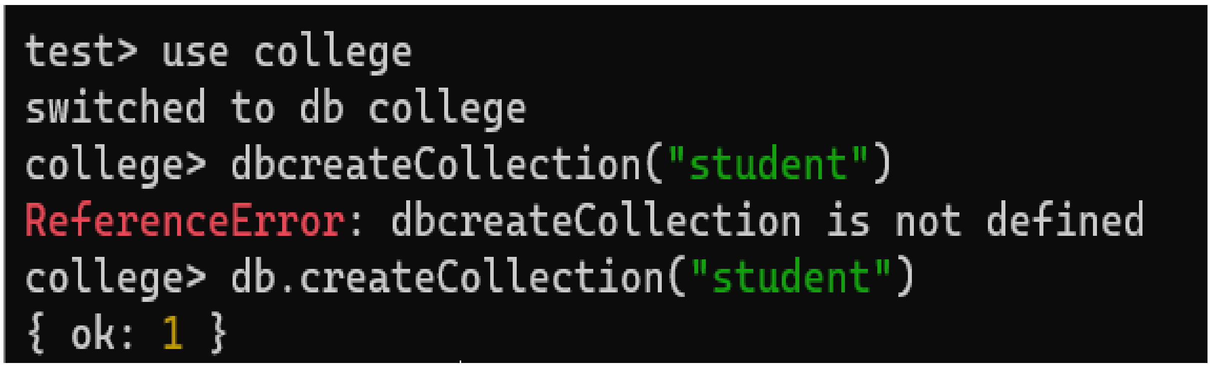
MongoDB Lab – 2

Q1)

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

Ans=

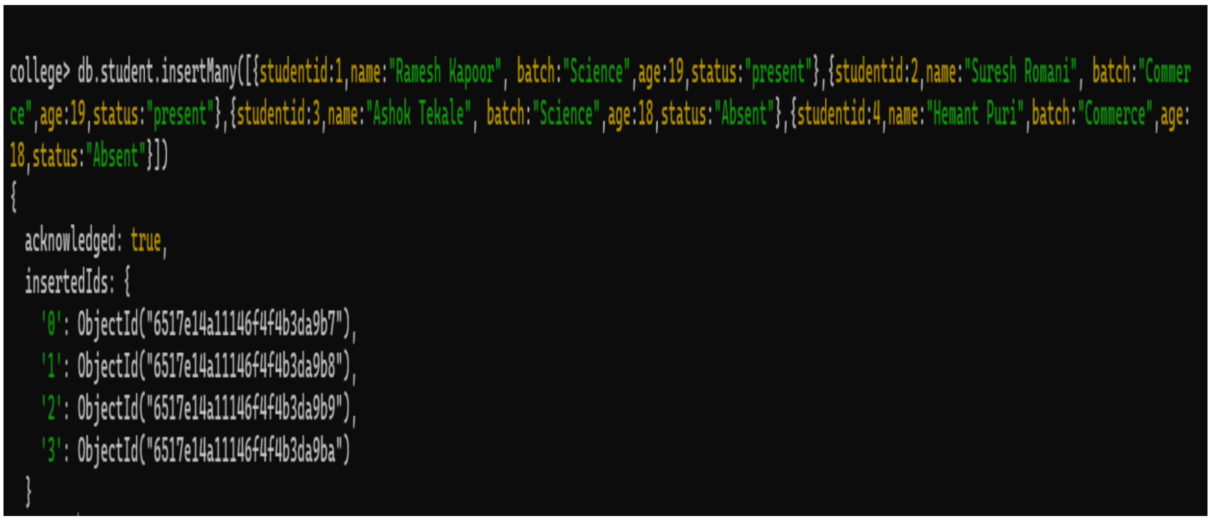


```
test> use college
switched to db college
college> dbcreateCollection("student")
ReferenceError: dbcreateCollection is not defined
college> db.createCollection("student")
{ ok: 1 }
```

b) Insert some documents to the collection with fields

studentid, name, batch(Science ,Commerce etc), age, status(present/absent).

