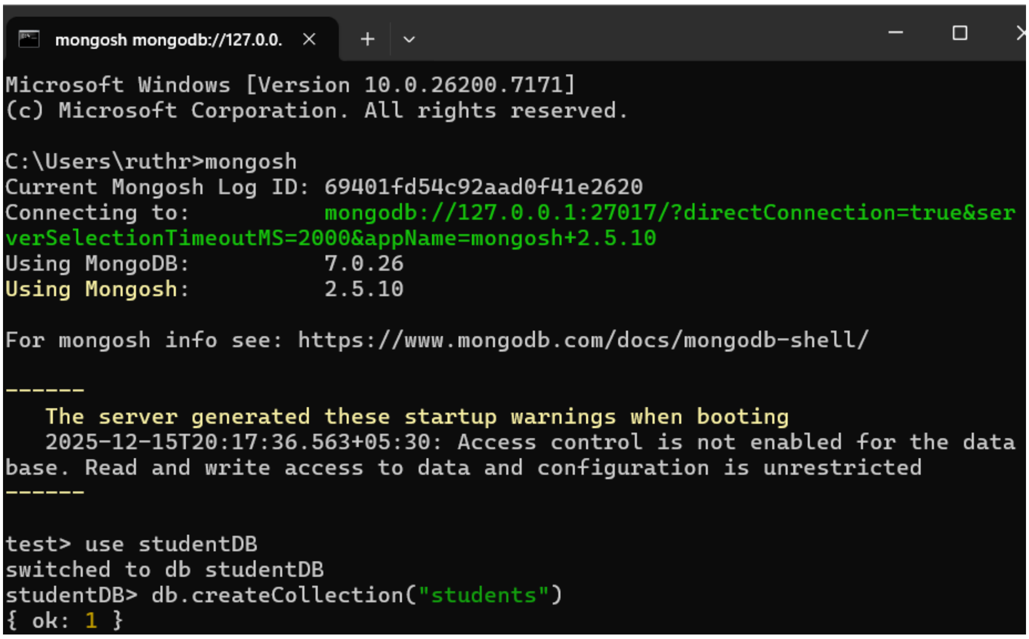
MongoDB Practice Task

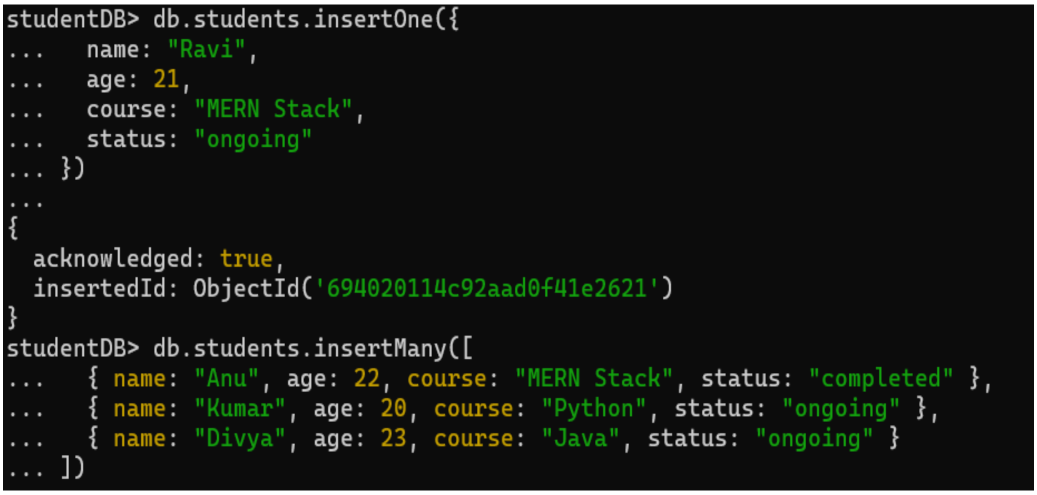
Student Management System

1. Database & Collection Setup



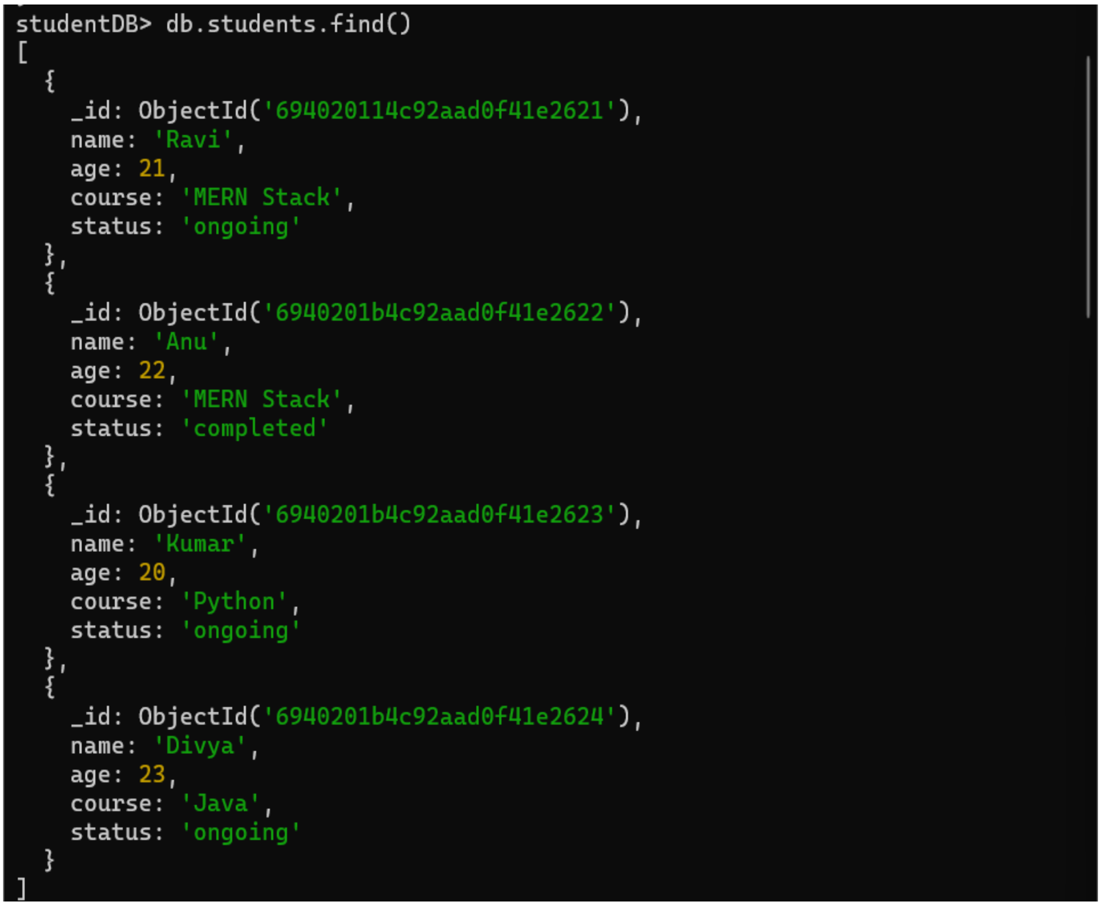
```
Microsoft Windows [Version 10.0.26200.7171]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ruthr>mongosh
Current Mongosh Log ID: 69401fd54c92aad0f41e2620
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&ser
verSelectionTimeoutMS=2000&appName=mongosh+2.5.10
Using MongoDB:          7.0.26
Using Mongosh:          2.5.10

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-12-15T20:17:36.563+05:30: Access control is not enabled for the data
base. Read and write access to data and configuration is unrestricted
------

test> use studentDB
switched to db studentDB
studentDB> db.createCollection("students")
{ ok: 1 }
```

