PROGRAM 14

AIM: To build a sample collections/documents to perform query operations. Create database college and collection students and insert student details into it.

```
test> use college;
switched to db college
college> db.createCollection("students");
{ ok: 1 }
college> db.students.insertMany([
{id: 1,name: "Meenuz",age: 21,gender: "Female",department: "Computer Science",gpa: 3.8},
{id: 2,name: "Anu",age: 28,gender: "Female",department: "Electrical   Engineering",gpa: 3.6},
{id: 3,name: "Binu",age: 19,gender: "Male",department: "Civil Engineering",gpa: 3.9},
{id: 4,name: "Pooja",age: 20,gender: "Female",department: "Computer Science",gpa: 3.7},
{id: 5,name: "Akhila",age: 23,gender: "Female",department: "Mechanical Engineering",gpa: 3.5}
]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660eccb0f96a2845bc9f990a'),
    '1': ObjectId('660eccb0f96a2845bc9f990b'),
    '2': ObjectId('660eccb0f96a2845bc9f990c'),
    '3': ObjectId('660eccb0f96a2845bc9f990d'),
    '4': ObjectId('660eccb0f96a2845bc9f990e')
  }
}
college> db.students.find()
[
  {
    _id: ObjectId('660eccb0f96a2845bc9f990a'),
    id: 1,
    name: 'Meenuz',
    age: 21,
    gender: 'Female',
    department: 'Computer Science',
    gpa: 3.8
  },
  {
    _id: ObjectId('660eccb0f96a2845bc9f990b'),
    id: 2,
    name: 'Anu',
    age: 28,
    gender: 'Female',
    department: 'Electrical Engineering',
    gpa: 3.6
  },
  {
    _id: ObjectId('660eccb0f96a2845bc9f990c'),
```

```
  id: 3,
  name: 'Binu',
  age: 19,
  gender: 'Male',
  department: 'Civil Engineering',
  gpa: 3.9
 },
 {
  _id: ObjectId('660eccb0f96a2845bc9f990d'),
  id: 4,
  name: 'Pooja',
  age: 20,
  gender: 'Female',
  department: 'Computer Science',
  gpa: 3.7
 },
 {
  _id: ObjectId('660eccb0f96a2845bc9f990e'),
  id: 5,
  name: 'Akhila',
  age: 23,
  gender: 'Female',
  department: 'Mechanical Engineering',
  gpa: 3.5
 }
]
```

PROGRAM 15

AIM: Create a Database ‘Student’ with the fields SRN, SName, degree, semester, CGPA and create a collection ‘Students’.

```
test>  use students
switched to db students
students>  db.createCollection("students")
{ ok: 1 }
students> db.students.insertOne({SRN:'101',SName:'Manu',Degree:'BCA',Sem:'6'
,CGPA:6.8})
{
  acknowledged: true,
  insertedId: ObjectId('661011372e8b33fef49f990a')
}
students> db.students.insertOne({SRN:'102',SName:'Binu',Degree:'BSc.Maths',Sem:'4',CGPA:6.2})
{
  acknowledged: true,
  insertedId: ObjectId('6610114e2e8b33fef49f990b')
}
students> db.students.insertOne({SRN:'103',SName:'Vinu',Degree:'BSc.CS',Sem:'6',CGPA:8.7})
{
  acknowledged: true,
  insertedId: ObjectId('661011632e8b33fef49f990c')
}
```

1. Display all the documents.

```
students> db.students.find().pretty()
[
  {
    _id: ObjectId('661011372e8b33fef49f990a'),
    SRN: '101',
    SName: 'Manu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.8
  },
  {
    _id: ObjectId('6610114e2e8b33fef49f990b'),
    SRN: '102',
    SName: 'Binu',
    Degree: 'BSc.Maths',
    Sem: '4',
    CGPA: 6.2
  },
  {
    _id: ObjectId('661011632e8b33fef49f990c'),
```

```
    SRN: '103',
    SName: 'Vinu',
    Degree: 'BSc.CS',
    Sem: '6',
    CGPA: 8.7
  },
  {
    _id: ObjectId('661011702e8b33fef49f990d'),
    SRN: '104',
    SName: 'Anu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 5.9
  },
  {
    _id: ObjectId('661012472e8b33fef49f990e'),
    SRN: '105',
    SName: 'Abhi',
    Degree: 'Bsc.Physics',
    Sem: '2',
    CGPA: 5
  },
  {
    _id: ObjectId('661012742e8b33fef49f990f'),
    SRN: '106',
    SName: 'Rahul',
    Degree: 'Bsc.Maths',
    Sem: '6',
    CGPA: 6.6
  },
  {
    _id: ObjectId('661012912e8b33fef49f9910'),
    SRN: '107',
    SName: 'Veenu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.9
  }
]
```

2. Display all the students in BCA.

```
students> db.students.find({Degree:'BCA'})
[
  {
    _id: ObjectId('661011372e8b33fef49f990a'),
    SRN: '101',
    SName: 'Manu',
    Degree: 'BCA',
```

```
   Sem: '6',
    CGPA: 6.8
  },
  {
    _id: ObjectId('661011702e8b33fef49f990d'),
    SRN: '104',
    SName: 'Anu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 5.9
  },
  {
    _id: ObjectId('661012912e8b33fef49f9910'),
    SRN: '107',
    SName: 'Veenu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.9
  }
]
```

