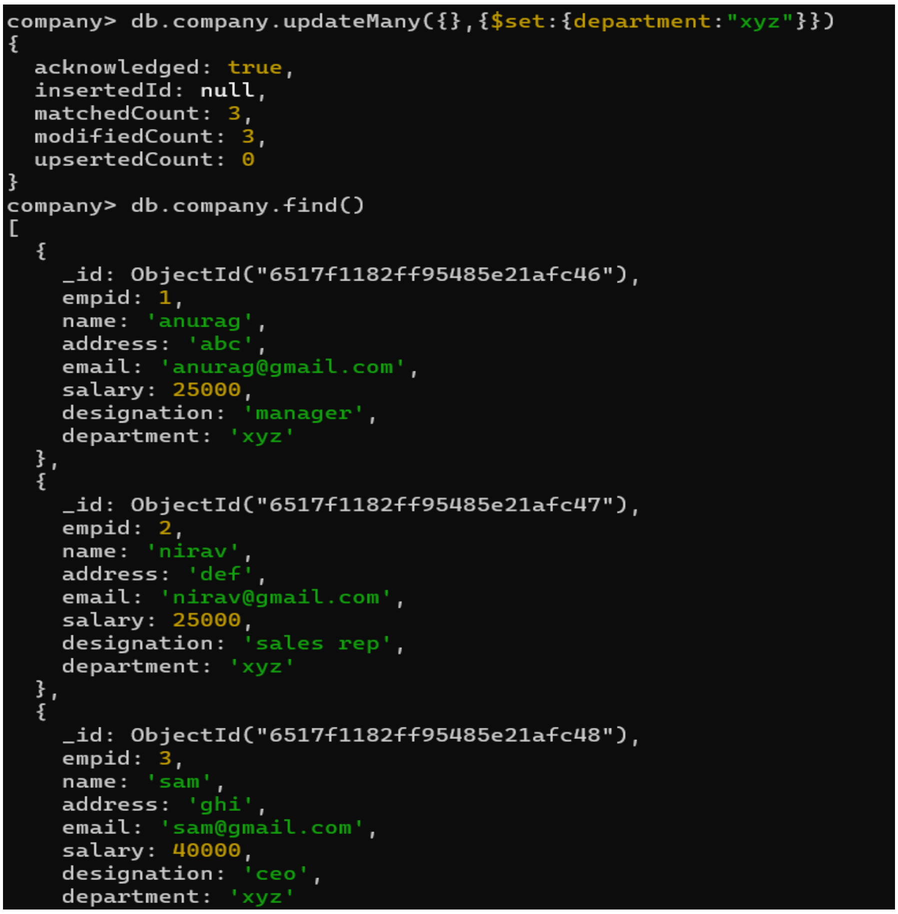
```
}
company> db.company.find()
[
  {
    _id: ObjectId("6517f1182ff95485e21afc46"),
    empid: 1,
    name: 'anurag',
    address: 'abc',
    email: 'anurag@gmail.com',
    salary: 25000,
    designation: 'manager'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    empid: 2,
    name: 'nirav',
    address: 'def',
    email: 'nirav@gmail.com',
    salary: 30000,
    designation: 'sales rep'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc48"),
    empid: 3,
    name: 'sam',
    address: 'ghi',
    email: 'sam@gmail.com',
    salary: 40000,
    designation: 'ceo'
  }
]
company> 
```

d) Update salary of a particular employee.



