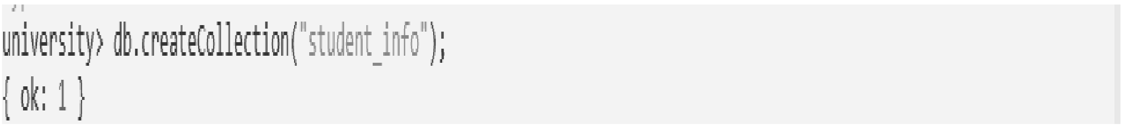
COURSE OUTCOME 4

Apply CRUD operations and retrieve data in a NoSQL environment

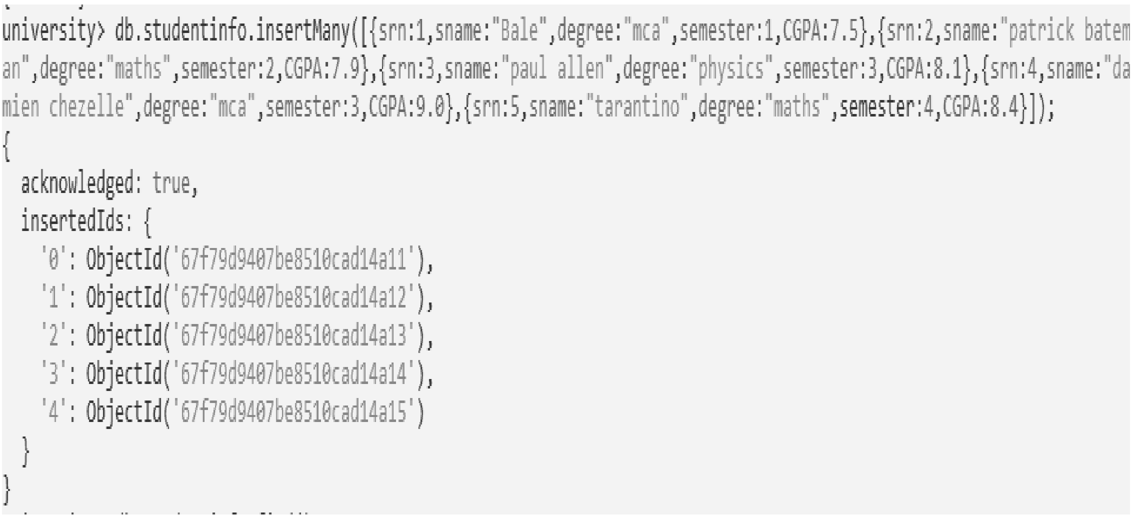
PROGRAM 1

1. Create a collection ‘student_info’



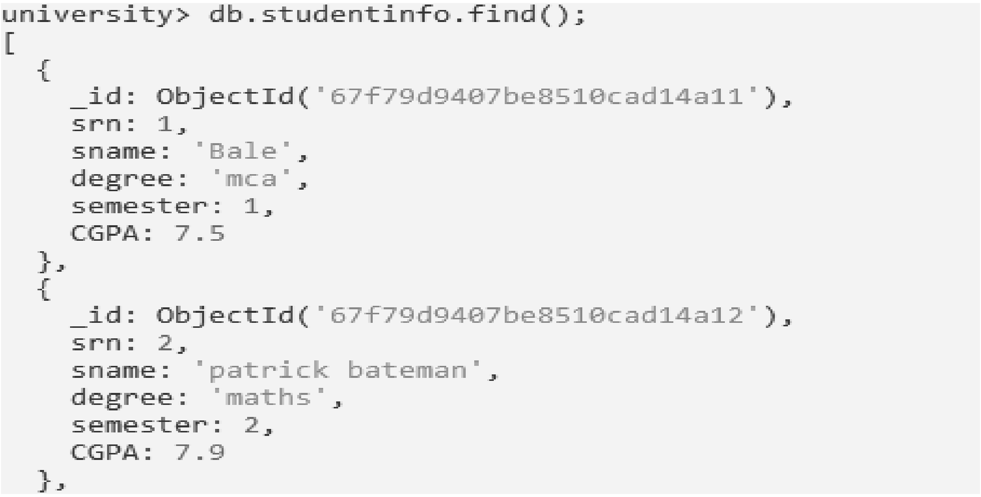
```
university> db.createCollection("student_info");
{ ok: 1 }
```

2. Insert values

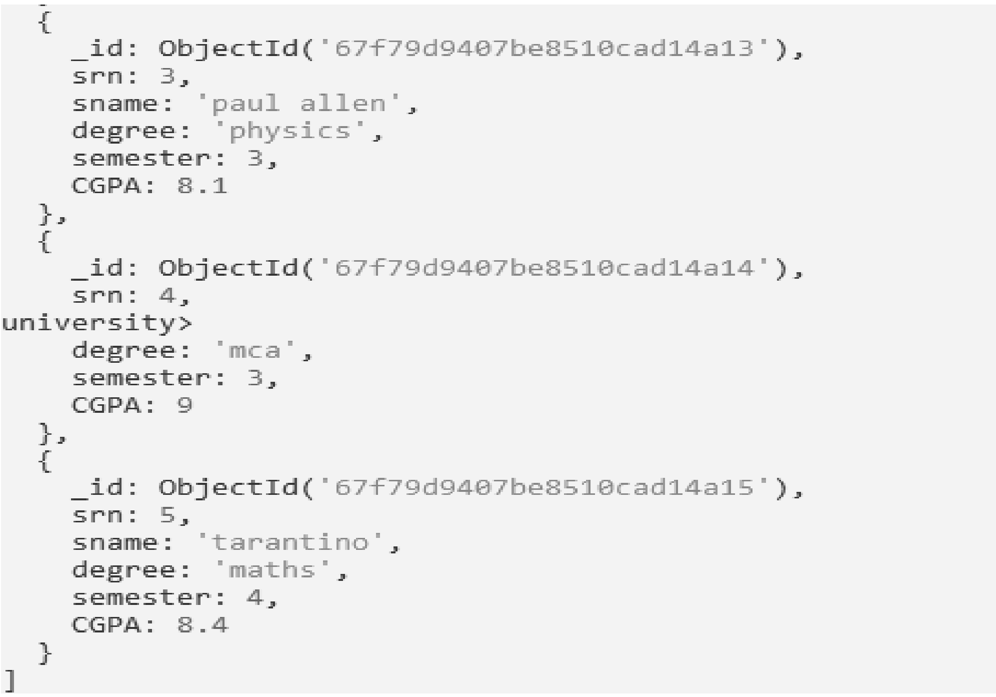


```
university> db.studentinfo.insertMany([{srn:1,sname:"Bale",degree:"mca",semester:1,CGPA:7.5},{srn:2,sname:"patrick batem
an",degree:"maths",semester:2,CGPA:7.9},{srn:3,sname:"paul allen",degree:"physics",semester:3,CGPA:8.1},{srn:4,sname:"da
mien chezelle",degree:"mca",semester:3,CGPA:9.0},{srn:5,sname:"tarantino",degree:"maths",semester:4,CGPA:8.4}]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67f79d9407be8510cad14a11'),
    '1': ObjectId('67f79d9407be8510cad14a12'),
    '2': ObjectId('67f79d9407be8510cad14a13'),
    '3': ObjectId('67f79d9407be8510cad14a14'),
    '4': ObjectId('67f79d9407be8510cad14a15')
  }
}
```

3. Display the collection

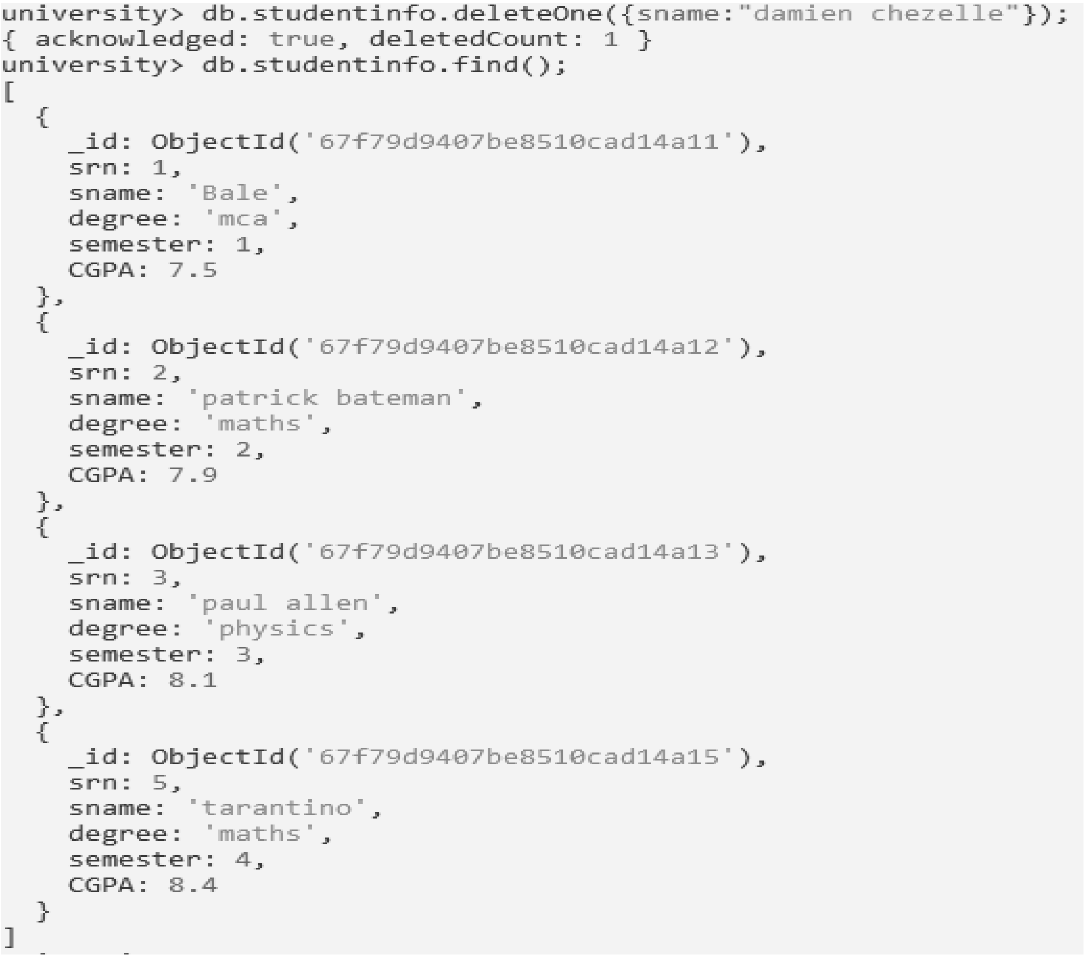


```
university> db.studentinfo.find();
[
  {
    _id: ObjectId('67f79d9407be8510cad14a11'),
    srn: 1,
    sname: 'Bale',
    degree: 'mca',
    semester: 1,
    CGPA: 7.5
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a12'),
    srn: 2,
    sname: 'patrick bateman',
    degree: 'maths',
    semester: 2,
    CGPA: 7.9
  },
```



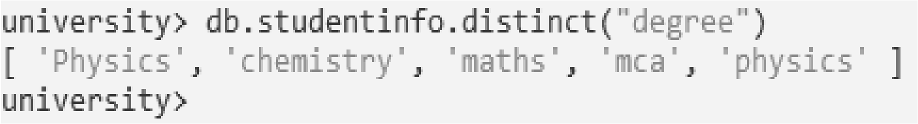
```
  {
    _id: ObjectId('67f79d9407be8510cad14a13'),
    srn: 3,
    sname: 'paul allen',
    degree: 'physics',
    semester: 3,
    CGPA: 8.1
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a14'),
    srn: 4,
university>
    degree: 'mca',
    semester: 3,
    CGPA: 9
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a15'),
    srn: 5,
    sname: 'tarantino',
    degree: 'maths',
    semester: 4,
    CGPA: 8.4
  }
]
```

4. Delete the student with name ‘damien chezelle’



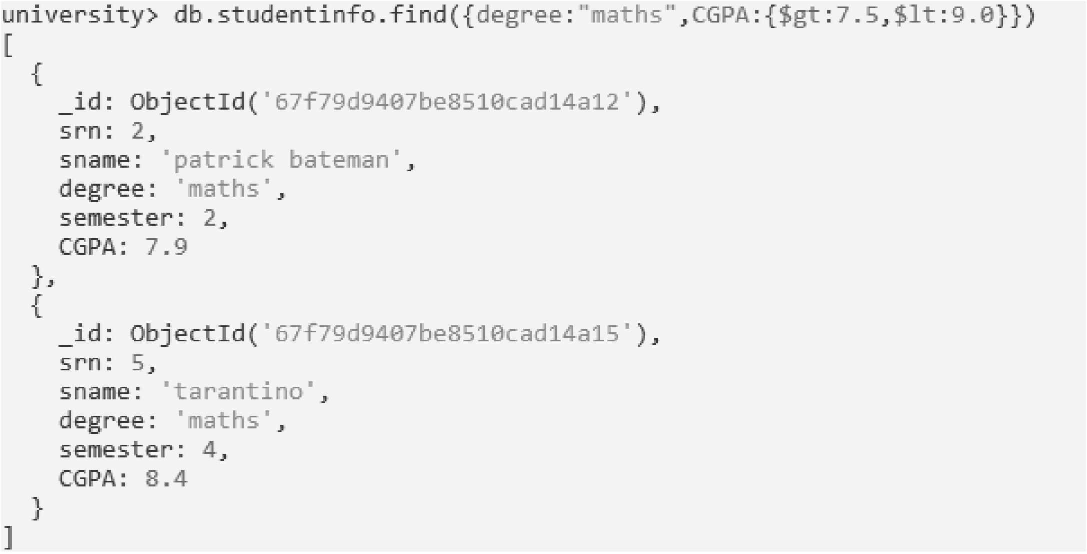
```
university> db.studentinfo.deleteOne({sname:"damien chezelle"});
{ acknowledged: true, deletedCount: 1 }
university> db.studentinfo.find();
[
  {
    _id: ObjectId('67f79d9407be8510cad14a11'),
    srn: 1,
    sname: 'Bale',
    degree: 'mca',
    semester: 1,
    CGPA: 7.5
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a12'),
    srn: 2,
    sname: 'patrick bateman',
    degree: 'maths',
    semester: 2,
    CGPA: 7.9
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a13'),
    srn: 3,
    sname: 'paul allen',
    degree: 'physics',
    semester: 3,
    CGPA: 8.1
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a15'),
    srn: 5,
    sname: 'tarantino',
    degree: 'maths',
    semester: 4,
    CGPA: 8.4
  }
]
```

