Assignment 1: Mastering MongoDB: From Basics to Advanced Operations and Schema Validation

Name: Raikhanov Nurbek
Course Group: SE-2312

MongoDB Fundamentals

What is MongoDB?

MongoDB is a NoSQL database that stores data in a flexible, JSON-like document format. It is widely used for modern applications due to its ability to scale horizontally and handle large volumes of unstructured data. Key features include:

- High performance and scalability.
- Schema-less architecture, which allows flexibility in data modeling.
- Support for replication and high availability.
- Compatibility with modern programming languages.

Structure of MongoDB

- **Database**: A container for collections, similar to a schema in relational databases.
- **Collection**: A group of documents, analogous to tables in SQL databases.
- **Document**: A record in a collection, represented in JSON or BSON format.

Data Representation

JSON and BSON Formats

- **JSON (JavaScript Object Notation)**: A lightweight data-interchange format, easy to read and write.
- **BSON (Binary JSON)**: A binary-encoded version of JSON, optimized for storage and retrieval in MongoDB.

Key Differences:

- BSON supports additional data types like `Date` and `Binary`, which JSON does not.
- BSON includes metadata for fast parsing and execution.

Example Document in JSON Format

```
{
    "courseName": "Introduction to MongoDB",
    "courseCode": "MNG101",
    "credits": 3
}
```

- **courseName**: The name of the course (string).
- **courseCode**: The unique code for the course (string).
- **credits**: The number of credits assigned to the course (integer).

CRUD Operations in MongoDB

Definition of CRUD

CRUD stands for:

- **Create**: Add new data to the database.
- **Read**: Retrieve data from the database.
- **Update**: Modify existing data.
- **Delete**: Remove data from the database.

MongoDB Command Examples

Create

```
db.Courses.insertOne({
    courseName: "Database Systems",
    courseCode: "DB101",
    credits: 3
})
```

Read

```
db.Courses.find({ credits: { $gte: 3 } })
```

Update

```
db.Courses.updateOne(
    { courseCode: "WEB102" },
    { $set: { credits: 5 } }
)
```

Delete

```
db.Courses.deleteOne({ courseCode: "ML103" })
```

Schema Validation

What is Schema Validation?

Schema validation in MongoDB enforces rules for the structure of documents within a collection. This ensures data consistency by defining requirements for fields (e.g., type, presence).

Schema Validation in NoSQL vs SQL

- In SQL, schemas are rigid, requiring a fixed table structure.
- In MongoDB, schemas are flexible but can include validation rules for stricter control when needed.

Example: Schema Validation Rules

```
db.createCollection("Courses", {
    validator: {
        $jsonSchema: {
            bsonType: "object",
            required: ["courseName", "courseCode", "credits"],
            properties: {
                courseName: {
                    bsonType: "string",
                    description: "must be a string and is required"
                },
                courseCode: {
                    bsonType: "string",
                    description: "must be a string and is required"
                },
                credits: {
                    bsonType: "int",
                    minimum: 1,
                    description: "must be an integer and is required"
                }
            }
        }
    }
})
```

Practical Component

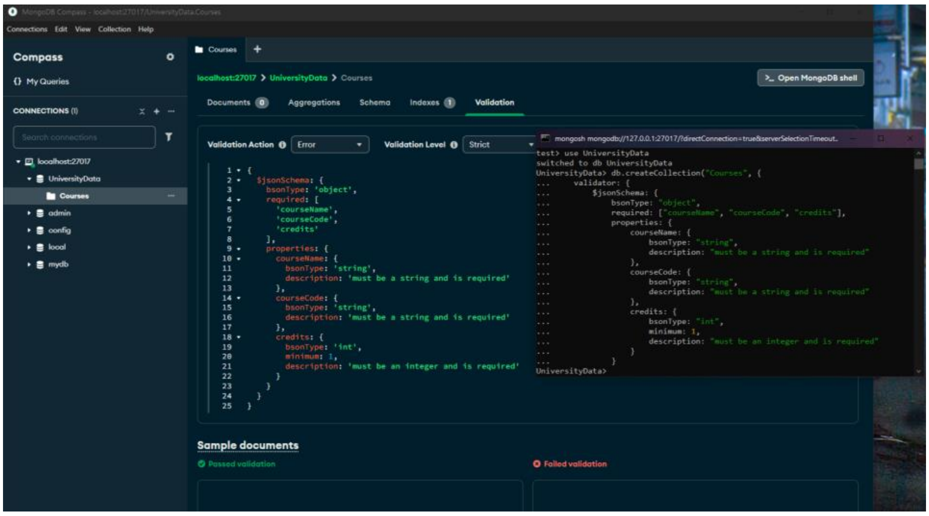
Tasks Completed

1. **Database Creation**:
   - Database Name: `UniversityData`
   - Collection Name: `Courses`
   - Schema Validation Rules Applied.
2. **Sample Documents Inserted**:
   - Document 1: `{ "courseName": "Database Systems", "courseCode": "DB101", "credits": 3 }`
   - Document 2: `{ "courseName": "Web Development", "courseCode": "WEB102", "credits": 4 }`
   - Document 3: `{ "courseName": "Machine Learning", "courseCode": "ML103", "credits": 3 }`
3. **CRUD Operations Executed**:
   - Updated Document: `{ "courseCode": "WEB102" }` to `{ "credits": 5 }`

- Queried Documents: Filtered by `credits >= 3`.
- Deleted Document: `{ "courseCode": "ML103" }`.

