MongoDB Excersices

Exercise 1 - Student Database

Agenda: Create database, Create collection, insert data, find, find one, sort, limit, skip, distinct, projection.

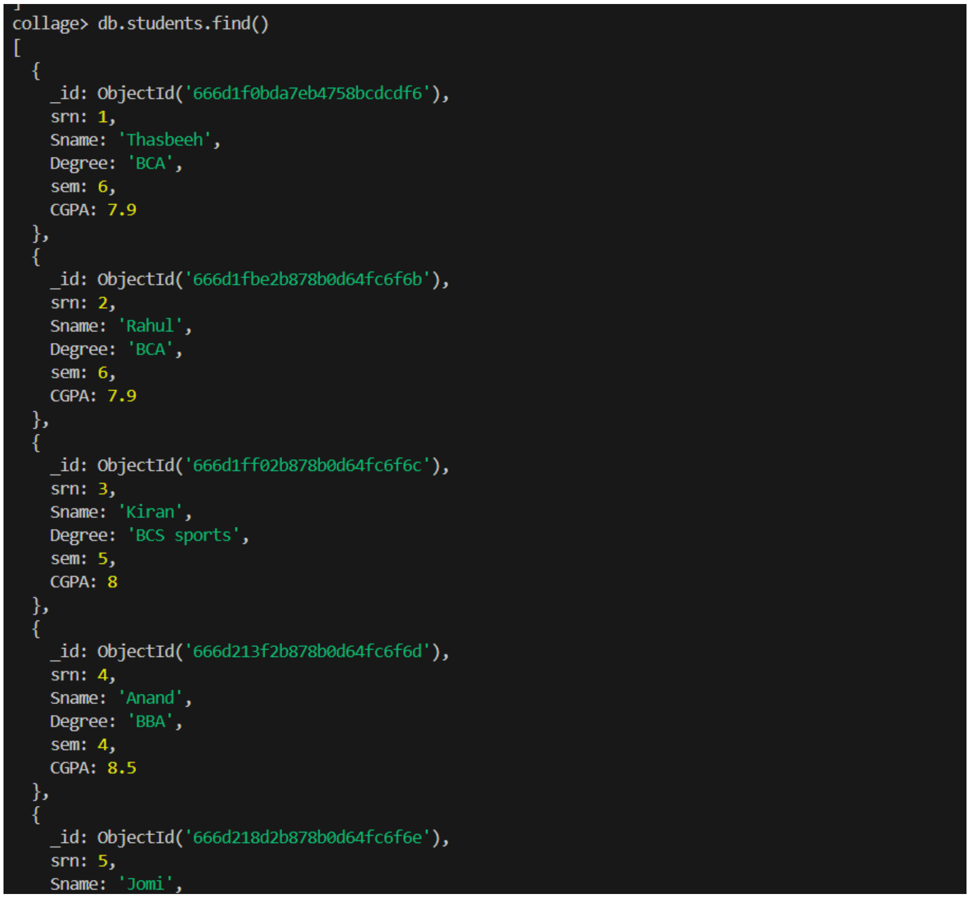
Create a student database with the fields: (SRN, Sname, Degree, Sem, CGPA)

```
> use studb9
switched to db studb9

> doc1=({srn:110,sname:"Rahul",degree:"BCA",sem:6,CGPA:7.9})
{
        "srn" : 110,
        "sname" : "Rahul",
        "degree" : "BCA",
        "sem" : 6,
        "CGPA" : 7.9
}
> db.stud09.insert(doc1)
```

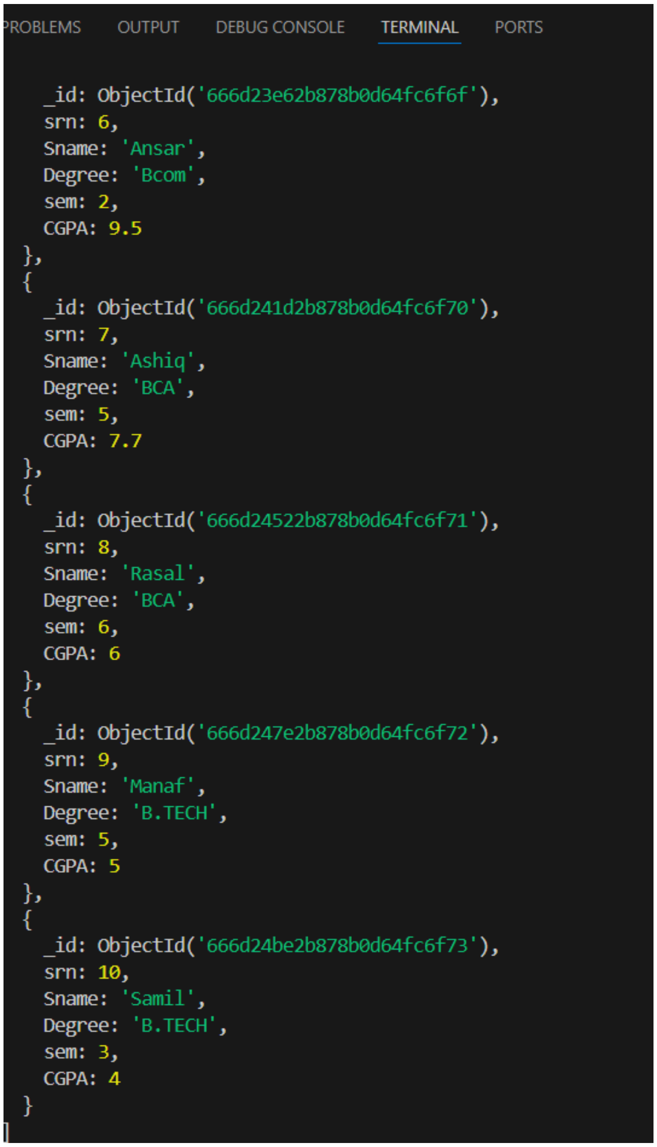
Note: insert 10 documents.

Questions:

1.display all the documents

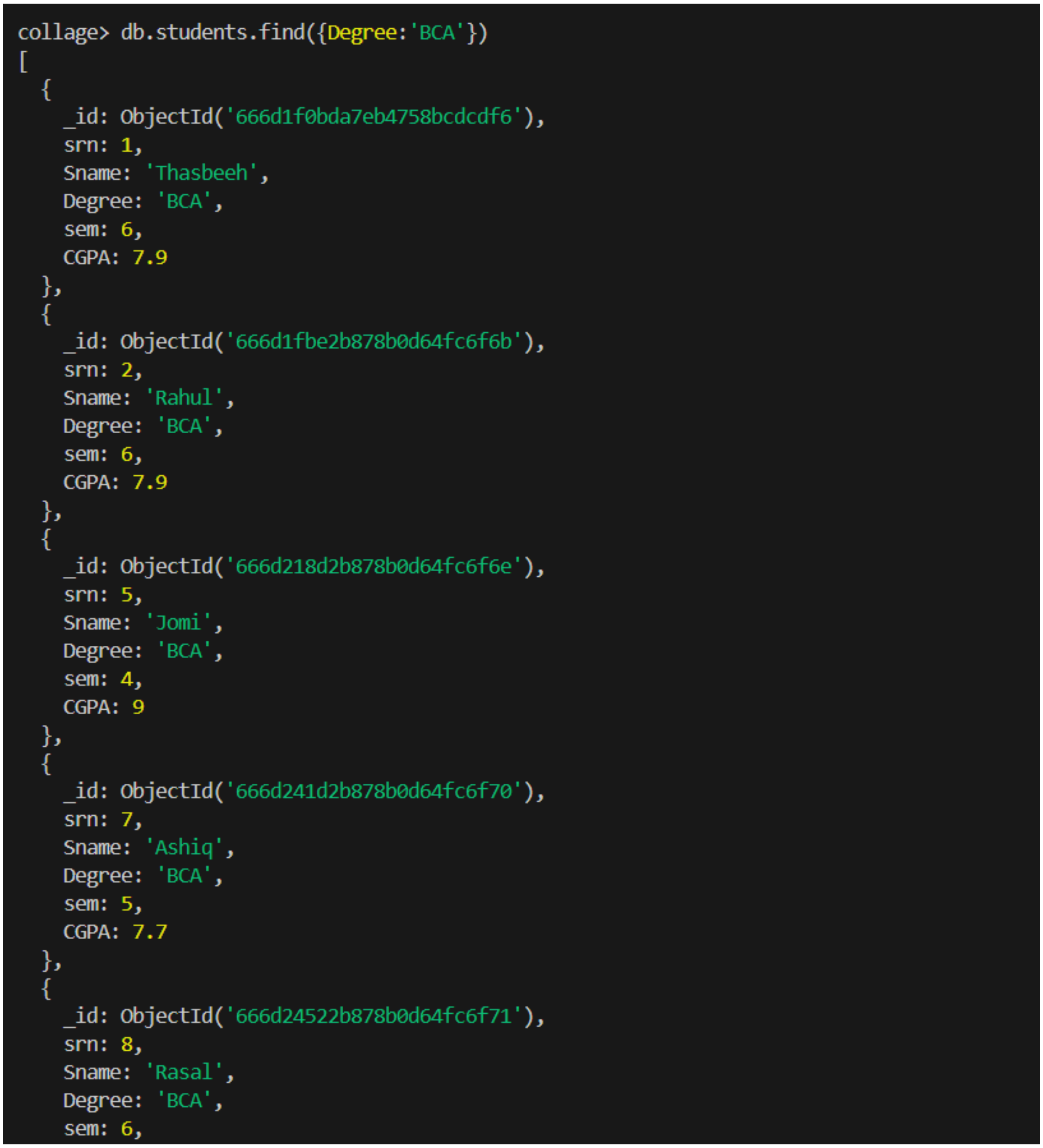


```
]
collage> db.students.find()
[
  {
    _id: ObjectId('666d1f0bda7eb4758bcdcdf6'),
    srn: 1,
    Sname: 'Thasbeeh',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1fbe2b878b0d64fc6f6b'),
    srn: 2,
    Sname: 'Rahul',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1ff02b878b0d64fc6f6c'),
    srn: 3,
    Sname: 'Kiran',
    Degree: 'BCS sports',
    sem: 5,
    CGPA: 8
  },
  {
    _id: ObjectId('666d213f2b878b0d64fc6f6d'),
    srn: 4,
    Sname: 'Anand',
    Degree: 'BBA',
    sem: 4,
    CGPA: 8.5
  },
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
```



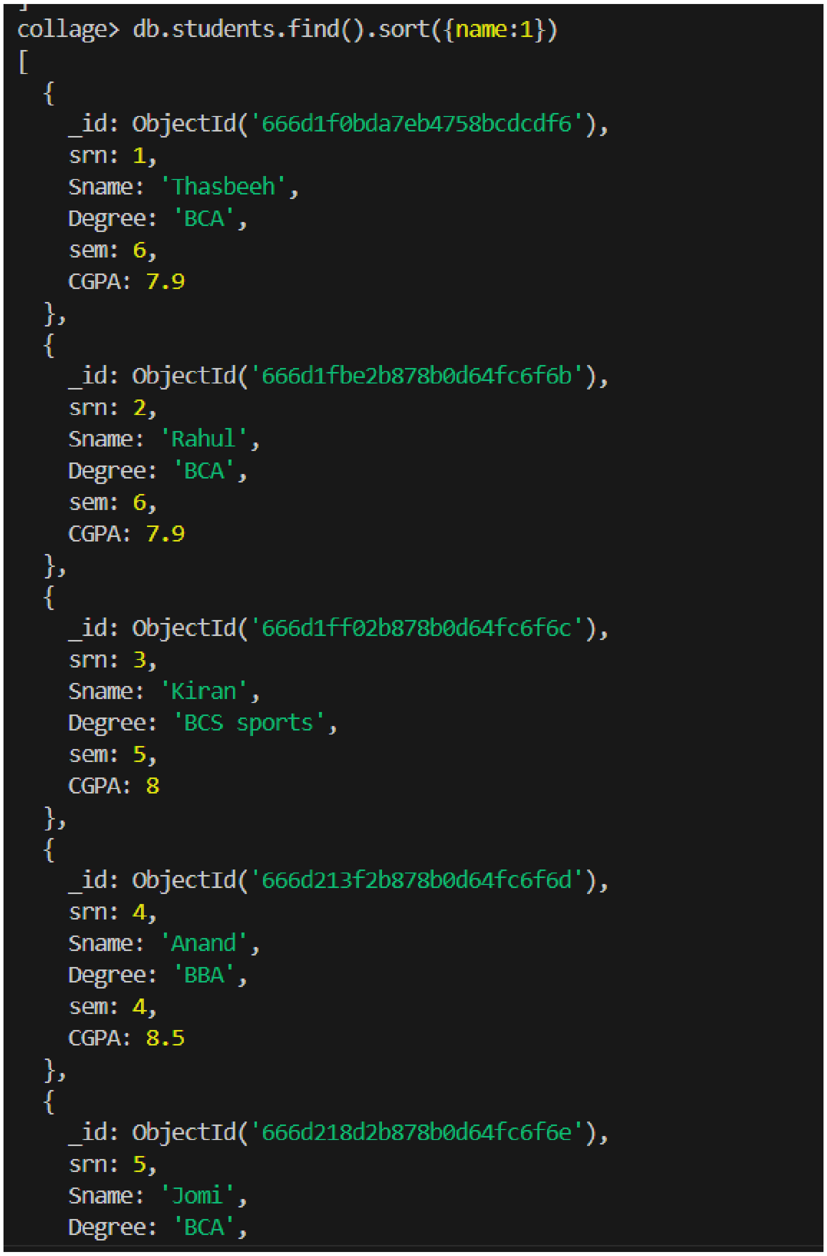
```
    _id: ObjectId('666d23e62b878b0d64fc6f6f'),
    srn: 6,
    Sname: 'Ansar',
    Degree: 'Bcom',
    sem: 2,
    CGPA: 9.5
  },
  {
    _id: ObjectId('666d241d2b878b0d64fc6f70'),
    srn: 7,
    Sname: 'Ashiq',
    Degree: 'BCA',
    sem: 5,
    CGPA: 7.7
  },
  {
    _id: ObjectId('666d24522b878b0d64fc6f71'),
    srn: 8,
    Sname: 'Rasal',
    Degree: 'BCA',
    sem: 6,
    CGPA: 6
  },
  {
    _id: ObjectId('666d247e2b878b0d64fc6f72'),
    srn: 9,
    Sname: 'Manaf',
    Degree: 'B.TECH',
    sem: 5,
    CGPA: 5
  },
  {
    _id: ObjectId('666d24be2b878b0d64fc6f73'),
    srn: 10,
    Sname: 'Samil',
    Degree: 'B.TECH',
    sem: 3,
    CGPA: 4
  }
]
```

