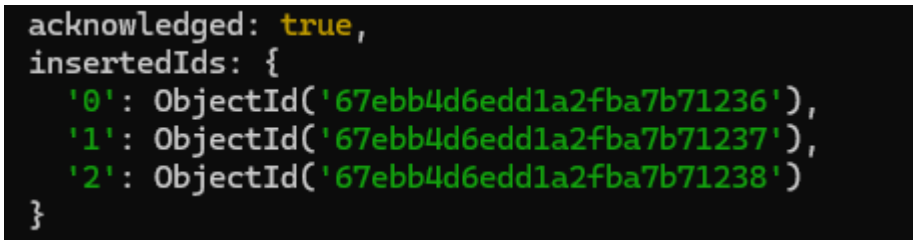
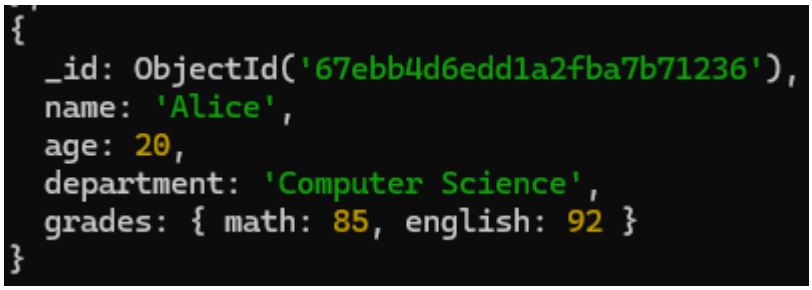
1.Create a database named university and a collection named students.Insert multiple student documents with fields:name,age,department,and grades.

```
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,physics:90}},{name:"Charlie",age:22,department:"Mathematics",grades:{math:95,statistics:89}}])
```



2.Write a query to display all students who are in the computer science department.

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



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"})

```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71236'),
  name: 'Alice',
  age: 20,
  department: 'Computer Science',
  grades: { math: 85, english: 92, programming: 93 }
}
```

4.Write a query to increment the age of all students by 1.

db.students.updateMany({},{$inc:{age:1}})

db.students.find()

```
acknowledged: true,
insertedId: null,
matchedCount: 19,
modifiedCount: 19,
upsertedCount: 0
```

```
  _id: ObjectId('67ebb4d6edd1a2fba7b71236'),
  name: 'Alice',
  age: 21,
  department: 'Computer Science',
  grades: { math: 85, english: 92, programming: 93 }
},
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71237'),
  name: 'Bob',
  age: 22,
  department: 'Physics',
  grades: { math: 88, physics: 90 }
},
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71238'),
  name: 'Charlie',
  age: 23,
  department: 'Mathematics',
  grades: { math: 95, statistics: 89 }
}
```

5.Write a query to delete all students who are 23 years old.

db.students.deleteMany({age:23})

db.students.find()

```
  _id: ObjectId('67ebb4d6edd1a2fba7b71236'),
  name: 'Alice',
  age: 21,
  department: 'Computer Science',
  grades: { math: 85, english: 92, programming: 93 }
},
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71237'),
  name: 'Bob',
  age: 22,
  department: 'Physics',
  grades: { math: 88, physics: 90 }
}
```

6.Write a query to create an index on the name field of the students collection.

```
db.students.createIndex({name:1})
```

```
name_1
```

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"}}}])
```

```
{ _id: null, averageAge: 28.5 },
{ _id: 'Physics', averageAge: 22 },
{ _id: 'Mathematics', averageAge: 22.5 },
{ _id: 'Computer Science', averageAge: 24.5 },
{ _id: 'Computer science', averageAge: 28 }
```

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}}]})
```

```
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71236'),
  name: 'Alice',
  age: 21,
  department: 'Computer Science',
  grades: { math: 85, english: 92, programming: 93 }
},
{
  _id: ObjectId('67ebd055507e91f88cb71236'),
  name: 'Charlie',
  age: 22,
  department: 'Mathematics',
  grades: { math: 95, english: 89 }
},
{
  _id: ObjectId('67ebd0f6507e91f88cb71237'),
  name: 'Charlie',
  age: 23,
  department: 'Mathematics',
  grades: { math: 95, english: 89 }
}
```

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"}})

```
acknowledged: true,
insertedId: null,
matchedCount: 2,
modifiedCount: 2,
upsertedCount: 0
```

db.students.find()

```
  _id: ObjectId('67ebb4d6edd1a2fba7b71236'),
  name: 'Alice',
  age: 21,
  department: 'Computer Science',
  grades: { math: 85, english: 92, programming: 93 }
},
{
  _id: ObjectId('67ebb4d6edd1a2fba7b71237'),
  name: 'Bob',
  age: 22,
  department: 'Physics',
  grades: { math: 88, physics: 90 }
},
{
  _id: ObjectId('67ebd055507e91f88cb71236'),
  name: 'Charlie',
  age: 22,
  department: 'Mathematics',
  grades: { math: 95, english: 89 },
  graduated: 'false'
},
{
  _id: ObjectId('67ebd0f6507e91f88cb71237'),
  name: 'Charlie',
  age: 23,
  department: 'Mathematics',
  grades: { math: 95, english: 89 },
  graduated: 'false'
}
```

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})

```
{ name: 'Alice', department: 'Computer Science' },
{ name: 'Bob', department: 'Physics' },
{ name: 'Charlie', department: 'Mathematics' },
```