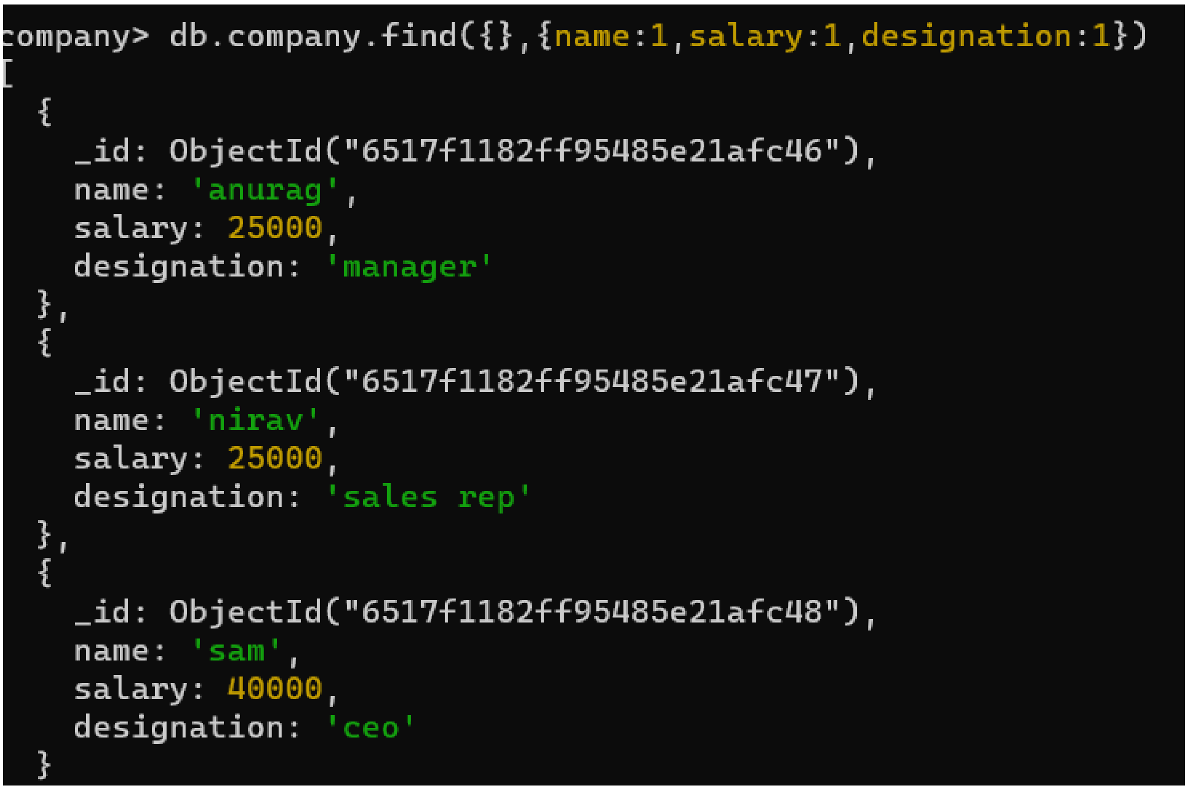
```
company> db.company.updateOne({empid:2},{$set:{salary:25000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.company.find({empid:2})
[
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    empid: 2,
    name: 'nirav',
    address: 'def',
    email: 'nirav@gmail.com',
    salary: 25000,
    designation: 'sales rep'
  }
]
company>
```

e) Add one more field department to the collection.



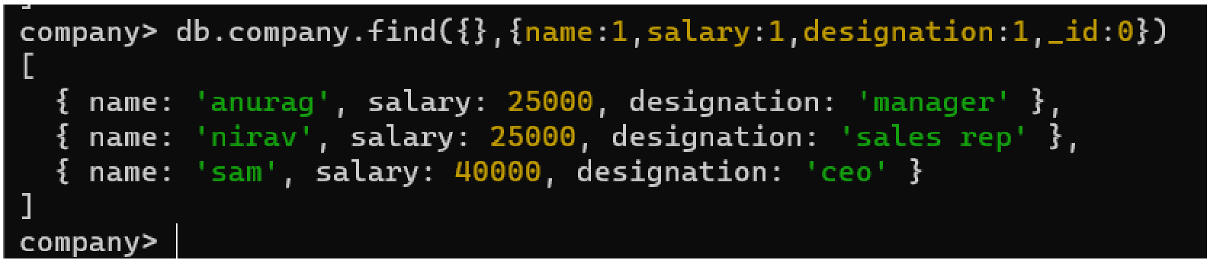
```
company> db.company.updateMany({},{$set:{department:"xyz"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
company> db.company.find()
[
  {
    _id: ObjectId("6517f1182ff95485e21afc46"),
    empid: 1,
    name: 'anurag',
    address: 'abc',
    email: 'anurag@gmail.com',
    salary: 25000,
    designation: 'manager',
    department: 'xyz'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    empid: 2,
    name: 'nirav',
    address: 'def',
    email: 'nirav@gmail.com',
    salary: 25000,
    designation: 'sales rep',
    department: 'xyz'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc48"),
    empid: 3,
    name: 'sam',
    address: 'ghi',
    email: 'sam@gmail.com',
    salary: 40000,
    designation: 'ceo',
    department: 'xyz'
```