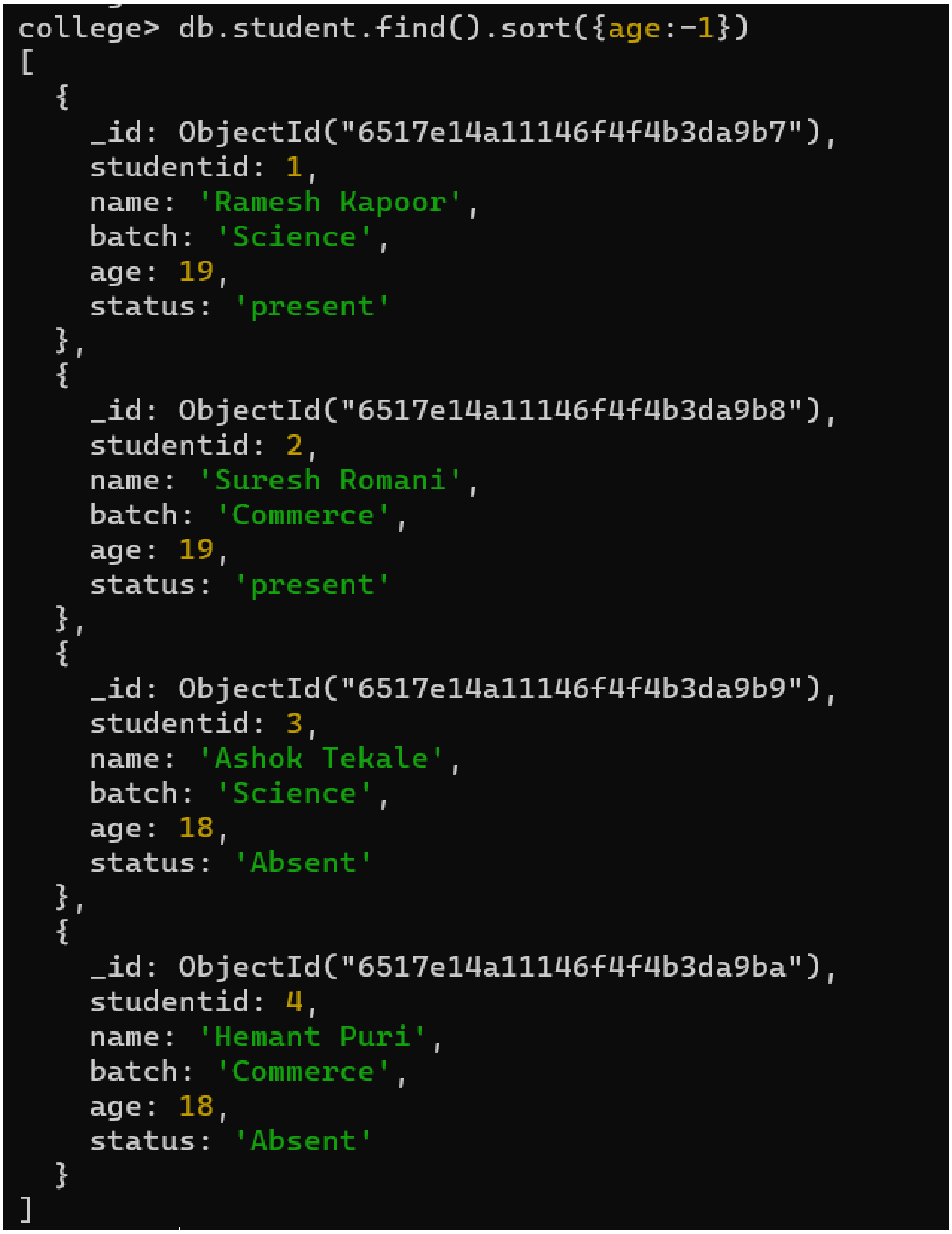
Ans=



```
college> db.student.insertMany([{studentid:1,name:"Ramesh Kapoor", batch:"Science",age:19,status:"present"},{studentid:2,name:"Suresh Romani", batch:"Commer
ce",age:19,status:"present"},{studentid:3,name:"Ashok Tekale", batch:"Science",age:18,status:"Absent"},{studentid:4,name:"Hemant Puri",batch:"Commerce",age:
18,status:"Absent"}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6517e14a11146f4f4b3da9b7"),
    '1': ObjectId("6517e14a11146f4f4b3da9b8"),
    '2': ObjectId("6517e14a11146f4f4b3da9b9"),
    '3': ObjectId("6517e14a11146f4f4b3da9ba")
  }
```

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

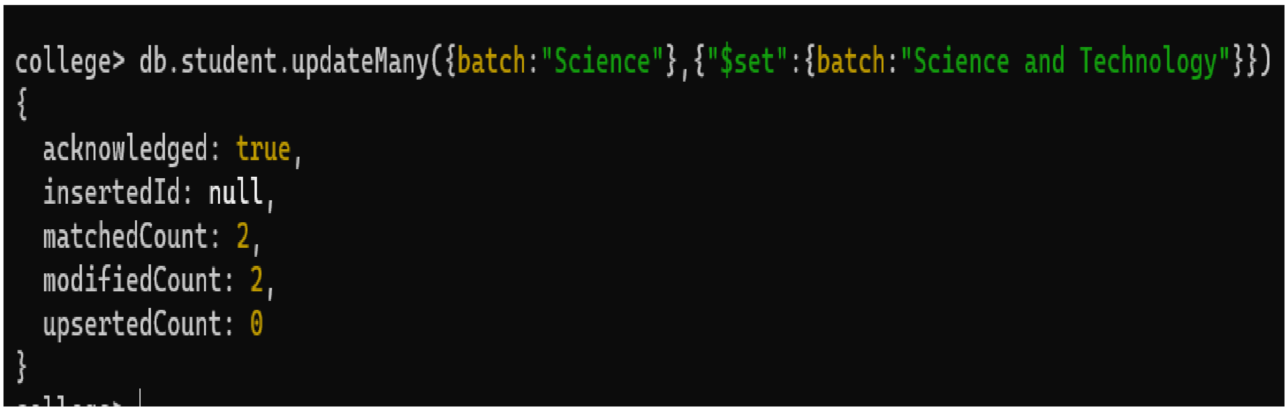
Ans=



```
college> db.student.find().sort({age:-1})
[
  {
    _id: ObjectId("6517e14a11146f4f4b3da9b7"),
    studentid: 1,
    name: 'Ramesh Kapoor',
    batch: 'Science',
    age: 19,
    status: 'present'
  },
  {
    _id: ObjectId("6517e14a11146f4f4b3da9b8"),
    studentid: 2,
    name: 'Suresh Romani',
    batch: 'Commerce',
    age: 19,
    status: 'present'
  },
  {
    _id: ObjectId("6517e14a11146f4f4b3da9b9"),
    studentid: 3,
    name: 'Ashok Tekale',
    batch: 'Science',
    age: 18,
    status: 'Absent'
  },
  {
    _id: ObjectId("6517e14a11146f4f4b3da9ba"),
    studentid: 4,
    name: 'Hemant Puri',
    batch: 'Commerce',
    age: 18,
    status: 'Absent'
  }
]
```

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

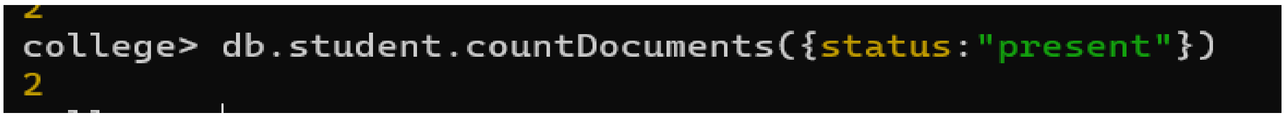
Ans=



```
college> db.student.updateMany({batch:"Science"},{"$set":{batch:"Science and Technology"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
```

e) Count the number of students who are present.