5. Display the distinct degrees



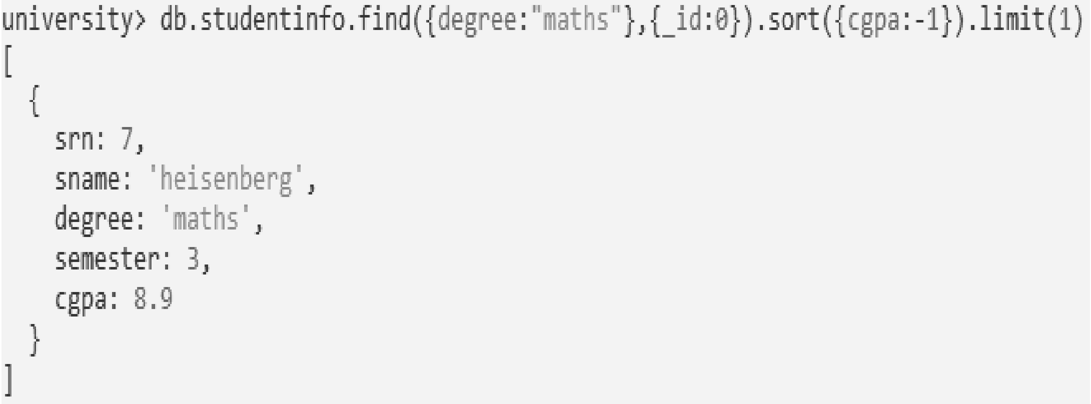
```
university> db.studentinfo.distinct("degree")
[ 'Physics', 'chemistry', 'maths', 'mca', 'physics' ]
university>
```

6. Display the details of students in degree ‘maths’ whose cgpa is between 7.5 and 8.5



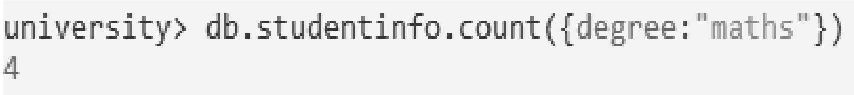
```
university> db.studentinfo.find({degree:"maths",CGPA:{$gt:7.5,$lt:9.0}})
[
  {
    _id: ObjectId('67f79d9407be8510cad14a12'),
    srn: 2,
    sname: 'patrick bateman',
    degree: 'maths',
    semester: 2,
    CGPA: 7.9
  },
  {
    _id: ObjectId('67f79d9407be8510cad14a15'),
    srn: 5,
    sname: 'tarantino',
    degree: 'maths',
    semester: 4,
    CGPA: 8.4
  }
]
```

7. Display the details of student who secured highest mark in the courses



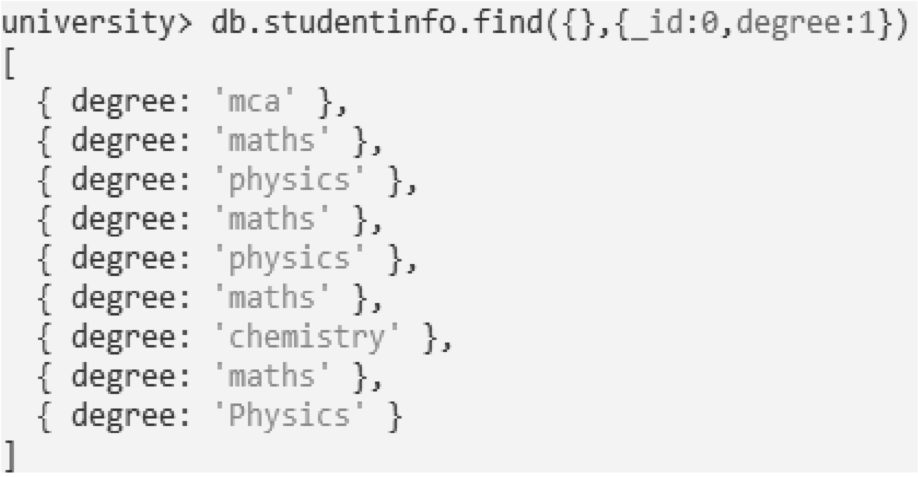
```
university> db.studentinfo.find({degree:"maths"},{_id:0}).sort({cgpa:-1}).limit(1)
[
  {
    srn: 7,
    sname: 'heisenberg',
    degree: 'maths',
    semester: 3,
    cgpa: 8.9
  }
]
```

8. Count the number of students in a particular degree



```
university> db.studentinfo.count({degree:"maths"})
4
```

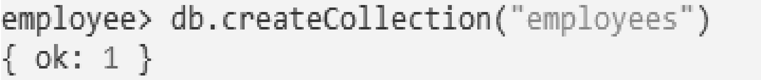
9. Display all the degree without id



```
university> db.studentinfo.find({},{_id:0,degree:1})
[
  { degree: 'mca' },
  { degree: 'maths' },
  { degree: 'physics' },
  { degree: 'maths' },
  { degree: 'physics' },
  { degree: 'maths' },
  { degree: 'chemistry' },
  { degree: 'maths' },
  { degree: 'Physics' }
]
```

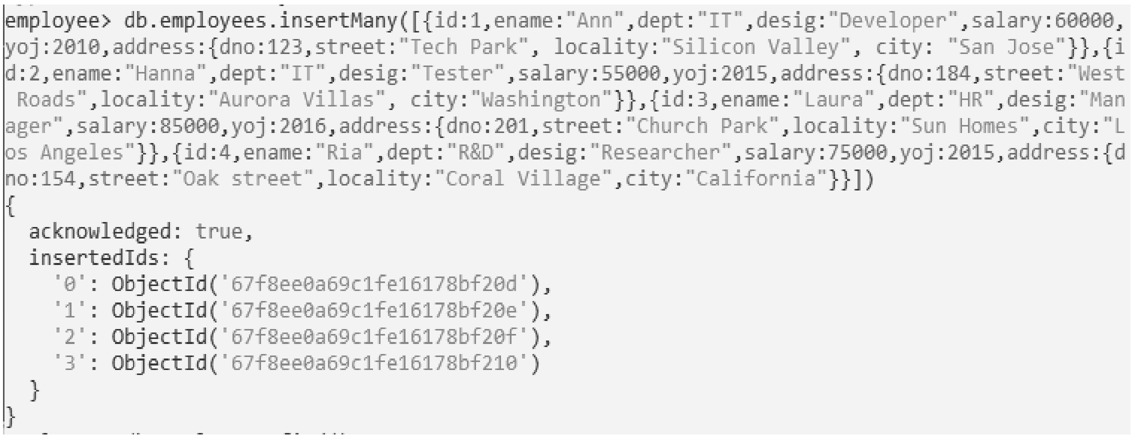
PROGRAM 2

1. Create a collection ‘employees’



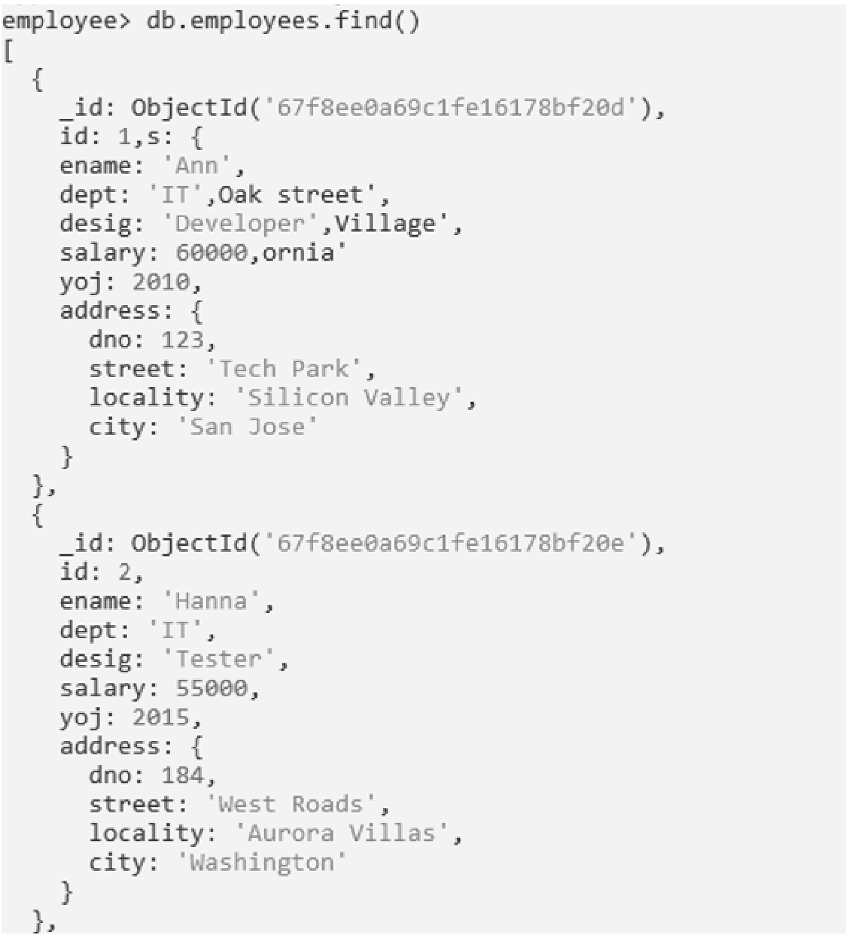
```
employee> db.createCollection("employees")
{ ok: 1 }
```

2. Insert values

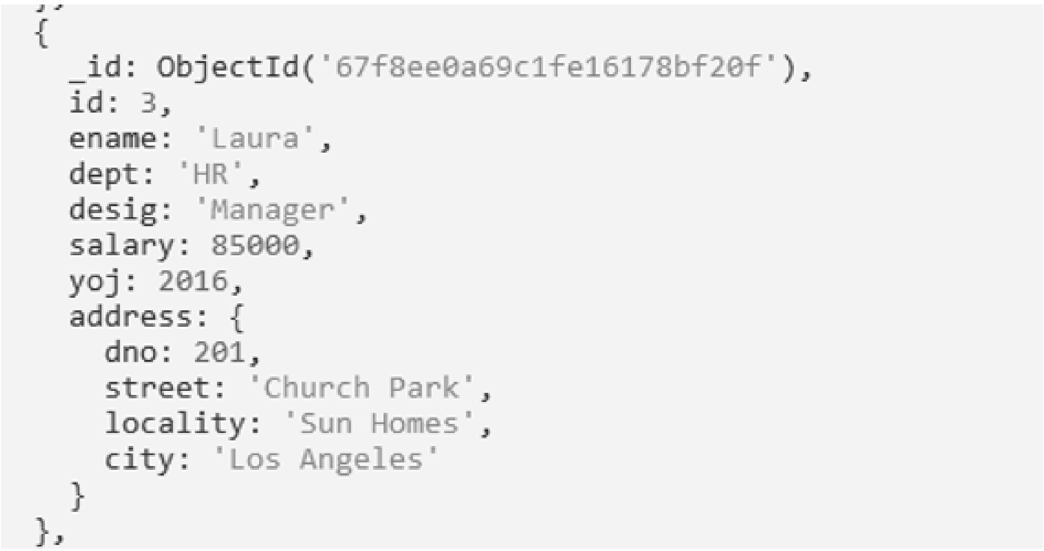


```
employee> db.employees.insertMany([{id:1,ename:"Ann",dept:"IT",desig:"Developer",salary:60000,
yoj:2010,address:{dno:123,street:"Tech Park", locality:"Silicon Valley", city: "San Jose"}},{i
d:2,ename:"Hanna",dept:"IT",desig:"Tester",salary:55000,yoj:2015,address:{dno:184,street:"West
 Roads",locality:"Aurora Villas", city:"Washington"}},{id:3,ename:"Laura",dept:"HR",desig:"Man
ager",salary:85000,yoj:2016,address:{dno:201,street:"Church Park",locality:"Sun Homes",city:"L
os Angeles"}},{id:4,ename:"Ria",dept:"R&D",desig:"Researcher",salary:75000,yoj:2015,address:{d
no:154,street:"Oak street",locality:"Coral Village",city:"California"}}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67f8ee0a69c1fe16178bf20d'),
    '1': ObjectId('67f8ee0a69c1fe16178bf20e'),
    '2': ObjectId('67f8ee0a69c1fe16178bf20f'),
    '3': ObjectId('67f8ee0a69c1fe16178bf210')
  }
}
```

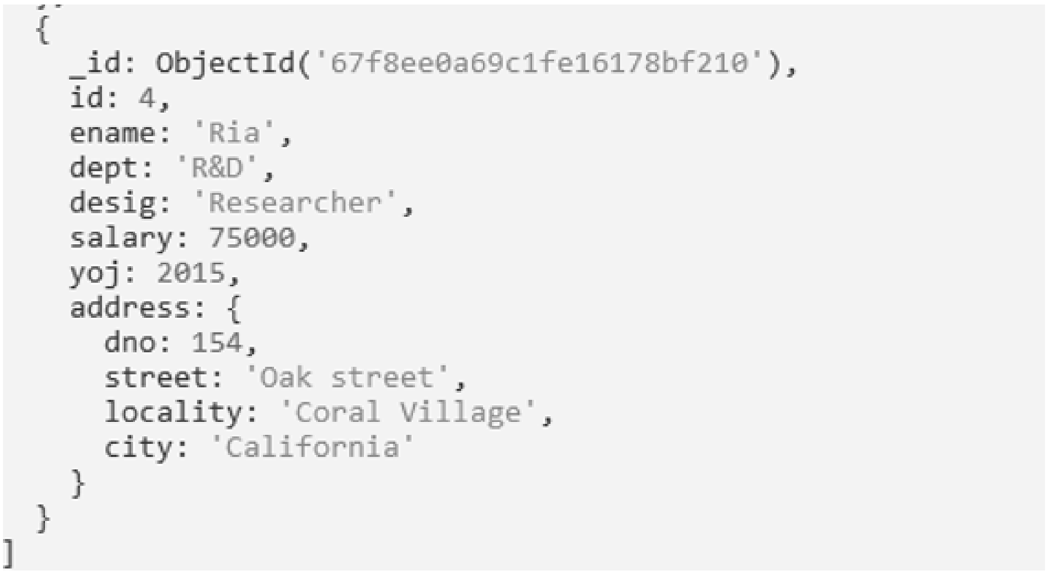
3. Display collection



```
employee> db.employees.find()
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,s: {
    ename: 'Ann',
    dept: 'IT',Oak street',
    desig: 'Developer',Village',
    salary: 60000,ornia'
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    }
  },
```