3. Display all the students in ascending order

```
students> db.students.find({},{SName:1,_id:0}).sort({SName:1})
[
  { SName: 'Abhi' },
  { SName: 'Anu' },
  { SName: 'Binu' },
  { SName: 'Manu' },
  { SName: 'Rahul' },
  { SName: 'Veenu' },
  { SName: 'Vinu' }
]
```

4. Display all the first five students.

```
students> db.students.find().limit(5)
[
  {
    _id: ObjectId('661011372e8b33fef49f990a'),
    SRN: '101',
    SName: 'Manu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.8
  },
  {
    _id: ObjectId('6610114e2e8b33fef49f990b'),
    SRN: '102',
```

```
  SName: 'Binu',
    Degree: 'BSc.Maths',
    Sem: '4',
    CGPA: 6.2
  },
  {
    _id: ObjectId('661011632e8b33fef49f990c'),
    SRN: '103',
    SName: 'Vinu',
    Degree: 'BSc.CS',
    Sem: '6',
    CGPA: 8.7
  },
  {
    _id: ObjectId('661011702e8b33fef49f990d'),
    SRN: '104',
    SName: 'Anu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 5.9
  },
  {
    _id: ObjectId('661012472e8b33fef49f990e'),
    SRN: '105',
    SName: 'Abhi',
    Degree: 'Bsc.Physics',
    Sem: '2',
    CGPA: 5
  }
]
```

5. Display students 5,6,7

```
students> db.students.find().skip(4).limit(3)
[
  {
    _id: ObjectId('661012472e8b33fef49f990e'),
    SRN: '105',
    SName: 'Abhi',
    Degree: 'Bsc.Physics',
    Sem: '2',
    CGPA: 5
  },
  {
    _id: ObjectId('661012742e8b33fef49f990f'),
    SRN: '106',
    SName: 'Rahul',
    Degree: 'Bsc.Maths',
    Sem: '6',
```

```
  CGPA: 6.6
  },
  {
   _id: ObjectId('661012912e8b33fef49f9910'),
   SRN: '107',
   SName: 'Veenu',
   Degree: 'BCA',
   Sem: '6',
   CGPA: 6.9
  }
]
```

6. Display the degree of student ‘Anu’.

```
students> db.students.find({SName:'Anu'},{Degree:1})
[ { _id: ObjectId('661011702e8b33fef49f990d'), Degree: 'BCA' } ]
```

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

```
students>  db.students.find().skip(4).limit(3).sort({CGPA:-1})
[
  {
   _id: ObjectId('6610114e2e8b33fef49f990b'),
   SRN: '102',
   SName: 'Binu',
   Degree: 'BSc.Maths',
   Sem: '4',
   CGPA: 6.2
  },
  {
   _id: ObjectId('661011702e8b33fef49f990d'),
   SRN: '104',
   SName: 'Anu',
   Degree: 'BCA',
   Sem: '6',
   CGPA: 5.9
  },
  {
   _id: ObjectId('661012472e8b33fef49f990e'),
   SRN: '105',
   SName: 'Abhi',
   Degree: 'Bsc.Physics',
   Sem: '2',
   CGPA: 5
  }
]
```

8. Display the number of students in BCA

```
students> db.students.count({Degree:'BCA'})
3
```

9.Display all the degrees without "_id"

```
students> db.students.find({},{_id:0})
[
  { SRN: '101', SName: 'Manu', Degree: 'BCA', Sem: '6', CGPA: 6.8 },
  { SRN: '102',SName: 'Binu', Degree: 'BSc.Maths',Sem: '4',CGPA: 6.2 },
   { SRN: '103', SName: 'Vinu', Degree: 'BSc.CS', Sem: '6', CGPA: 8.7 },
  { SRN: '104', SName: 'Anu', Degree: 'BCA', Sem: '6', CGPA: 5.9 },
  { SRN: '105',SName: 'Abhi', Degree: 'Bsc.Physics', Sem: '2',CGPA: 5},
  { SRN: '106', SName: 'Rahul',Degree: 'Bsc.Maths',Sem: '6',CGPA: 6.6},
   { SRN: '107', SName: 'Veenu', Degree: 'BCA', Sem: '6', CGPA: 6.9 }]
```

10. Display the distinct degrees.

```
students>  db.students.distinct("Degree")
[ 'BCA', 'BSc.CS', 'BSc.Maths', 'Bsc.Maths', 'Bsc.Physics' ]
```

12. Display all the BCA students and in 2nd sem.

```
students>  db.students.find({Degree:'BCA',Sem:'6'})
[
  {
    _id: ObjectId('661011372e8b33fef49f990a'),
    SRN: '101',
    SName: 'Manu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.8
  },
  {
    _id: ObjectId('661011702e8b33fef49f990d'),
    SRN: '104',
    SName: 'Anu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 5.9
  },
  {
    _id: ObjectId('661012912e8b33fef49f9910'),
    SRN: '107',
    SName: 'Veenu',
    Degree: 'BCA',
    Sem: '6',
    CGPA: 6.9  }]
```