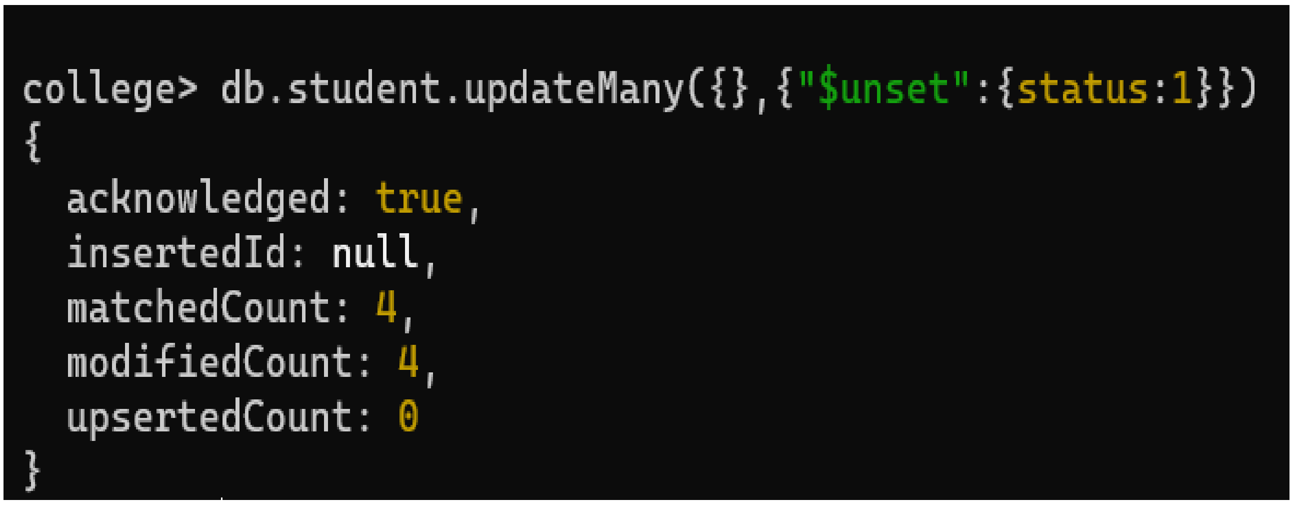
Ans=



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

f) Remove the status field.

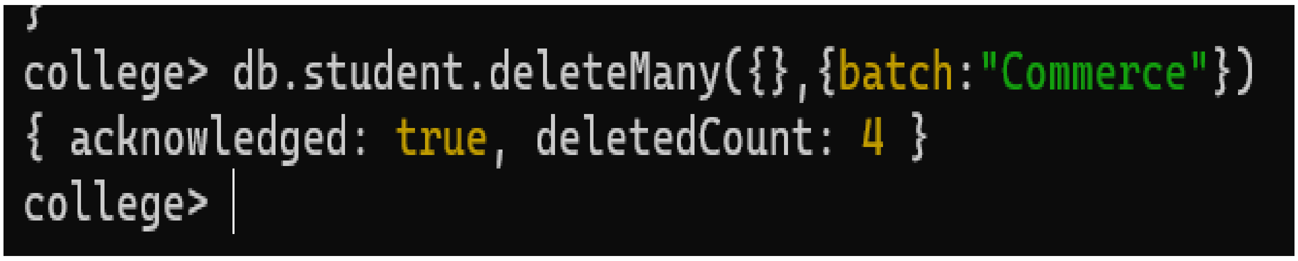
Ans=



```
college> db.student.updateMany({},{"$unset":{status:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

g) Remove all students from commerce batch.

Ans=

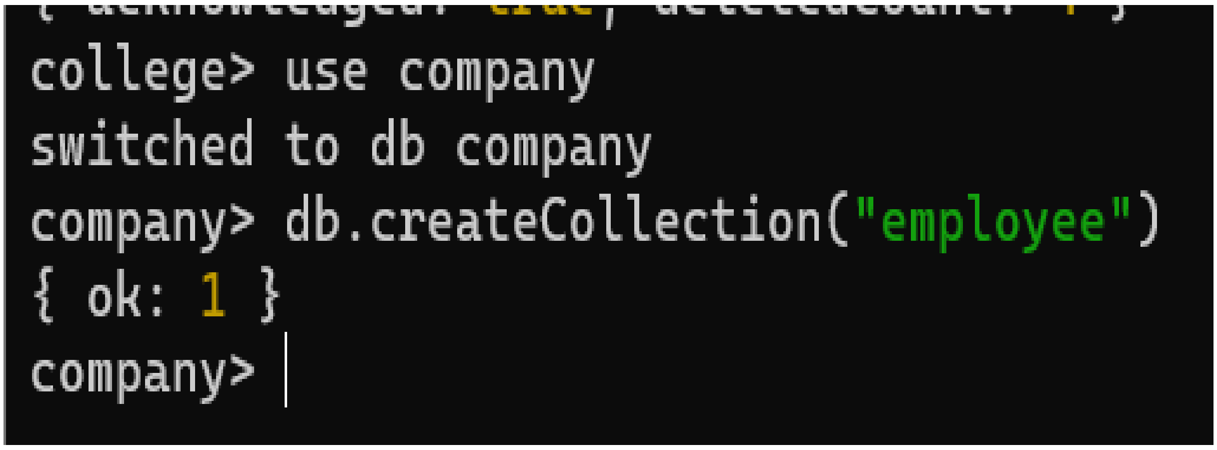