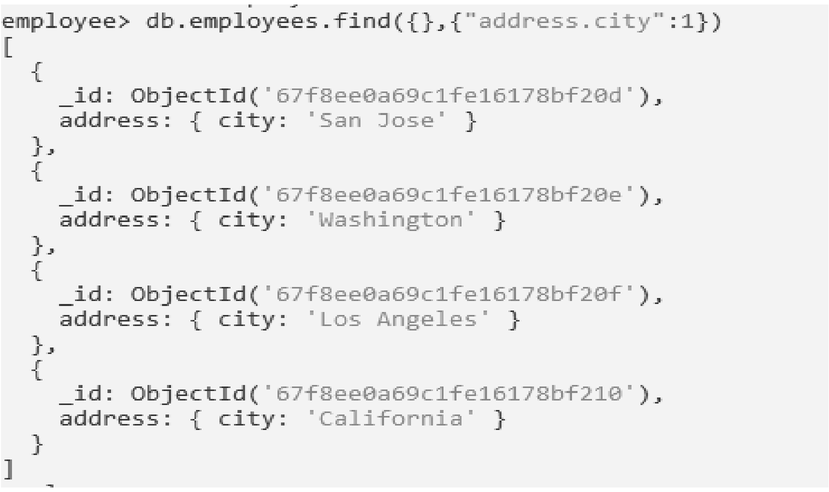


```
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20f'),
    id: 3,
    ename: 'Laura',
    dept: 'HR',
    desig: 'Manager',
    salary: 85000,
    yoj: 2016,
    address: {
      dno: 201,
      street: 'Church Park',
      locality: 'Sun Homes',
      city: 'Los Angeles'
    }
  },
```



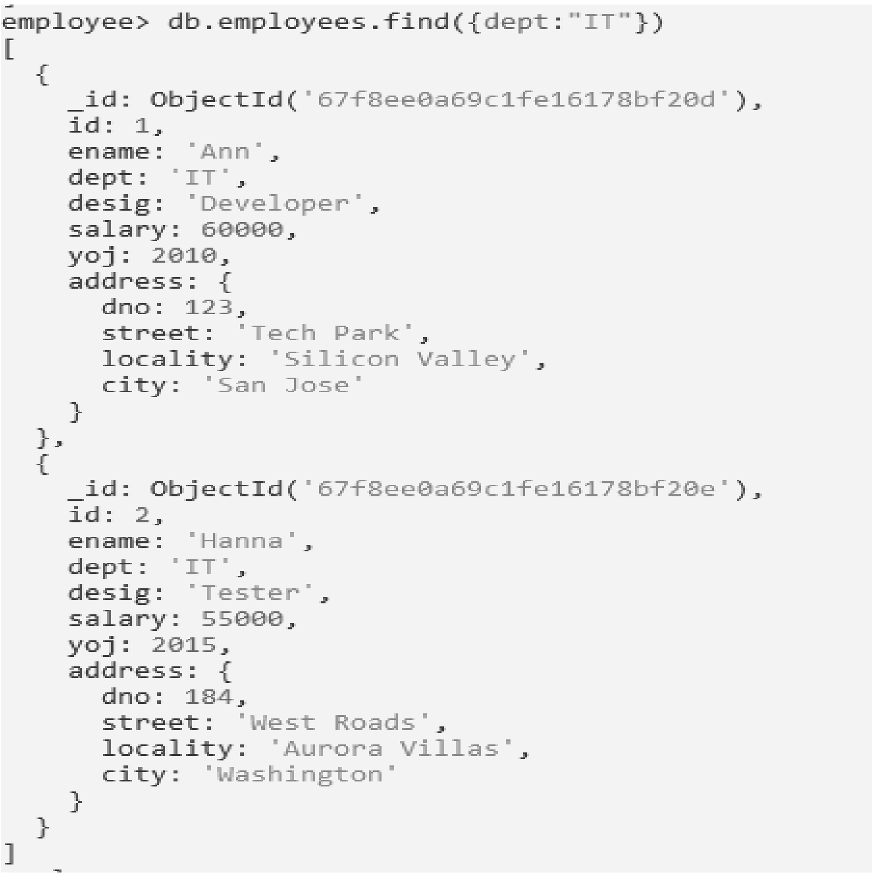
```
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    yoj: 2015,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    }
  }
]
```

4. Display city of the employees



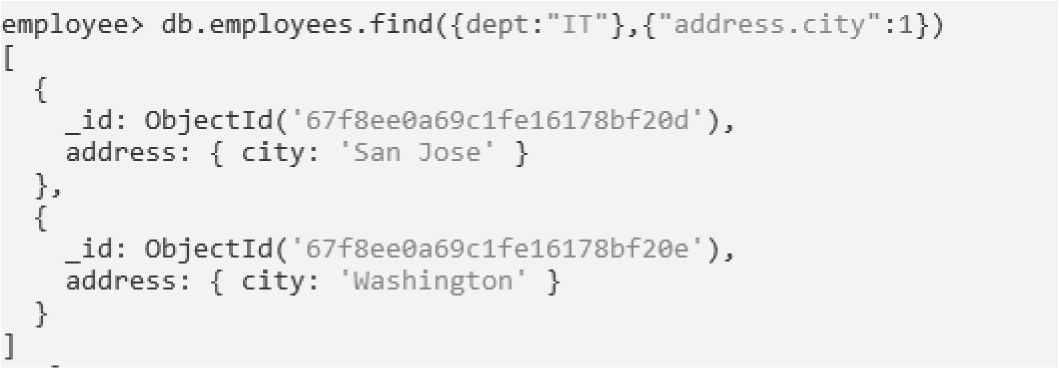
```
employee> db.employees.find({},{"address.city":1})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    address: { city: 'San Jose' }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    address: { city: 'Washington' }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20f'),
    address: { city: 'Los Angeles' }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    address: { city: 'California' }
  }
]
```

5. Display the details of the employees in IT department



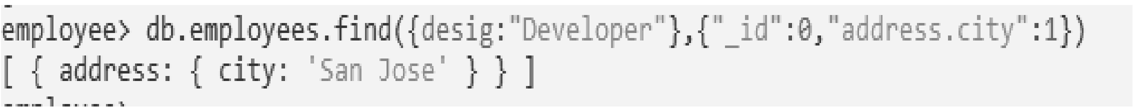
```
employee> db.employees.find({dept:"IT"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 60000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    }
  }
]
```

6. Display the city of employees in IT department



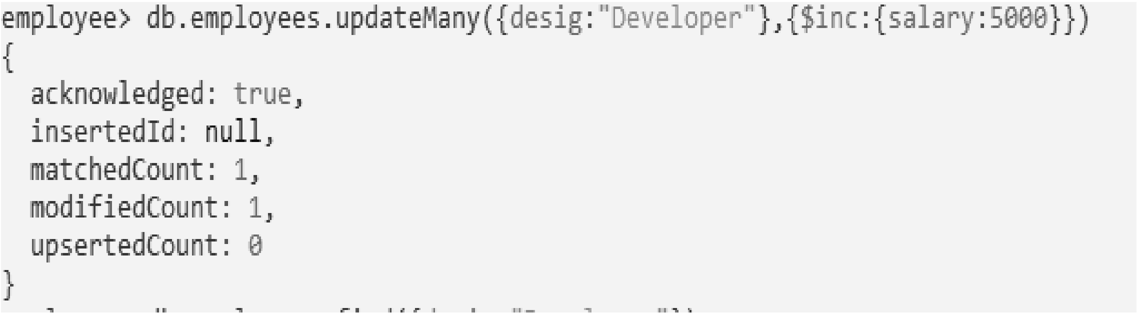
```
employee> db.employees.find({dept:"IT"},{"address.city":1})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    address: { city: 'San Jose' }
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    address: { city: 'Washington' }
  }
]
```

7. Display the city of employees with designation ‘Developer’

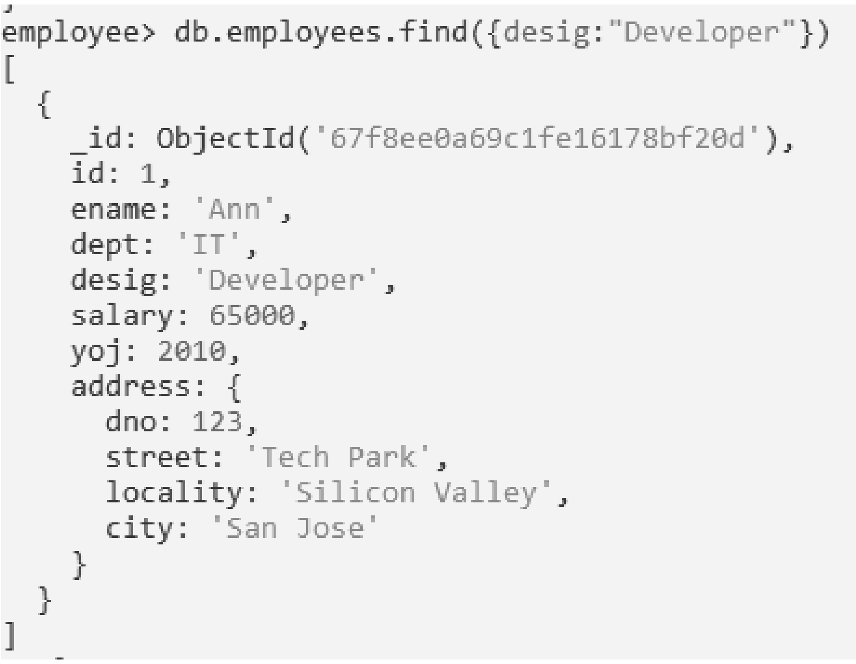


```
employee> db.employees.find({desig:"Developer"},{"_id":0,"address.city":1})
[ { address: { city: 'San Jose' } } ]
```

8. Update the salary of developers by ₹5000

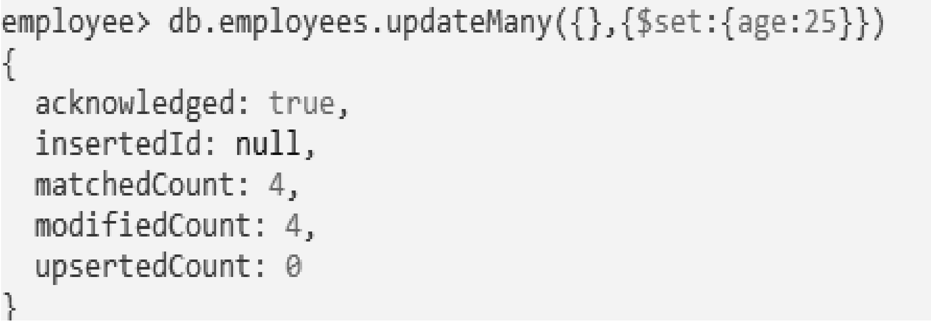


```
employee> db.employees.updateMany({desig:"Developer"},{$inc:{salary:5000}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

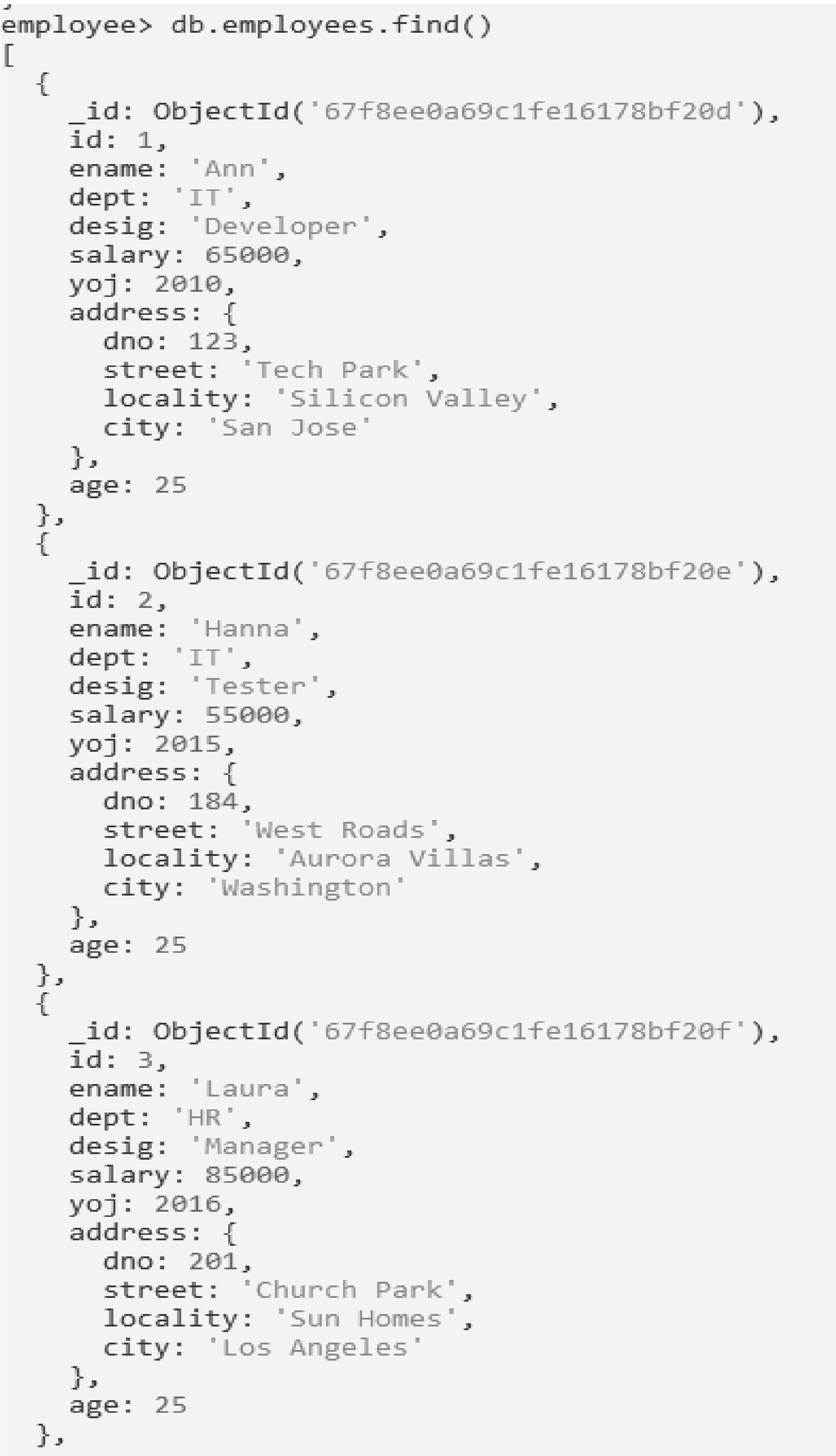