PROGRAM 16

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

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

```
test> use employee;
switched to db employee
employee> db.createCollection("employees");
{ ok: 1 }
employee> db.employees.insertMany([
{id: 1,ename: "Rahul",dept: "IT",desig: "Developer",salary: 60000,yoj: 2010,address: {dno:
123,street: "Tech Park",locality: "Silicon Valley",city: "San Jose"}},
{id: 2,ename: "Anu",dept: "HR",desig: "Manager",salary: 80000,yoj: 2005,address: {dno:
456,street: "Corporate Park",locality: "Do
wntown",city: "San Francisco"}},
{id: 3,ename: "Binu",dept: "IT",desig: "Tester",salary: 55000,yoj: 2012,address: {dno: 789,street:
"Tech Plaza",locality: "Silicon Oasis",city: "San Jose"}}
]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660ee074ef0f76ffb19f990a'),
    '1': ObjectId('660ee074ef0f76ffb19f990b'),
    '2': ObjectId('660ee074ef0f76ffb19f990c')
  }
}
    employee> db.employees.find({salary:{$gte:50000,$lt:75000}});
    [
      {
        _id: ObjectId('660ee074ef0f76ffb19f990a'),
        id: 1,
        ename: 'Rahul',
        dept: 'IT',
        desig: 'Developer',
        salary: 60000,
        yoj: 2010,
        address: {
          dno: 123,
          street: 'Tech Park',
          locality: 'Silicon Valley',
          city: 'San Jose'
        }
      },
      {
        _id: ObjectId('660ee074ef0f76ffb19f990c'),
```

```
 id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
      dno: 789,
      street: 'Tech Plaza',
      locality: 'Silicon Oasis',
      city: 'San Jose'
    }
  }
]
```

2. Display all the employees with design developer

```
employee> db.employees.find({desig:"Developer"});
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 60000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    }
  }
]
```

3. Display the salary of Rahul

```
employee> db.employees.findOne({ename:"Rahul"},{salary:1});
{ _id: ObjectId('660ee074ef0f76ffb19f990a'), salary: 60000 }
```

4. Display the city of employee.

```
employee> db.employees.find({},{"address.city":1});
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    address: { city: 'San Jose' }
  },
```

```
  {
   _id: ObjectId('660ee074ef0f76ffb19f990b'),
   address: { city: 'San Francisco' }
  },
  {
   _id: ObjectId('660ee074ef0f76ffb19f990c'),
   address: { city: 'San Jose' }
  },
]
```

5. Update the salary of developers by 5000.

```
employee> db.employees.updateMany({desig:"Developer"},{$inc:{salary:5000}});
{
 acknowledged: true,
 insertedId: null,
 matchedCount: 2,
 modifiedCount: 2,
 upsertedCount: 0
}
employee> db.employees.find()
[
 {
  _id: ObjectId('660ee074ef0f76ffb19f990a'),
  id: 1,
  ename: 'Rahul',
  dept: 'IT',
  desig: 'Developer',
  salary: 65000,
  yoj: 2010,
  address: {
   dno: 123,
   street: 'Tech Park',
   locality: 'Silicon Valley',
   city: 'San Jose'
  }
 },
 {
  _id: ObjectId('660ee074ef0f76ffb19f990b'),
  id: 2,
  ename: 'Anu',
  dept: 'HR',
  desig: 'Manager',
  salary: 80000,
  yoj: 2005,
  address: {
   dno: 456,
   street: 'Corporate Park',
   locality: 'Downtown',
```

```
    city: 'San Francisco'
    }
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
      dno: 789,
      street: 'Tech Plaza',
      locality: 'Silicon Oasis',
      city: 'San Jose'
    }
  }
]
```

6. Add field age to employee.

```
employee> db.employees.updateMany({},{$set:{age:30}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 4,
  modifiedCount: 4,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    yoj: 2010,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 30
  },
  {
```

```
    _id: ObjectId('660ee074ef0f76ffb19f990b'),
    id: 2,
    ename: 'Anu',
    dept: 'HR',
    desig: 'Manager',
    salary: 80000,
    yoj: 2005,
    address: {
      dno: 456,
      street: 'Corporate Park',
      locality: 'Downtown',
      city: 'San Francisco'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
      dno: 789,
      street: 'Tech Plaza',
      locality: 'Silicon Oasis',
      city: 'San Jose'
    },
    age: 30
  }
]
```

7. Remove yoj from Rahul

```
employee> db.employees.updateOne({ename:"Rahul"},{$unset:{yoj:""}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
```

```
  dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990b'),
    id: 2,
    ename: 'Anu',
    dept: 'HR',
    desig: 'Manager',
    salary: 80000,
    yoj: 2005,
    address: {
      dno: 456,
      street: 'Corporate Park',
      locality: 'Downtown',
      city: 'San Francisco'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
      dno: 789,
      street: 'Tech Plaza',
      locality: 'Silicon Oasis',
      city: 'San Jose'
    },
    age: 30
  }
]
```

8. Add an array field project to Rahul.

```
employee> db.employees.updateOne({ename:"Rahul"},{$push:{projects:"p1"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 30,
    projects: [ 'p1' ]
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990b'),
    id: 2,
    ename: 'Anu',
    dept: 'HR',
    desig: 'Manager',
    salary: 80000,
    yoj: 2005,
    address: {
      dno: 456,
      street: 'Corporate Park',
      locality: 'Downtown',
      city: 'San Francisco'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
```

```
  desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
      dno: 789,
      street: 'Tech Plaza',
      locality: 'Silicon Oasis',
      city: 'San Jose'
    },
    age: 30
  }
]
```