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



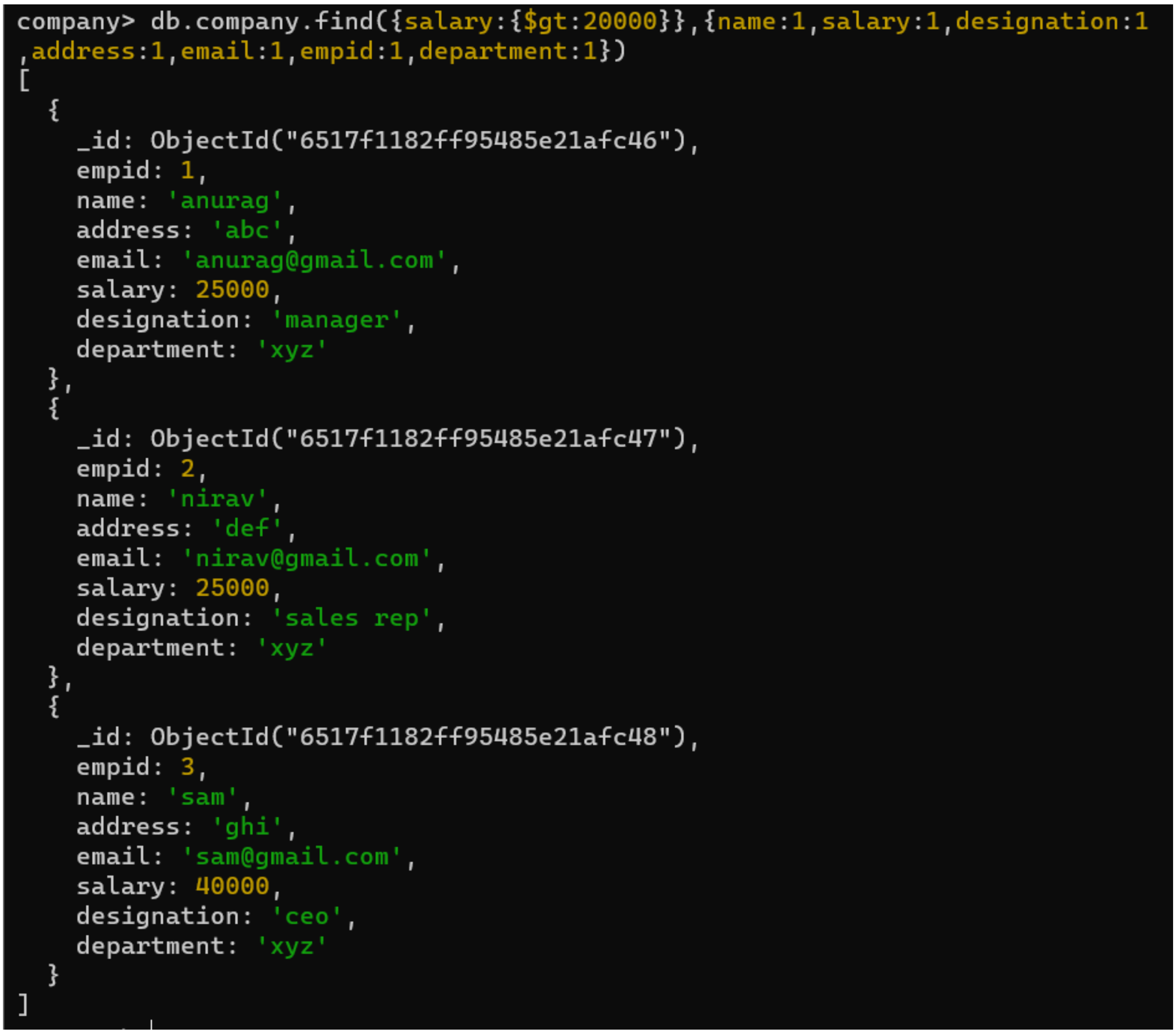
```
company> db.company.find({},{name:1,salary:1,designation:1})
[
  {
    _id: ObjectId("6517f1182ff95485e21afc46"),
    name: 'anurag',
    salary: 25000,
    designation: 'manager'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    name: 'nirav',
    salary: 25000,
    designation: 'sales rep'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc48"),
    name: 'sam',
    salary: 40000,
    designation: 'ceo'
  }
```