2. Insert Operations

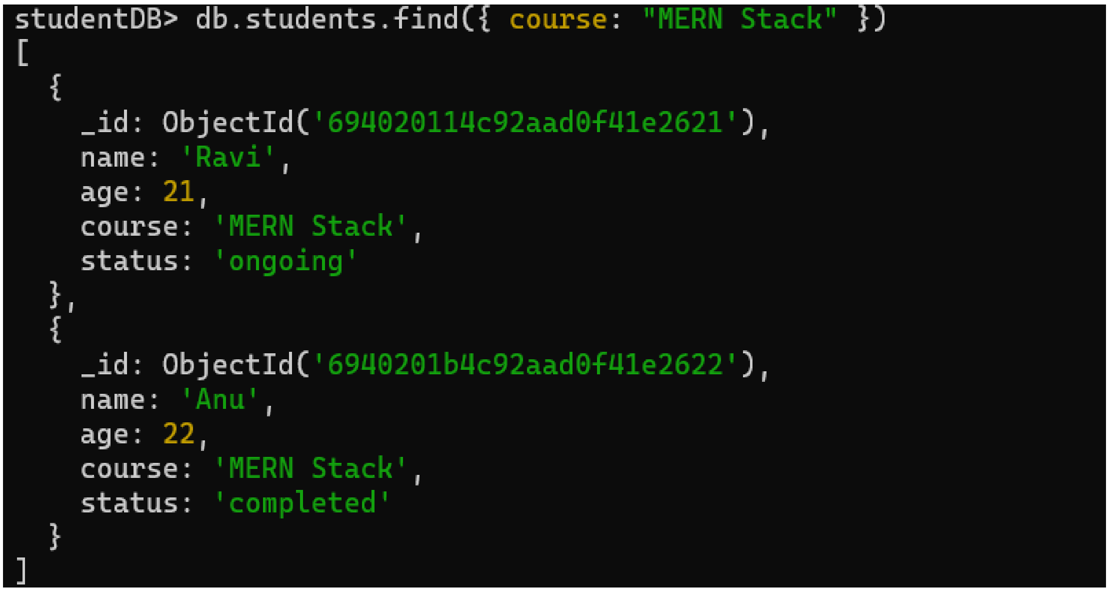


```
studentDB> db.students.insertOne({
...     name: "Ravi",
...     age: 21,
...     course: "MERN Stack",
...     status: "ongoing"
... })
...
{
  acknowledged: true,
  insertedId: ObjectId('694020114c92aad0f41e2621')
}
studentDB> db.students.insertMany([
...     { name: "Anu", age: 22, course: "MERN Stack", status: "completed" },
...     { name: "Kumar", age: 20, course: "Python", status: "ongoing" },
...     { name: "Divya", age: 23, course: "Java", status: "ongoing" }
... ])
```