9. Add p2 and p3 project to Rahul

```
employee> db.employees.updateOne({ename:"Rahul"},{$push:{projects:{$each:["p2","p3"]}}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 30,
    projects: [ 'p1', 'p2', 'p3' ]
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990b'),
    id: 2,
    ename: 'Anu',
    dept: 'HR',
    desig: 'Manager',
    salary: 80000,
    yoj: 2005,
```

```
    address: {
     dno: 456,
     street: 'Corporate Park',
     locality: 'Downtown',
     city: 'San Francisco'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
    address: {
     dno: 789,
     street: 'Tech Plaza',
     locality: 'Silicon Oasis',
     city: 'San Jose'
    },
    age: 30
  }
]
```

10. Remove p3 from Rahul.

```
employee> db.employees.updateOne({ename:"Rahul"},{$pull:{projects:"p3"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
```

```
    city: 'San Jose'
   },
   age: 30,
   projects: [ 'p1', 'p2' ]
  },
  {
   _id: ObjectId('660ee074ef0f76ffb19f990b'),
   id: 2,
   ename: 'Anu',
   dept: 'HR',
   desig: 'Manager',
   salary: 80000,
   yoj: 2005,
   address: {
    dno: 456,
    street: 'Corporate Park',
    locality: 'Downtown',
    city: 'San Francisco'
   },
   age: 30
  },
  {
   _id: ObjectId('660ee074ef0f76ffb19f990c'),
   id: 3,
   ename: 'Binu',
   dept: 'IT',
   desig: 'Tester',
   salary: 55000,
   yoj: 2012,
   address: {
    dno: 789,
    street: 'Tech Plaza',
    locality: 'Silicon Oasis',
    city: 'San Jose'
   },
   age: 30
  }
]
```

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

```
employee>
db.employees.updateOne({ename:"Rahul"},{$set:{contacts:{email:["rahul@gmail.com"],phon
e:["04829-262234","04829-225678"]}}});
{
 acknowledged: true,
 insertedId: null,
 matchedCount: 1,
```

```
 modifiedCount: 1,
  upsertedCount: 0
}
employee> db.employees.find()
[
  {
    _id: ObjectId('660ee074ef0f76ffb19f990a'),
    id: 1,
    ename: 'Rahul',
    dept: 'IT',
    desig: 'Developer',
    salary: 65000,
    address: {
      dno: 123,
      street: 'Tech Park',
      locality: 'Silicon Valley',
      city: 'San Jose'
    },
    age: 30,
    projects: [ 'p1', 'p2' ],
    contacts: {
      email: [ 'rahul@gmail.com' ],
      phone: [ '04829-262234', '04829-225678' ]
    }
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990b'),
    id: 2,
    ename: 'Anu',
    dept: 'HR',
    desig: 'Manager',
    salary: 80000,
    yoj: 2005,
    address: {
      dno: 456,
      street: 'Corporate Park',
      locality: 'Downtown',
      city: 'San Francisco'
    },
    age: 30
  },
  {
    _id: ObjectId('660ee074ef0f76ffb19f990c'),
    id: 3,
    ename: 'Binu',
    dept: 'IT',
    desig: 'Tester',
    salary: 55000,
    yoj: 2012,
```

```
  address: {
     dno: 789,
     street: 'Tech Plaza',
     locality: 'Silicon Oasis',
     city: 'San Jose'
    },
    age: 30
  }
]
```

PROGRAM 17

AIM: Create a database named college and then create a collection named students. Insert some values into it. Write a MongoDB Query to:

1. Display details of students who have their name starting with the letter 'C' using $regex operator

```
test> use college;
switched to db college
colleg> db.createCollection("students");
{ ok: 1 }
college> db.students.insertMany([
{id: 1, name: "Chris", dept: "CS", age: 21, gender: "Male" },
{id: 2, name: "Doanl", dept: "EE", age: 22, gender: "Male" },
{id: 3, name: "Anu", dept: "CS", age: 23, gender: "Female" },
{id: 4, name: "Karthika", dept: "ME", age: 24, gender: "Female" },
{id: 5, name: "Jency", dept: "EC", age: 25, gender: "Female" },
{id: 6, name: "Ryan", dept: "CS", age: 26, gender: "Male" },
{id: 7, name: "Ameer", dept: "EC", age: 27, gender: "Male" }
]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('660ee84b4181dffc519f990a'),
    '1': ObjectId('660ee84b4181dffc519f990b'),
    '2': ObjectId('660ee84b4181dffc519f990c'),
    '3': ObjectId('660ee84b4181dffc519f990d'),
    '4': ObjectId('660ee84b4181dffc519f990e'),
    '5': ObjectId('660ee84b4181dffc519f990f'),
    '6': ObjectId('660ee84b4181dffc519f9910')
  }
}
college> db.students.find()
[
  {
    _id: ObjectId('660ee84b4181dffc519f990a'),
    id: 1,
    name: 'Chris',
    dept: 'CS',
    age: 21,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990b'),
    id: 2,
    name: 'Doanl',
    dept: 'EE',
```

```
  age: 22,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990c'),
    id: 3,
    name: 'Anu',
    dept: 'CS',
    age: 23,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990d'),
    id: 4,
    name: 'Karthika',
    dept: 'ME',
    age: 24,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990e'),
    id: 5,
    name: 'Jency',
    dept: 'EC',
    age: 25,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990f'),
    id: 6,
    name: 'Ryan',
    dept: 'CS',
    age: 26,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f9910'),
    id: 7,
    name: 'Ameer',
    dept: 'EC',
    age: 27,
    gender: 'Male'
  }
]

college> db.students.find({name:{$regex:/^C/i}});
[
  {
    _id: ObjectId('660ee84b4181dffc519f990a'),
```

```
  id: 1,
    name: 'Chris',
    dept: 'CS',
    age: 21,
    gender: 'Male'
  }
]
```