```
employee> db.employees.find({desig:"Developer"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    }
  }
]
```

9. Add age field to the collection ‘employees’

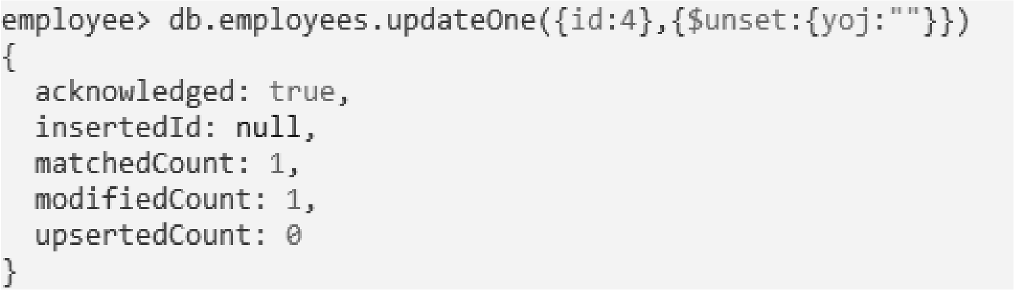


```
employee> db.employees.updateMany({},{$set:{age:25}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
```

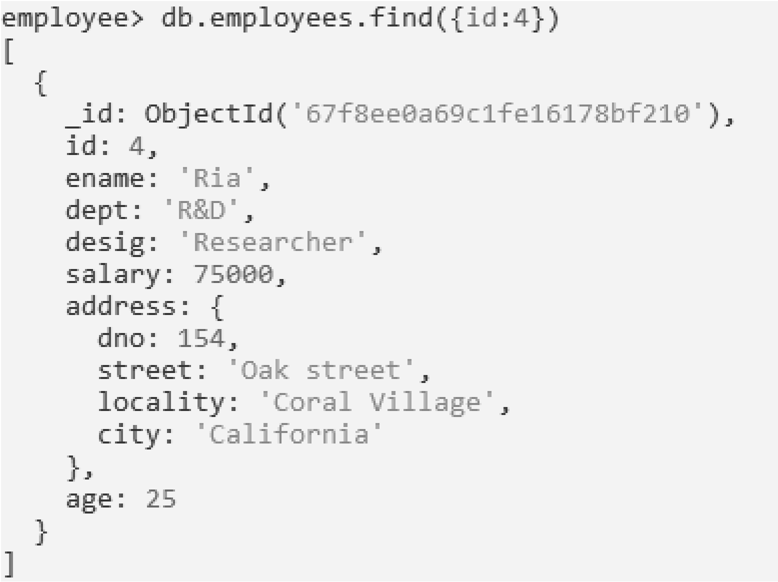


```
,
employee> db.employees.find()
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20f'),
    id: 3,
    ename: 'Laura',
    dept: 'HR',
    desig: 'Manager',
    salary: 85000,
    yoj: 2016,
    address: {
      dno: 201,
      street: 'Church Park',
      locality: 'Sun Homes',
      city: 'Los Angeles'
    },
    age: 25
  },
```

10. Delete a field from an employee

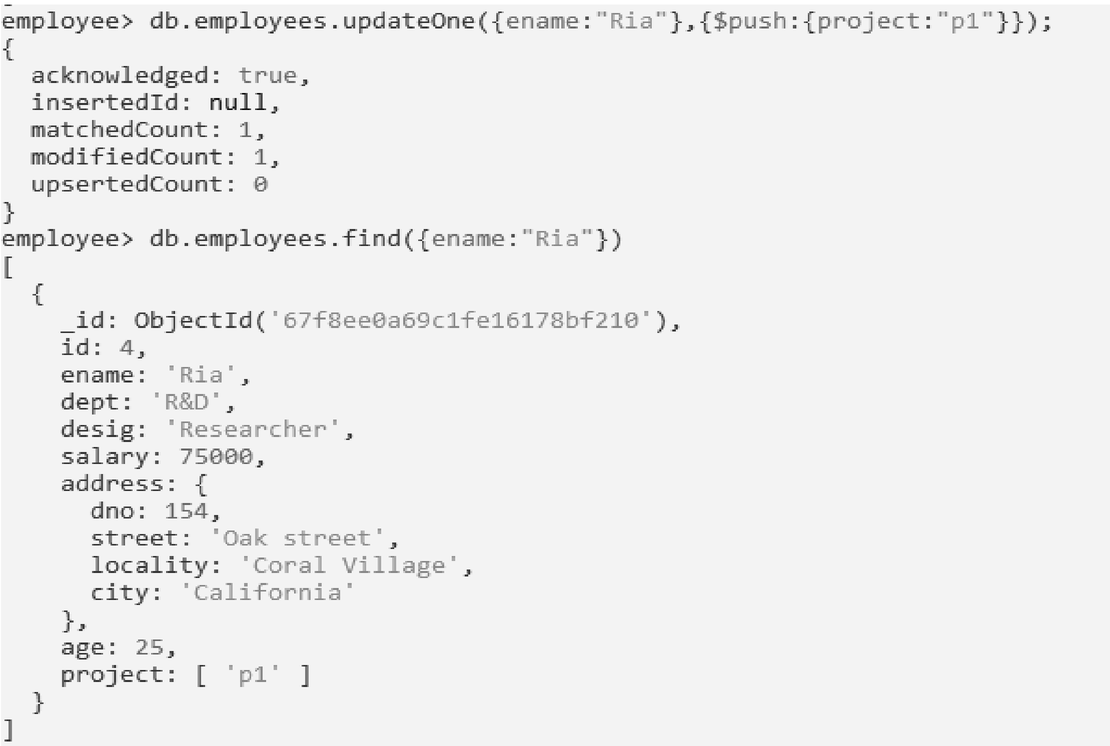


```
employee> db.employees.updateOne({id:4},{$unset:{yoj:""}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



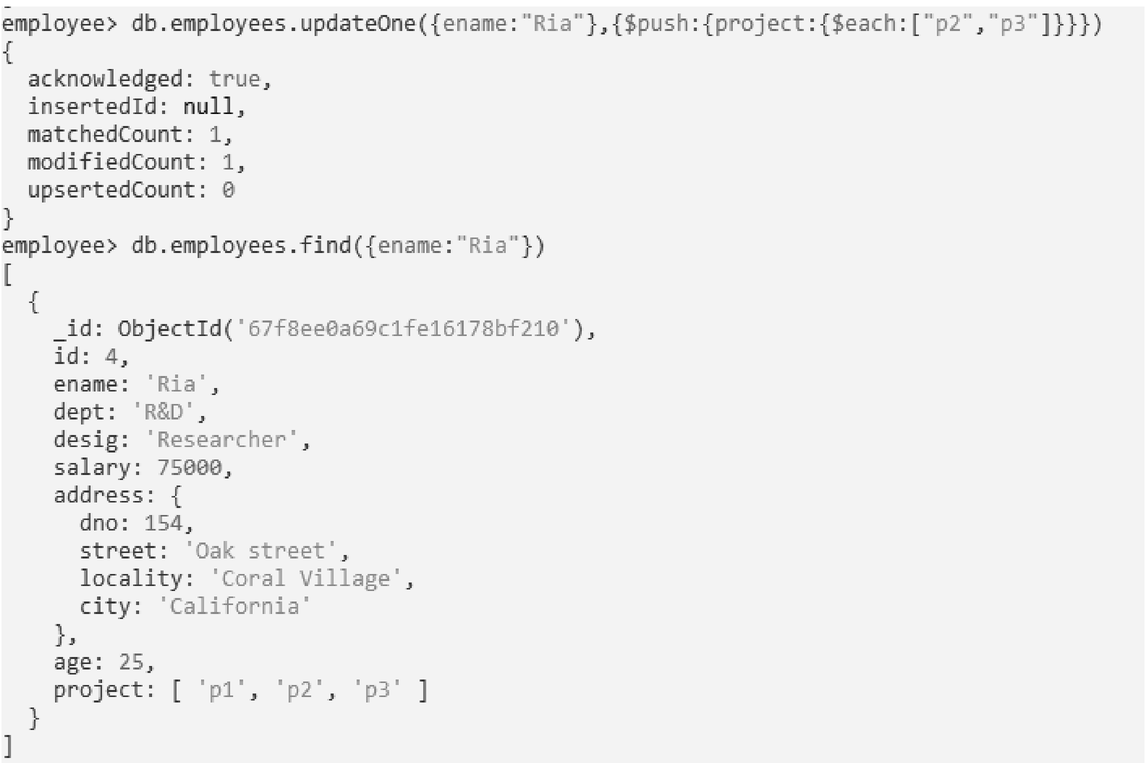
```
employee> db.employees.find({id:4})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25
  }
]
```

11. Add an array field ‘project’ to employee ‘Ria’



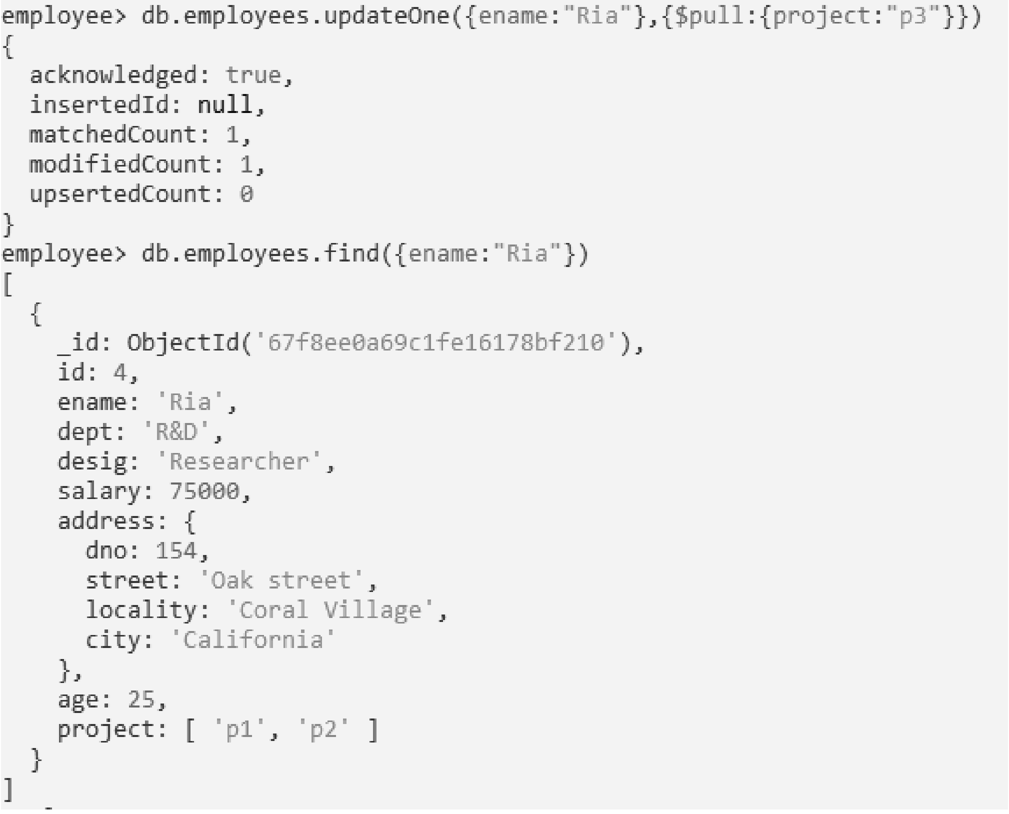
```
employee> db.employees.updateOne({ename:"Ria"},{$push:{project:"p1"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find({ename:"Ria"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25,
    project: [ 'p1' ]
  }
]
```

12. Add two more projects to employee ‘Ria’



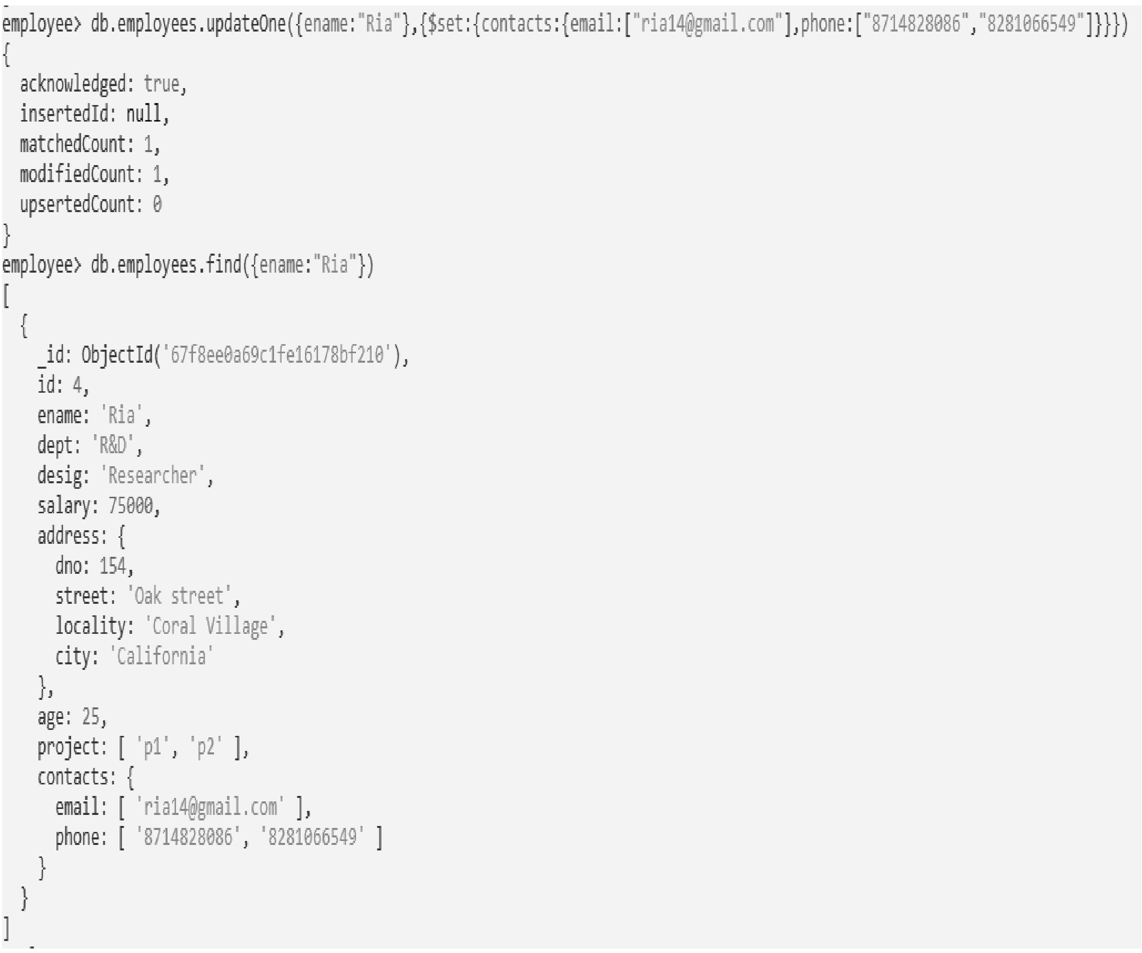
```
employee> db.employees.updateOne({ename:"Ria"},{$push:{project:{$each:["p2","p3"]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find({ename:"Ria"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25,
    project: [ 'p1', 'p2', 'p3' ]
  }
]
```