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



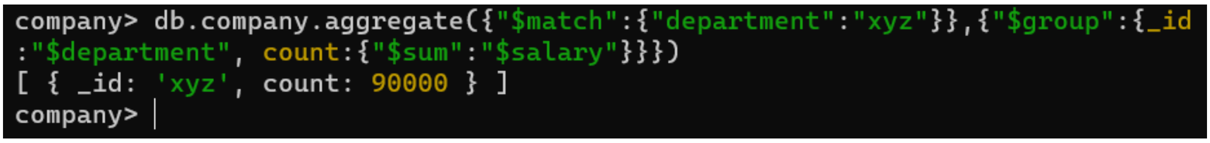
```
company> db.company.find({},{name:1,salary:1,designation:1,_id:0})
[
  { name: 'anurag', salary: 25000, designation: 'manager' },
  { name: 'nirav', salary: 25000, designation: 'sales rep' },
  { name: 'sam', salary: 40000, designation: 'ceo' }
]
company>
```

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



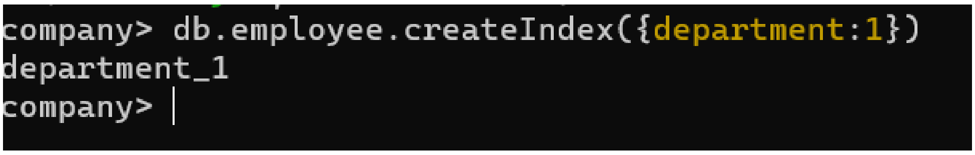
```
company> db.company.find({salary:{$gt:20000}},{name:1,salary:1,designation:1
,address:1,email:1,empid:1,department:1})
[
  {
    _id: ObjectId("6517f1182ff95485e21afc46"),
    empid: 1,
    name: 'anurag',
    address: 'abc',
    email: 'anurag@gmail.com',
    salary: 25000,
    designation: 'manager',
    department: 'xyz'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    empid: 2,
    name: 'nirav',
    address: 'def',
    email: 'nirav@gmail.com',
    salary: 25000,
    designation: 'sales rep',
    department: 'xyz'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc48"),
    empid: 3,
    name: 'sam',
    address: 'ghi',
    email: 'sam@gmail.com',
    salary: 40000,
    designation: 'ceo',
    department: 'xyz'
  }
]
```

i)Find department wise total salary of employees.



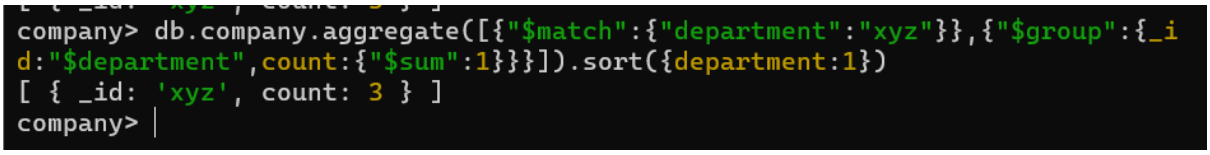
```
company> db.company.aggregate({"$match":{"department":"xyz"}},{"$group":{_id
:"$department", count:{"$sum":"$salary"}}})
[ { _id: 'xyz', count: 90000 } ]
company> |
```

j) Create an index for department field.



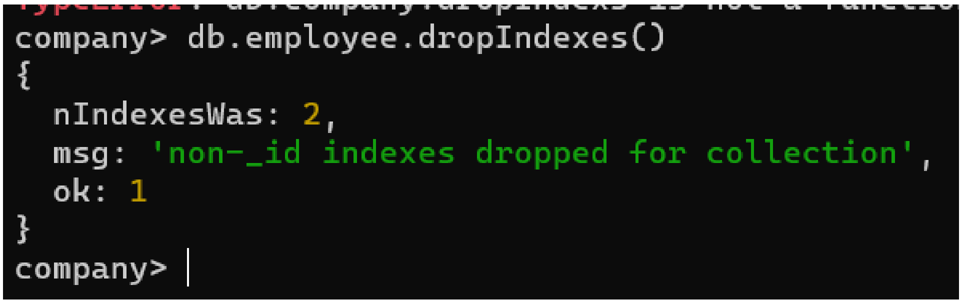
```
company> db.employee.createIndex({department:1})
department_1
company> |
```