```
college> db.student.deleteMany({},{batch:"Commerce"})
{ acknowledged: true, deletedCount: 4 }
college>
```

Q2).

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

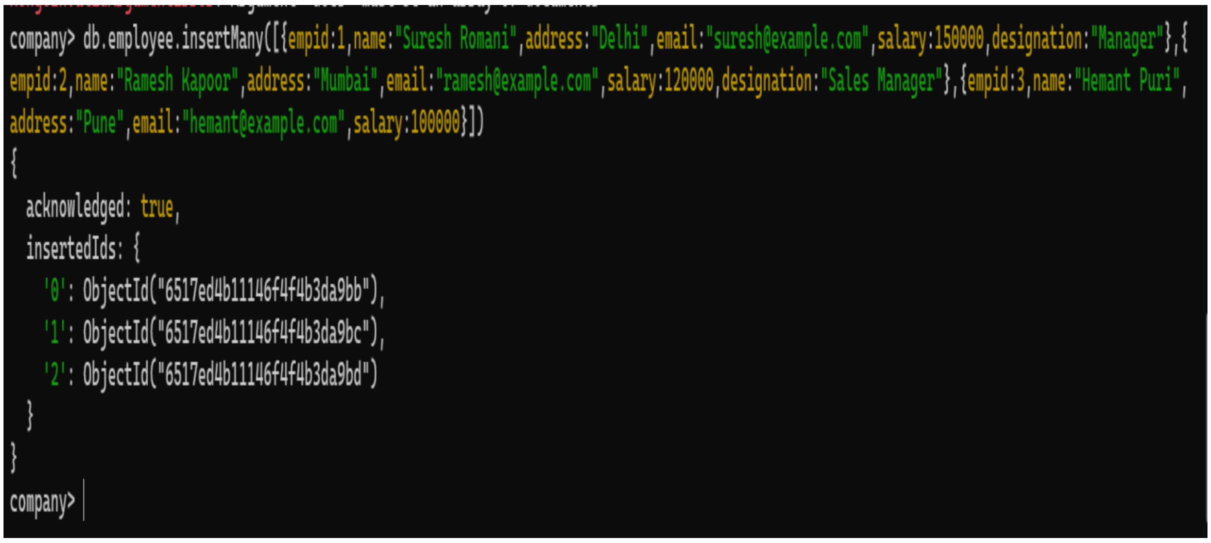
Ans=



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

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

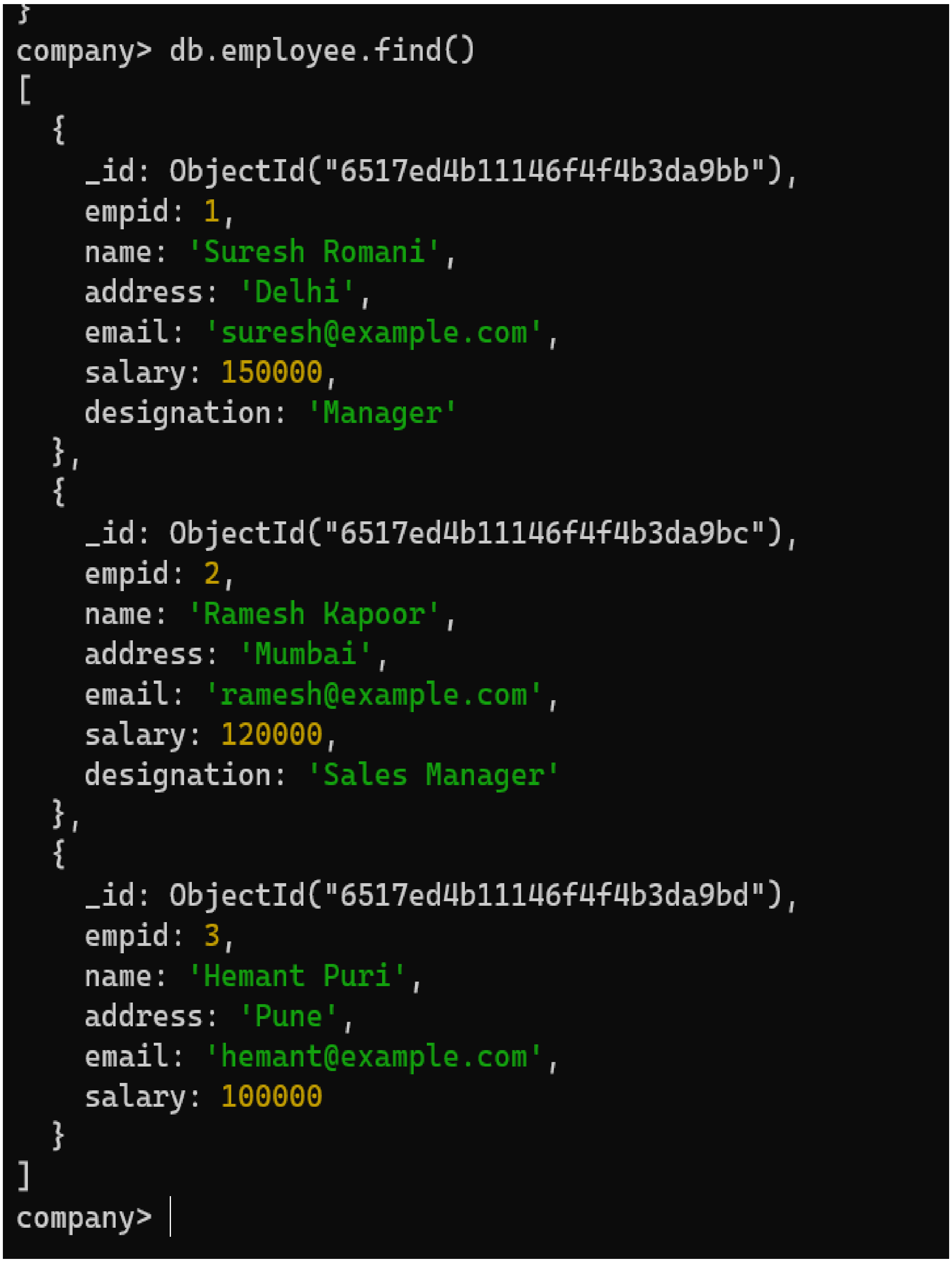
Ans=