13. Remove project ‘p3’ from ‘Ria’



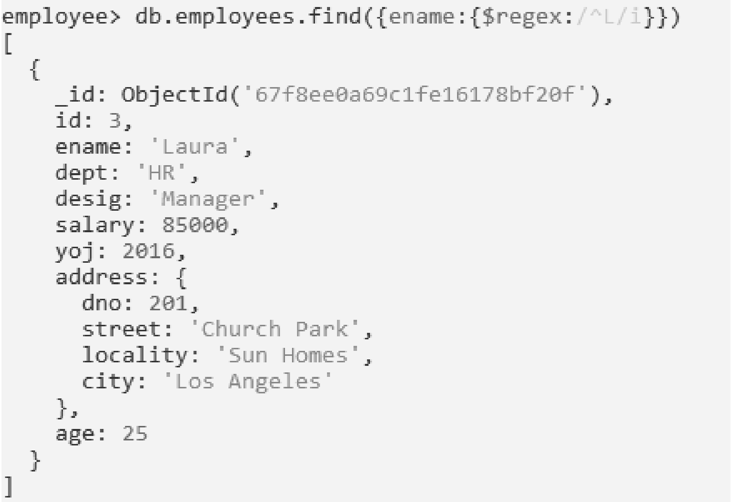
```
employee> db.employees.updateOne({ename:"Ria"},{$pull:{project:"p3"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find({ename:"Ria"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25,
    project: [ 'p1', 'p2' ]
  }
]
```

14. Add a new embedded object ‘contacts’ with email, phone as array objects to employee ‘Ria’



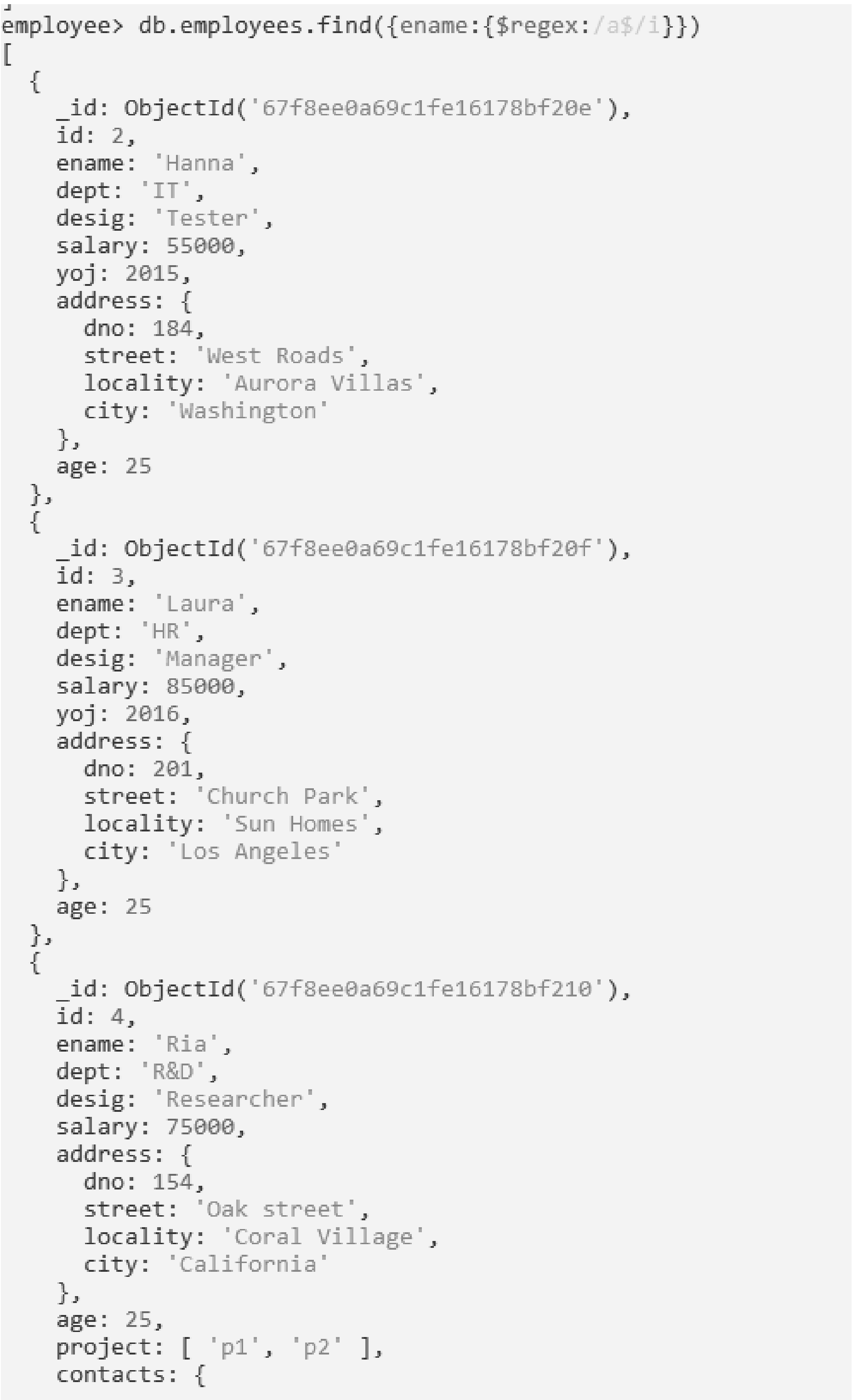
```
employee> db.employees.updateOne({ename:"Ria"},{$set:{contacts:{email:["ria14@gmail.com"],phone:["8714828086","8281066549"]}}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find({ename:"Ria"})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25,
    project: [ 'p1', 'p2' ],
    contacts: {
      email: [ 'ria14@gmail.com' ],
      phone: [ '8714828086', '8281066549' ]
    }
  }
]
```

15. Display details of employees who have their name starting with the letter ‘L’ using $regex operator



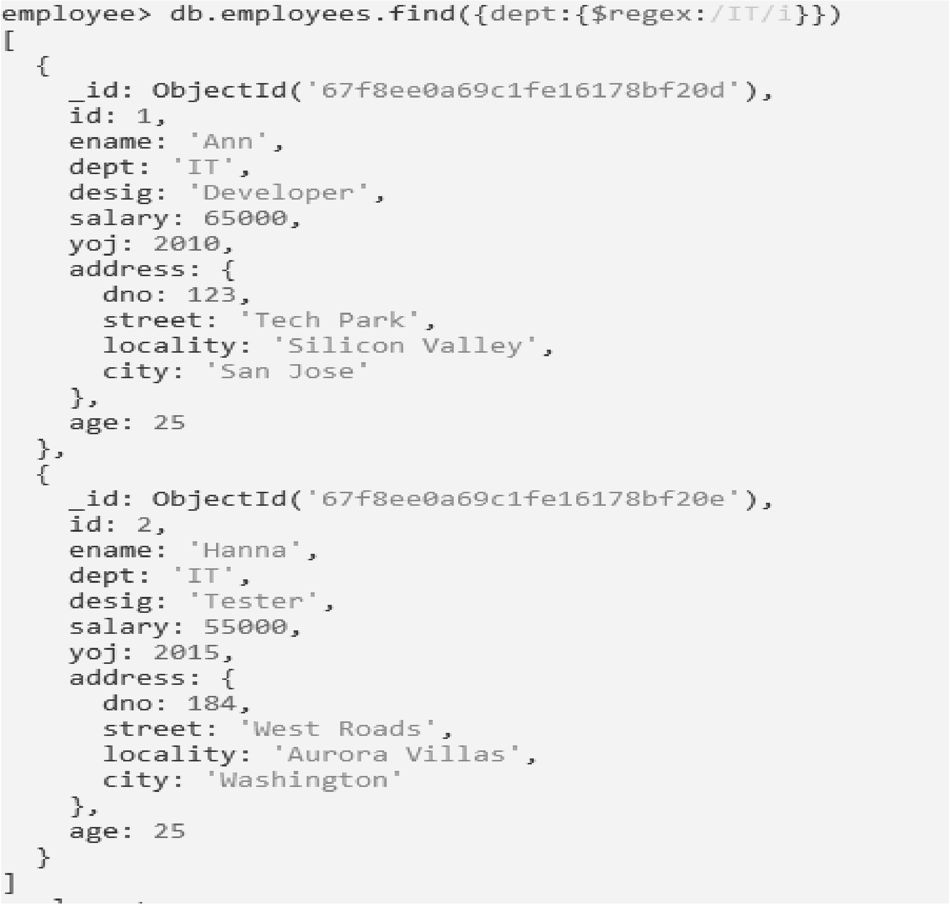
```
employee> db.employees.find({ename:{$regex:/^L/i}})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20f'),
    id: 3,
    ename: 'Laura',
    dept: 'HR',
    desig: 'Manager',
    salary: 85000,
    yoj: 2016,
    address: {
      dno: 201,
      street: 'Church Park',
      locality: 'Sun Homes',
      city: 'Los Angeles'
    },
    age: 25
  }
]
```

16. Display the details of employees with name ending with letter ‘a’ using $regex operator



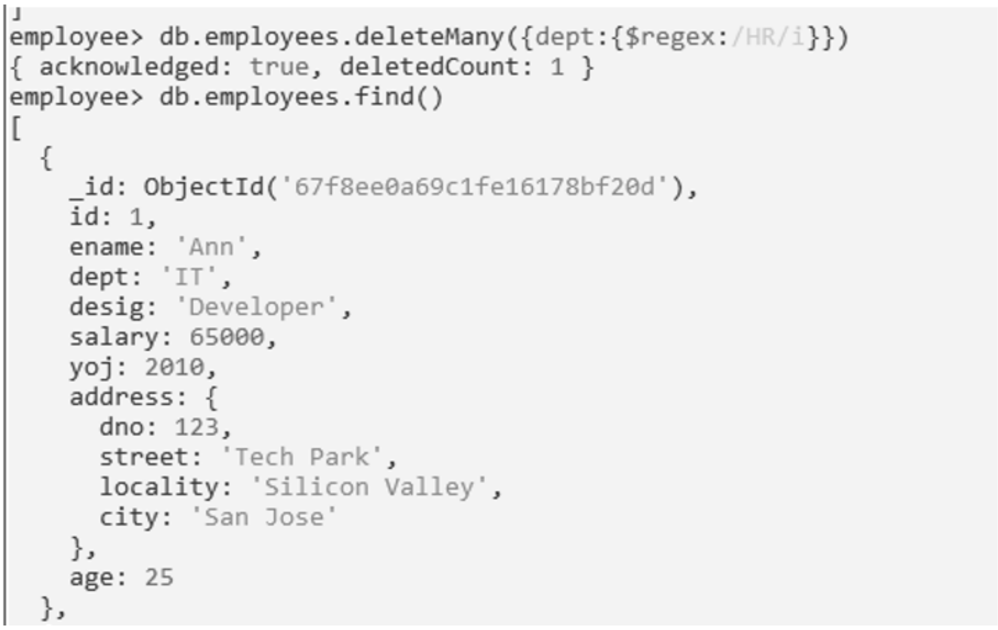
```
employee> db.employees.find({ename:{$regex:/a$/i}})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20f'),
    id: 3,
    ename: 'Laura',
    dept: 'HR',
    desig: 'Manager',
    salary: 85000,
    yoj: 2016,
    address: {
      dno: 201,
      street: 'Church Park',
      locality: 'Sun Homes',
      city: 'Los Angeles'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
    age: 25,
    project: [ 'p1', 'p2' ],
    contacts: {
```