2. Display details of students who have their name ending with the letter ‘r’ using $regex Operator

```
college> db.students.find({name:{$regex:/r$/i}});
[
  {
    _id: ObjectId('660ee84b4181dffc519f9910'),
    id: 7,
    name: 'Ameer',
    dept: 'EC',
    age: 27,
    gender: 'Male'
  }
]
```

3. Display details of students who are having ‘CS’ as their department using $regex operator

```
college> db.students.find({dept:{$regex:/CS/i}});
[
  {
    _id: ObjectId('660ee84b4181dffc519f990a'),
    id: 1,
    name: 'Chris',
    dept: 'CS',
    age: 21,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990c'),
    id: 3,
    name: 'Anu',
    dept: 'CS',
    age: 23,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990f'),
    id: 6,
    name: 'Ryan',
    dept: 'CS',
    age: 26,
```

```
    gender: 'Male'
  }
]
```

4. Remove details of student who are having ‘EC’ as their department

```
college> db.students.deleteMany({dept:{$regex:/EC/i}});
{ acknowledged: true, deletedCount: 2 }
college> db.students.find()
[
  {
    _id: ObjectId('660ee84b4181dffc519f990a'),
    id: 1,
    name: 'Chris',
    dept: 'CS',
    age: 21,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990b'),
    id: 2,
    name: 'Doanl',
    dept: 'EE',
    age: 22,
    gender: 'Male'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990c'),
    id: 3,
    name: 'Anu',
    dept: 'CS',
    age: 23,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990d'),
    id: 4,
    name: 'Karthika',
    dept: 'ME',
    age: 24,
    gender: 'Female'
  },
  {
    _id: ObjectId('660ee84b4181dffc519f990f'),
    id: 6,
    name: 'Ryan',
    dept: 'CS',
    age: 26,
    gender: 'Male' }
```

PROGRAM 18

AIM: Create database ‘candidate’ and collection ‘details’.

```
test> use candidate
switched to db candidate
candidate> db.createCollection("Details")
{ ok: 1 }
candidate>db.details.insert({"name":"Anu","age":21,"gender":"female","amount":7000});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f6da955ffdfc4748bf202') }
}
candidate>db.details.insert({"name":"Akhila","age":22,"gender":"female","amount":6000});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f6da955ffdfc4748bf203') }
}
candidate>db.details.insert({"name":"Arjun","age":32,"gender":"male","amount":20000});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f6da955ffdfc4748bf204') }
}
candidate>db.details.insert({"name":"Amal","age":45,"gender":"male","amount":40000});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f6da955ffdfc4748bf205') }
}
candidate>db.details.insert({"name":"Akash","age":53,"gender":"male","amount":50000});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f6da955ffdfc4748bf206') }
}
candidate> db.details.find()
[
  {cknowledged: true,
    _id: ObjectId('660f6da955ffdfc4748bf202'),c4748bf206') }
    name: 'Anu',
    age: 21,
    gender: 'female',
    amount: 7000
  },
  {
    _id: ObjectId('660f6da955ffdfc4748bf203'),
    name: 'Akhila',
    age: 22,
    gender: 'female',
```

```
  amount: 6000
 },
 {
  _id: ObjectId('660f6da955ffdfc4748bf204'),
  name: 'Arjun',
  age: 32,
  gender: 'male',
  amount: 20000
 },
 {
  _id: ObjectId('660f6da955ffdfc4748bf205'),
  name: 'Amal',
  age: 45,
  gender: 'male',
  amount: 40000
 },
 {
  _id: ObjectId('660f6dac55ffdfc4748bf206'),
  name: 'Akash',
  age: 53,
  gender: 'male',
  amount: 50000
 }
]
```

1. Query customer who are either male or younger than 25?

```
candidate> db.details.find({$or:[{'gender':'male'},{'age':{$lt:25}}]})
[
 {
  _id: ObjectId('660f6da955ffdfc4748bf202'),
  name: 'Anu',
  age: 21,
  gender: 'female',
  amount: 7000
 },
 {
  _id: ObjectId('660f6da955ffdfc4748bf203'),
  name: 'Akhila',
  age: 22,
  gender: 'female',
  amount: 6000
 },
 {
  _id: ObjectId('660f6da955ffdfc4748bf204'),
  name: 'Arjun',
  age: 32,
  gender: 'male',
  amount: 20000
```

```
 },
{
   _id: ObjectId('660f6da955ffdfc4748bf205'),
   name: 'Amal',
   age: 45,
   gender: 'male',
   amount: 40000
  },
  {
   _id: ObjectId('660f6dac55ffdfc4748bf206'),
   name: 'Akash',
   age: 53,
   gender: 'male',
   amount: 50000
  }
]
```