```
company> db.employee.insertMany([{empid:1,name:"Suresh Romani",address:"Delhi",email:"suresh@example.com",salary:150000,designation:"Manager"},{
empid:2,name:"Ramesh Kapoor",address:"Mumbai",email:"ramesh@example.com",salary:120000,designation:"Sales Manager"},{empid:3,name:"Hemant Puri",
address:"Pune",email:"hemant@example.com",salary:100000}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("6517ed4b11146f4f4b3da9bb"),
    '1': ObjectId("6517ed4b11146f4f4b3da9bc"),
    '2': ObjectId("6517ed4b11146f4f4b3da9bd")
  }
}
company> 
```

c) Display all the employee details.

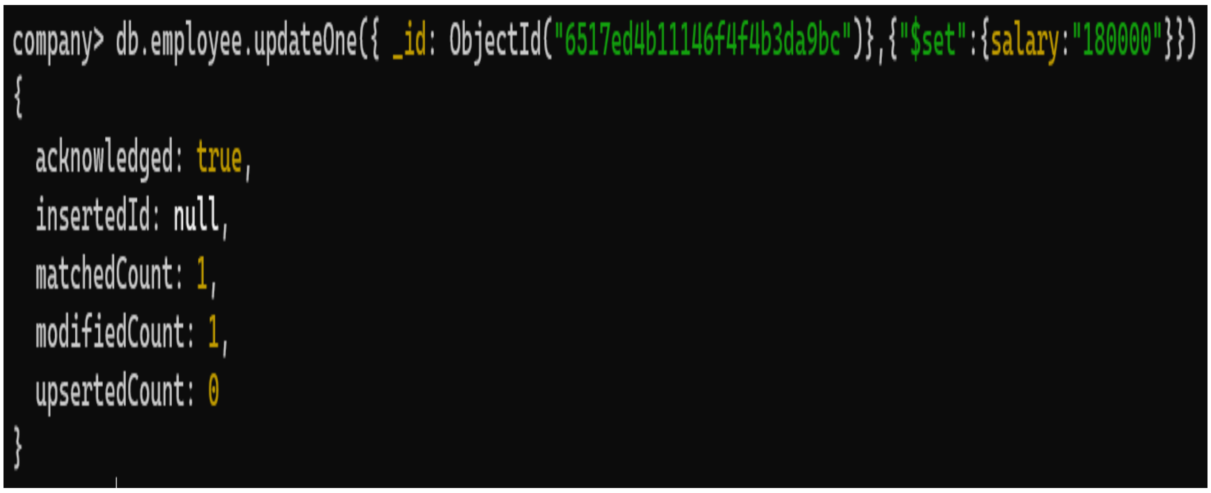
Ans=



```
}
company> db.employee.find()
[
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bb"),
    empid: 1,
    name: 'Suresh Romani',
    address: 'Delhi',
    email: 'suresh@example.com',
    salary: 150000,
    designation: 'Manager'
  },
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bc"),
    empid: 2,
    name: 'Ramesh Kapoor',
    address: 'Mumbai',
    email: 'ramesh@example.com',
    salary: 120000,
    designation: 'Sales Manager'
  },
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bd"),
    empid: 3,
    name: 'Hemant Puri',
    address: 'Pune',
    email: 'hemant@example.com',
    salary: 100000
  }
]
company> 
```

d) Update salary of a particular employee.

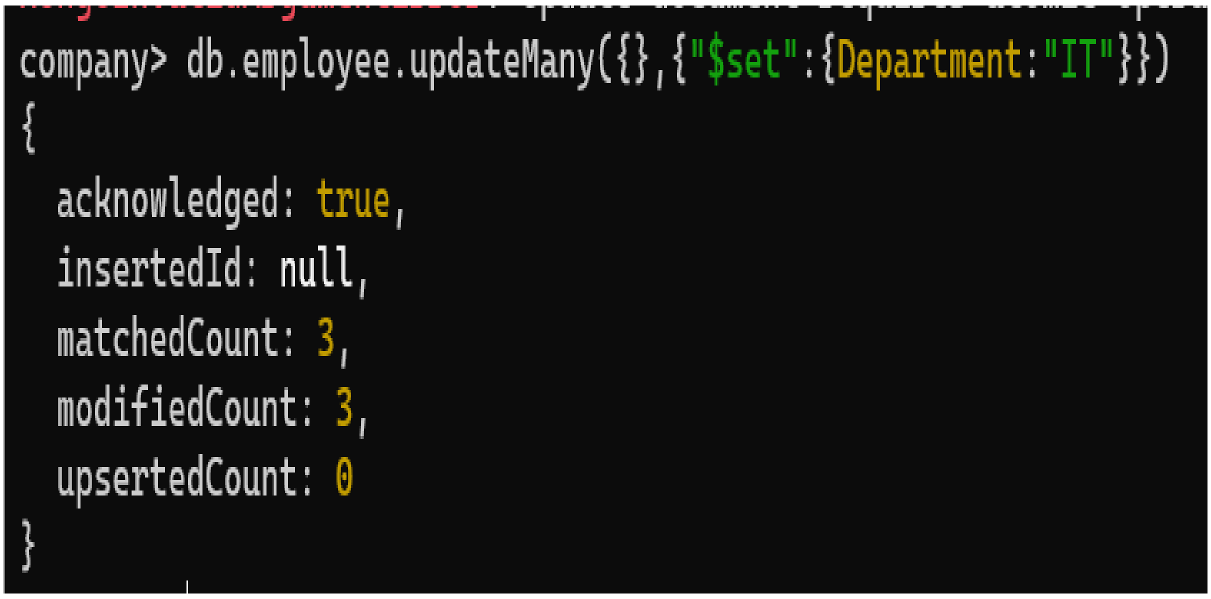
Ans=



```
company> db.employee.updateOne({ _id: ObjectId("6517ed4b11146f4f4b3da9bc")},{"$set":{salary:"180000"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

e) Add one more field department to the collection.