17. Display details of employees who are having ‘IT’ as their department using $regex operator

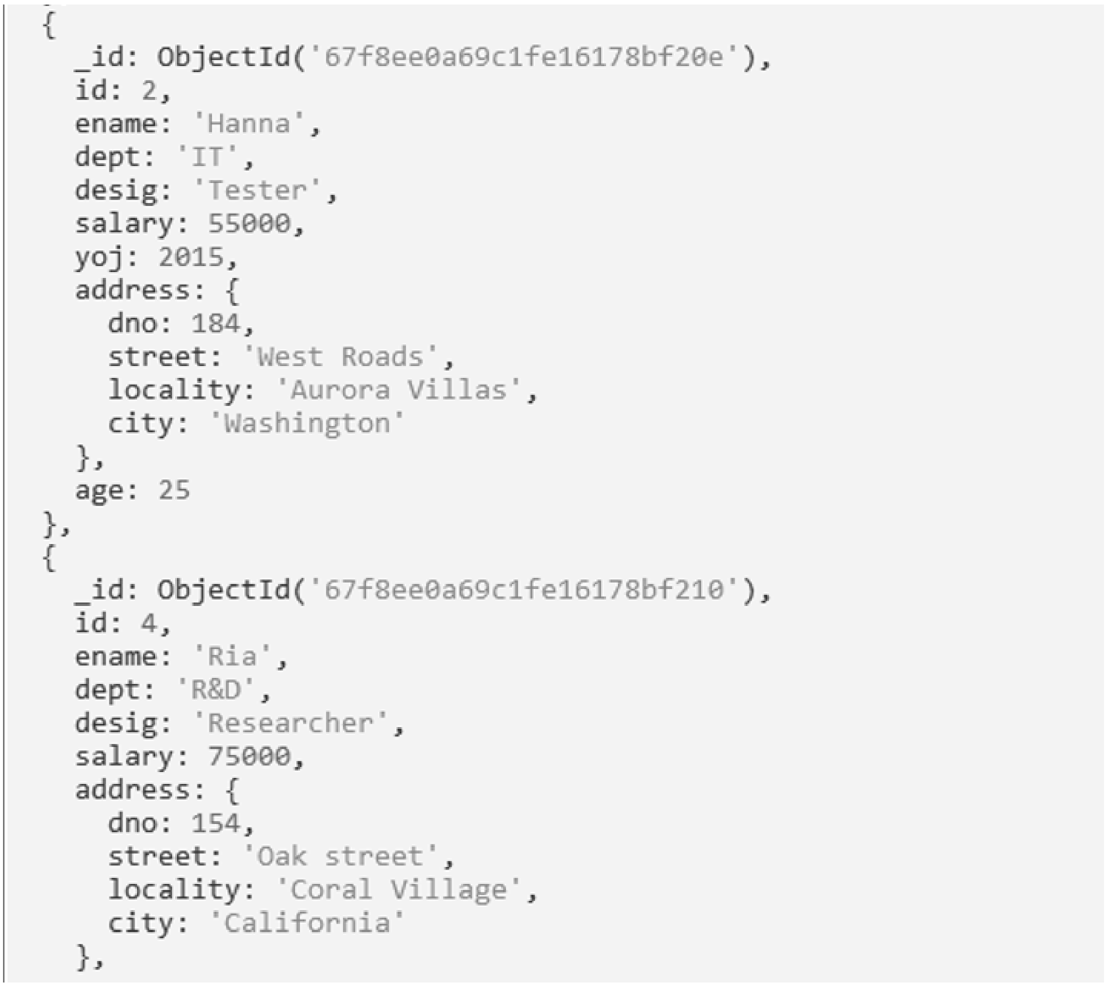


```
employee> db.employees.find({dept:{$regex:/IT/i}})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    },
    age: 25
  }
]
```

18. Remove details of student who are having ‘HR’ as their department using $regex operator

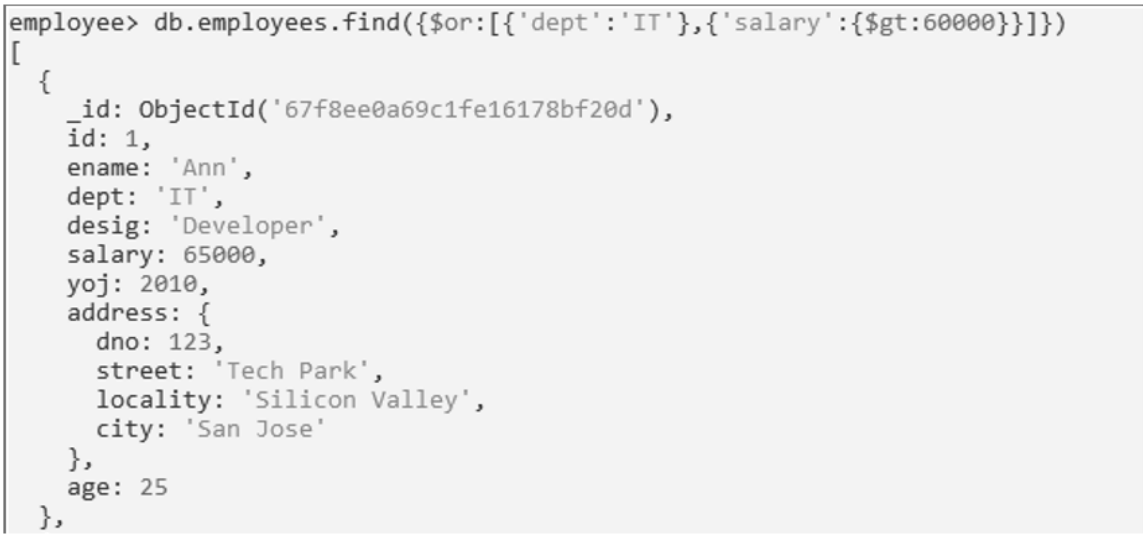


```
]
employee> db.employees.deleteMany({dept:{$regex:/HR/i}})
{ acknowledged: true, deletedCount: 1 }
employee> db.employees.find()
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 25
  },
```

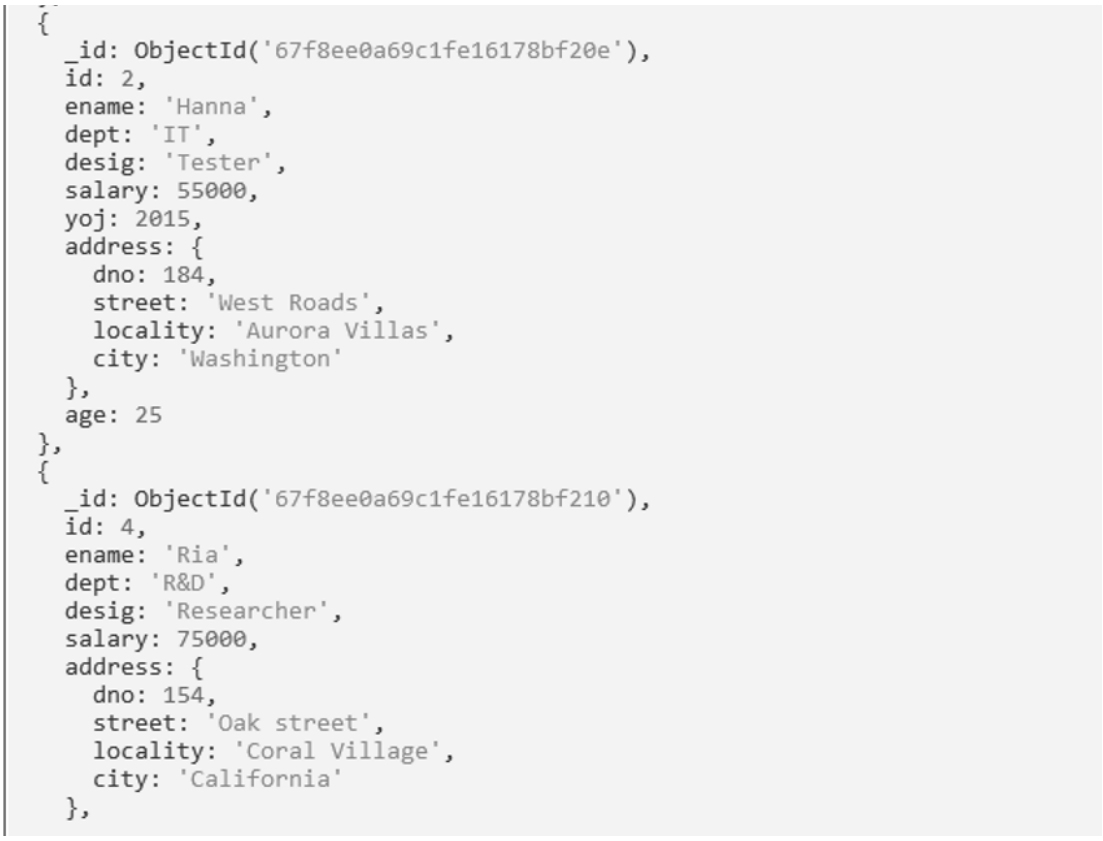


```
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
```

19. Display employees who have either department: IT or salary greater than ₹60000



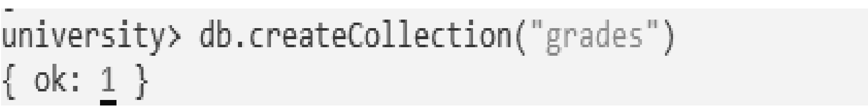
```
employee> db.employees.find({$or:[{'dept':'IT'},{'salary':{$gt:60000}}]})
[
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20d'),
    id: 1,
    ename: 'Ann',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 25
  },
```



```
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf20e'),
    id: 2,
    ename: 'Hanna',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2015,
    address: {
      dno: 184,
      street: 'West Roads',
      locality: 'Aurora Villas',
      city: 'Washington'
    },
    age: 25
  },
  {
    _id: ObjectId('67f8ee0a69c1fe16178bf210'),
    id: 4,
    ename: 'Ria',
    dept: 'R&D',
    desig: 'Researcher',
    salary: 75000,
    address: {
      dno: 154,
      street: 'Oak street',
      locality: 'Coral Village',
      city: 'California'
    },
```

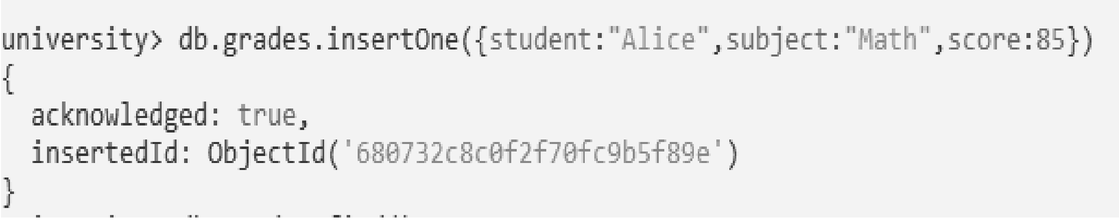
PROGRAM 3

1. Create a collection ‘grades’

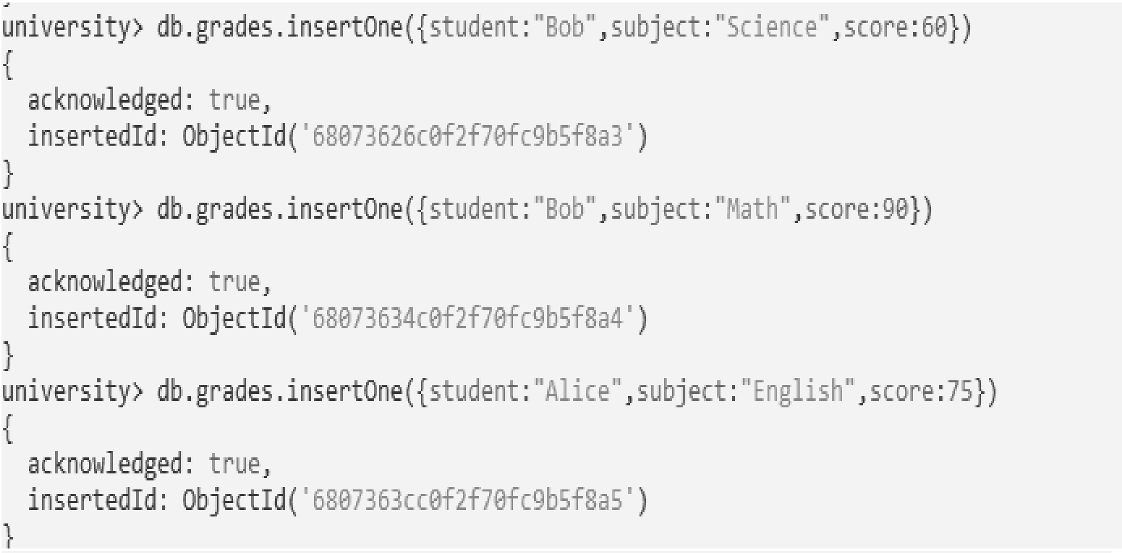


```
university> db.createCollection("grades")
{ ok: 1 }
```

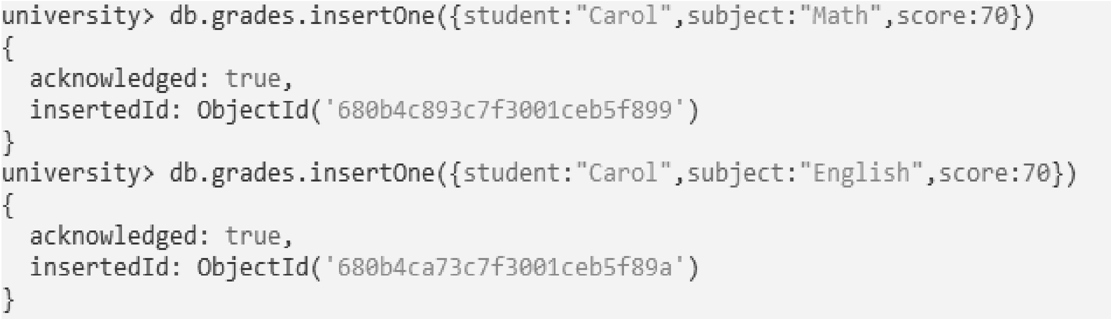
2. Insert values



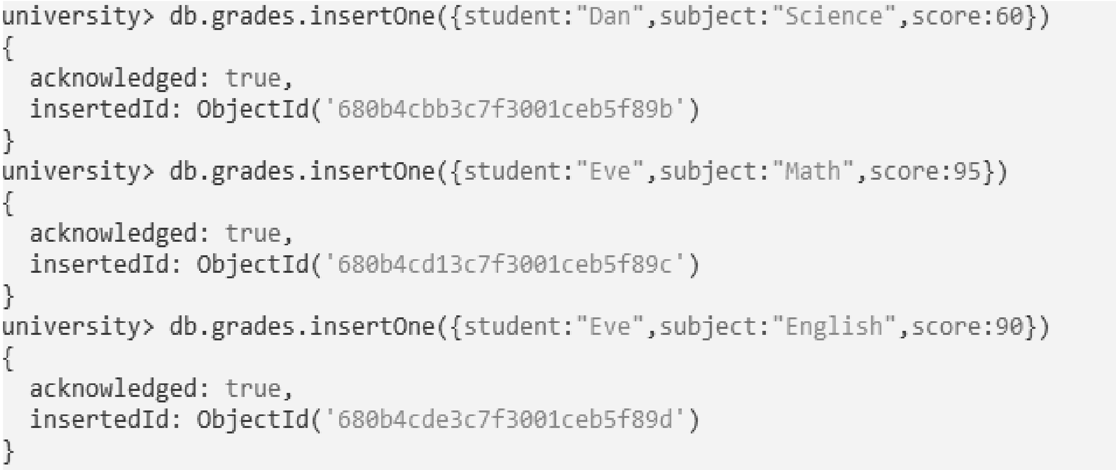
```
university> db.grades.insertOne({student:"Alice",subject:"Math",score:85})
{
  acknowledged: true,
  insertedId: ObjectId('680732c8c0f2f70fc9b5f89e')
}
```