2.Display all the students in BCA



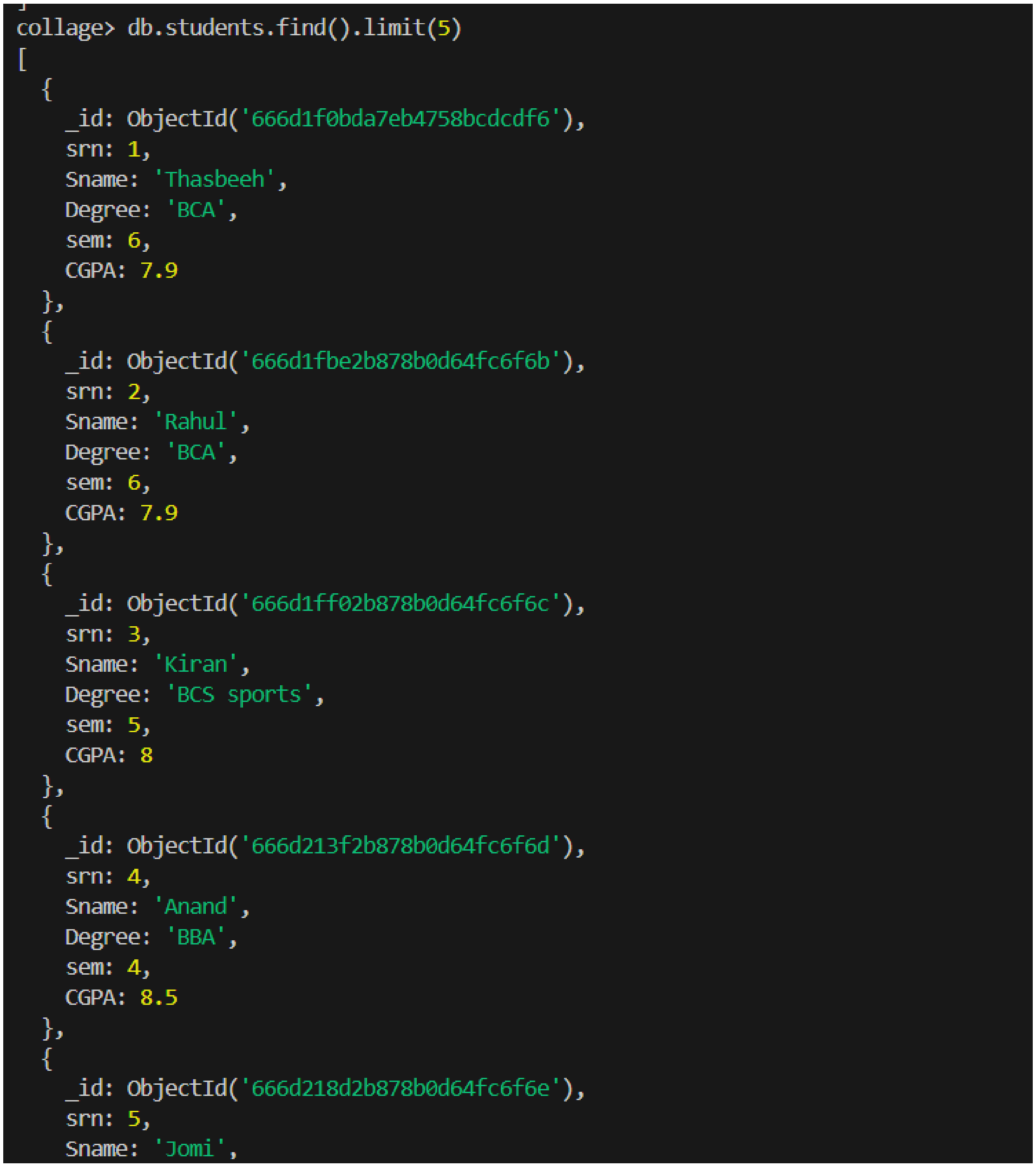
```
collage> db.students.find({Degree:'BCA'})
[
  {
    _id: ObjectId('666d1f0bda7eb4758bcdcdf6'),
    srn: 1,
    Sname: 'Thasbeeh',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1fbe2b878b0d64fc6f6b'),
    srn: 2,
    Sname: 'Rahul',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
    Degree: 'BCA',
    sem: 4,
    CGPA: 9
  },
  {
    _id: ObjectId('666d241d2b878b0d64fc6f70'),
    srn: 7,
    Sname: 'Ashiq',
    Degree: 'BCA',
    sem: 5,
    CGPA: 7.7
  },
  {
    _id: ObjectId('666d24522b878b0d64fc6f71'),
    srn: 8,
    Sname: 'Rasal',
    Degree: 'BCA',
    sem: 6,
```

3.Display all the students in ascending order



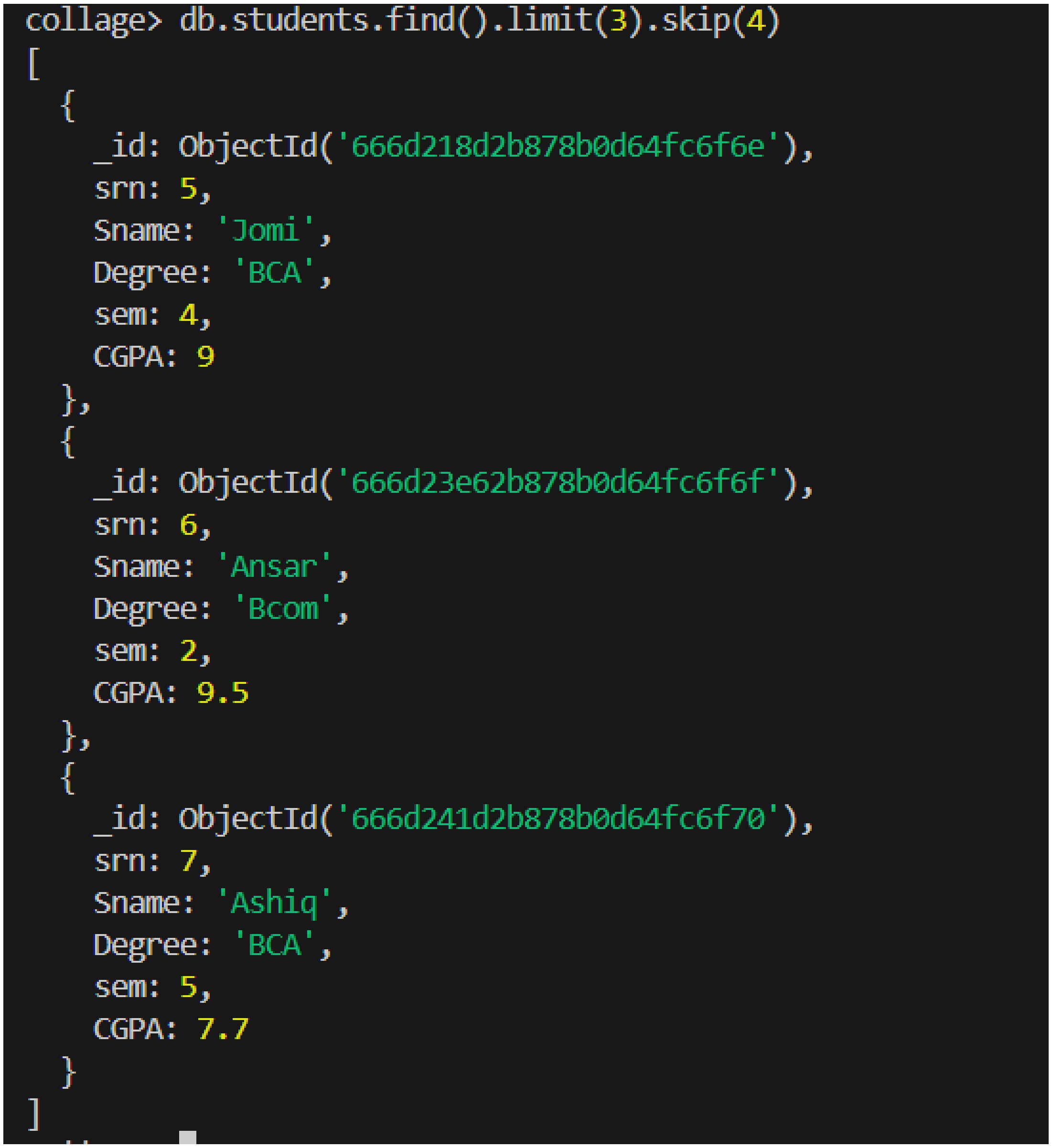
```
]
collage> db.students.find().sort({name:1})
[
  {
    _id: ObjectId('666d1f0bda7eb4758bcdcdf6'),
    srn: 1,
    Sname: 'Thasbeeh',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1fbe2b878b0d64fc6f6b'),
    srn: 2,
    Sname: 'Rahul',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1ff02b878b0d64fc6f6c'),
    srn: 3,
    Sname: 'Kiran',
    Degree: 'BCS sports',
    sem: 5,
    CGPA: 8
  },
  {
    _id: ObjectId('666d213f2b878b0d64fc6f6d'),
    srn: 4,
    Sname: 'Anand',
    Degree: 'BBA',
    sem: 4,
    CGPA: 8.5
  },
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
    Degree: 'BCA',
```

4.Display first 5 students



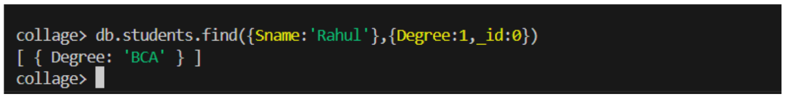
```
]
collage> db.students.find().limit(5)
[
  {
    _id: ObjectId('666d1f0bda7eb4758bcdcdf6'),
    srn: 1,
    Sname: 'Thasbeeh',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1fbe2b878b0d64fc6f6b'),
    srn: 2,
    Sname: 'Rahul',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1ff02b878b0d64fc6f6c'),
    srn: 3,
    Sname: 'Kiran',
    Degree: 'BCS sports',
    sem: 5,
    CGPA: 8
  },
  {
    _id: ObjectId('666d213f2b878b0d64fc6f6d'),
    srn: 4,
    Sname: 'Anand',
    Degree: 'BBA',
    sem: 4,
    CGPA: 8.5
  },
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
```

5.display students 5,6,7



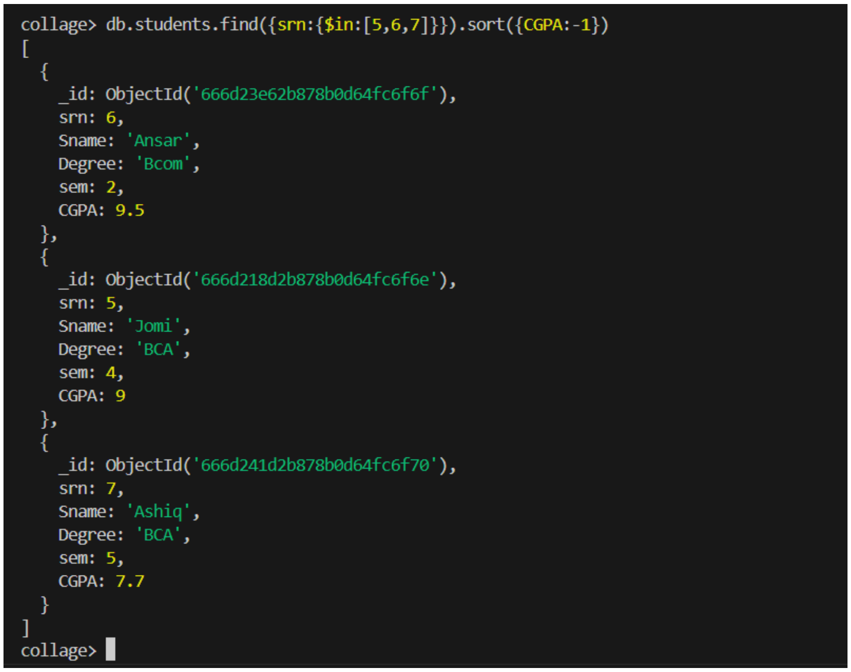
```
collage> db.students.find().limit(3).skip(4)
[
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
    Degree: 'BCA',
    sem: 4,
    CGPA: 9
  },
  {
    _id: ObjectId('666d23e62b878b0d64fc6f6f'),
    srn: 6,
    Sname: 'Ansar',
    Degree: 'Bcom',
    sem: 2,
    CGPA: 9.5
  },
  {
    _id: ObjectId('666d241d2b878b0d64fc6f70'),
    srn: 7,
    Sname: 'Ashiq',
    Degree: 'BCA',
    sem: 5,
    CGPA: 7.7
  }
]
```

6.list the degree of student "Rahul"



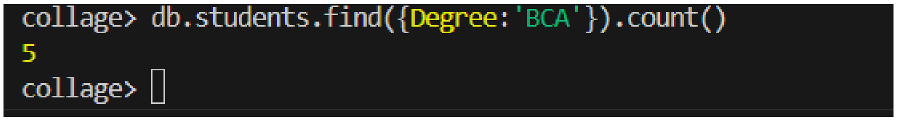
```
collage> db.students.find({Sname:'Rahul'},{Degree:1,_id:0})
[ { Degree: 'BCA' } ]
collage> 
```

7.Display students details of 5,6,7 in descending order of percentage



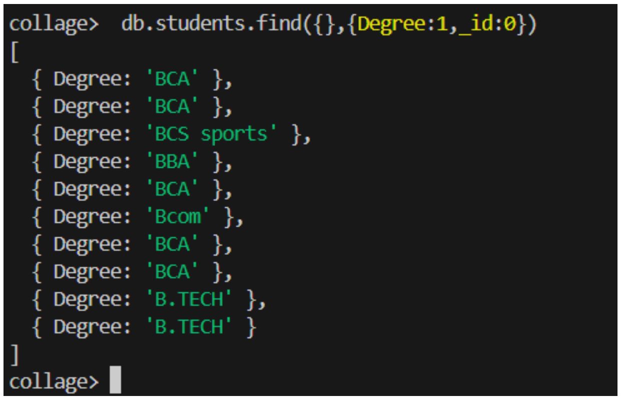
```
collage> db.students.find({srn:{$in:[5,6,7]}}).sort({CGPA:-1})
[
  {
    _id: ObjectId('666d23e62b878b0d64fc6f6f'),
    srn: 6,
    Sname: 'Ansar',
    Degree: 'Bcom',
    sem: 2,
    CGPA: 9.5
  },
  {
    _id: ObjectId('666d218d2b878b0d64fc6f6e'),
    srn: 5,
    Sname: 'Jomi',
    Degree: 'BCA',
    sem: 4,
    CGPA: 9
  },
  {
    _id: ObjectId('666d241d2b878b0d64fc6f70'),
    srn: 7,
    Sname: 'Ashiq',
    Degree: 'BCA',
    sem: 5,
    CGPA: 7.7
  }
]
collage>
```

8.Display the number of students in BCA



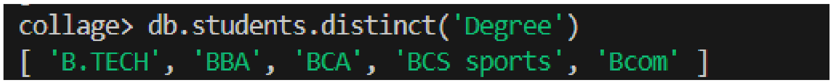
```
collage> db.students.find({Degree:'BCA'}).count()
5
collage>
```