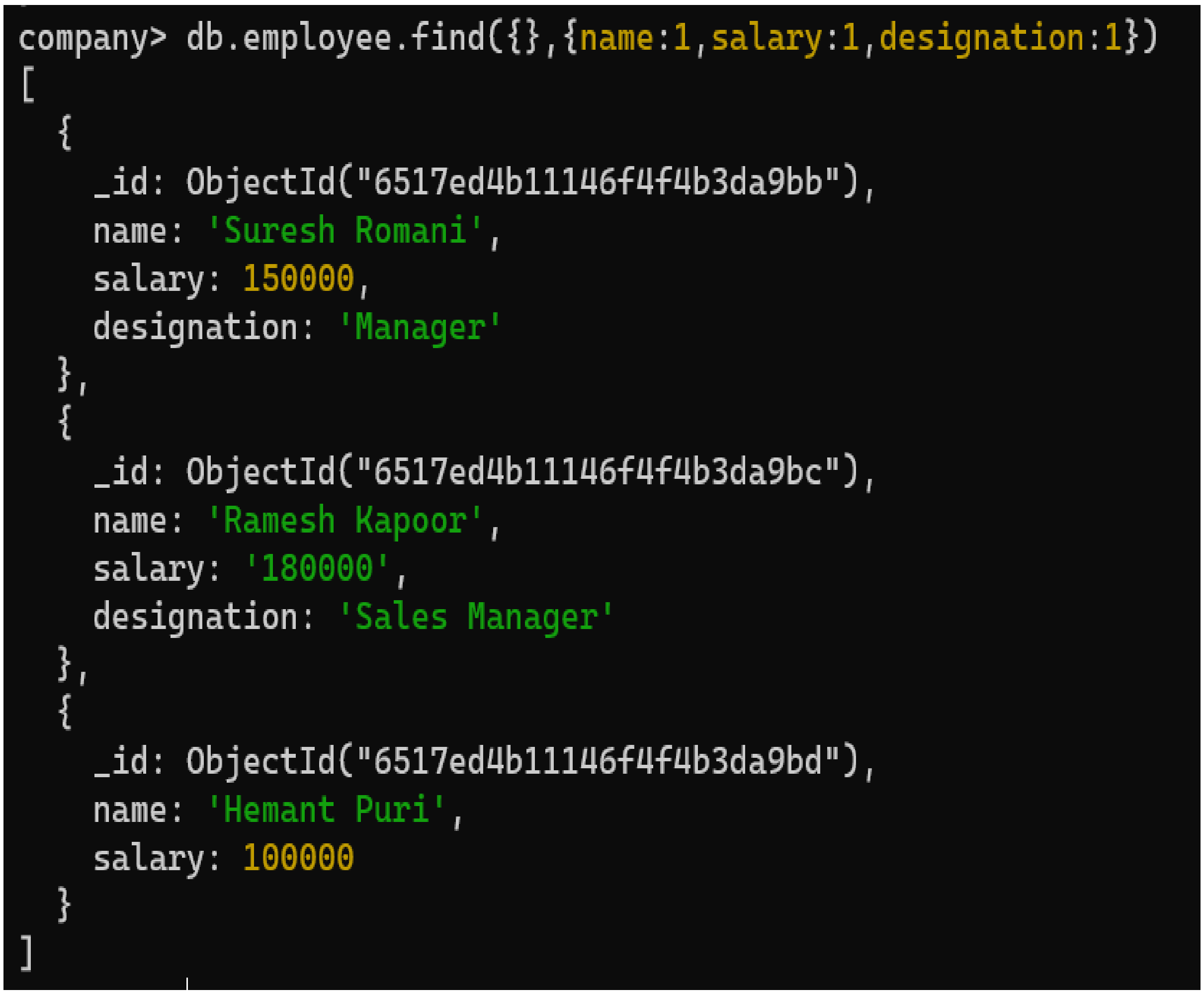
Ans=



```
company> db.employee.updateMany({},{"$set":{Department:"IT"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

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

Ans=

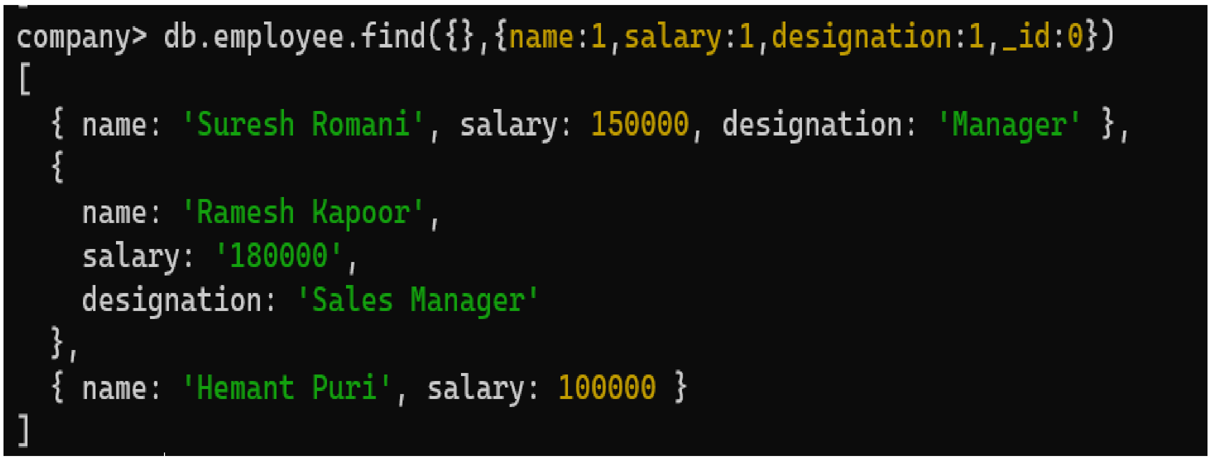