3. Read Operations

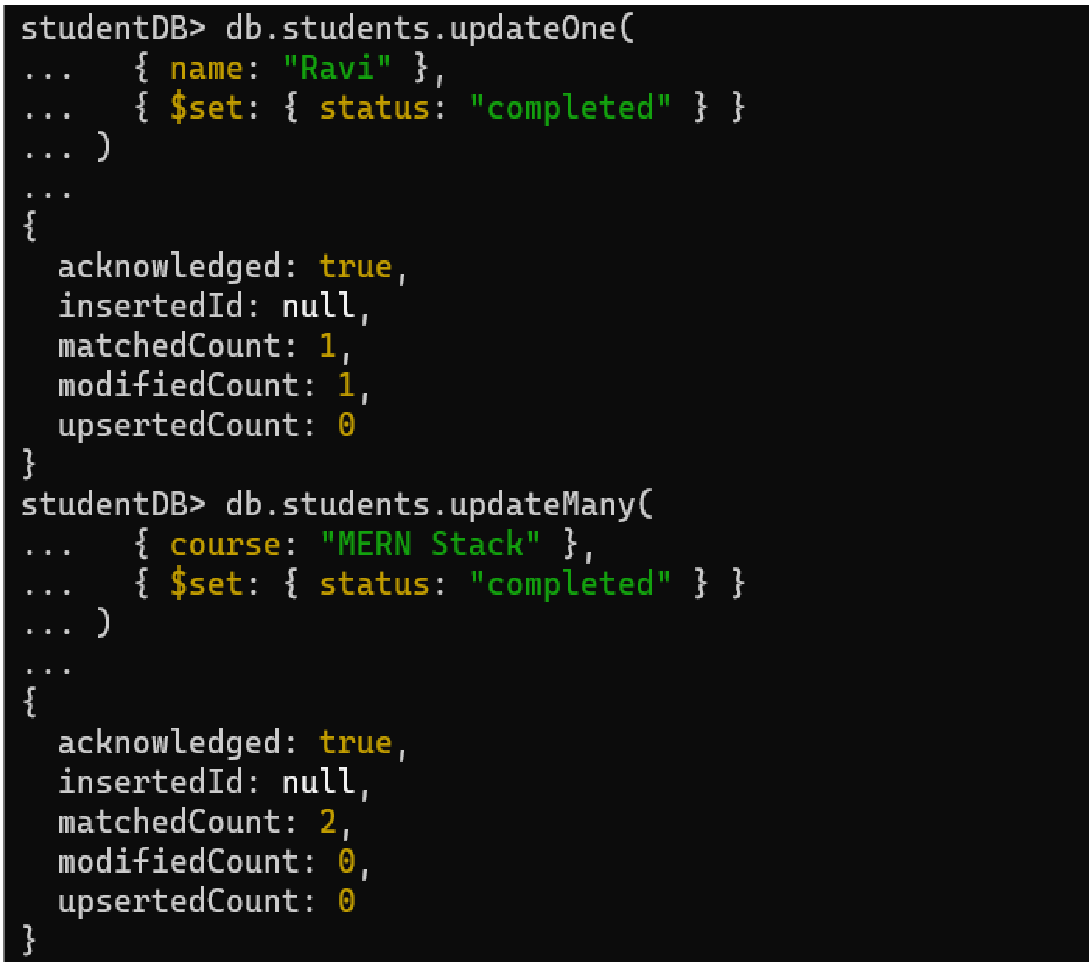


```
studentDB> db.students.find()
[
  {
    _id: ObjectId('694020114c92aad0f41e2621'),
    name: 'Ravi',
    age: 21,
    course: 'MERN Stack',
    status: 'ongoing'
  },
  {
    _id: ObjectId('6940201b4c92aad0f41e2622'),
    name: 'Anu',
    age: 22,
    course: 'MERN Stack',
    status: 'completed'
  },
  {
    _id: ObjectId('6940201b4c92aad0f41e2623'),
    name: 'Kumar',
    age: 20,
    course: 'Python',
    status: 'ongoing'
  },
  {
    _id: ObjectId('6940201b4c92aad0f41e2624'),
    name: 'Divya',
    age: 23,
    course: 'Java',
    status: 'ongoing'
  }
]
```



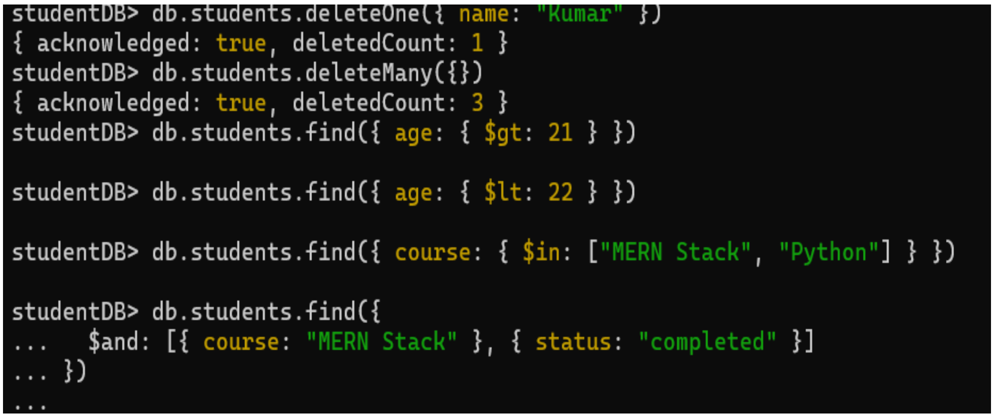
```
studentDB> db.students.find({ course: "MERN Stack" })
[
  {
    _id: ObjectId('694020114c92aad0f41e2621'),
    name: 'Ravi',
    age: 21,
    course: 'MERN Stack',
    status: 'ongoing'
  },
  {
    _id: ObjectId('6940201b4c92aad0f41e2622'),
    name: 'Anu',
    age: 22,
    course: 'MERN Stack',
    status: 'completed'
  }
]
```