9.Display all the degrees without _id



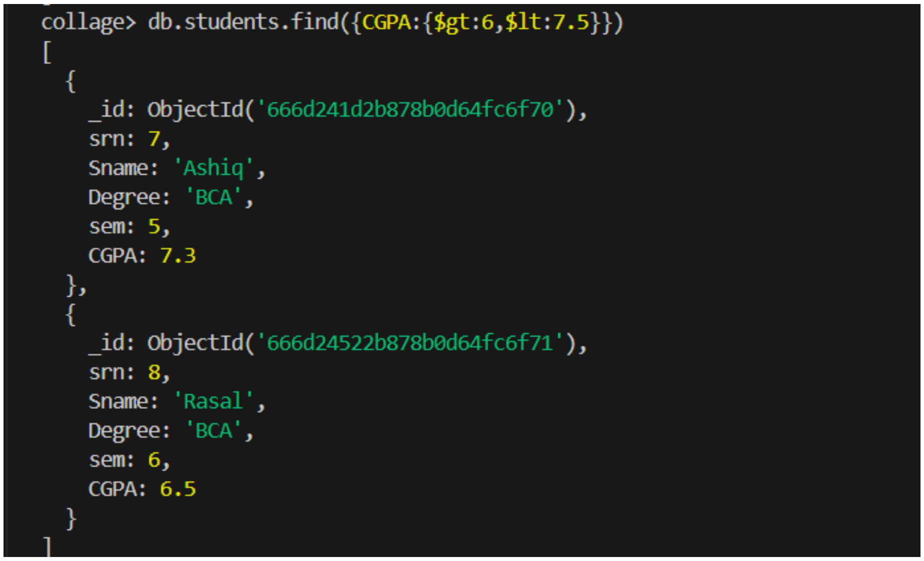
```
collage>  db.students.find({},{Degree:1,_id:0})
[
  { Degree: 'BCA' },
  { Degree: 'BCA' },
  { Degree: 'BCS sports' },
  { Degree: 'BBA' },
  { Degree: 'BCA' },
  { Degree: 'Bcom' },
  { Degree: 'BCA' },
  { Degree: 'BCA' },
  { Degree: 'B.TECH' },
  { Degree: 'B.TECH' }
]
collage>
```

10.Display all the distinct degrees



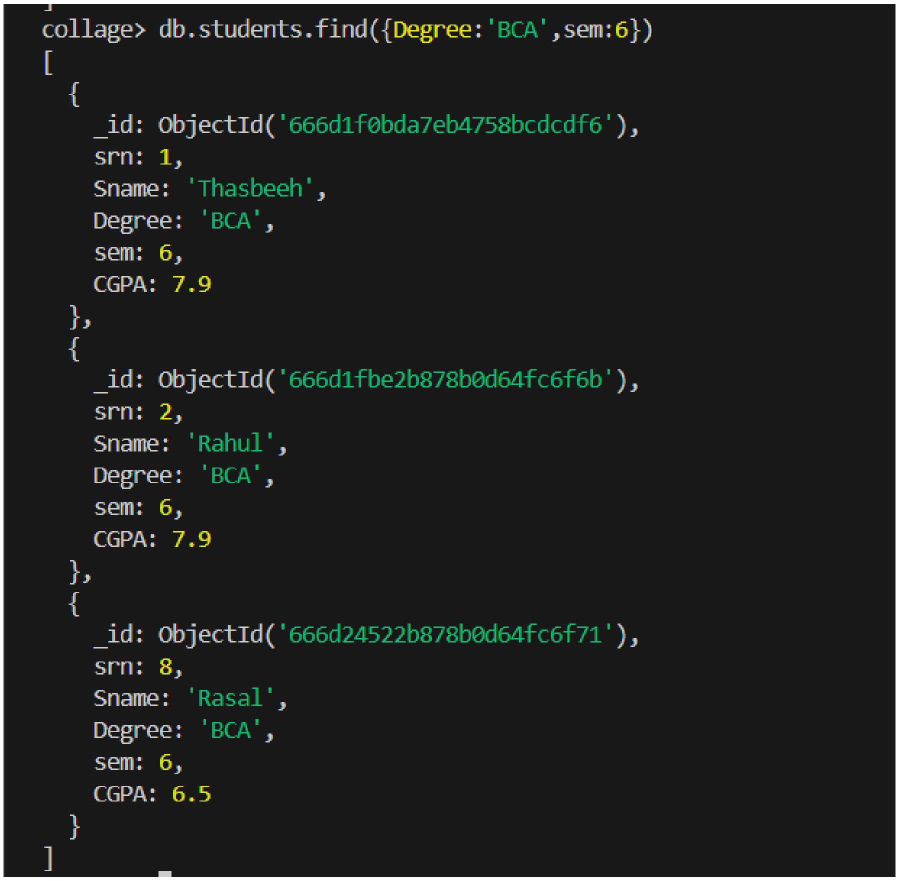
```
collage> db.students.distinct('Degree')
[ 'B.TECH', 'BBA', 'BCA', 'BCS sports', 'Bcom' ]
```

11.Display all the BCA students with CGPA greater than 6, but less than 7.5



```
collage> db.students.find({CGPA:{$gt:6,$lt:7.5}})
[
  {
    _id: ObjectId('666d241d2b878b0d64fc6f70'),
    srn: 7,
    Sname: 'Ashiq',
    Degree: 'BCA',
    sem: 5,
    CGPA: 7.3
  },
  {
    _id: ObjectId('666d24522b878b0d64fc6f71'),
    srn: 8,
    Sname: 'Rasal',
    Degree: 'BCA',
    sem: 6,
    CGPA: 6.5
  }
]
```

12.Display all the students in BCA and in 6th Sem



```
collage> db.students.find({Degree:'BCA',sem:6})
[
  {
    _id: ObjectId('666d1f0bda7eb4758bcdcdf6'),
    srn: 1,
    Sname: 'Thasbeeh',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d1fbe2b878b0d64fc6f6b'),
    srn: 2,
    Sname: 'Rahul',
    Degree: 'BCA',
    sem: 6,
    CGPA: 7.9
  },
  {
    _id: ObjectId('666d24522b878b0d64fc6f71'),
    srn: 8,
    Sname: 'Rasal',
    Degree: 'BCA',
    sem: 6,
    CGPA: 6.5
  }
]
```

Exercise 2 - Employee Database

Agenda: Update modifiers ($set, $unset, $inc, $push, $pushAll, $pull, $pullAll, $addToSet)

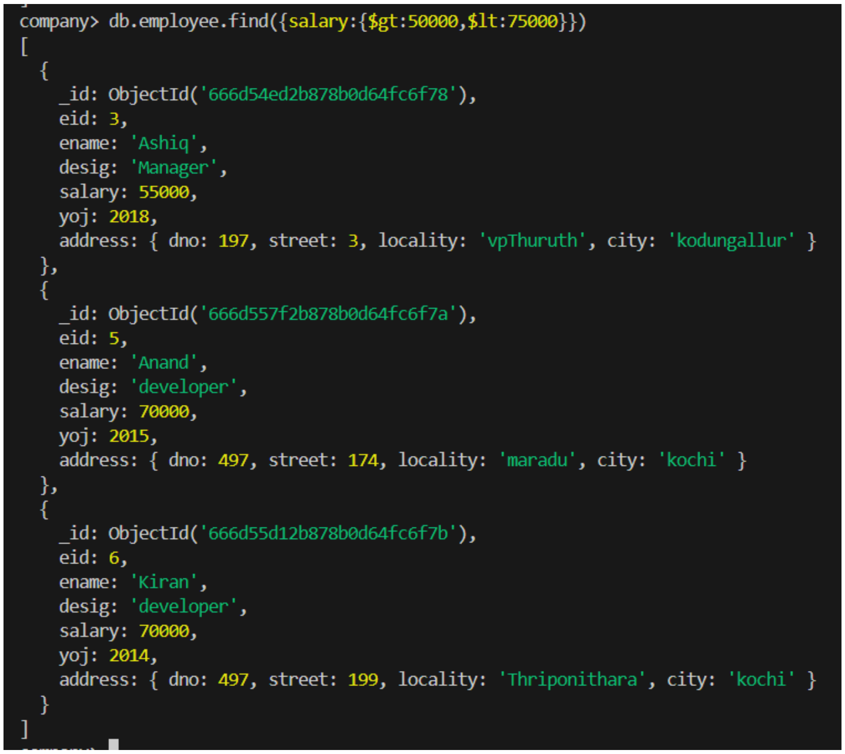
Create an employee database with the fields: {eid, ename, dept, desig, salary, yoj, address{dno, street, locality, city}}

```
> use empdb9
switched to db empdb9

> doc1 = {eid:001, ename:"Rahul", dept:"production", desig:"developer", salary:30000, yoj:2015,
            address:{dno:397, street:2, locality:"rmnagar", city:"bangalore"} }
{
        "eid" : 1,
        "ename" : "Rahul",
        "dept" : "production",
        "desig" : "developer",
        "salary" : 30000,
        "yoj" : 2015,
        "address" : {
                "dno" : 397,
                "street" : 2,
                "locality" : "rmnagar",
                "city" : "bangalore"
        }
}
> db.emp09.insert(doc1)
WriteResult({ "nInserted" : 1 })
```

Note: insert 10 documents.

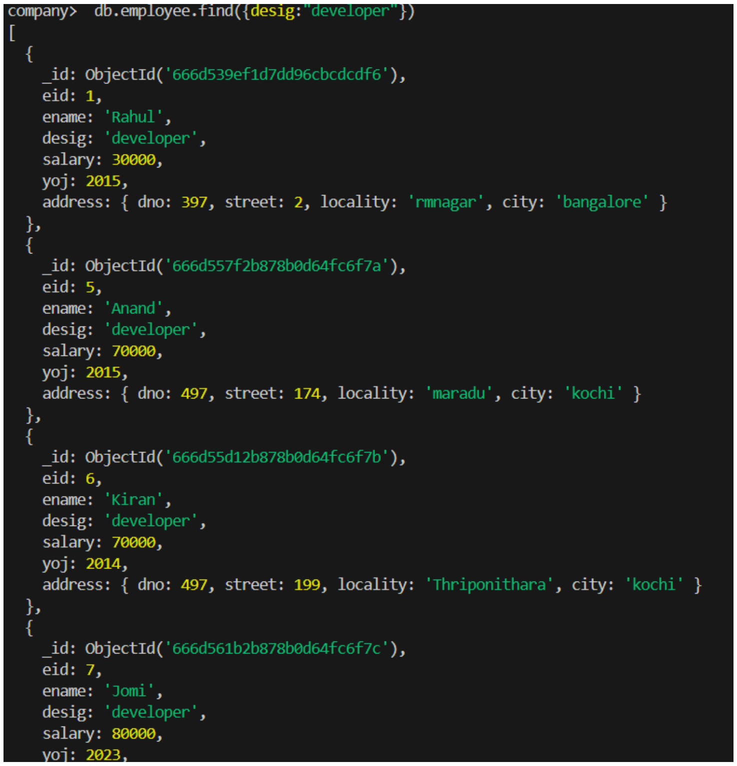
Questions:

1.Display all the employees with salary in range (50000, 75000)



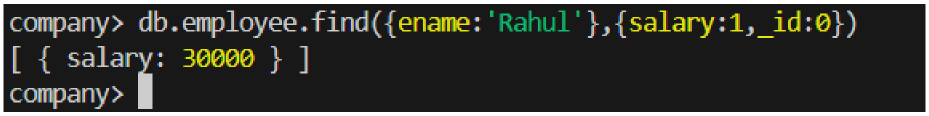
```
company> db.employee.find({salary:{$gt:50000,$lt:75000}})
[
  {
    _id: ObjectId('666d54ed2b878b0d64fc6f78'),
    eid: 3,
    ename: 'Ashiq',
    desig: 'Manager',
    salary: 55000,
    yoj: 2018,
    address: { dno: 197, street: 3, locality: 'vpThuruth', city: 'kodungallur' }
  },
  {
    _id: ObjectId('666d557f2b878b0d64fc6f7a'),
    eid: 5,
    ename: 'Anand',
    desig: 'developer',
    salary: 70000,
    yoj: 2015,
    address: { dno: 497, street: 174, locality: 'maradu', city: 'kochi' }
  },
  {
    _id: ObjectId('666d55d12b878b0d64fc6f7b'),
    eid: 6,
    ename: 'Kiran',
    desig: 'developer',
    salary: 70000,
    yoj: 2014,
    address: { dno: 497, street: 199, locality: 'Thriponithara', city: 'kochi' }
  }
]
```

2.Display all the employees with desig developer



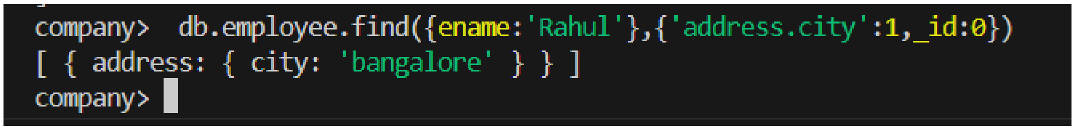
```
company>  db.employee.find({desig:"developer"})
[
  {
    _id: ObjectId('666d539ef1d7dd96cbcdcdf6'),
    eid: 1,
    ename: 'Rahul',
    desig: 'developer',
    salary: 30000,
    yoj: 2015,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'bangalore' }
  },
  {
    _id: ObjectId('666d557f2b878b0d64fc6f7a'),
    eid: 5,
    ename: 'Anand',
    desig: 'developer',
    salary: 70000,
    yoj: 2015,
    address: { dno: 497, street: 174, locality: 'maradu', city: 'kochi' }
  },
  {
    _id: ObjectId('666d55d12b878b0d64fc6f7b'),
    eid: 6,
    ename: 'Kiran',
    desig: 'developer',
    salary: 70000,
    yoj: 2014,
    address: { dno: 497, street: 199, locality: 'Thriponithara', city: 'kochi' }
  },
  {
    _id: ObjectId('666d561b2b878b0d64fc6f7c'),
    eid: 7,
    ename: 'Jomi',
    desig: 'developer',
    salary: 80000,
    yoj: 2023,
```

3.Display the Salary of “Rahul”



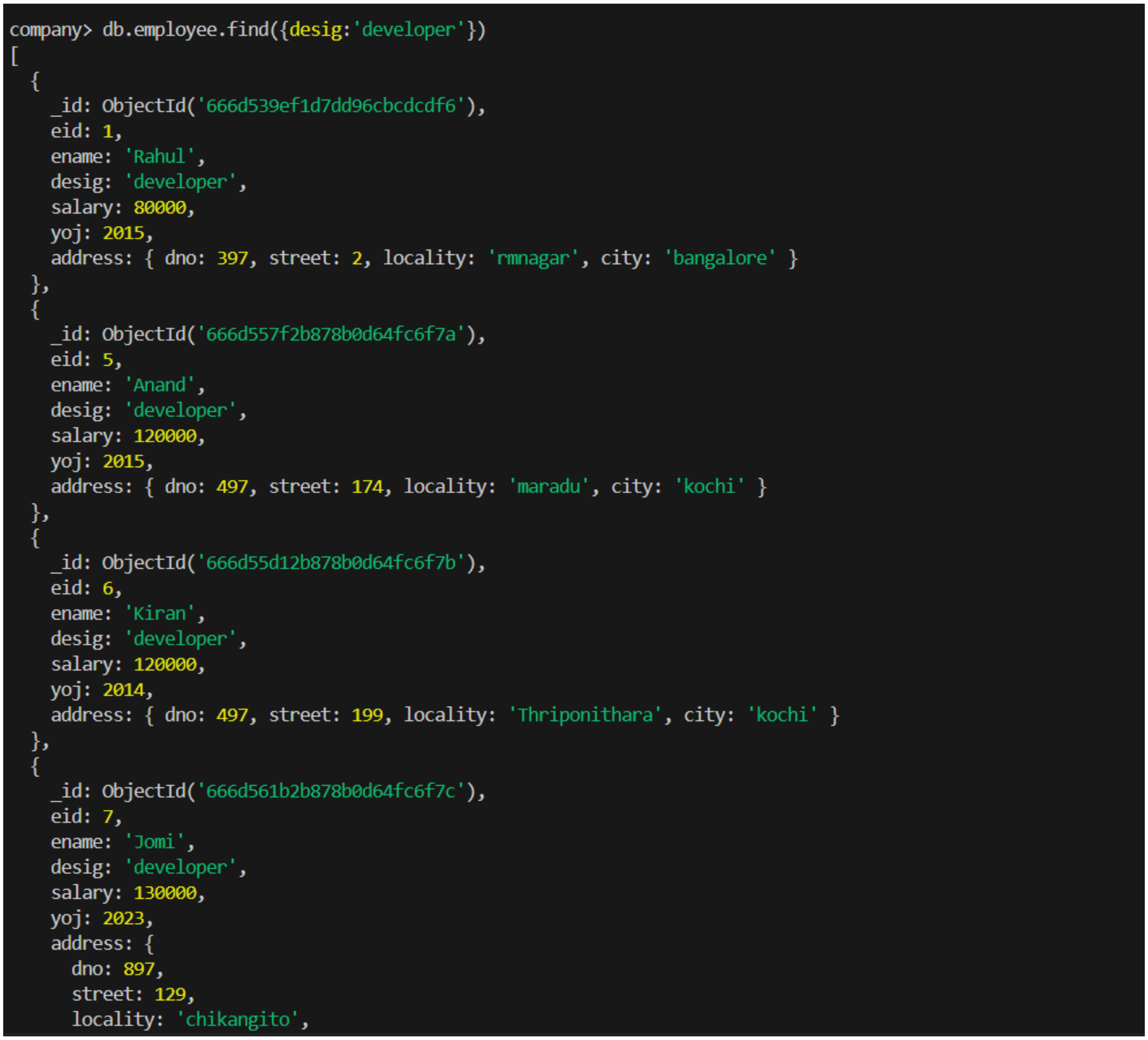
```
company> db.employee.find({ename:'Rahul'},{salary:1,_id:0})
[ { salary: 30000 } ]
company> 
```