2. Calculate total purchase amount for males and females using aggregate method

```
candidate> db.details.find({$or:[{'gender':'male'},{'age':{$lt:25}}]})
[
{ _id: 'male', 'total amount': 110000 },
  { _id: 'female', 'total amount': 13000 }
]
```

3. Select customers who are older than 25 and calculate the average purchase amount for males and females

```
candidate>
db.details.aggregate([{$match:{"age":{$gt:25}}},{$group:{_id:"$gender",'totalamount':{$avg:'$a
mount'}}}])
[ { _id: 'male', totalamount: 36666.666666666664 } ]
```

4. sort the data based on average amount.

```
candidate>
db.details.aggregate([{$match:{"age":{$gt:25}}},{$group:{_id:"$gender",'totalamount':{$avg:'$amoun
t'}}},{$sor$sort:{avg:1}}])
[ { _id: 'male', totalamount: 36666.666666666664 } ]
```

PROGRAM 19

AIM: Create a database named college and then create a collection named studlist. Insert some values into it .Write a MongoDB Query to:

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

college>
db.details.insertMany([{"fname":"Akhila","lname":"T","mark":"95","gender":"F","dept":"MCA","grad
e":"A+","contact":"9947558569","loc":"kollam"}])
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f7448401dc347898bf203') }
}
college>
db.details.insertMany([{"fname":"Arjun","lname":"P","mark":"82","gender":"M","dept":"mech","grad
e":"A","contact":"9947558569","loc":"kannur"}])
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f7448401dc347898bf204') }
}
college>
db.details.insertMany([{"fname":"Anu","lname":"S","mark":"85","gender":"F","dept":"mech","grade":
"A","contact":"9947558569","loc":"tvm"}])
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f7448401dc347898bf205') }
}
college>
db.details.insertMany([{"fname":"Pooja","lname":"m","mark":"85","gender":"F","dept":"mech","grade
":"A","contact":"9947558569","loc":"tvm"}])
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f7448401dc347898bf206') }
}
college>
db.details.insertMany([{"fname":"Adarsh","lname":"h","mark":"85","gender":"M","dept":"mech","gra
de":"A","contact":"9947558569","loc":"pkd"}])
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f7448401dc347898bf207') }
}
college>
db.details.insertMany([{"fname":"Amal","lname":"h","mark":"78","gender":"M","dept":"MCA","grad
e":"B","contact":"9947558569","loc":"pkd"}])
```

```
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660f744a401dc347898bf208') }
}
college> db.details.find()
[
  {
    _id: ObjectId('660f7413401dc347898bf202'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
    loc: 'kollam'
  },
  {
    _id: ObjectId('660f7448401dc347898bf204'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf205'),
    fname: 'Anu',
    lname: 'S',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
```

```
  },
  {
    _id: ObjectId('660f7448401dc347898bf206'),
    fname: 'Pooja',
    lname: 'm',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  },
  {
    _id: ObjectId('660f7448401dc347898bf207'),
    fname: 'Adarsh',
    lname: 'h',
    mark: '85',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'pkd'
  },
  {
    _id: ObjectId('660f744a401dc347898bf208'),
    fname: 'Amal',
    lname: 'h',
    mark: '78',
    gender: 'M',
    dept: 'MCA',
    grade: 'B',
    contact: '9947558569',
    loc: 'pkd'
  }
]
```

1. Display name (both fname and lname) and mark of all female students in MCA department.

```
college> db.details.find({dept:"MCA",gender:"F"}).pretty()
[
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
```

```
    loc: 'kollam'
  }
]
```

2. Display the details of student who secured highest mark in the course MCA

```
college> db.details.find({dept:"MCA"},{_id:0}).sort({mark:-1}).limit(1)
[
  {
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
    loc: 'kollam'
  }
]
```

3. Display all male students who secured A+ grade.

```
college>
db.details.find({grade:"A+",gender:"F"},{fname:1,lname:1,gender:1,dept:1,loc:1,grade:1,mark:1})
[
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    loc: 'kollam'
  }
]
```

4. Display the names of the top three students in Mechanical department.

```
college> db.details.find({dept:"mech"}).sort({mark:-1}).limit(3)
[
  {
    _id: ObjectId('660f7448401dc347898bf207'),
    fname: 'Adarsh',
    lname: 'h',
    mark: '85',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
```

```
    contact: '9947558569',

    loc: 'pkd'
  },
  {
    _id: ObjectId('660f7448401dc347898bf206'),
    fname: 'Pooja',
    lname: 'm',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  },
  {
    _id: ObjectId('660f7448401dc347898bf205'),
    fname: 'Anu',
    lname: 'S',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  }
]
```

5. Display the details of female students [fname,lname,grade,mark,contact] who achieved a mark more than 90.

```
college> db.details.find({mark:{$gt:90},gender:'F'},{fname:1,lname:1,mark:1,contact:1,grade:1})
[
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    loc: 'kollam'
  }
]
```

6. Display the details of students who secured mark, more than 80 but less than 90.

