MongoDB - 2

1. Create a database named university and a collection named students. Insert multiple student documents with fields: name, age, department, and grades.

```
db.students.insertMany([
    {
        "name": "Alice",
        "age": 20,
        "department": "Computer Science",
        "grades": { "math": 85, "english": 92 }
    },
    {
        "name": "Bob",
        "age": 21,
        "department": "Physics",
        "grades": { "math": 88, "physics": 90 }
    },
    {
        "name": "Charlie",
        "age": 22,
        "department": "Mathematics",
        "grades": { "math": 95, "statistics": 89 }
    }
]);
```

use university

db.createCollection("students")

db.students.insertMany([

{name:"Alice",age:20,department:"Computer Science",grades:{math:85,english:92}},

{name:"Bob",age:21,department:"Physics",grades:{math:88,english:90}},

{name:"Charlie",age:22,department:"Mathematics",grades:{math:95,english:89}}])

db.students.find()

```
 mycompiler_mongodb> { ok: 1 }
mycompiler_mongodb> ... ... ... {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67dd941343f20ebeed6b128c'),
    '1': ObjectId('67dd941343f20ebeed6b128d'),
    '2': ObjectId('67dd941343f20ebeed6b128e')
  }
}
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd941343f20ebeed6b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  },
  {
    _id: ObjectId('67dd941343f20ebeed6b128d'),
    name: 'Bob',
    age: 21,
    department: 'Physics',
    grades: { math: 88, english: 90 }
  },
  {
    _id: ObjectId('67dd941343f20ebeed6b128e'),
    name: 'Charlie',
    age: 22,
    department: 'Mathematics',
    grades: { math: 95, english: 89 }
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

2. Write a query to display all students who are in the Computer Science department.
   db.students.find({department:"Computer Science"})

```
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd978b3ff00b2a046b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

3. Write a query to update the grades of a student named Alice by adding a new subject programming with a grade of 93.

db.students.updateOne({name:"Alice"},{$set:{ "grades.programming": 93}})

db.students.find({name:"Alice"})

```
mycompiler_mongodb> {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd982c70299beec06b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92, programming: 93 }
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

4. Write a query to increment the age of all students by 1.
   db.students.updateMany({},{$inc:{age:1}})
   db.students.find()

```
mycompiler_mongodb> {
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd993df76d8b52d66b128c'),
    name: 'Alice',
    age: 21,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  },
  {
    _id: ObjectId('67dd993df76d8b52d66b128d'),
    name: 'Bob',
    age: 22,
    department: 'Physics',
    grades: { math: 88, english: 90 }
  },
  {
    _id: ObjectId('67dd993df76d8b52d66b128e'),
    name: 'Charlie',
    age: 23,
    department: 'Mathematics',
    grades: { math: 95, english: 89 }
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

5. Write a query to delete all students who are 23 years old.
   db.students.deleteMany({age:23})
   db.students.find()

```
mycompiler_mongodb> { acknowledged: true, deletedCount: 0 }
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd9a5284d3cbeeba6b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  },
  {
    _id: ObjectId('67dd9a5284d3cbeeba6b128d'),
    name: 'Bob',
    age: 21,
    department: 'Physics',
    grades: { math: 88, english: 90 }
  },
  {
    _id: ObjectId('67dd9a5284d3cbeeba6b128e'),
    name: 'Charlie',
    age: 22,
    department: 'Mathematics',
    grades: { math: 95, english: 89 }
  }
]
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

6. Write a query to create an index on the name field of the students collection.
   db.students.createIndex({ name: 1 })

```
 mycompiler_mongodb> ... ... ... {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67dd9afff9c31407ee6b128c'),
    '1': ObjectId('67dd9afff9c31407ee6b128d'),
    '2': ObjectId('67dd9afff9c31407ee6b128e')
  }
}
mycompiler_mongodb> name_1
mycompiler_mongodb>
mycompiler_mongodb>

[Execution complete with exit code 0]
```

7. Write an aggregation query to group students by their department and calculate the average age in each department.
   db.students.aggregate([{$group:{_id:"$department",averageAge:{$avg:"$age"}}}])

```
 mycompiler_mongodb> ... ... ... {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('67dd9c5d657963b07c6b128c'),
    '1': ObjectId('67dd9c5d657963b07c6b128d'),
    '2': ObjectId('67dd9c5d657963b07c6b128e')
  }
}
mycompiler_mongodb> [
  { _id: 'Physics', averageAge: 21 },
  { _id: 'Computer Science', averageAge: 20 },
  { _id: 'Mathematics', averageAge: 22 }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

8. Write a query to find all students who have scored more than 90 in any subject.
   db.students.find({$or:[{"grades.math":{$gt:90}},{"grades.english":{$gt:90}}]})

```
mycompiler_mongodb> [
  {
    _id: ObjectId('67dda26e3ed4385bb56b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  },
  {
    _id: ObjectId('67dda26e3ed4385bb56b128e'),
    name: 'Charlie',
    age: 22,
    department: 'Mathematics',
    grades: { math: 95, english: 89 }
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

9. Write a query to add a new field graduated set to false for all students who are in the Mathematics department.

db.students.updateMany({department:"Mathematics"},{$set:{graduated:false}})

db.students.find()

```
mycompiler_mongodb> {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
mycompiler_mongodb> [
  {
    _id: ObjectId('67dd9dea2f922c6dc66b128c'),
    name: 'Alice',
    age: 20,
    department: 'Computer Science',
    grades: { math: 85, english: 92 }
  },
  {
    _id: ObjectId('67dd9dea2f922c6dc66b128d'),
    name: 'Bob',
    age: 21,
    department: 'Physics',
    grades: { math: 88, english: 90 }
  },
  {
    _id: ObjectId('67dd9dea2f922c6dc66b128e'),
    name: 'Charlie',
    age: 22,
    department: 'Mathematics',
    grades: { math: 95, english: 89 },
    graduated: false
  }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```

10. How can you retrieve only the name and department fields for all students, excluding the _id field?

db.students.find({},{_id:0,name:1,department:1})

```
mycompiler_mongodb> [
  { name: 'Alice', department: 'Computer Science' },
  { name: 'Bob', department: 'Physics' },
  { name: 'Charlie', department: 'Mathematics' }
]
mycompiler_mongodb>

[Execution complete with exit code 0]
```