#Create a new database called student_management

use student_management

#Create a collection called students in the student_management database

db.createCollection("students")

#Insert at least five student records into the students collection.Each record should have the following fields:

db.students.insertMany([

{student_id:101,name:"Akshay kumar",age:23,department:"Computer Science",courses:["HTML","DATABASE SYSTEM"],grade:"A"},

{student_id:102,name:"Salman khan",age:24,departement:"Information Technology",courses:["Btech","BCOM"],grade:"E"},

{student_id:103,name:"Shahid kapoor",age:20,department:"Hotel Manager",courses:["Bcom","BSC"],grade:"A"},

{student_id:104,name:"John Abraham",age:19,department:"Computer Science",courses:["DATABASE SYSTEM","BSC"],grade:"A"},

{student_id:105,name:"Amir khan",age:24,department:"Computer Science",courses:["PTHON","HTML"],grade:"D"}

])

#1 Retrieve all students who are in the "Computer Science" department.

db.students.find({department:"Computer Science"})

```
student_management> db.students.find({department:"Computer Science"})
[
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'),
    student_id: 101,
    name: 'Akshay kumar',
    age: 23,
    department: 'Computer Science',
    courses: [ 'HTML', 'DATABASE SYSTEM' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce5'),
    student_id: 104,
    name: 'John Abraham',
    age: 19,
    department: 'Computer Science',
    courses: [ 'DATABASE SYSTEM', 'BSC' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce6'),
    student_id: 105,
    name: 'Amir khan',
    age: 24,
    department: 'Computer Science',
    courses: [ 'PTHON', 'HTML' ],
    grade: 'D'
  }
]
```

#2 Retrieve students who have an age greater than 21.

db.students.find({age:{$gt:21}})

```
student_management>  db.students.find({age:{$gt:21}})
[
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'),
    student_id: 101,
    name: 'Akshay kumar',
    age: 23,
    department: 'Computer Science',
    courses: [ 'HTML', 'DATABASE SYSTEM' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce3'),
    student_id: 102,
    name: 'Salman khan',
    age: 24,
    departement: 'Information Technology',
    courses: [ 'Btech', 'BCOM' ],
    grade: 'E'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce6'),
    student_id: 105,
    name: 'Amir khan',
    age: 24,
    department: 'Computer Science',
    courses: [ 'PTHON', 'HTML' ],
    grade: 'D'
  }
]
```

#3 Retrieve students who are taking the "Database Systems" course.

db.students.find({courses:"DATABASE SYSTEM"})

```
student_management>  db.students.find({courses:"DATABASE SYSTEM"})
[
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'),
    student_id: 101,
    name: 'Akshay kumar',
    age: 23,
    department: 'Computer Science',
    courses: [ 'HTML', 'DATABASE SYSTEM' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce5'),
    student_id: 104,
    name: 'John Abraham',
    age: 19,
    department: 'Computer Science',
    courses: [ 'DATABASE SYSTEM', 'BSC' ],
    grade: 'A'
  }
]
```

#4 Retrieve students with a grade of "A".

db.students.find({grade:"A"})

```
student_management> db.students.find({grade:"A"})
[
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'),
    student_id: 101,
    name: 'Akshay kumar',
    age: 23,
    department: 'Computer Science',
    courses: [ 'HTML', 'DATABASE SYSTEM' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce4'),
    student_id: 103,
    name: 'Shahid kapoor',
    age: 20,
    department: 'Hotel Manager',
    courses: [ 'Bcom', 'BSC' ],
    grade: 'A'
  },
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce5'),
    student_id: 104,
    name: 'John Abraham',
    age: 19,
    department: 'Computer Science',
    courses: [ 'DATABASE SYSTEM', 'BSC' ],
    grade: 'A'
  }
]
```

#5 Update the age of a student with student_id 101 to 21.

db.students.updateOne({student_id:101},{$set:{age:21}})

```
student_management> db.students.updateOne({student_id:101},{$set:{age:21}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
student_management> db.students.find()
[
  {
    _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'),
    student_id: 101,
    name: 'Akshay kumar',
    age: 21,
    department: 'Computer Science',
    courses: [ 'HTML', 'DATABASE SYSTEM' ],
    grade: 'A'
  },
```

#6 Add a new course, "Machine Learning", to the courses array for students in the "Computer Science" department.

db.students.updateMany({department:"Computer Science"},{$addToSet:{courses:"Machine Learning"}})

```
student_management> db.students.updateMany({department:"Computer Science"},{$addToSet:{courses:"Machine Learning"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

```
[
  {
    name: 'Akshay kumar',
    courses: [ 'HTML', 'DATABASE SYSTEM', 'Machine Learning' ]
  },
  {
    name: 'John Abraham',
    courses: [ 'DATABASE SYSTEM', 'BSC', 'Machine Learning' ]
  },
  {
    name: 'Amir khan',
    courses: [ 'PTHON', 'HTML', 'Machine Learning' ]
  }
]
```

#7 Delete a student record with student_id 105.

db.student.deleteOne({student_id:105})

```
student_management> db.students.deleteOne({student_id:105})
{ acknowledged: true, deletedCount: 1 }
```

```
student_management> db.students.find({},{student_id:101})
[
  { _id: ObjectId('67d6f62b0ebbbfe5cacb0ce2'), student_id: 101 },
  { _id: ObjectId('67d6f62b0ebbbfe5cacb0ce3'), student_id: 102 },
  { _id: ObjectId('67d6f62b0ebbbfe5cacb0ce4'), student_id: 103 },
  { _id: ObjectId('67d6f62b0ebbbfe5cacb0ce5'), student_id: 104 }
]
```

#8 Delete all students who have a grade lower than "C".

db.students.deleteMany({grade:{$lt:"C"}})

```
student_management>  db.students.deleteMany({grade:{$lt:"C"}})
{ acknowledged: true, deletedCount: 3 }
```

```
{ _id: ObjectId('67cc573c1ac2c7c857cb0cfb'), grade: 'A' },
{ _id: ObjectId('67cc573c1ac2c7c857cb0cfd'), grade: 'A' },
{ _id: ObjectId('67cc573c1ac2c7c857cb0cfe'), grade: 'A' }
```