```
college>
db.details.find({$and:[{mark:{$gt:80}},{mark:{$lt:90}}]},{fname:1,lname:1,mark:1,contact:1,grade:1}
)
[
  {
    _id: ObjectId('660f7413401dc347898bf202'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
    loc: 'kollam'
  },
  {
    _id: ObjectId('660f7448401dc347898bf204'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf205'),
    fname: 'Anu',
    lname: 'S',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
```

```
  },
  {
    _id: ObjectId('660f7448401dc347898bf206'),
    fname: 'Pooja',
    lname: 'm',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  },
  {
    _id: ObjectId('660f7448401dc347898bf207'),
    fname: 'Adarsh',
    lname: 'h',
    mark: '85',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'pkd'
  }
 ]
```

7. Display the details of students whose name starts with ‘P’

```
college> db.details.find({fname:{$regex:'^P'}},{})
[
  {
    _id: ObjectId('660f7448401dc347898bf206'),
    fname: 'Pooja',
    lname: 'm',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  }
]
```

8. Display all students from Kollam

```
college> db.details.find({loc:"kollam"},{})
[
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
```

```
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
    loc: 'kollam'
  }
]
```

9. Display all students who does not belong to neither Kollam nor Thiruvananthapuram

```
college> db.details.find({$nor:[{loc:"kollam"},{loc:"tvm"}]},{})
[
  {
    _id: ObjectId('660f7413401dc347898bf202'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf204'),
    fname: 'Arjun',
    lname: 'P',
    mark: '82',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'kannur'
  },
  {
    _id: ObjectId('660f7448401dc347898bf207'),
    fname: 'Adarsh',
    lname: 'h',
    mark: '85',
    gender: 'M',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'pkd'
  },
  {
    _id: ObjectId('660f744a401dc347898bf208'),
```

```
    fname: 'Amal',
    lname: 'h',
    mark: '78',
    gender: 'M',
    dept: 'MCA',
    grade: 'B',
    contact: '9947558569',
    loc: 'pkd'
  }
]
```

10. Display all female students who belong to either Kollam or Thiruvananthapuram

```
college> db.details.find({$or:[{loc:"kollam"},{loc:"tvm"}],gender:"F"},{})
[
  {
    _id: ObjectId('660f7448401dc347898bf203'),
    fname: 'Akhila',
    lname: 'T',
    mark: '95',
    gender: 'F',
    dept: 'MCA',
    grade: 'A+',
    contact: '9947558569',
    loc: 'kollam'
  },
  {
    _id: ObjectId('660f7448401dc347898bf205'),
    fname: 'Anu',
    lname: 'S',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  },
  {
    _id: ObjectId('660f7448401dc347898bf206'),
    fname: 'Pooja',
    lname: 'm',
    mark: '85',
    gender: 'F',
    dept: 'mech',
    grade: 'A',
    contact: '9947558569',
    loc: 'tvm'
  }
]
```

PROGRAM 20

AIM: Create a database in MongoDB named "mcadb" with collections named "course" and "students" and perform aggregate functions, and regular expressions on it.

```
test>  use mcadb
switched to db mcadb
mcadb> db.createCollection('course')
{ ok: 1 }
mcadb> db.createCollection('students')
{ ok: 1 }
mcadb>
db.students.insertMany([{name:'Jency',age:25,gender:'female'},{name:'Ganga',age:21,gender:'female'},
{name:'Sravan',age:24,gender:'male'},{name:'Hari',age:23,gender:'male'},{name:'David',age:23,gender:
'male'}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('661524ac6c2f90bded9f990a'),
    '1': ObjectId('661524ac6c2f90bded9f990b'),
    '2': ObjectId('661524ac6c2f90bded9f990c'),
    '3': ObjectId('661524ac6c2f90bded9f990d'),
    '4': ObjectId('661524ac6c2f90bded9f990e')
  }
}
mcadb> db.students.find()
[
  {
    _id: ObjectId('661524ac6c2f90bded9f990a'),
    name: 'Jency',
    age: 25,
    gender: 'female'
  },
  {
    _id: ObjectId('661524ac6c2f90bded9f990b'),
    name: 'Ganga',
    age: 21,
    gender: 'female'
  },
  {
    _id: ObjectId('661524ac6c2f90bded9f990c'),
    name: 'Sravan',
    age: 24,
    gender: 'male'
  },
  {
    _id: ObjectId('661524ac6c2f90bded9f990d'),
    name: 'Hari',
```

```
    age: 23,
    gender: 'male'
  },
  {
    _id: ObjectId('661524ac6c2f90bded9f990e'),
    name: 'David',
    age: 23,
    gender: 'male'
  }
]
mcadb>
db.course.insertMany([{course:'ENG',C_credict:2},{course:'MCA',C_credict:3},{course:'BCA',c_credi
ct:2},{course:'ENG',C_credict:3},{course:'BCA',C_credict:2}])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('661524cb6c2f90bded9f990f'),
    '1': ObjectId('661524cb6c2f90bded9f9910'),
    '2': ObjectId('661524cb6c2f90bded9f9911'),
    '3': ObjectId('661524cb6c2f90bded9f9912'),
    '4': ObjectId('661524cb6c2f90bded9f9913')
  }
}
mcadb> db.course.find()
[
  {
    _id: ObjectId('661524cb6c2f90bded9f990f'),
    course: 'ENG',
    C_credict: 2
  },
  {
    _id: ObjectId('661524cb6c2f90bded9f9910'),
    course: 'MCA',
    C_credict: 3
  },
  {
    _id: ObjectId('661524cb6c2f90bded9f9911'),
    course: 'BCA',
    c_credict: 2
  },
  {
    _id: ObjectId('661524cb6c2f90bded9f9912'),
    course: 'ENG',
    C_credict: 3
  },
  {
    _id: ObjectId('661524cb6c2f90bded9f9913'),
    course: 'BCA',
    C_credict: 2 }]
```

1.Calculate the average age of students