```
company> db.employee.find({},{name:1,salary:1,designation:1})
[
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bb"),
    name: 'Suresh Romani',
    salary: 150000,
    designation: 'Manager'
  },
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bc"),
    name: 'Ramesh Kapoor',
    salary: '180000',
    designation: 'Sales Manager'
  },
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bd"),
    name: 'Hemant Puri',
    salary: 100000
  }
]
```

g) Display the fields name, email and designation for all the documents but

exclude the field _id.

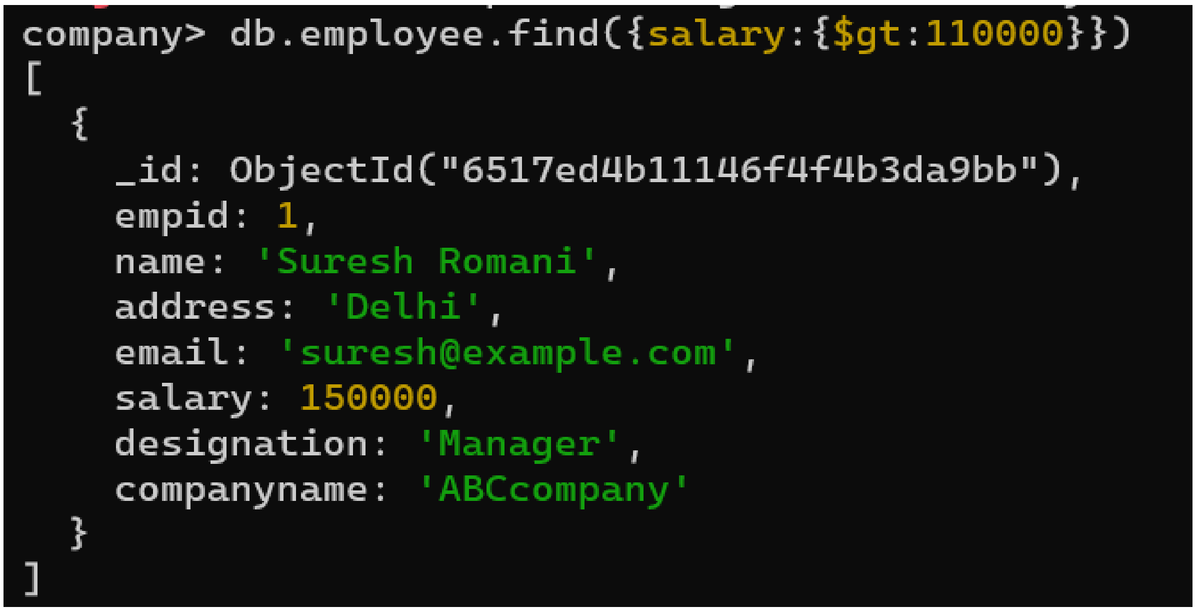
Ans=



```
company> db.employee.find({},{name:1,salary:1,designation:1,_id:0})
[
  { name: 'Suresh Romani', salary: 150000, designation: 'Manager' },
  {
    name: 'Ramesh Kapoor',
    salary: '180000',
    designation: 'Sales Manager'
  },
  { name: 'Hemant Puri', salary: 100000 }
]
```

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

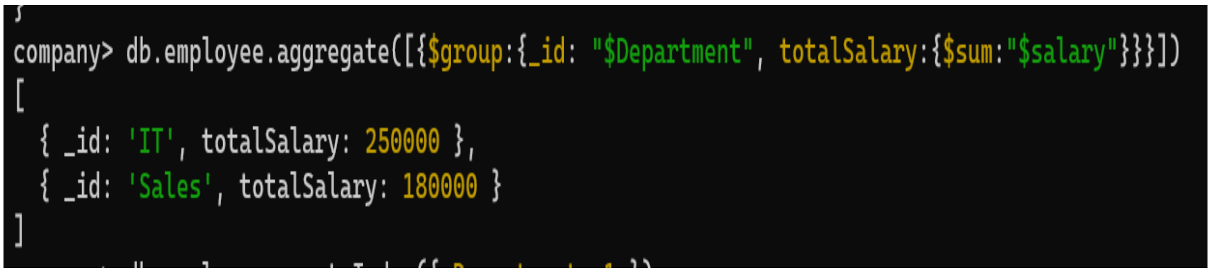
Ans=



```
company> db.employee.find({salary:{$gt:110000}})
[
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bb"),
    empid: 1,
    name: 'Suresh Romani',
    address: 'Delhi',
    email: 'suresh@example.com',
    salary: 150000,
    designation: 'Manager',
    companyname: 'ABCcompany'
  }
]
```

i) Find department wise total salary of employees.

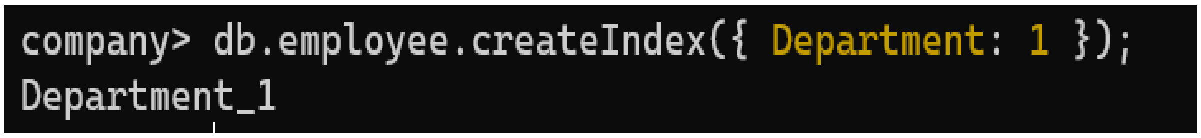
Ans=



```
company> db.employee.aggregate([{$group:{_id: "$Department", totalSalary:{$sum:"$salary"}}}])
[
  { _id: 'IT', totalSalary: 250000 },
  { _id: 'Sales', totalSalary: 180000 }
]
```

j) Create an index for department field.

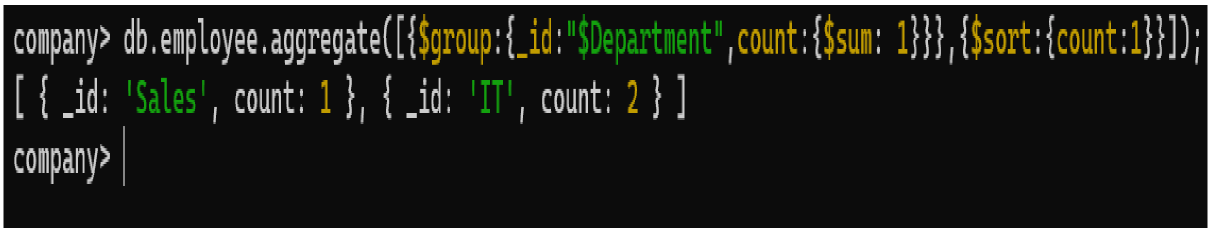
Ans=