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



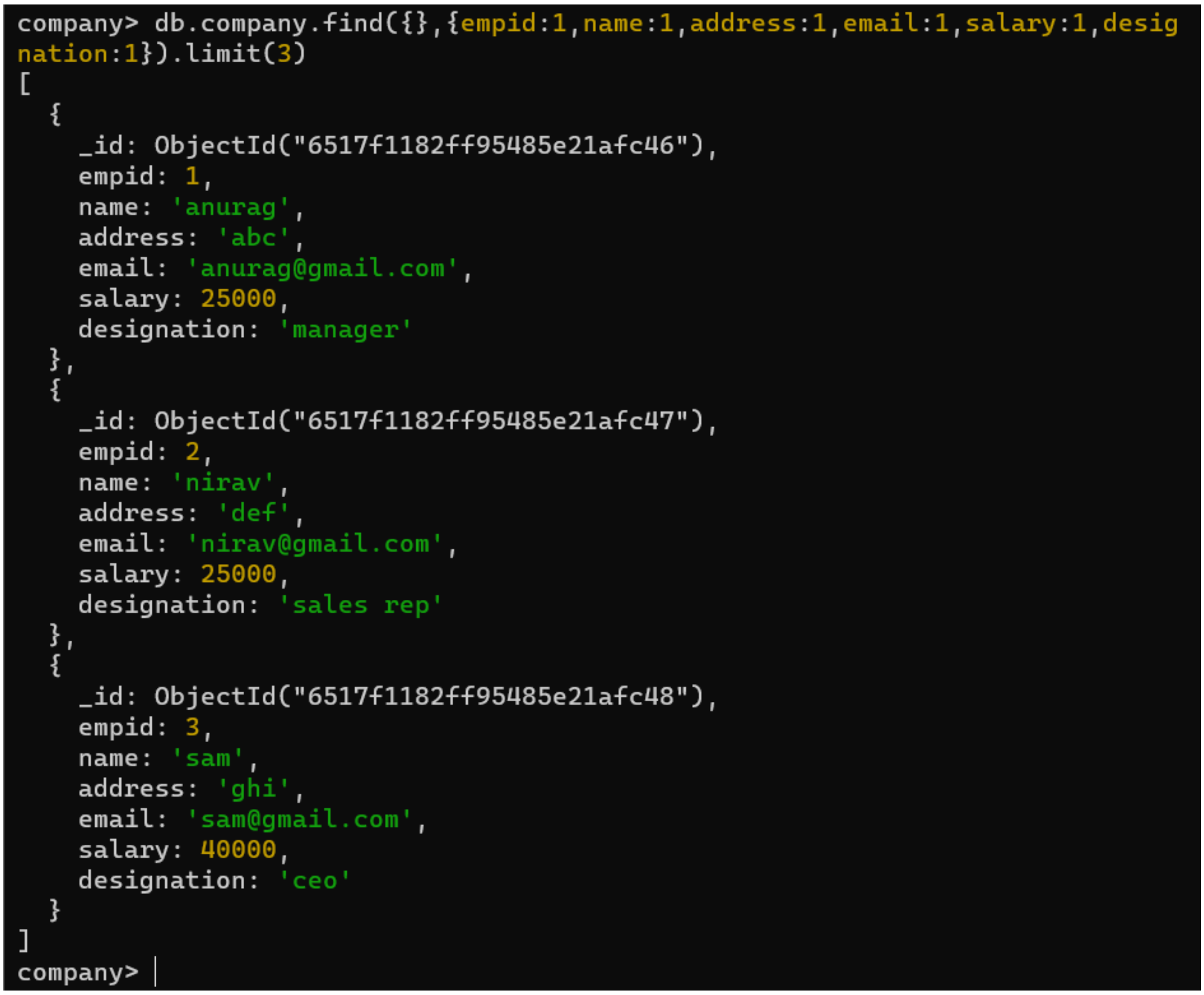
```
company> db.company.aggregate([{"$match":{"department":"xyz"}},{"$group":{_i
d:"$department",count:{"$sum":1}}}]).sort({department:1})
[ { _id: 'xyz', count: 3 } ]
company> |
```

l) Remove all indexes from employee collection.



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

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



```
company> db.company.find({},{empid:1,name:1,address:1,email:1,salary:1,desig
nation:1}).limit(3)
[
  {
    _id: ObjectId("6517f1182ff95485e21afc46"),
    empid: 1,
    name: 'anurag',
    address: 'abc',
    email: 'anurag@gmail.com',
    salary: 25000,
    designation: 'manager'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc47"),
    empid: 2,
    name: 'nirav',
    address: 'def',
    email: 'nirav@gmail.com',
    salary: 25000,
    designation: 'sales rep'
  },
  {
    _id: ObjectId("6517f1182ff95485e21afc48"),
    empid: 3,
    name: 'sam',
    address: 'ghi',
    email: 'sam@gmail.com',
    salary: 40000,
    designation: 'ceo'
  }
]
company> |
```