```
mcadb> db.students.aggregate([{$group:{_id:null,averageAge:{$avg:"$age"}}}])
[ { _id: null, averageAge: 23.2 } ]
```

2.Count the number of male and female students

```
mcadb> db.students.aggregate([{$group:{_id:"$gender",count:{$sum:1}}}]);
[ { _id: 'female', count: 2 }, { _id: 'male', count: 3 } ]
```

3.Find the courses with the highest number of credits

```
mcadb> db.course.aggregate([{$sort:{C_credits:-1}},{$limit:1}]);
[
  {
    _id: ObjectId('661524cb6c2f90bded9f990f'),
    course: 'ENG',
    C_credict: 2
  }
]
```

4. Find students whose names start with "J"

```
mcadb> db.students.find({name:{$regex:"^J"}})
[
  {
    _id: ObjectId('6614db4c52ff5f31438bf202'),
    name: 'Jency',
    age: 25,
    gender: 'female'
  }
]
```

5. Find courses with codes containing "ENG"

```
mcadb> db.course.find({course:{$regex:"^ENG"}})
[
  {
    _id: ObjectId('6614dde752ff5f31438bf207'),
    course: 'ENG',
    C_credict: 2
  },
  {
    _id: ObjectId('6614dde752ff5f31438bf20a'),
    course: 'ENG',
    C_credict: 3
  }
]
```

Course Outcome 4

Understand the basic storage architecture of distributed file systems

PROGRAM 21

AIM: Build collections mcaDB documents students,course and perform shell commands to create replicaset, indexing etc

```
test> use mca
switched to db mca
mca> db.createCollection("students")
{ ok: 1 }
mca> db.createCollection("course")
{ ok: 1 }
mca> db.students.insert({ "name": "John", "age": 25, "course": "MCA" })
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('663526abd2a4a954b39f990a') }
}
mca> db.course.insert({ "courseName": "Database Systems", "credits": 3 })
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('663526abd2a4a954b39f990b') }
}
mca> db.students.createIndex({ "name": 1 })
name_1
mca> db.students.getIndexes()
[
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { rollNo: 1 }, name: 'rollNo_1' },
  { v: 2, key: { name: 1 }, name: 'name_1' }
]
```

Course Outcome 5

Design and deployment of NoSQL databases with real time requirements.

PROGRAM 22

AIM: Develop students' marks calculation applications using Python and MongoDB

```
import pymongo

# Connect to MongoDB
client = pymongo.MongoClient("mongodb://localhost:27017/")
db = client["student_marks"]
collection = db["students"]

def enter_student_data():
   name = input("Enter student name: ")
   dbms = float(input("Enter DBMS marks: "))
   oops = float(input("Enter OOPS marks: "))
   networks = float(input("Enter Networks marks: "))

   student_data = {
   "name": name,
   "dbms": dbms,

   "oops": oops,
   "networks": networks
   }
   collection.insert_one(student_data)
   print("Student data entered successfully!")

def calculate_student_marks():
   name = input("Enter student name to calculate marks: ")
   student = collection.find_one({"name": name})

   if student:
      total_marks = student["dbms"] + student["oops"] + student["networks"] print(f"Total
      marks for {name}: {total_marks}")
   else:
      print(f"Student '{name}' not found!")

def view_students():
   print("List of Students:")
```

```
for student in collection.find():
print(f"Name: {student['name']}, DBMS: {student['dbms']}, OOPS:{student['oops']}, Networks:
{student['networks']}")
def main():

  while True:
    print("\nStudent Marks Calculation Application") print("1. Enter
    student data")
    print("2. Calculate student marks")
    print("3. View students")
    print("4. Exit")

    choice = input("Enter your choice: ")

    if choice == "1":
      enter_student_data()
    elif choice == "2":
      calculate_student_marks()
    elif choice == "3":
      view_students()
    elif choice == "4":
      print("Exiting the application.")
      break
    else:
      print("Invalid choice. Please try again.")
  if __name__ == "__main__":
  main()
```

OUTPUT

```
Student Marks Calculation Application
1. Enter student data
2. Calculate student marks
3. View students
4. Exit
Enter your choice: 1
Enter student name: Meenu
Enter DBMS marks: 50
Enter OOPS marks: 30
Enter Networks marks: 70
Student data entered successfully!

Student Marks Calculation Application
```

1. Enter student data
2. Calculate student marks
3. View students
4. Exit

Enter your choice: 2
Enter student name to calculate marks: Meenu
Total marks for Meenu: 150.0

Student Marks Calculation Application
1. Enter student data
2. Calculate student marks
3. View students
4. Exit

Enter your choice: 3
List of Students:
Name: Meenu, DBMS: 50.0, OOPS:30.0, Networks: 70.0

Student Marks Calculation Application
1. Enter student data
2. Calculate student marks
3. View students
4. Exit

Enter your choice:4