4. Update Operations



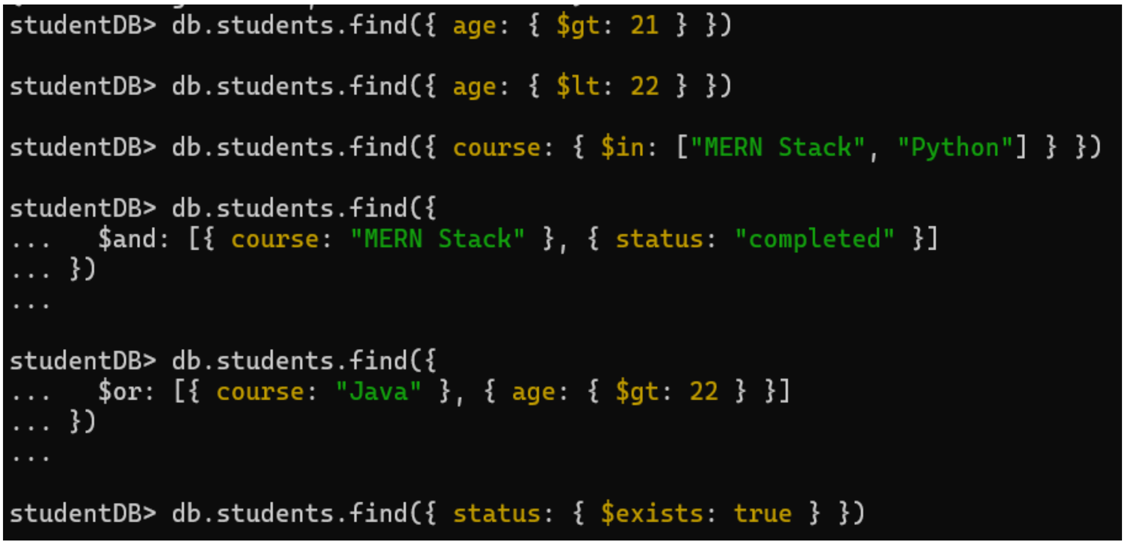
```
studentDB> db.students.updateOne(
...    { name: "Ravi" },
...    { $set: { status: "completed" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
studentDB> db.students.updateMany(
...    { course: "MERN Stack" },
...    { $set: { status: "completed" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 0,
  upsertedCount: 0
}
```

