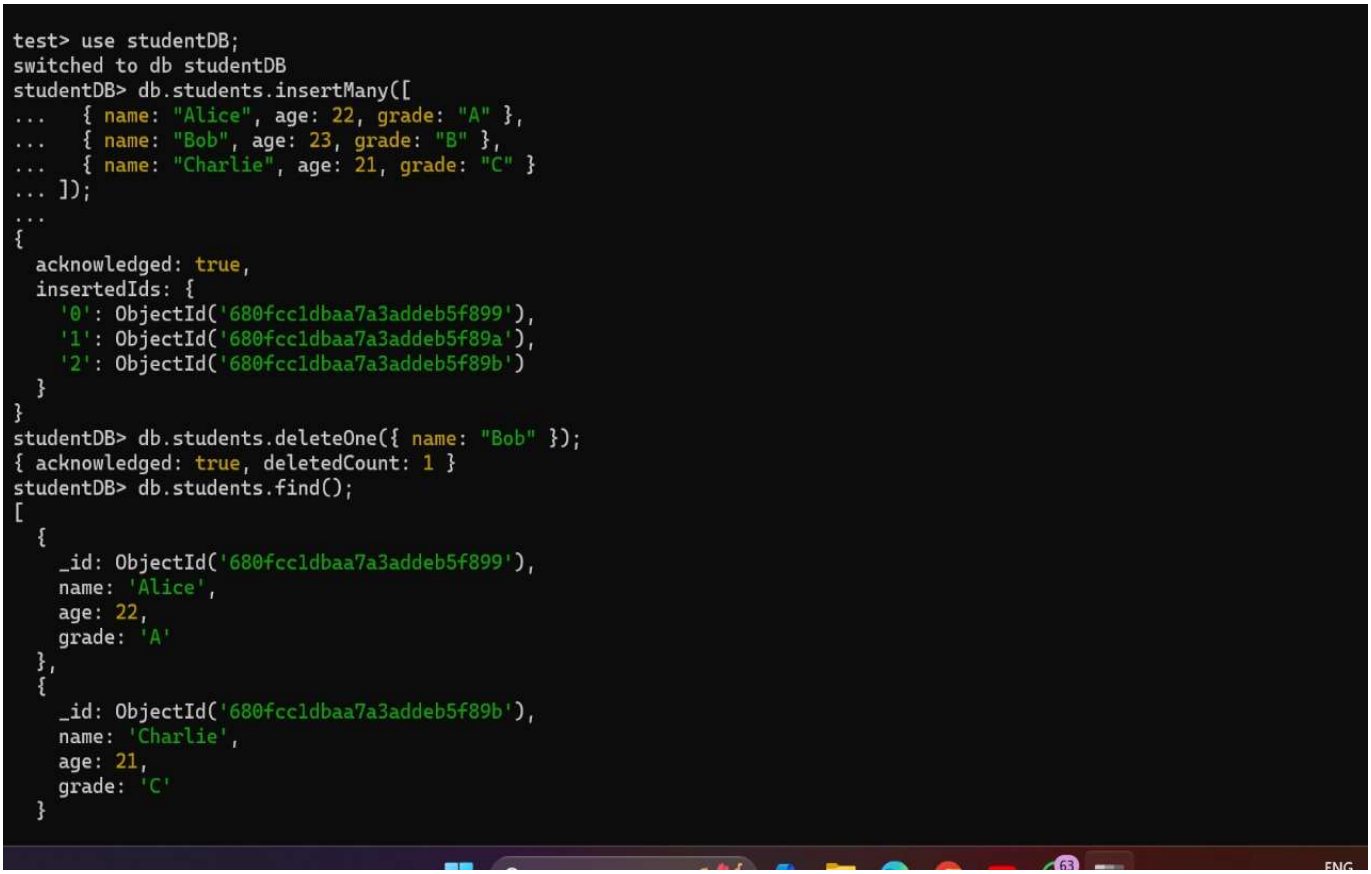
MongoDB Practicals

13. Create, Insert and Delete a Document from a Collection

1. Create a new database named studentDB.
2. Create a collection named students.
3. Insert multiple documents with fields like name, age, and grade.
4. Delete the document where name is Bob.
5. Use deleteOne() to remove the document.



14. Retrieve, Query and update Documents

1. Retrieve all documents from the students collection.
2. Query documents where grade is A.
3. Update the age of a student named Alice to 21.
4. Use updateOne() to modify the document.

```
studentDB> db.students.find();
[
  {
    _id: ObjectId('680fcc1dbaa7a3addeb5f899'),
    name: 'Alice',
    age: 22,
    grade: 'A'
  },
  {
    _id: ObjectId('680fcc1dbaa7a3addeb5f89b'),
    name: 'Charlie',
    age: 21,
    grade: 'C'
  },
  {
    _id: ObjectId('680fccbdbaa7a3addeb5f89c'),
    name: 'Alice',
    age: 22,
    grade: 'A'
  },
  {
    _id: ObjectId('680fccbdbaa7a3addeb5f89d'),
    name: 'Bob',
    age: 23,
    grade: 'B'
  },
  {
    _id: ObjectId('680fccbdbaa7a3addeb5f89e'),
    name: 'Charlie',
    age: 21,
    grade: 'C'
  }
]
studentDB> db.students.find({ grade: "A" });
[
  {
    _id: ObjectId('680fcc1dbaa7a3addeb5f899'),
    name: 'Alice',
    age: 22,
    grade: 'A'
  },
  {
    _id: ObjectId('680fccbdbaa7a3addeb5f89c'),
    name: 'Alice',
    age: 22,
    grade: 'A'
  }
]
studentDB> db.students.updateOne(
...   { name: "Alice" },  // Filter condition (find student named Alice)
...   { $set: { age: 21 } }  // Update the age to 21
... );
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
studentDB>
```

15. Create and Apply Indexes for Faster Queries

1. Create an index on the name field for faster search.
2. Verify the created index using getIndexes().

16. Aggregation and Data Analysis

1. Perform aggregation using $match, $group, and $sort.
2. Generate summarized data reports.

```
studentDB> [
...   { name: "Alice", age: 21, grade: "A" },
...   { name: "Bob", age: 23, grade: "B" },
...   { name: "Charlie", age: 21, grade: "C" },
...   { name: "David", age: 22, grade: "A" },
...   { name: "Eva", age: 23, grade: "B" }
... ]
...
[
  { name: 'Alice', age: 21, grade: 'A' },
  { name: 'Bob', age: 23, grade: 'B' },
  { name: 'Charlie', age: 21, grade: 'C' },
  { name: 'David', age: 22, grade: 'A' },
  { name: 'Eva', age: 23, grade: 'B' }
]
studentDB> db.students.aggregate([
...   {
...     $match: { grade: "A" }   // Filter students with grade "A"
...   },
...   {
...     $group: {
...       _id: "$grade",          // Group by grade
...       total_students: { $sum: 1 }   // Count students in each grade
...     }
...   },
...   {
...     $sort: { total_students: -1 }  // Sort by total_students descending
...   }
... ]);
...
[ { _id: 'A', total_students: 2 } ]
studentDB>
```