4.Display the city of employee “Rahul”



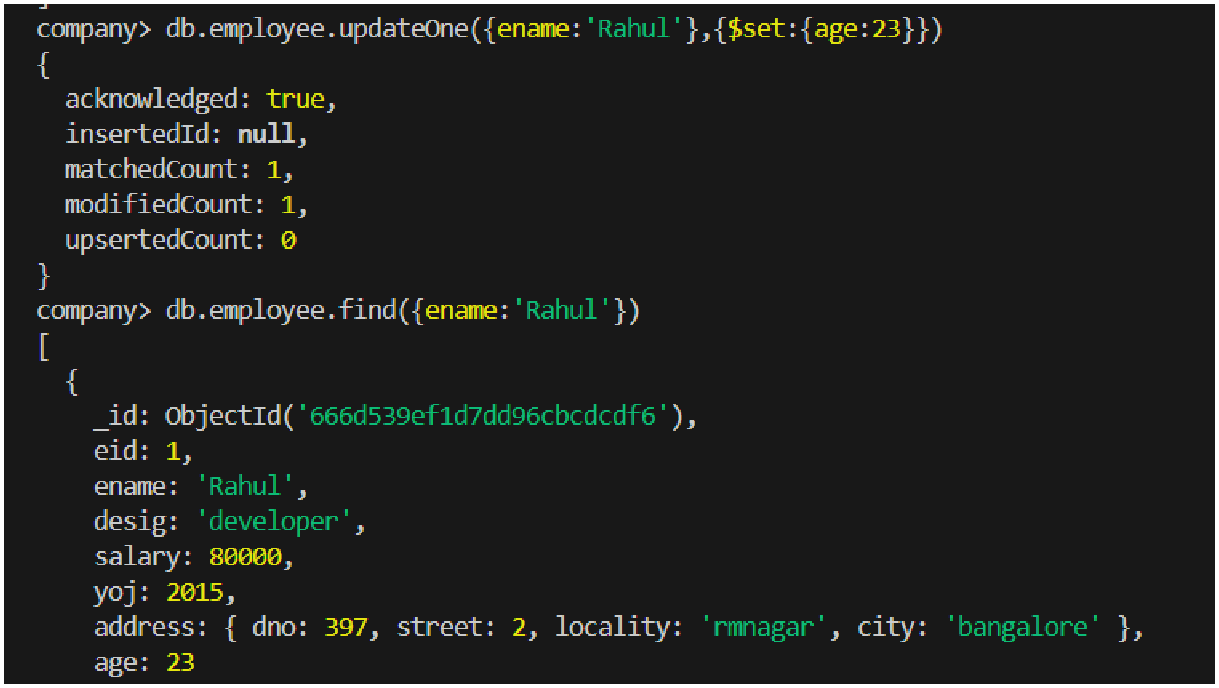
```
company>  db.employee.find({ename:'Rahul'},{'address.city':1,_id:0})
[ { address: { city: 'bangalore' } } ]
company> 
```

5. Update the salary of developers by 5000 increment



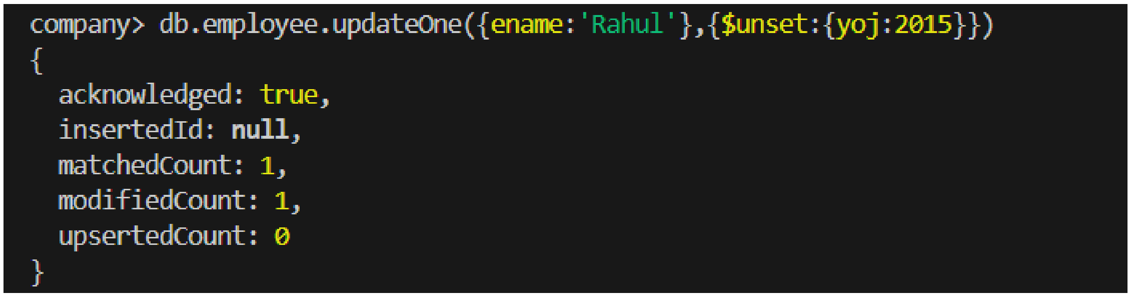
```
company> db.employee.find({desig:'developer'})
[
  {
    _id: ObjectId('666d539ef1d7dd96cbcdcdf6'),
    eid: 1,
    ename: 'Rahul',
    desig: 'developer',
    salary: 80000,
    yoj: 2015,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'bangalore' }
  },
  {
    _id: ObjectId('666d557f2b878b0d64fc6f7a'),
    eid: 5,
    ename: 'Anand',
    desig: 'developer',
    salary: 120000,
    yoj: 2015,
    address: { dno: 497, street: 174, locality: 'maradu', city: 'kochi' }
  },
  {
    _id: ObjectId('666d55d12b878b0d64fc6f7b'),
    eid: 6,
    ename: 'Kiran',
    desig: 'developer',
    salary: 120000,
    yoj: 2014,
    address: { dno: 497, street: 199, locality: 'Thriponithara', city: 'kochi' }
  },
  {
    _id: ObjectId('666d561b2b878b0d64fc6f7c'),
    eid: 7,
    ename: 'Jomi',
    desig: 'developer',
    salary: 130000,
    yoj: 2023,
    address: {
      dno: 897,
      street: 129,
      locality: 'chikangito',
```

6. Add field age to employee “Rahul”



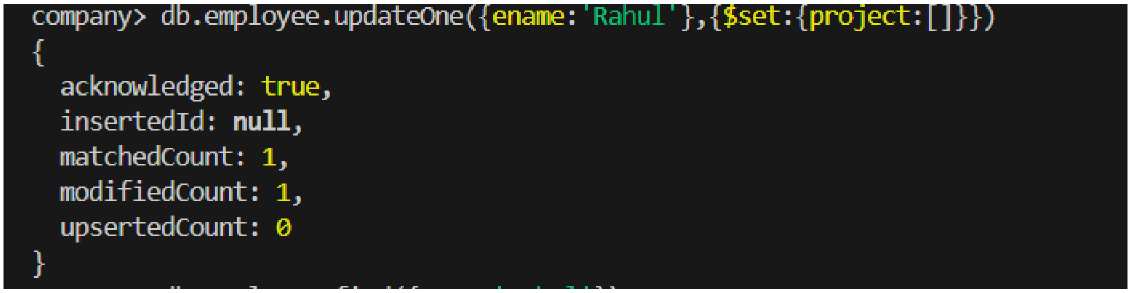
```
company> db.employee.updateOne({ename:'Rahul'},{$set:{age:23}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.employee.find({ename:'Rahul'})
[
  {
    _id: ObjectId('666d539ef1d7dd96cbcdcdf6'),
    eid: 1,
    ename: 'Rahul',
    desig: 'developer',
    salary: 80000,
    yoj: 2015,
    address: { dno: 397, street: 2, locality: 'rmnagar', city: 'bangalore' },
    age: 23
```

7. Remove YOJ from “Rahul”



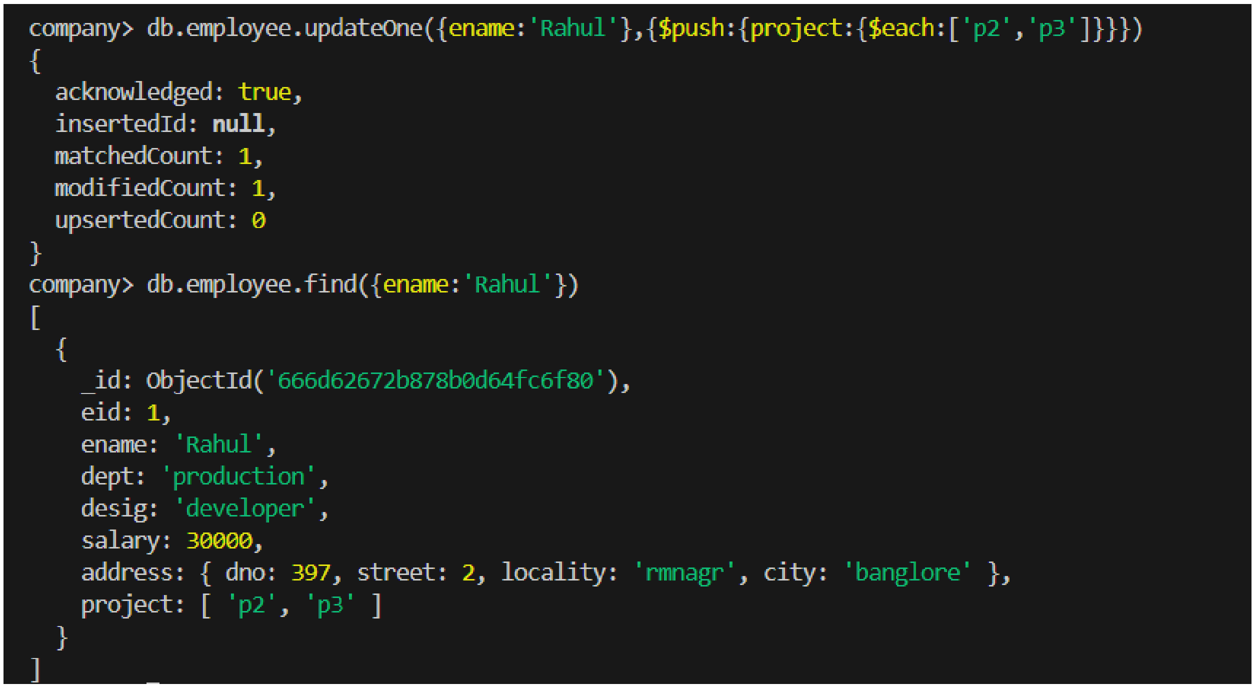
```
company> db.employee.updateOne({ename:'Rahul'},{$unset:{yoj:2015}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. Add an array field project to “Rahul”



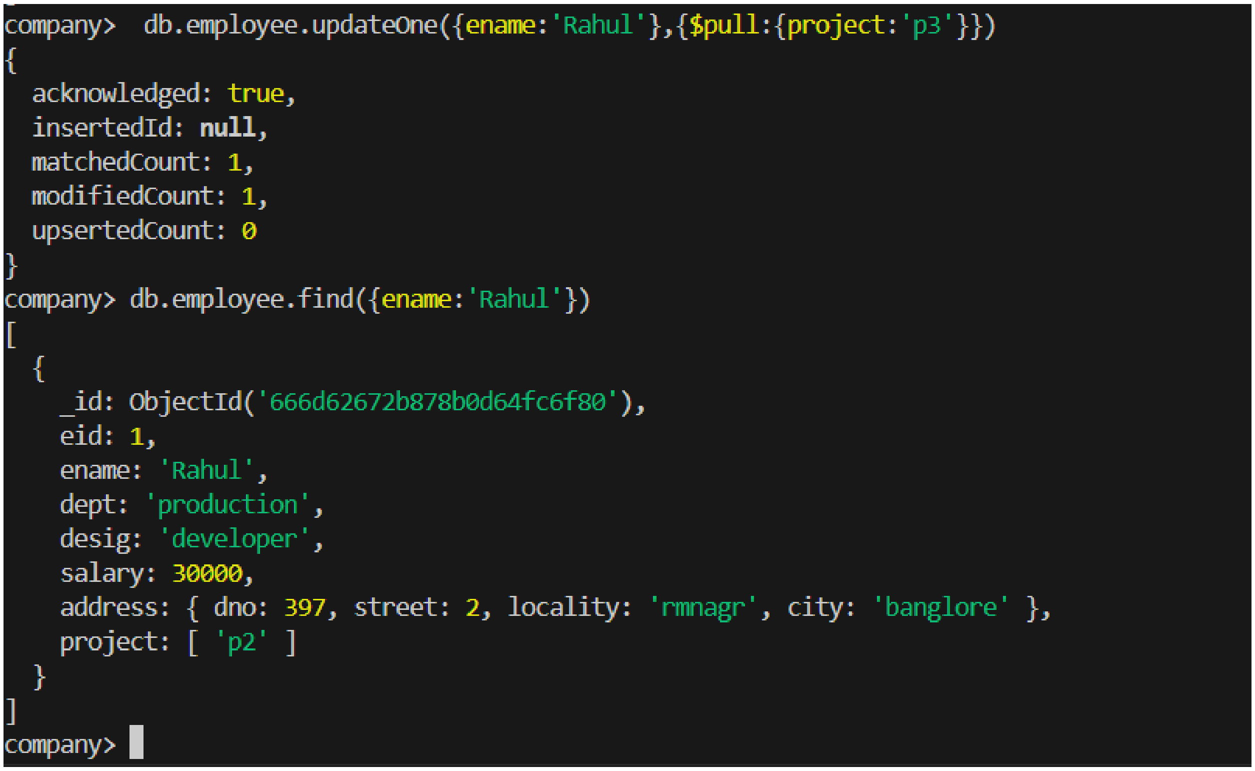
```
company> db.employee.updateOne({ename:'Rahul'},{$set:{project:[]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