5. Delete Operation

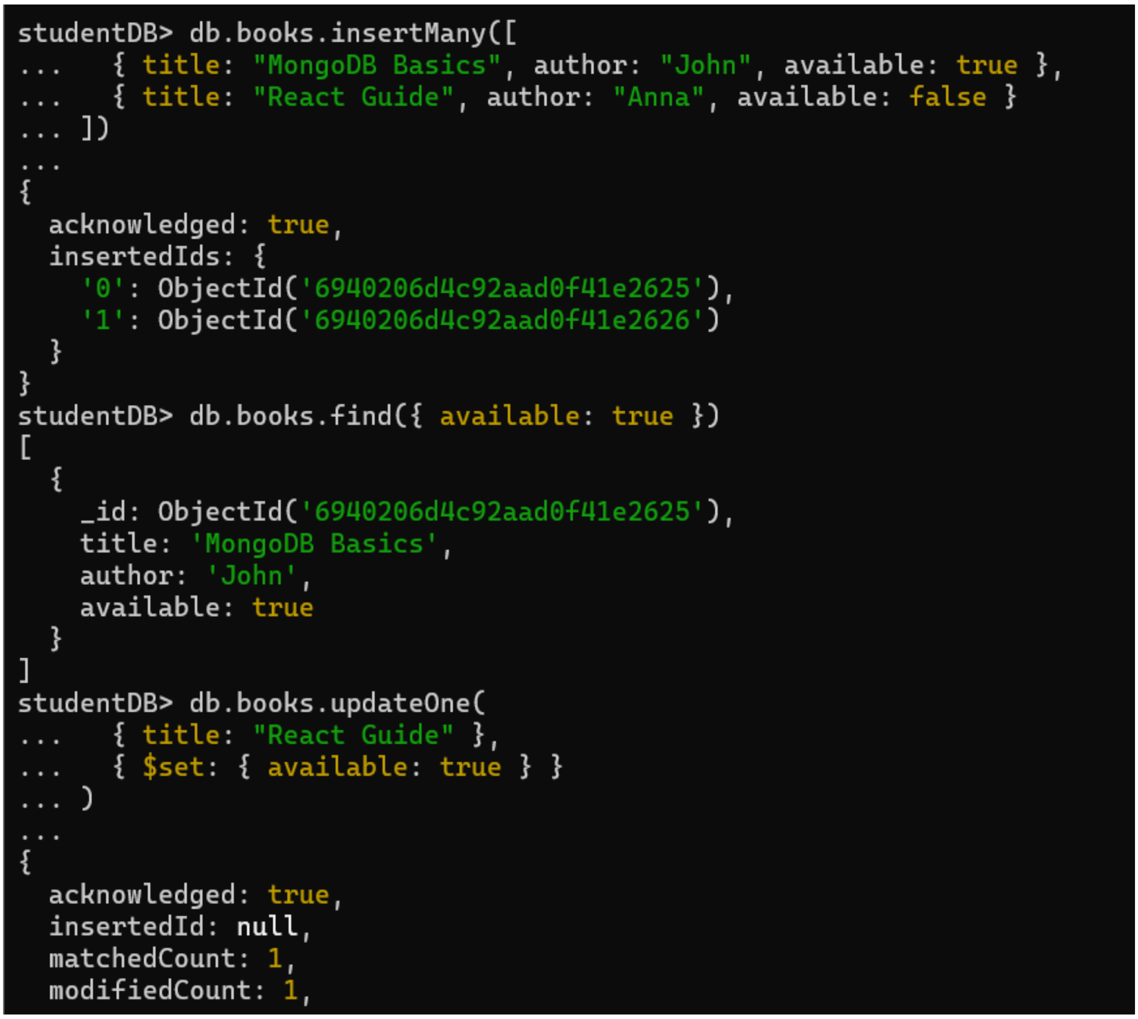


```
studentDB> db.students.deleteOne({ name: "Kumar" })
{ acknowledged: true, deletedCount: 1 }
studentDB> db.students.deleteMany({})
{ acknowledged: true, deletedCount: 3 }
studentDB> db.students.find({ age: { $gt: 21 } })

studentDB> db.students.find({ age: { $lt: 22 } })

studentDB> db.students.find({ course: { $in: ["MERN Stack", "Python"] } })

studentDB> db.students.find({
...   $and: [{ course: "MERN Stack" }, { status: "completed" }]
... })
...
```

6. Query Operators



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

studentDB> db.students.find({ age: { $lt: 22 } })

studentDB> db.students.find({ course: { $in: ["MERN Stack", "Python"] } })

studentDB> db.students.find({
...   $and: [{ course: "MERN Stack" }, { status: "completed" }]
... })
...

studentDB> db.students.find({
...   $or: [{ course: "Java" }, { age: { $gt: 22 } }]
... })
...

studentDB> db.students.find({ status: { $exists: true } })
```



```
studentDB> db.books.insertMany([
...   { title: "MongoDB Basics", author: "John", available: true },
...   { title: "React Guide", author: "Anna", available: false }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6940206d4c92aad0f41e2625'),
    '1': ObjectId('6940206d4c92aad0f41e2626')
  }
}
studentDB> db.books.find({ available: true })
[
  {
    _id: ObjectId('6940206d4c92aad0f41e2625'),
    title: 'MongoDB Basics',
    author: 'John',
    available: true
  }
]
studentDB> db.books.updateOne(
...   { title: "React Guide" },
...   { $set: { available: true } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
```