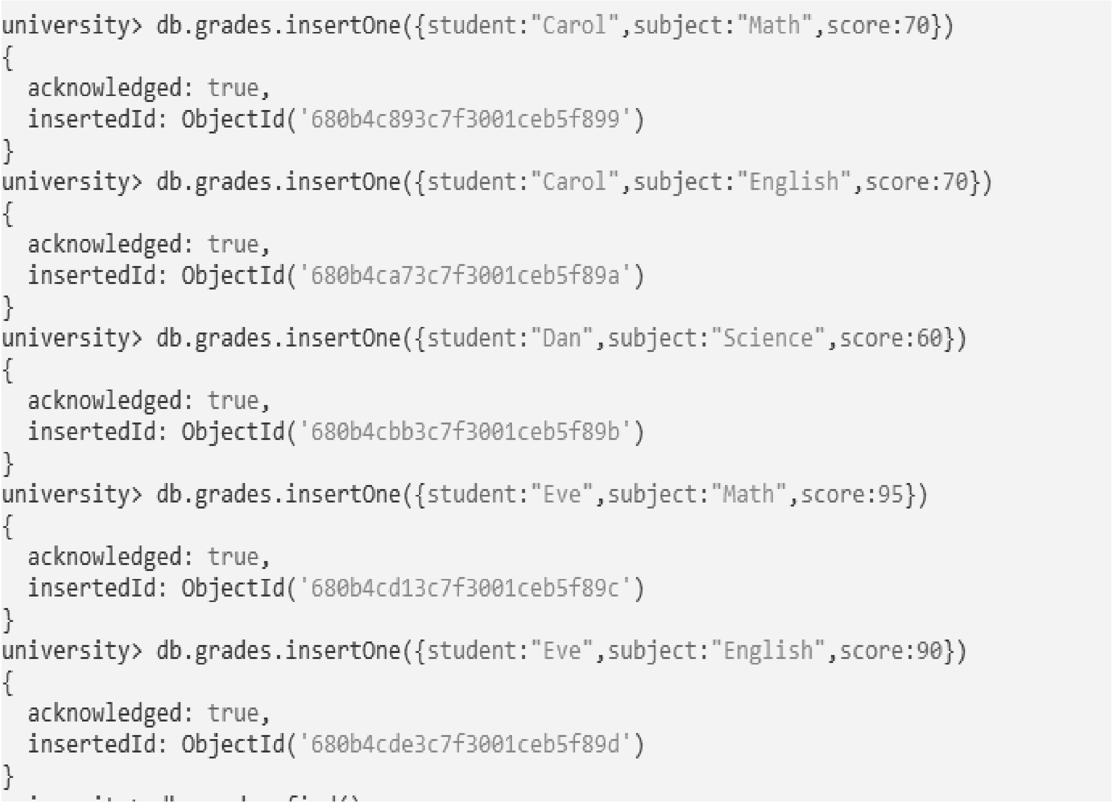
```
university> db.grades.insertOne({student:"Bob",subject:"Science",score:60})
{
  acknowledged: true,
  insertedId: ObjectId('68073626c0f2f70fc9b5f8a3')
}
university> db.grades.insertOne({student:"Bob",subject:"Math",score:90})
{
  acknowledged: true,
  insertedId: ObjectId('68073634c0f2f70fc9b5f8a4')
}
university> db.grades.insertOne({student:"Alice",subject:"English",score:75})
{
  acknowledged: true,
  insertedId: ObjectId('6807363cc0f2f70fc9b5f8a5')
}
```



```
university> db.grades.insertOne({student:"Carol",subject:"Math",score:70})
{
  acknowledged: true,
  insertedId: ObjectId('680b4c893c7f3001ceb5f899')
}
university> db.grades.insertOne({student:"Carol",subject:"English",score:70})
{
  acknowledged: true,
  insertedId: ObjectId('680b4ca73c7f3001ceb5f89a')
}
```

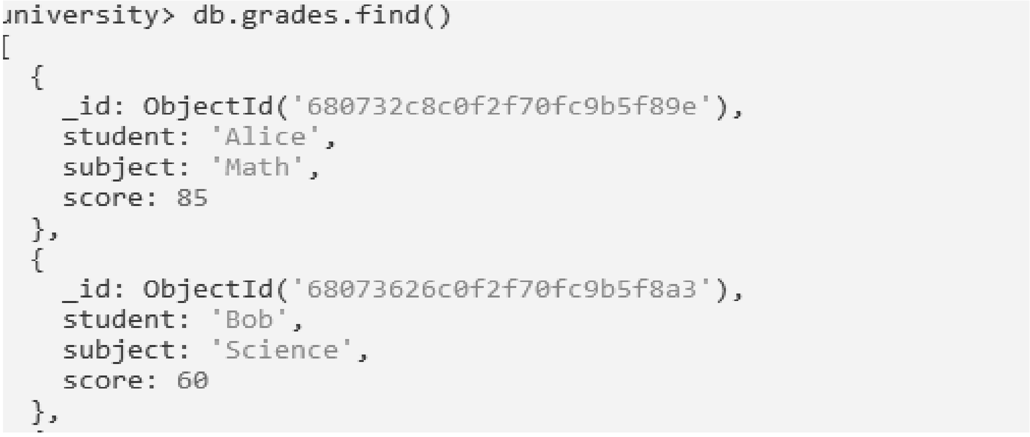


```
university> db.grades.insertOne({student:"Dan",subject:"Science",score:60})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cbb3c7f3001ceb5f89b')
}
university> db.grades.insertOne({student:"Eve",subject:"Math",score:95})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cd13c7f3001ceb5f89c')
}
university> db.grades.insertOne({student:"Eve",subject:"English",score:90})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cde3c7f3001ceb5f89d')
}
```

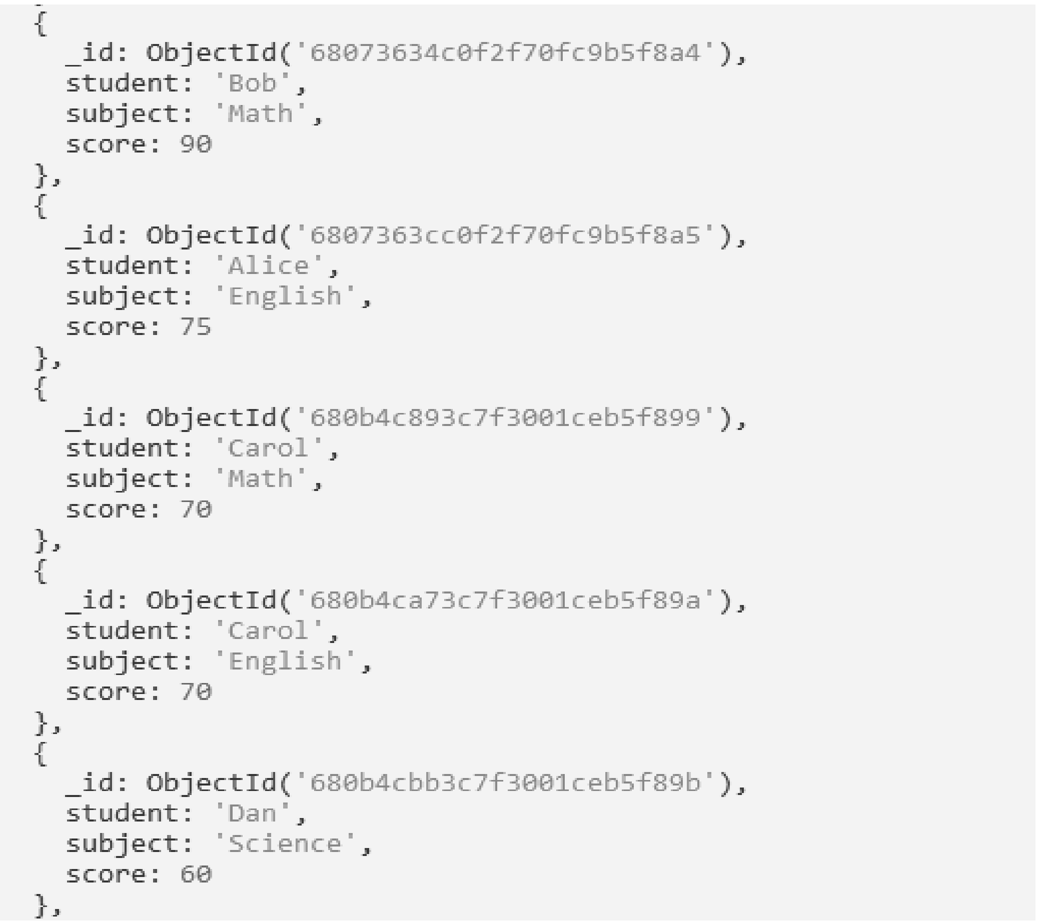


```
university> db.grades.insertOne({student:"Carol",subject:"Math",score:70})
{
  acknowledged: true,
  insertedId: ObjectId('680b4c893c7f3001ceb5f899')
}
university> db.grades.insertOne({student:"Carol",subject:"English",score:70})
{
  acknowledged: true,
  insertedId: ObjectId('680b4ca73c7f3001ceb5f89a')
}
university> db.grades.insertOne({student:"Dan",subject:"Science",score:60})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cbb3c7f3001ceb5f89b')
}
university> db.grades.insertOne({student:"Eve",subject:"Math",score:95})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cd13c7f3001ceb5f89c')
}
university> db.grades.insertOne({student:"Eve",subject:"English",score:90})
{
  acknowledged: true,
  insertedId: ObjectId('680b4cde3c7f3001ceb5f89d')
}
```

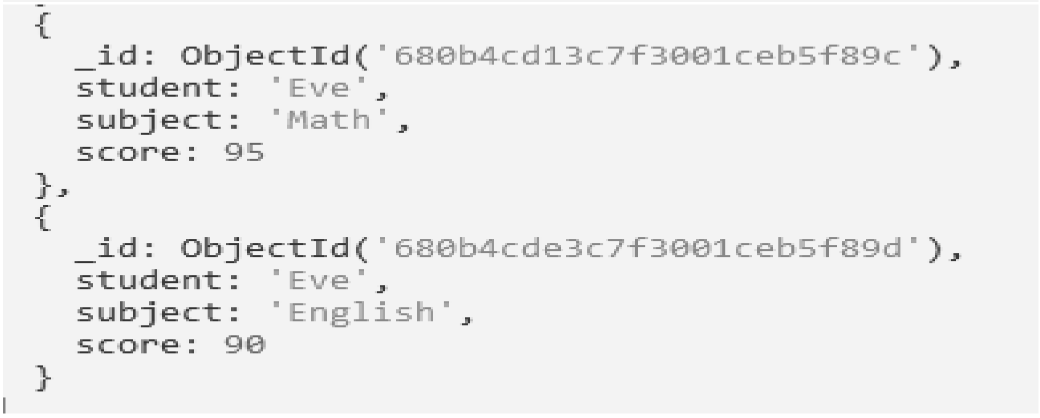
3. Display collection



```
university> db.grades.find()
[
  {
    _id: ObjectId('680732c8c0f2f70fc9b5f89e'),
    student: 'Alice',
    subject: 'Math',
    score: 85
  },
  {
    _id: ObjectId('68073626c0f2f70fc9b5f8a3'),
    student: 'Bob',
    subject: 'Science',
    score: 60
  },
```



```
  {
    _id: ObjectId('68073634c0f2f70fc9b5f8a4'),
    student: 'Bob',
    subject: 'Math',
    score: 90
  },
  {
    _id: ObjectId('6807363cc0f2f70fc9b5f8a5'),
    student: 'Alice',
    subject: 'English',
    score: 75
  },
  {
    _id: ObjectId('680b4c893c7f3001ceb5f899'),
    student: 'Carol',
    subject: 'Math',
    score: 70
  },
  {
    _id: ObjectId('680b4ca73c7f3001ceb5f89a'),
    student: 'Carol',
    subject: 'English',
    score: 70
  },
  {
    _id: ObjectId('680b4cbb3c7f3001ceb5f89b'),
    student: 'Dan',
    subject: 'Science',
    score: 60
  },
```

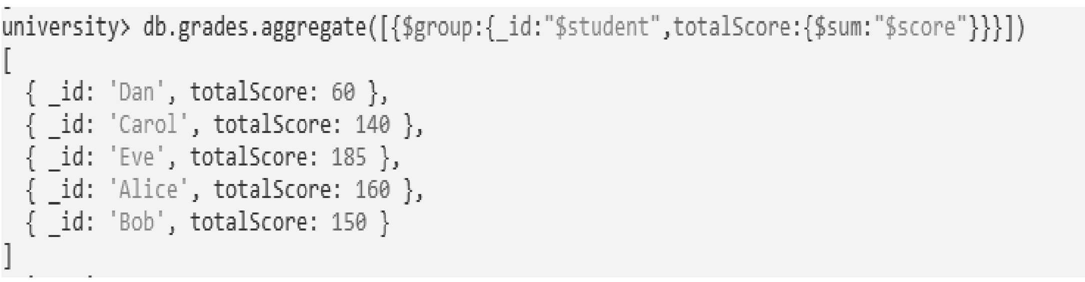


```
  {
    _id: ObjectId('680b4cd13c7f3001ceb5f89c'),
    student: 'Eve',
    subject: 'Math',
    score: 95
  },
  {
    _id: ObjectId('680b4cde3c7f3001ceb5f89d'),
    student: 'Eve',
    subject: 'English',
    score: 90
  }
```