```
company> db.employee.createIndex({ Department: 1 });
Department_1
```

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

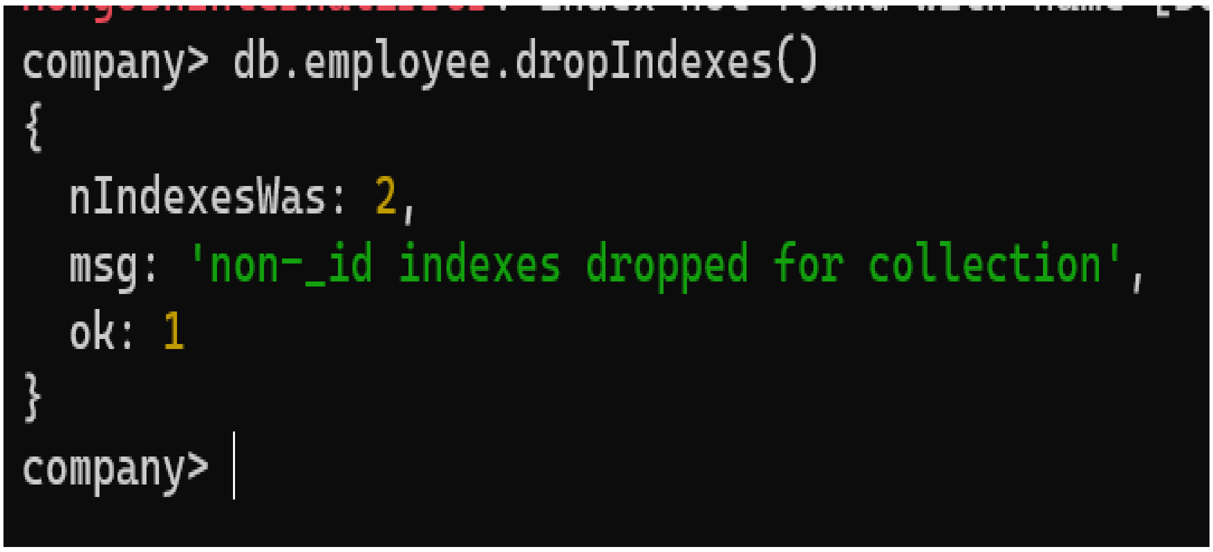
Ans=



```
company> db.employee.aggregate([{$group:{_id:"$Department",count:{$sum: 1}}},{$sort:{count:1}}]);
[ { _id: 'Sales', count: 1 }, { _id: 'IT', count: 2 } ]
company>
```

l) Remove all indexes from employee collection.

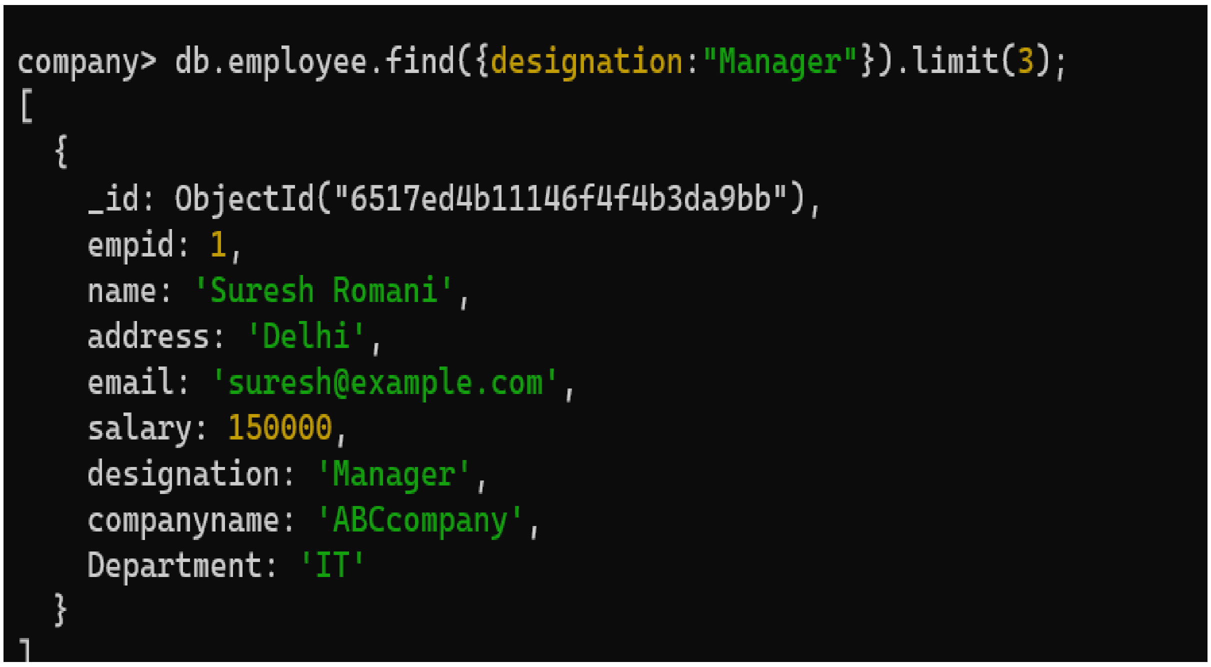
Ans=



```
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.

Ans=



```
company> db.employee.find({designation:"Manager"}).limit(3);
[
  {
    _id: ObjectId("6517ed4b11146f4f4b3da9bb"),
    empid: 1,
    name: 'Suresh Romani',
    address: 'Delhi',
    email: 'suresh@example.com',
    salary: 150000,
    designation: 'Manager',
    companyname: 'ABCcompany',
    Department: 'IT'
  }
]
```