9. Add p2 and p3 project to “Rahul”



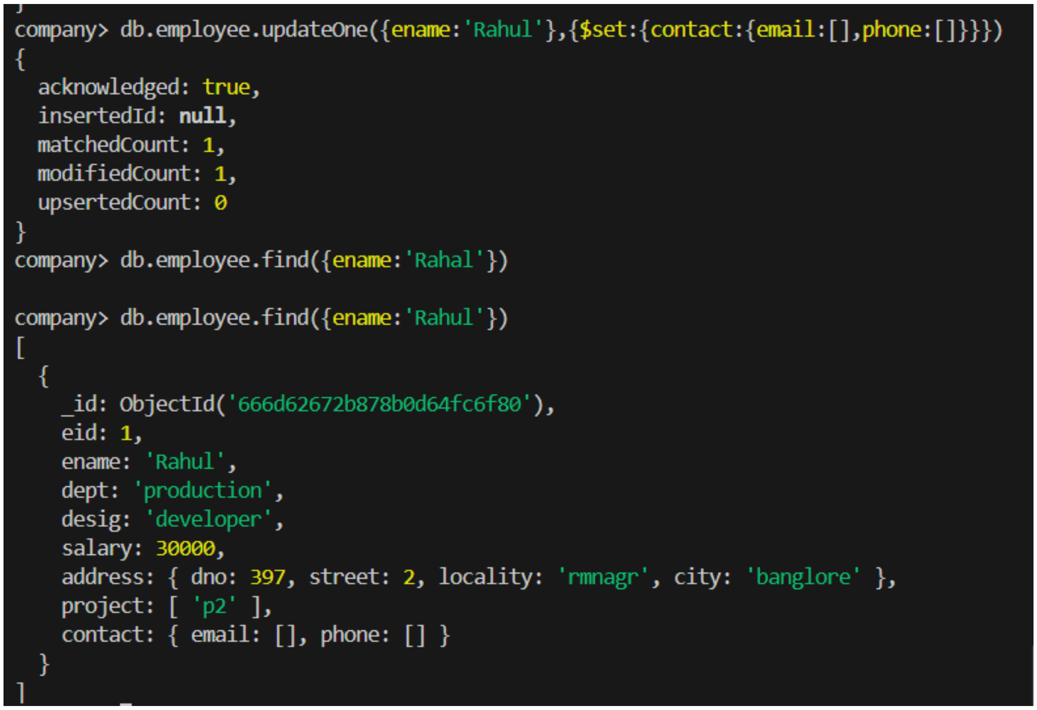
```
company> db.employee.updateOne({ename:'Rahul'},{$push:{project:{$each:['p2','p3']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.employee.find({ename:'Rahul'})
[
  {
    _id: ObjectId('666d62672b878b0d64fc6f80'),
    eid: 1,
    ename: 'Rahul',
    dept: 'production',
    desig: 'developer',
    salary: 30000,
    address: { dno: 397, street: 2, locality: 'rmnagr', city: 'banglore' },
    project: [ 'p2', 'p3' ]
  }
]
```

10. Remove p3 from “Rahul”



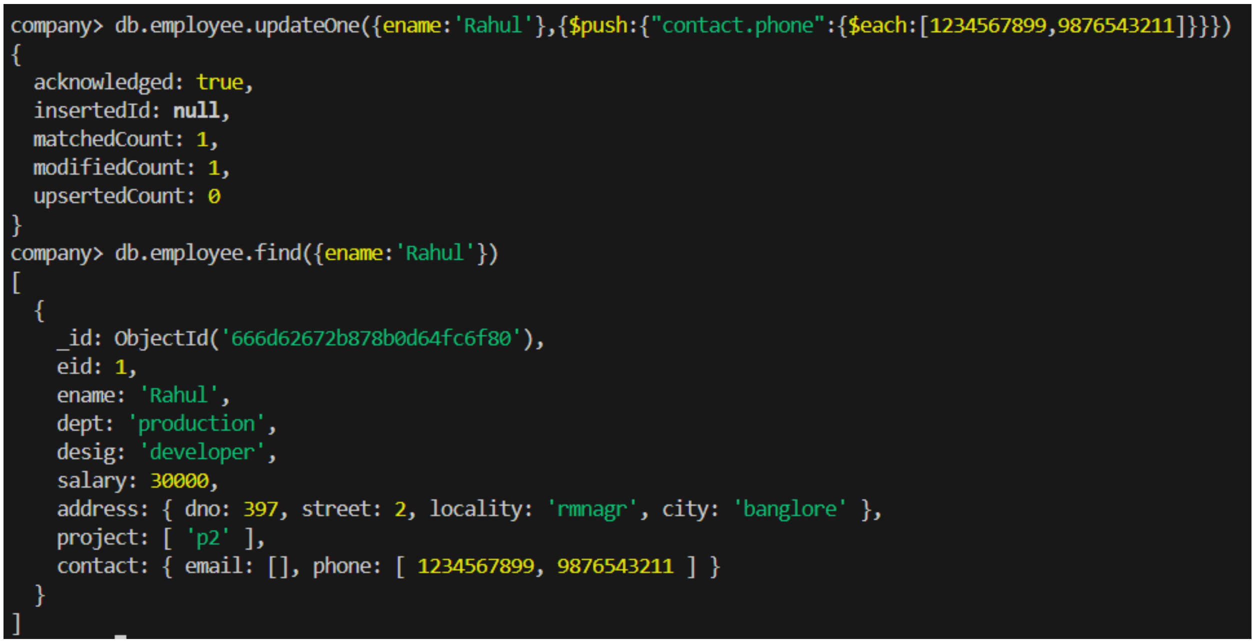
```
company>  db.employee.updateOne({ename:'Rahul'},{$pull:{project:'p3'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.employee.find({ename:'Rahul'})
[
  {
    _id: ObjectId('666d62672b878b0d64fc6f80'),
    eid: 1,
    ename: 'Rahul',
    dept: 'production',
    desig: 'developer',
    salary: 30000,
    address: { dno: 397, street: 2, locality: 'rmnagr', city: 'banglore' },
    project: [ 'p2' ]
  }
]
company>
```

11. Add a new embedded object “contacts” with “email” and “phone” as array objects to “Rahul”



```
company> db.employee.updateOne({ename:'Rahul'},{$set:{contact:{email:[],phone:[]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.employee.find({ename:'Rahal'})

company> db.employee.find({ename:'Rahul'})
[
  {
    _id: ObjectId('666d62672b878b0d64fc6f80'),
    eid: 1,
    ename: 'Rahul',
    dept: 'production',
    desig: 'developer',
    salary: 30000,
    address: { dno: 397, street: 2, locality: 'rmnagr', city: 'banglore' },
    project: [ 'p2' ],
    contact: { email: [], phone: [] }
  }
]
```

12. Add two phone numbers to “Rahul”



```
company> db.employee.updateOne({ename:'Rahul'},{$push:{"contact.phone":{$each:[1234567899,9876543211]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
company> db.employee.find({ename:'Rahul'})
[
  {
    _id: ObjectId('666d62672b878b0d64fc6f80'),
    eid: 1,
    ename: 'Rahul',
    dept: 'production',
    desig: 'developer',
    salary: 30000,
    address: { dno: 397, street: 2, locality: 'rmnagr', city: 'banglore' },
    project: [ 'p2' ],
    contact: { email: [], phone: [ 1234567899, 9876543211 ] }
  }
]
```

Exercise 3 - Book Database

Agenda: Create database, Create collection, insert data, find, sort, limit, $all, $in.

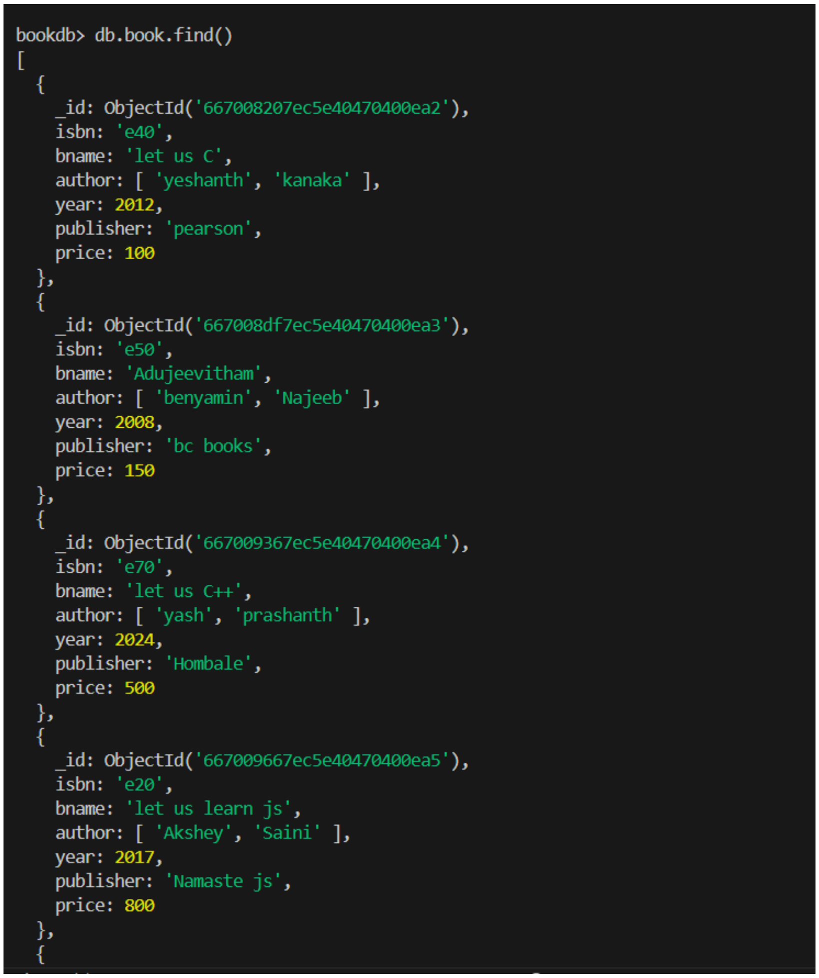
Create a book Database with the fields: (isbn, bname, author[], year, publisher, price)

```
> use bookdb
switched to db bookdb

> doc1=({isbn:"e40", bname:"let us C", author:["yeshanth", "kanaka"], year:2012,
publisher:"pearson", price:100})
{
        "isbn" : "e40",
        "bname" : "let us C",
        "author" : [
                "yeshanth",
                "kanaka"
        ],
        "year" : 2012,
        "publisher" : "pearson",
        "price" : 100
}
> db.book.insert(doc1)
```

Note: insert 5 documents.

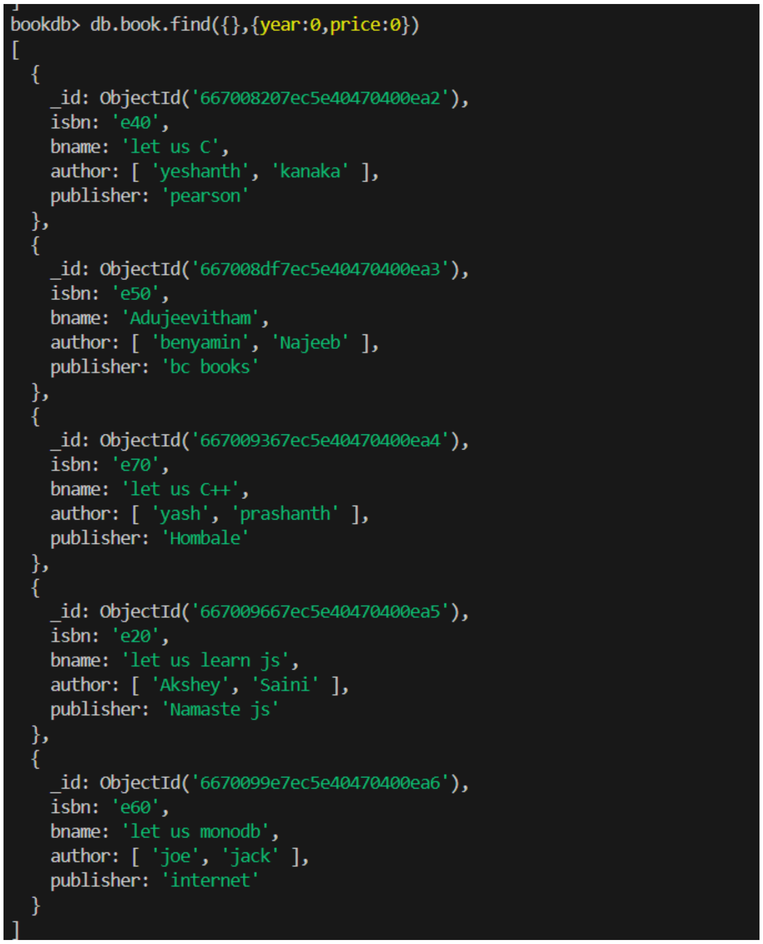
Questions:

1. list all the documents.



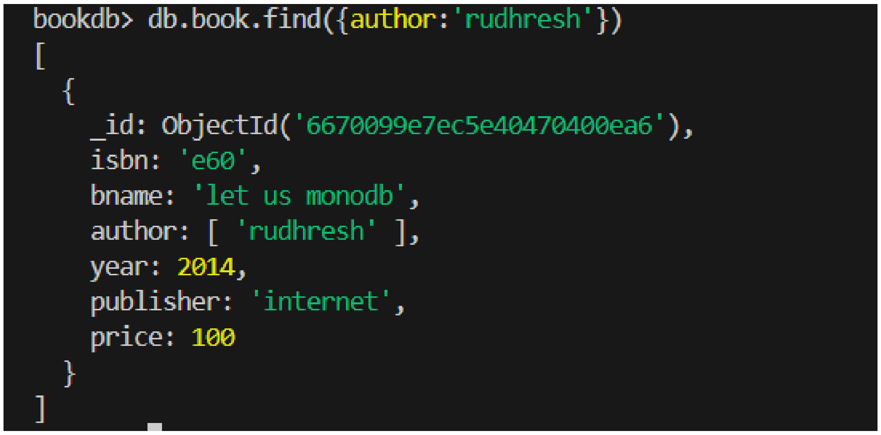
```
bookdb> db.book.find()
[
  {
    _id: ObjectId('667008207ec5e40470400ea2'),
    isbn: 'e40',
    bname: 'let us C',
    author: [ 'yeshanth', 'kanaka' ],
    year: 2012,
    publisher: 'pearson',
    price: 100
  },
  {
    _id: ObjectId('667008df7ec5e40470400ea3'),
    isbn: 'e50',
    bname: 'Adujeevitham',
    author: [ 'benyamin', 'Najeeb' ],
    year: 2008,
    publisher: 'bc books',
    price: 150
  },
  {
    _id: ObjectId('667009367ec5e40470400ea4'),
    isbn: 'e70',
    bname: 'let us C++',
    author: [ 'yash', 'prashanth' ],
    year: 2024,
    publisher: 'Hombale',
    price: 500
  },
  {
    _id: ObjectId('667009667ec5e40470400ea5'),
    isbn: 'e20',
    bname: 'let us learn js',
    author: [ 'Akshey', 'Saini' ],
    year: 2017,
    publisher: 'Namaste js',
    price: 800
  },
  {
```

2. list all the book name except year and price.