4. Aggregate Pipeline:

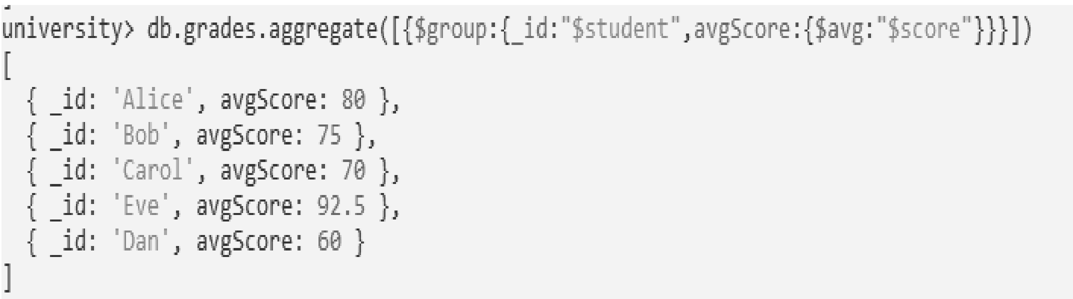
   a) With respect to student

   Sum



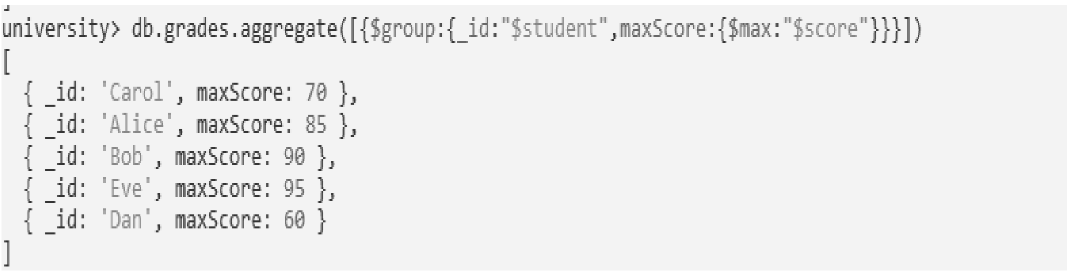
```
university> db.grades.aggregate([{$group:{_id:"$student",totalScore:{$sum:"$score"}}}])
[
  { _id: 'Dan', totalScore: 60 },
  { _id: 'Carol', totalScore: 140 },
  { _id: 'Eve', totalScore: 185 },
  { _id: 'Alice', totalScore: 160 },
  { _id: 'Bob', totalScore: 150 }
]
```

Average



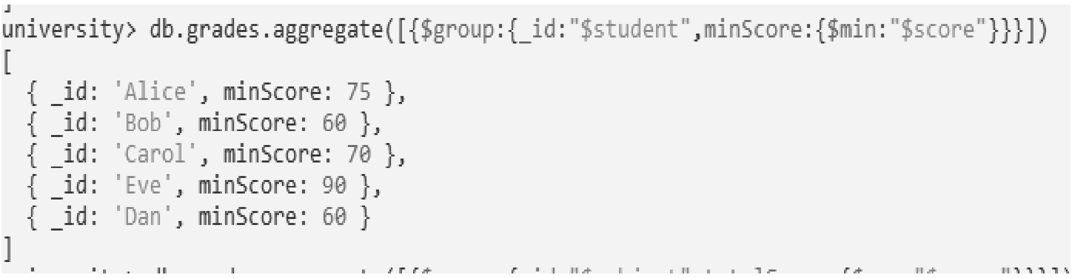
```
university> db.grades.aggregate([{$group:{_id:"$student",avgScore:{$avg:"$score"}}}])
[
  { _id: 'Alice', avgScore: 80 },
  { _id: 'Bob', avgScore: 75 },
  { _id: 'Carol', avgScore: 70 },
  { _id: 'Eve', avgScore: 92.5 },
  { _id: 'Dan', avgScore: 60 }
]
```

Maximum



```
university> db.grades.aggregate([{$group:{_id:"$student",maxScore:{$max:"$score"}}}])
[
  { _id: 'Carol', maxScore: 70 },
  { _id: 'Alice', maxScore: 85 },
  { _id: 'Bob', maxScore: 90 },
  { _id: 'Eve', maxScore: 95 },
  { _id: 'Dan', maxScore: 60 }
]
```

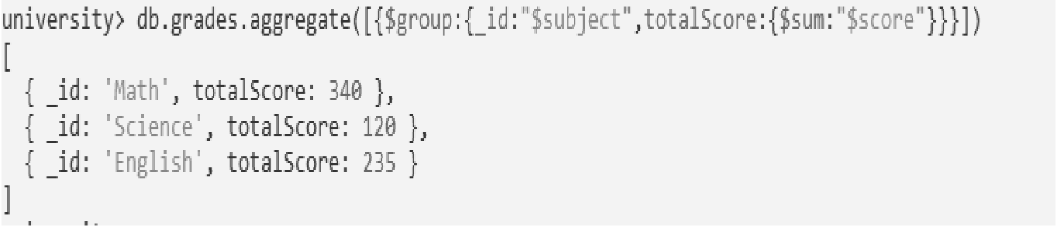
Minimum



```
university> db.grades.aggregate([{$group:{_id:"$student",minScore:{$min:"$score"}}}])
[
  { _id: 'Alice', minScore: 75 },
  { _id: 'Bob', minScore: 60 },
  { _id: 'Carol', minScore: 70 },
  { _id: 'Eve', minScore: 90 },
  { _id: 'Dan', minScore: 60 }
]
```

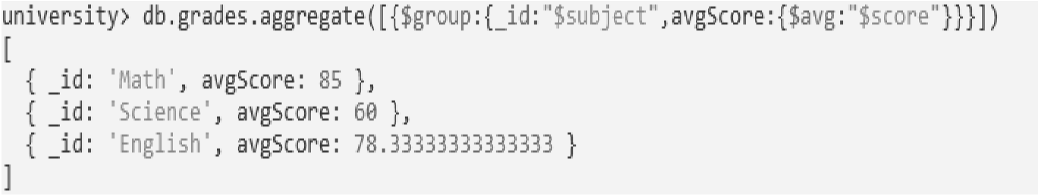
b) With respect to subject

Sum



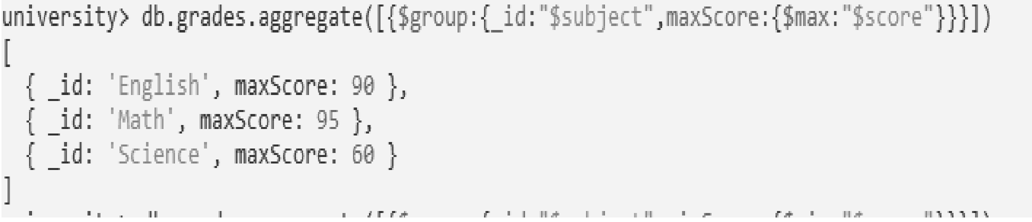
```
university> db.grades.aggregate([{$group:{_id:"$subject",totalScore:{$sum:"$score"}}}])
[
  { _id: 'Math', totalScore: 340 },
  { _id: 'Science', totalScore: 120 },
  { _id: 'English', totalScore: 235 }
]
```

Average



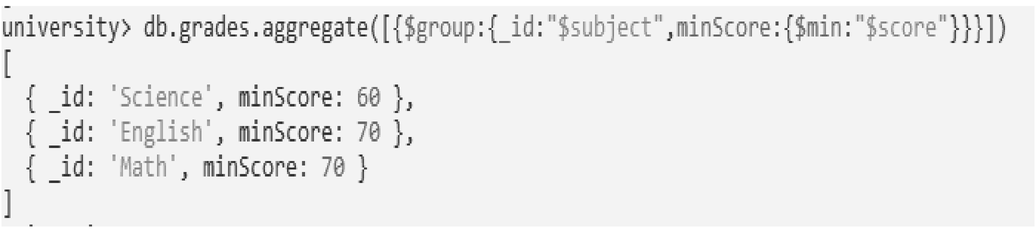
```
university> db.grades.aggregate([{$group:{_id:"$subject",avgScore:{$avg:"$score"}}}])
[
  { _id: 'Math', avgScore: 85 },
  { _id: 'Science', avgScore: 60 },
  { _id: 'English', avgScore: 78.33333333333333 }
]
```

Maximum



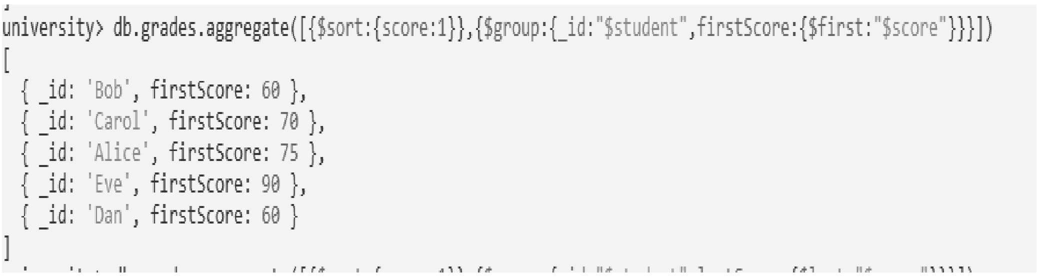
```
university> db.grades.aggregate([{$group:{_id:"$subject",maxScore:{$max:"$score"}}}])
[
  { _id: 'English', maxScore: 90 },
  { _id: 'Math', maxScore: 95 },
  { _id: 'Science', maxScore: 60 }
]
```

Minimum



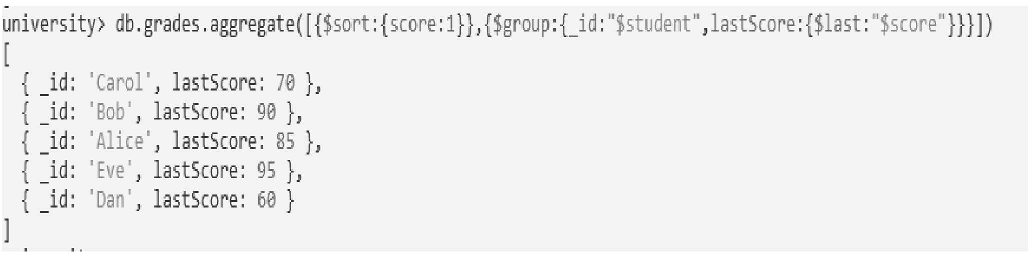
```
university> db.grades.aggregate([{$group:{_id:"$subject",minScore:{$min:"$score"}}}])
[
  { _id: 'Science', minScore: 60 },
  { _id: 'English', minScore: 70 },
  { _id: 'Math', minScore: 70 }
]
```

c) Sort the scores of each student and display the minimum score using ‘first’



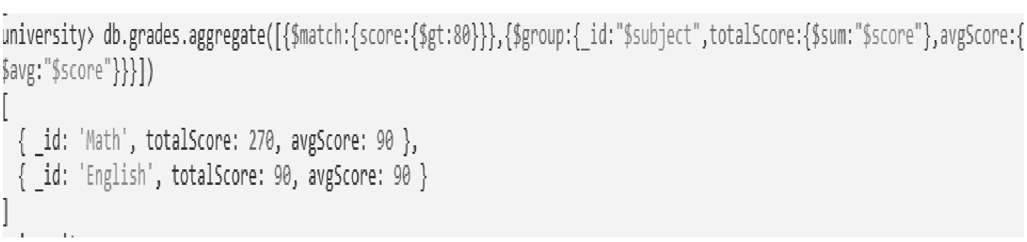
```
university> db.grades.aggregate([{$sort:{score:1}},{$group:{_id:"$student",firstScore:{$first:"$score"}}}])
[
  { _id: 'Bob', firstScore: 60 },
  { _id: 'Carol', firstScore: 70 },
  { _id: 'Alice', firstScore: 75 },
  { _id: 'Eve', firstScore: 90 },
  { _id: 'Dan', firstScore: 60 }
]
```

d) Sort the scores of each student and display the maximum score using ‘last’



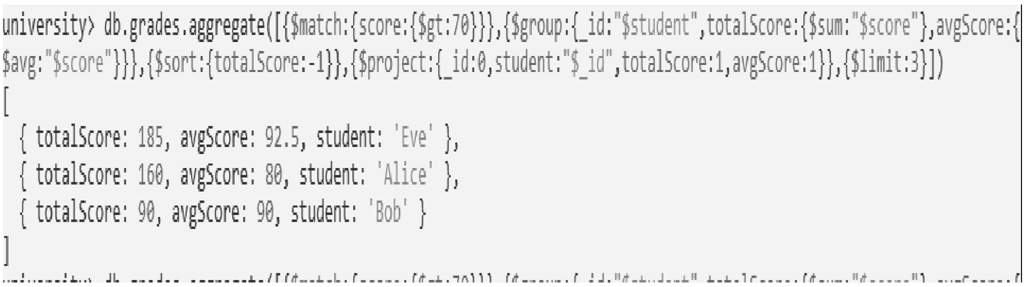
```
university> db.grades.aggregate([{$sort:{score:1}},{$group:{_id:"$student",lastScore:{$last:"$score"}}}])
[
  { _id: 'Carol', lastScore: 70 },
  { _id: 'Bob', lastScore: 90 },
  { _id: 'Alice', lastScore: 85 },
  { _id: 'Eve', lastScore: 95 },
  { _id: 'Dan', lastScore: 60 }
]
```

e) Find students who scored more than 80, then group the results by subject, and calculate: total score per subject, average score per subject



```
university> db.grades.aggregate([{$match:{score:{$gt:80}}},{$group:{_id:"$subject",totalScore:{$sum:"$score"},avgScore:{
$avg:"$score"}}}])
[
  { _id: 'Math', totalScore: 270, avgScore: 90 },
  { _id: 'English', totalScore: 90, avgScore: 90 }
]
```

f) Display the total score of three students in descending order of their total scores



```
university> db.grades.aggregate([{$match:{score:{$gt:70}}},{$group:{_id:"$student",totalScore:{$sum:"$score"},avgScore:{
$avg:"$score"}}},{$sort:{totalScore:-1}},{$project:{_id:0,student:"$_id",totalScore:1,avgScore:1}},{$limit:3}])
[
  { totalScore: 185, avgScore: 92.5, student: 'Eve' },
  { totalScore: 160, avgScore: 80, student: 'Alice' },
  { totalScore: 90, avgScore: 90, student: 'Bob' }
]
```