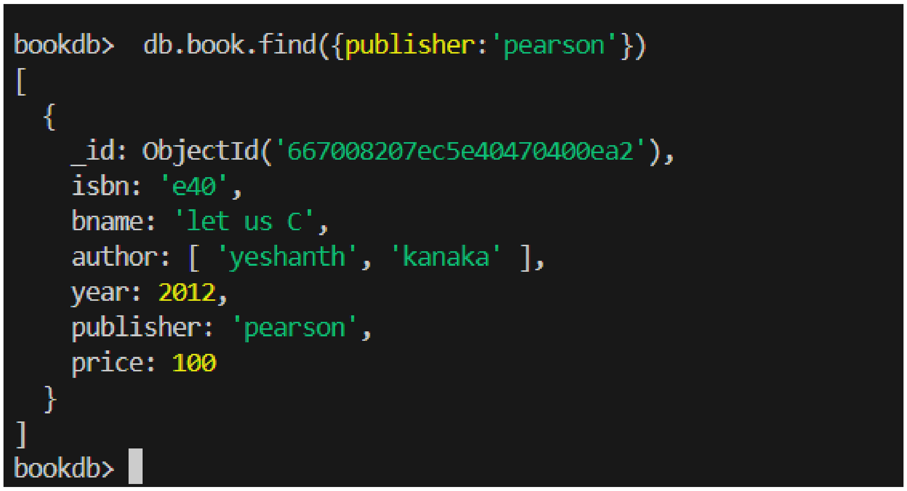
```
bookdb> db.book.find({},{year:0,price:0})
[
  {
    _id: ObjectId('667008207ec5e40470400ea2'),
    isbn: 'e40',
    bname: 'let us C',
    author: [ 'yeshanth', 'kanaka' ],
    publisher: 'pearson'
  },
  {
    _id: ObjectId('667008df7ec5e40470400ea3'),
    isbn: 'e50',
    bname: 'Adujeevitham',
    author: [ 'benyamin', 'Najeeb' ],
    publisher: 'bc books'
  },
  {
    _id: ObjectId('667009367ec5e40470400ea4'),
    isbn: 'e70',
    bname: 'let us C++',
    author: [ 'yash', 'prashanth' ],
    publisher: 'Hombale'
  },
  {
    _id: ObjectId('667009667ec5e40470400ea5'),
    isbn: 'e20',
    bname: 'let us learn js',
    author: [ 'Akshey', 'Saini' ],
    publisher: 'Namaste js'
  },
  {
    _id: ObjectId('6670099e7ec5e40470400ea6'),
    isbn: 'e60',
    bname: 'let us monodb',
    author: [ 'joe', 'jack' ],
    publisher: 'internet'
  }
]
```

3.display all the books authored by rudresh



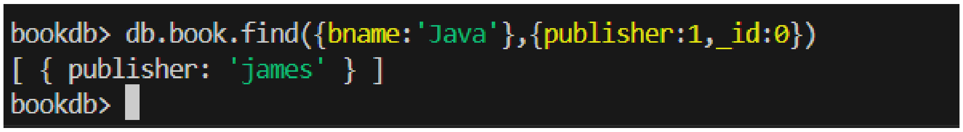
```
bookdb> db.book.find({author:'rudhresh'})
[
  {
    _id: ObjectId('6670099e7ec5e40470400ea6'),
    isbn: 'e60',
    bname: 'let us monodb',
    author: [ 'rudhresh' ],
    year: 2014,
    publisher: 'internet',
    price: 100
  }
]
```

4.list all the books published by pearson



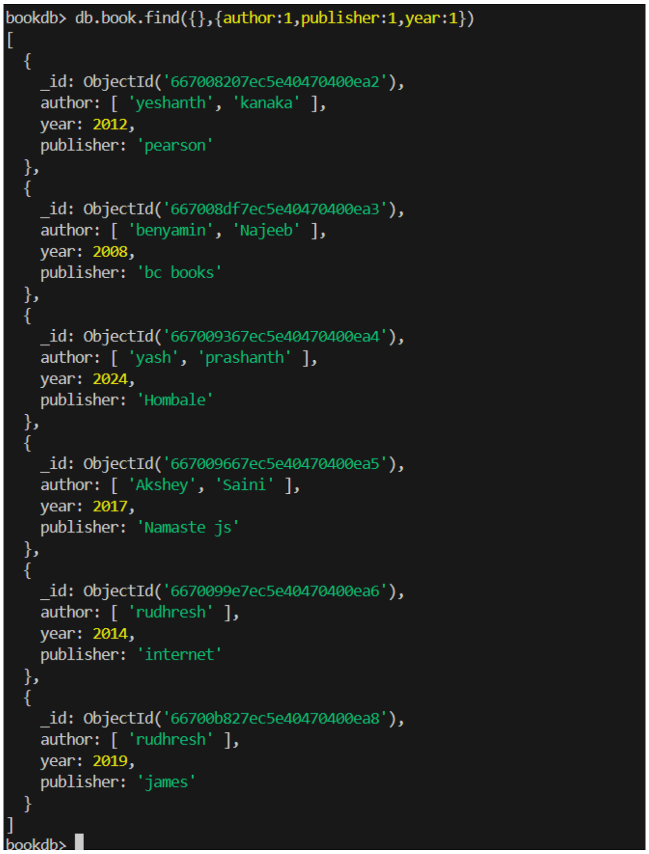
```
bookdb>  db.book.find({publisher:'pearson'})
[
  {
    _id: ObjectId('667008207ec5e40470400ea2'),
    isbn: 'e40',
    bname: 'let us C',
    author: [ 'yeshanth', 'kanaka' ],
    year: 2012,
    publisher: 'pearson',
    price: 100
  }
]
bookdb> 
```

5.list the publisher of book java



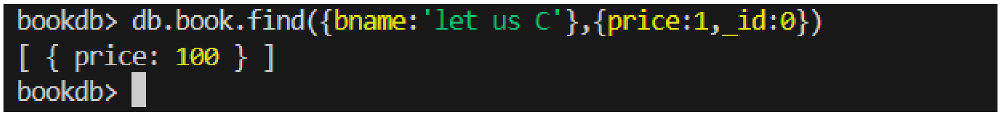
```
bookdb> db.book.find({bname:'Java'},{publisher:1,_id:0})
[ { publisher: 'james' } ]
bookdb> 
```

6.list the author,publisher and year of the book let us see.



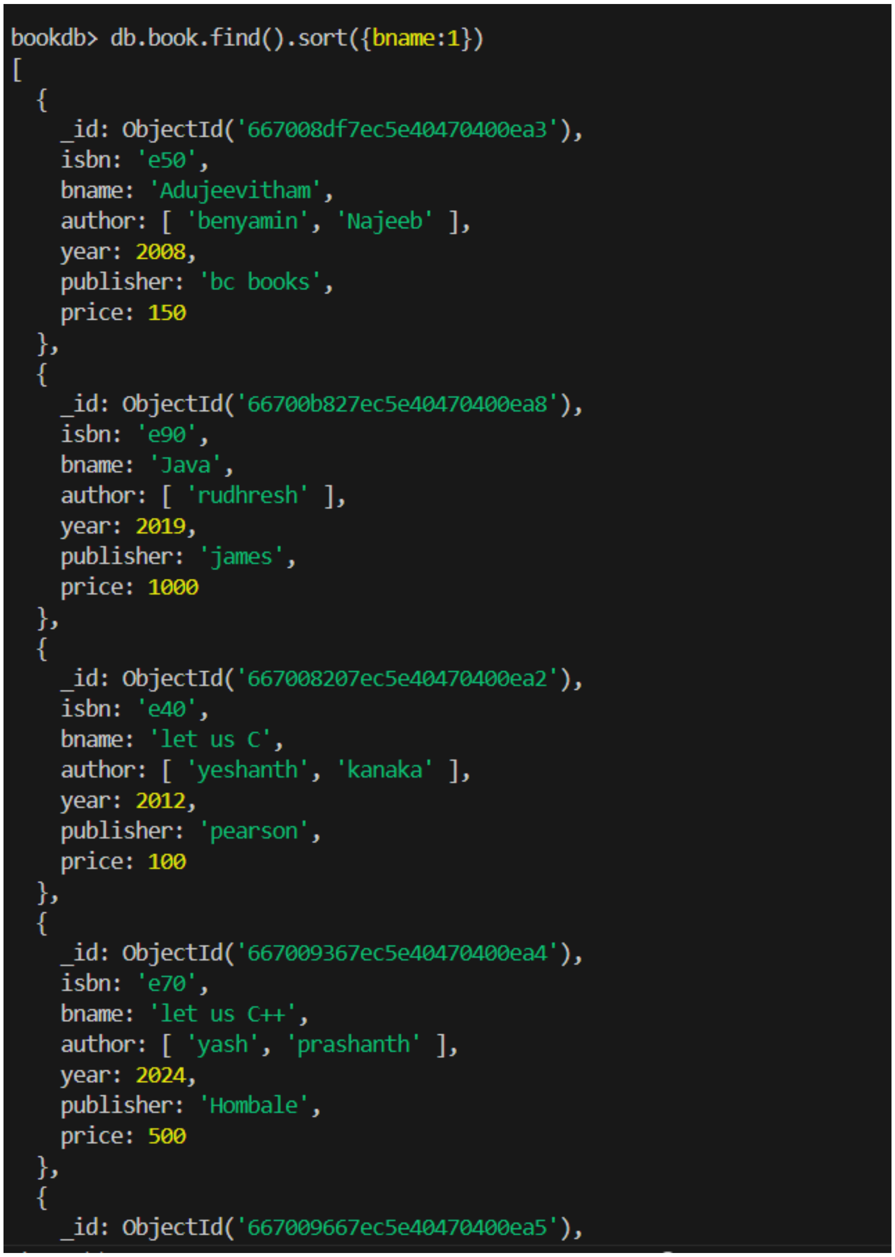
```
bookdb> db.book.find({},{author:1,publisher:1,year:1})
[
  {
    _id: ObjectId('667008207ec5e40470400ea2'),
    author: [ 'yeshanth', 'kanaka' ],
    year: 2012,
    publisher: 'pearson'
  },
  {
    _id: ObjectId('667008df7ec5e40470400ea3'),
    author: [ 'benyamin', 'Najeeb' ],
    year: 2008,
    publisher: 'bc books'
  },
  {
    _id: ObjectId('667009367ec5e40470400ea4'),
    author: [ 'yash', 'prashanth' ],
    year: 2024,
    publisher: 'Hombale'
  },
  {
    _id: ObjectId('667009667ec5e40470400ea5'),
    author: [ 'Akshey', 'Saini' ],
    year: 2017,
    publisher: 'Namaste js'
  },
  {
    _id: ObjectId('6670099e7ec5e40470400ea6'),
    author: [ 'rudhresh' ],
    year: 2014,
    publisher: 'internet'
  },
  {
    _id: ObjectId('66700b827ec5e40470400ea8'),
    author: [ 'rudhresh' ],
    year: 2019,
    publisher: 'james'
  }
]
bookdb> 
```

7.Display the price of “let us see” except _id



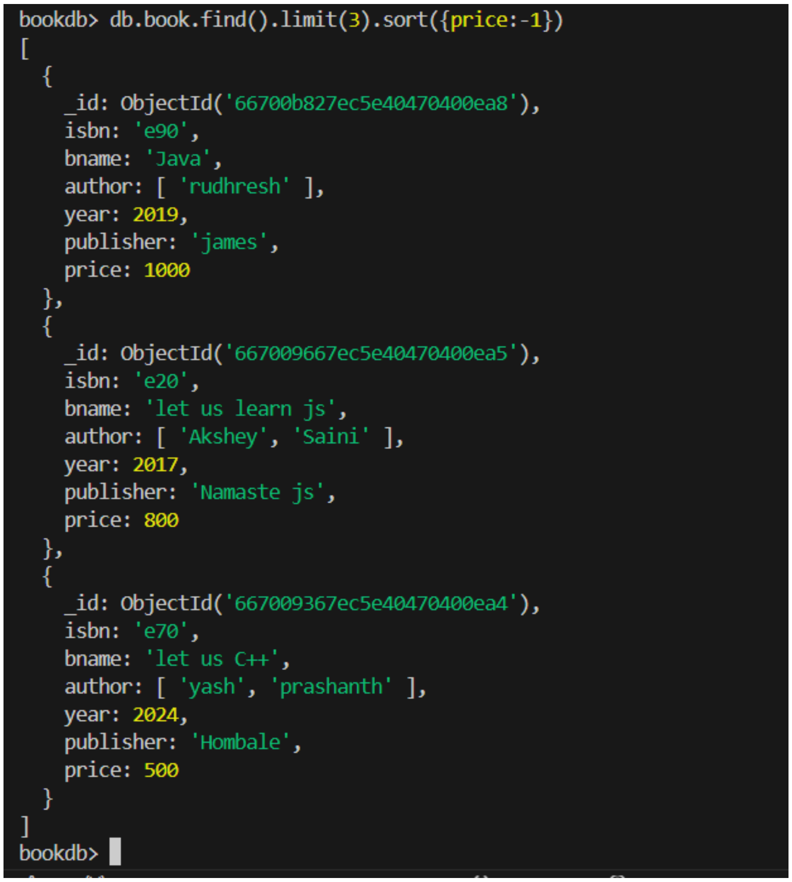
```
bookdb> db.book.find({bname:'let us C'},{price:1,_id:0})
[ { price: 100 } ]
bookdb> 
```

8.sort and display all books in ascending order of book names



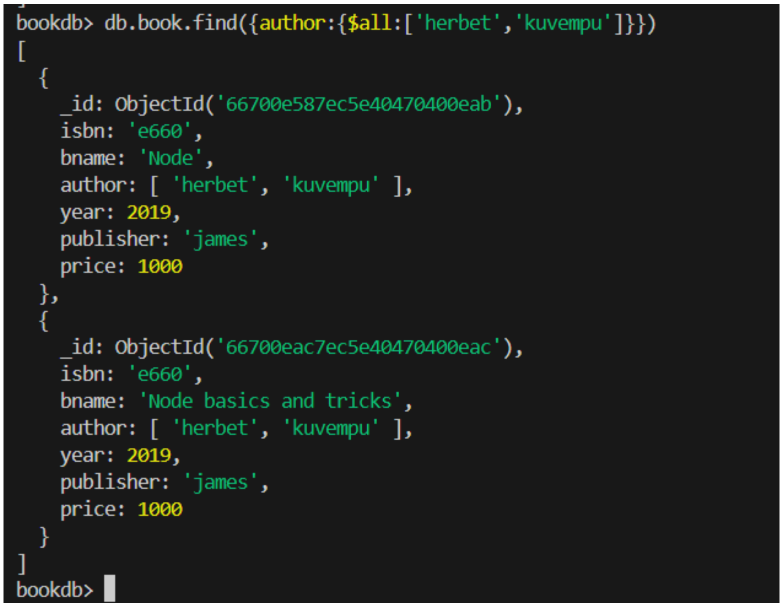
```
bookdb> db.book.find().sort({bname:1})
[
  {
    _id: ObjectId('667008df7ec5e40470400ea3'),
    isbn: 'e50',
    bname: 'Adujeevitham',
    author: [ 'benyamin', 'Najeeb' ],
    year: 2008,
    publisher: 'bc books',
    price: 150
  },
  {
    _id: ObjectId('66700b827ec5e40470400ea8'),
    isbn: 'e90',
    bname: 'Java',
    author: [ 'rudhresh' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  },
  {
    _id: ObjectId('667008207ec5e40470400ea2'),
    isbn: 'e40',
    bname: 'let us C',
    author: [ 'yeshanth', 'kanaka' ],
    year: 2012,
    publisher: 'pearson',
    price: 100
  },
  {
    _id: ObjectId('667009367ec5e40470400ea4'),
    isbn: 'e70',
    bname: 'let us C++',
    author: [ 'yash', 'prashanth' ],
    year: 2024,
    publisher: 'Hombale',
    price: 500
  },
  {
    _id: ObjectId('667009667ec5e40470400ea5'),
```

9.sort and display only 3 books in descending order of price.



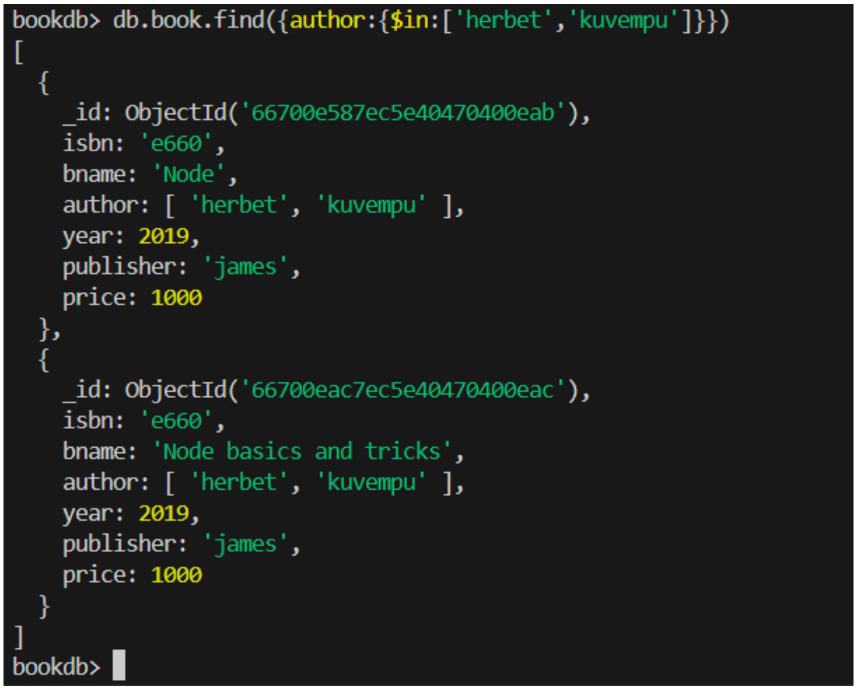
```
bookdb> db.book.find().limit(3).sort({price:-1})
[
  {
    _id: ObjectId('66700b827ec5e40470400ea8'),
    isbn: 'e90',
    bname: 'Java',
    author: [ 'rudhresh' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  },
  {
    _id: ObjectId('667009667ec5e40470400ea5'),
    isbn: 'e20',
    bname: 'let us learn js',
    author: [ 'Akshey', 'Saini' ],
    year: 2017,
    publisher: 'Namaste js',
    price: 800
  },
  {
    _id: ObjectId('667009367ec5e40470400ea4'),
    isbn: 'e70',
    bname: 'let us C++',
    author: [ 'yash', 'prashanth' ],
    year: 2024,
    publisher: 'Hombale',
    price: 500
  }
]
bookdb> █
```

10.Display all the books written by herbet and kuvempu



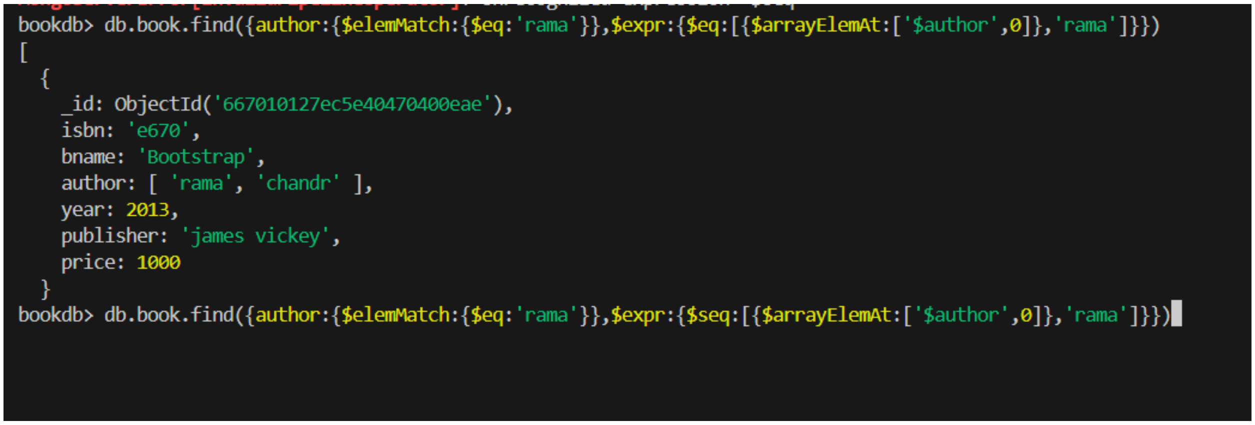
```
bookdb> db.book.find({author:{$all:['herbet','kuvempu']}})
[
  {
    _id: ObjectId('66700e587ec5e40470400eab'),
    isbn: 'e660',
    bname: 'Node',
    author: [ 'herbet', 'kuvempu' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  },
  {
    _id: ObjectId('66700eac7ec5e40470400eac'),
    isbn: 'e660',
    bname: 'Node basics and tricks',
    author: [ 'herbet', 'kuvempu' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  }
]
bookdb> █
```

11. Display all the books either written by herbet and kuvempu



```
bookdb> db.book.find({author:{$in:['herbet','kuvempu']}})
[
  {
    _id: ObjectId('66700e587ec5e40470400eab'),
    isbn: 'e660',
    bname: 'Node',
    author: [ 'herbet', 'kuvempu' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  },
  {
    _id: ObjectId('66700eac7ec5e40470400eac'),
    isbn: 'e660',
    bname: 'Node basics and tricks',
    author: [ 'herbet', 'kuvempu' ],
    year: 2019,
    publisher: 'james',
    price: 1000
  }
]
bookdb> 
```

12.display all the books where rama is the first author



```
bookdb> db.book.find({author:{$elemMatch:{$eq:'rama'}},$expr:{$eq:[{$arrayElemAt:['$author',0]},'rama']}})
[
  {
    _id: ObjectId('667010127ec5e40470400eae'),
    isbn: 'e670',
    bname: 'Bootstrap',
    author: [ 'rama', 'chandr' ],
    year: 2013,
    publisher: 'james vickey',
    price: 1000
  }
bookdb> db.book.find({author:{$elemMatch:{$eq:'rama'}},$expr:{$seq:[{$arrayElemAt:['$author',0]},'rama']}})
```

Exercise 4 - Food Database

Agenda: Create database, Create collection, insert data, find, find one, update, upsert, multi.

Create a Food Database with the fields: (food id, food cat, food name, chef name[ ], price, ingredients[], hotel name, hotel address { no, street, locality, city})

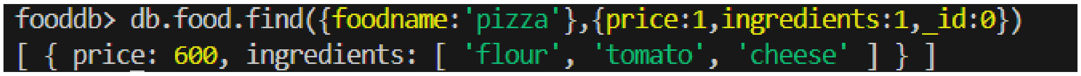
```
> use fooddb
switched to db fooddb
```

```
> doc1= {foodid:1, foodcat:"fastfood", foodname:"burger", chefname:["naveen","rakesh"], price:500,ingredients:["chees","corn"], hotelname:"mcburger", address:{no:31, street:"belroad", locality:"yelahanka", city:"bangalore"}}
{
        "foodid" : 1,
        "foodcat" : "fast food",
        "foodname" : "burger",
        "chefname" : [
                "naveen",
                "rakesh"
        ],
        "price" : 500,
        "ingredients" : [
                "chees",
                "corn"
        ],
        "hotelname" : "mcburger",
        "address" : {
                "no" : 31,
                "street" : "belroad",
                "locality" : "yelahanka",
                "city" : "bangalore"
        }
}
```

Note: insert 5 documents.

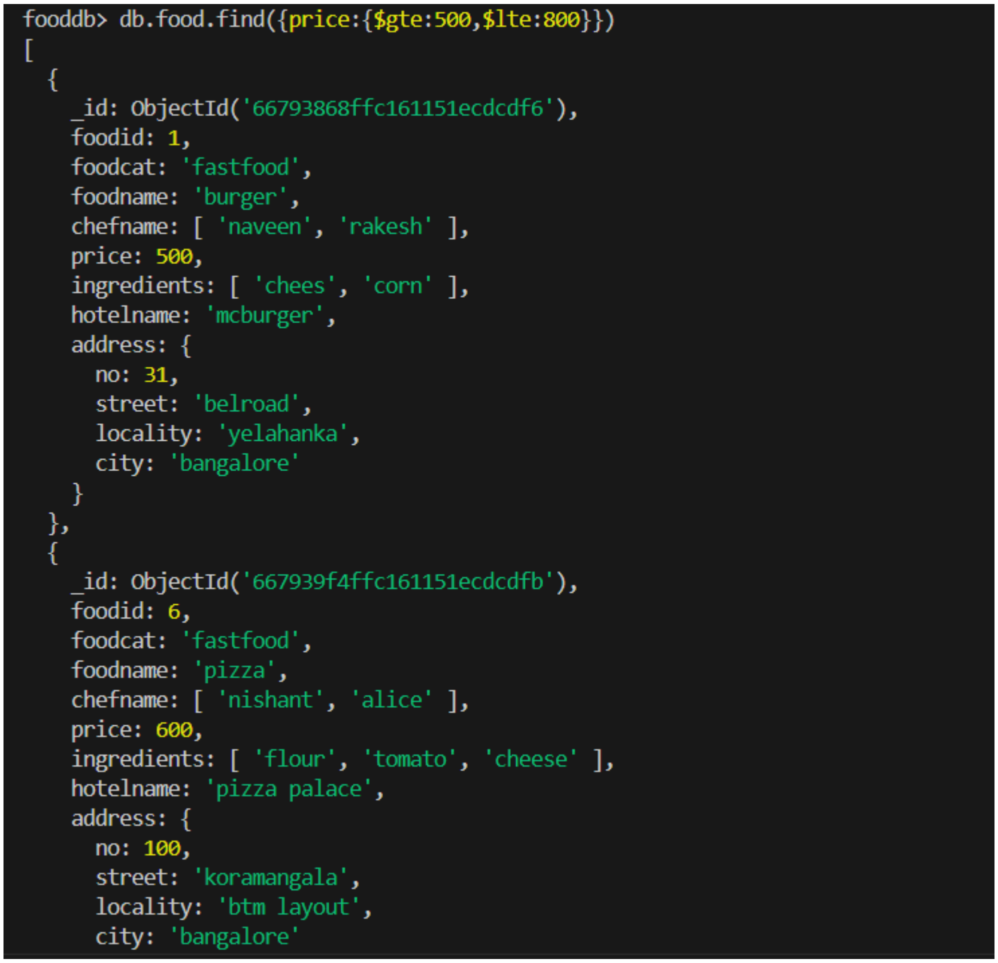
Questions:

1. list the price of pizza with ingredients.



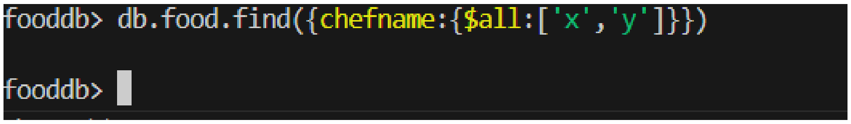
```
fooddb> db.food.find({foodname:'pizza'},{price:1,ingredients:1,_id:0})
[ { price: 600, ingredients: [ 'flour', 'tomato', 'cheese' ] } ]
```

2. display the item in the price range(500,800)



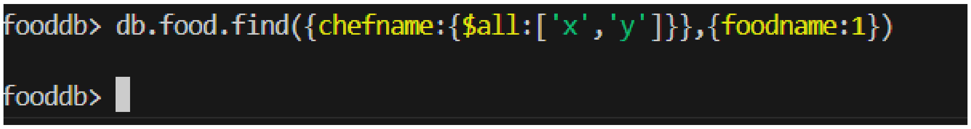
```
fooddb> db.food.find({price:{$gte:500,$lte:800}})
[
  {
    _id: ObjectId('66793868ffc161151ecdcdf6'),
    foodid: 1,
    foodcat: 'fastfood',
    foodname: 'burger',
    chefname: [ 'naveen', 'rakesh' ],
    price: 500,
    ingredients: [ 'chees', 'corn' ],
    hotelname: 'mcburger',
    address: {
      no: 31,
      street: 'belroad',
      locality: 'yelahanka',
      city: 'bangalore'
    }
  },
  {
    _id: ObjectId('667939f4ffc161151ecdcdfb'),
    foodid: 6,
    foodcat: 'fastfood',
    foodname: 'pizza',
    chefname: [ 'nishant', 'alice' ],
    price: 600,
    ingredients: [ 'flour', 'tomato', 'cheese' ],
    hotelname: 'pizza palace',
    address: {
      no: 100,
      street: 'koramangala',
      locality: 'btm layout',
      city: 'bangalore'
```

3.display the item prepared by x and y



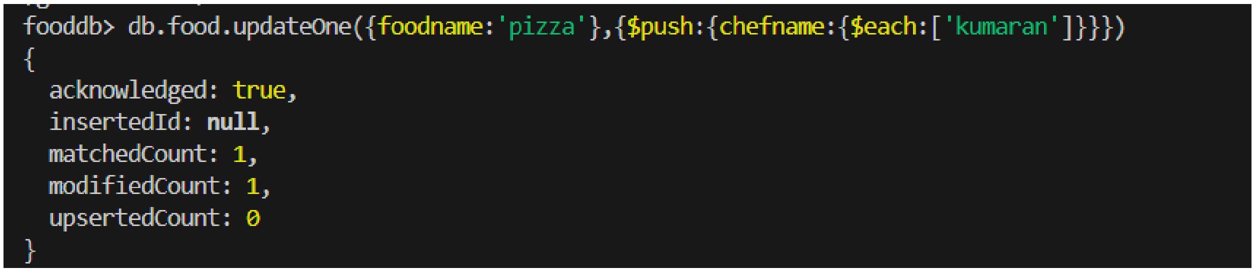
```
fooddb> db.food.find({chefname:{$all:['x','y']}})

fooddb> 
```

4. Display the item prepared by x or y



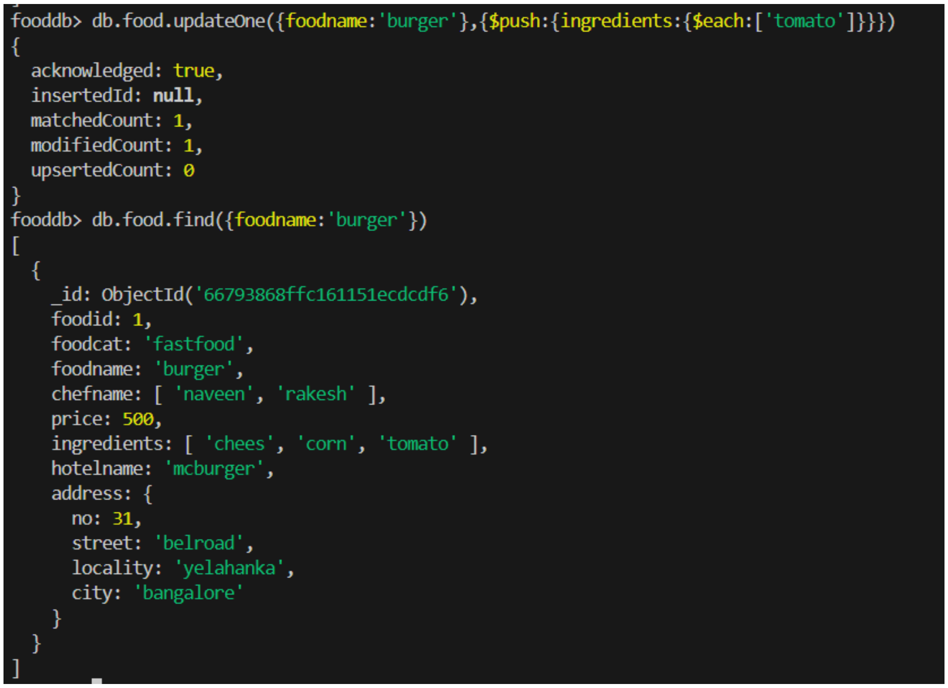
```
fooddb> db.food.find({chefname:{$all:['x','y']}},{foodname:1})

fooddb> 
```

5. Add one chef to the food pizza



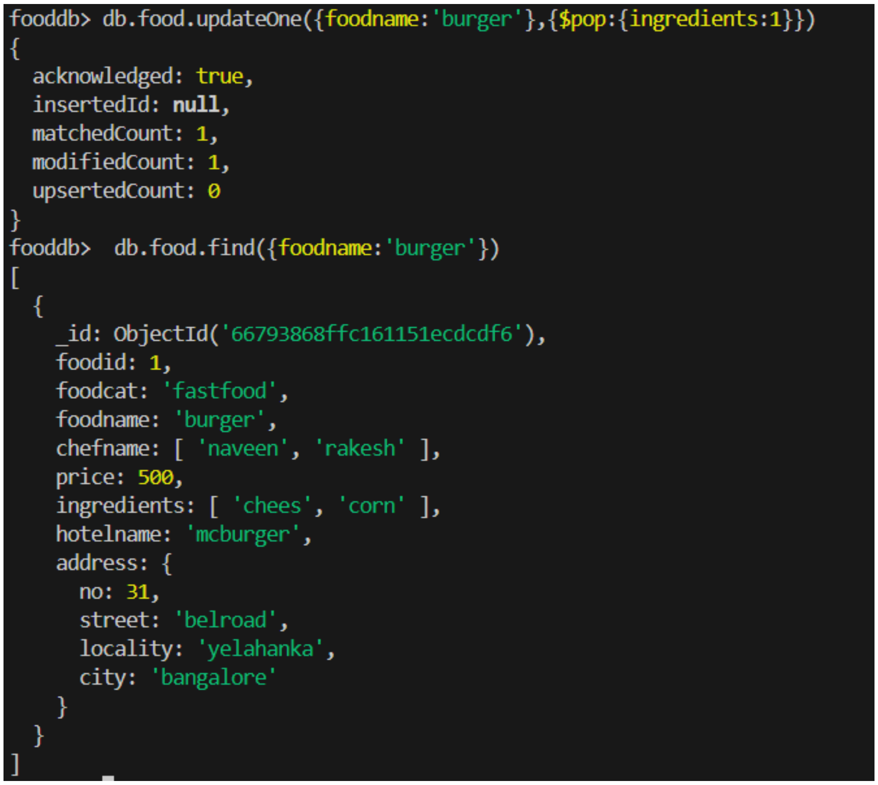
```
fooddb> db.food.updateOne({foodname:'pizza'},{$push:{chefname:{$each:['kumaran']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

6. Add ingredients to the food Burger



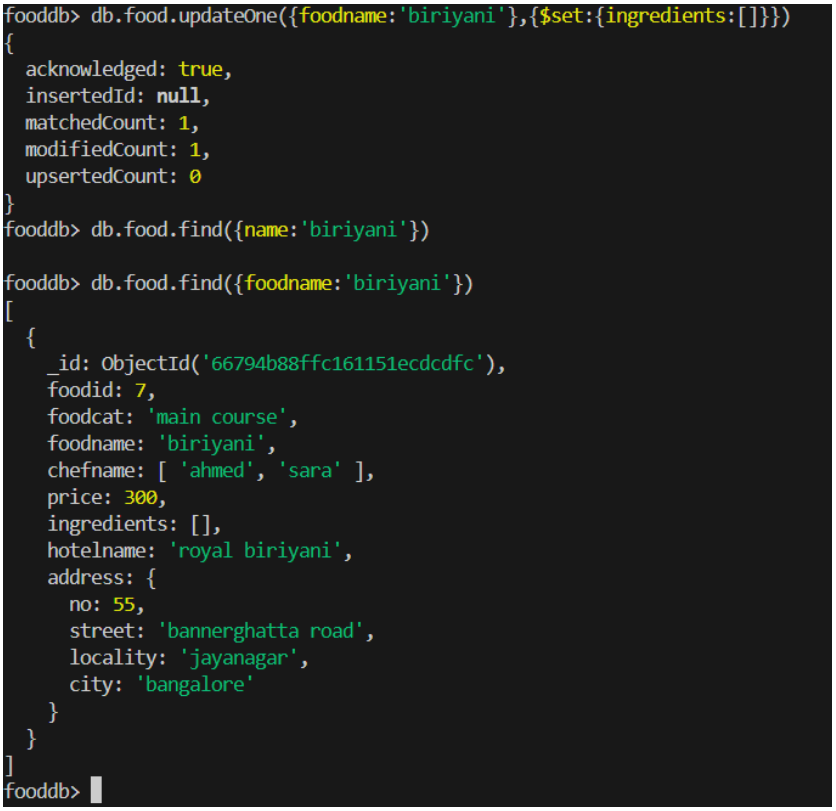
```
fooddb> db.food.updateOne({foodname:'burger'},{$push:{ingredients:{$each:['tomato']}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
fooddb> db.food.find({foodname:'burger'})
[
  {
    _id: ObjectId('66793868ffc161151ecdcdf6'),
    foodid: 1,
    foodcat: 'fastfood',
    foodname: 'burger',
    chefname: [ 'naveen', 'rakesh' ],
    price: 500,
    ingredients: [ 'chees', 'corn', 'tomato' ],
    hotelname: 'mcburger',
    address: {
      no: 31,
      street: 'belroad',
      locality: 'yelahanka',
      city: 'bangalore'
    }
  }
]
```

7. Delete last ingredient added to the food burger



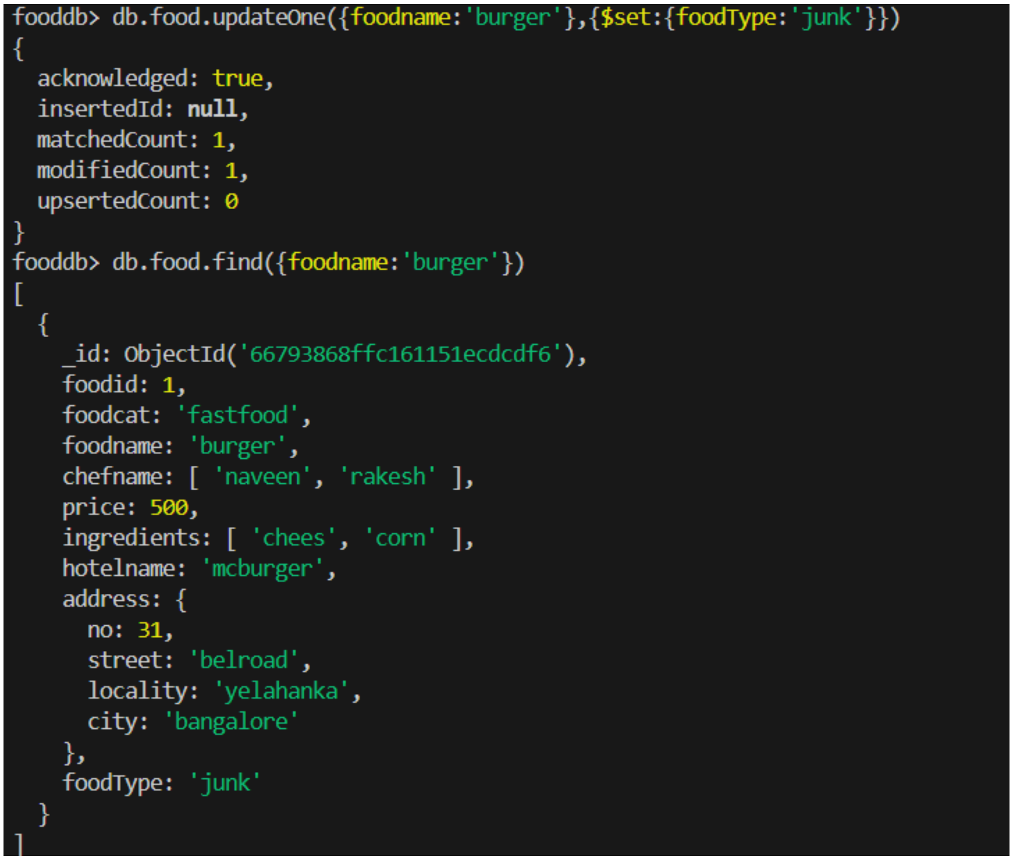
```
fooddb> db.food.updateOne({foodname:'burger'},{$pop:{ingredients:1}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
fooddb>  db.food.find({foodname:'burger'})
[
  {
    _id: ObjectId('66793868ffc161151ecdcdf6'),
    foodid: 1,
    foodcat: 'fastfood',
    foodname: 'burger',
    chefname: [ 'naveen', 'rakesh' ],
    price: 500,
    ingredients: [ 'chees', 'corn' ],
    hotelname: 'mcburger',
    address: {
      no: 31,
      street: 'belroad',
      locality: 'yelahanka',
      city: 'bangalore'
    }
  }
]
```

8.Delete All the ingredients from the food biryani



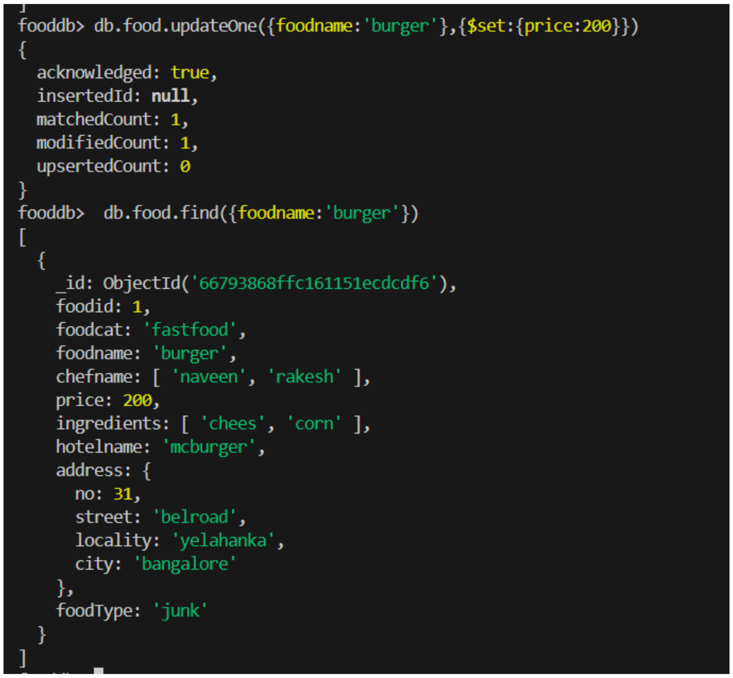
```
fooddb> db.food.updateOne({foodname:'biriyani'},{$set:{ingredients:[]}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
fooddb> db.food.find({name:'biriyani'})

fooddb> db.food.find({foodname:'biriyani'})
[
  {
    _id: ObjectId('66794b88ffc161151ecdcdfc'),
    foodid: 7,
    foodcat: 'main course',
    foodname: 'biriyani',
    chefname: [ 'ahmed', 'sara' ],
    price: 300,
    ingredients: [],
    hotelname: 'royal biriyani',
    address: {
      no: 55,
      street: 'bannerghatta road',
      locality: 'jayanagar',
      city: 'bangalore'
    }
  }
]
fooddb> 
```

9. Add food type to the food Burger.



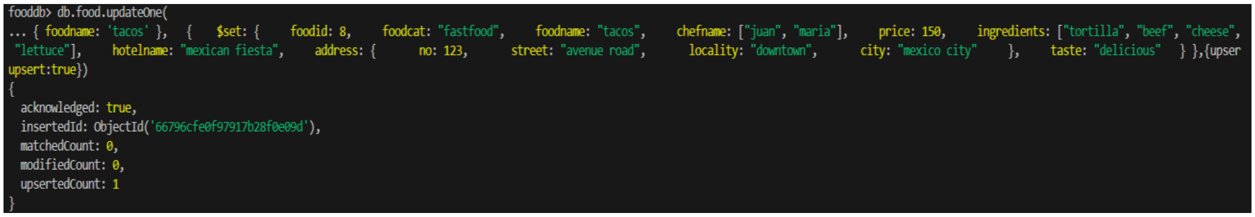
```
fooddb> db.food.updateOne({foodname:'burger'},{$set:{foodType:'junk'}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
fooddb> db.food.find({foodname:'burger'})
[
  {
    _id: ObjectId('66793868ffc161151ecdcdf6'),
    foodid: 1,
    foodcat: 'fastfood',
    foodname: 'burger',
    chefname: [ 'naveen', 'rakesh' ],
    price: 500,
    ingredients: [ 'chees', 'corn' ],
    hotelname: 'mcburger',
    address: {
      no: 31,
      street: 'belroad',
      locality: 'yelahanka',
      city: 'bangalore'
    },
    foodType: 'junk'
  }
]
```

10. Modify the burger price by 200



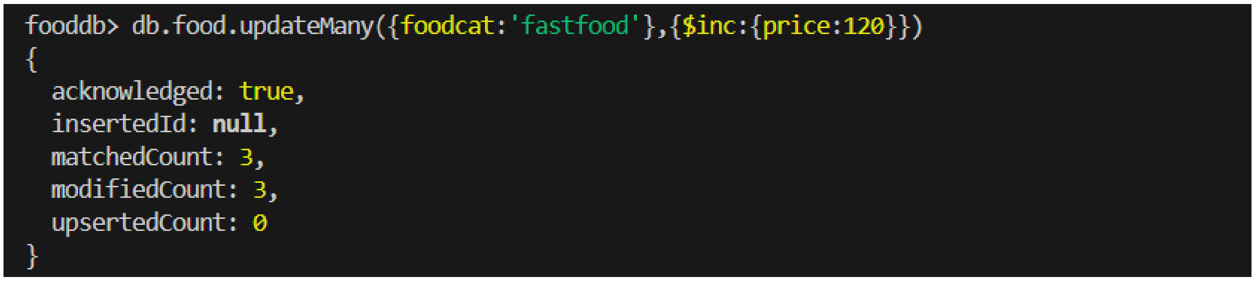
```
]
fooddb> db.food.updateOne({foodname:'burger'},{$set:{price:200}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
fooddb>  db.food.find({foodname:'burger'})
[
  {
    _id: ObjectId('66793868ffc161151ecdcdf6'),
    foodid: 1,
    foodcat: 'fastfood',
    foodname: 'burger',
    chefname: [ 'naveen', 'rakesh' ],
    price: 200,
    ingredients: [ 'chees', 'corn' ],
    hotelname: 'mcburger',
    address: {
      no: 31,
      street: 'belroad',
      locality: 'yelahanka',
      city: 'bangalore'
    },
    foodType: 'junk'
  }
]
```

11. Add or insert a new food item with the food Id “f08 “ using upsert as True.



```
fooddb> db.food.updateOne(
... { foodname: 'tacos' },   {    $set: {     foodid: 8,     foodcat: "fastfood",     foodname: "tacos",     chefname: ["juan", "maria"],     price: 150,     ingredients: ["tortilla", "beef", "cheese",
 "lettuce"],     hotelname: "mexican fiesta",     address: {       no: 123,       street: "avenue road",       locality: "downtown",       city: "mexico city"     },     taste: "delicious"   } },{upser
upsert:true})
{
  acknowledged: true,
  insertedId: ObjectId('66796cfe0f97917b28f0e09d'),
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 1
}
```

12. Increment the price of all food item in food cat: fastfood by 120.



```
fooddb> db.food.updateMany({foodcat:'fastfood'},{$inc:{price:120}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```