Screenshots

- **Database Creation:**



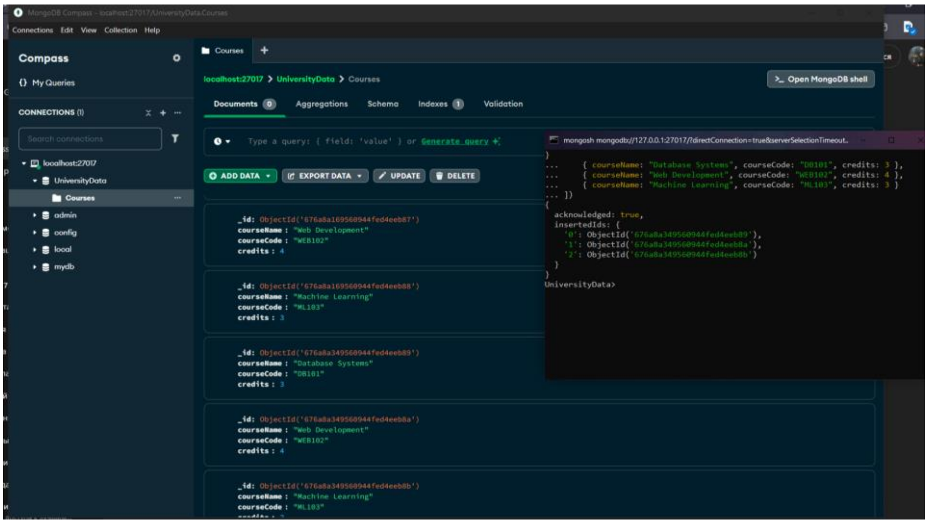
Validation Action: Error
Validation Level: Strict

```
{
  $jsonSchema: {
    bsonType: 'object',
    required: [
      'courseName',
      'courseCode',
      'credits'
    ],
    properties: {
      courseName: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      courseCode: {
        bsonType: 'string',
        description: 'must be a string and is required'
      },
      credits: {
        bsonType: 'int',
        minimum: 1,
        description: 'must be an integer and is required'
      }
    }
  }
}
```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeout…

```
test> use UniversityData
switched to db UniversityData
UniversityData> db.createCollection("Courses", {
...     validator: {
...         $jsonSchema: {
...             bsonType: "object",
...             required: ["courseName", "courseCode", "credits"],
...             properties: {
...                 courseName: {
...                     bsonType: "string",
...                     description: "must be a string and is required"
...                 },
...                 courseCode: {
...                     bsonType: "string",
...                     description: "must be a string and is required"
...                 },
...                 credits: {
...                     bsonType: "int",
...                     minimum: 1,
...                     description: "must be an integer and is required"
...                 }
...             }
UniversityData>
```

- **Inserted Documents:**



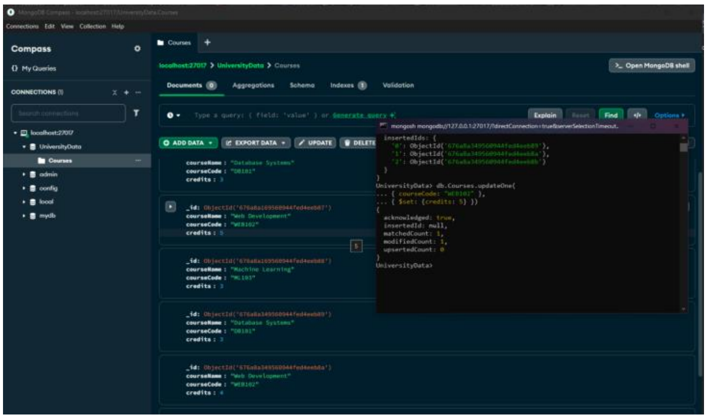
```
_id: ObjectId('676a8a169560944fed4eeb87')
courseName : "Web Development"
courseCode : "WEB102"
credits : 4

_id: ObjectId('676a8a169560944fed4eeb88')
courseName : "Machine Learning"
courseCode : "ML103"
credits : 3

_id: ObjectId('676a8a349560944fed4eeb89')
courseName : "Database Systems"
courseCode : "DB101"
credits : 3

_id: ObjectId('676a8a349560944fed4eeb8a')
courseName : "Web Development"
courseCode : "WEB102"
credits : 4

_id: ObjectId('676a8a349560944fed4eeb8b')
courseName : "Machine Learning"
courseCode : "ML103"
```

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeout…

```
}
...     { courseName: "Database Systems", courseCode: "DB101", credits: 3 },
...     { courseName: "Web Development", courseCode: "WEB102", credits: 4 },
...     { courseName: "Machine Learning", courseCode: "ML103", credits: 3 }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('676a8a349560944fed4eeb89'),
    '1': ObjectId('676a8a349560944fed4eeb8a'),
    '2': ObjectId('676a8a349560944fed4eeb8b')
  }
}
UniversityData>
```

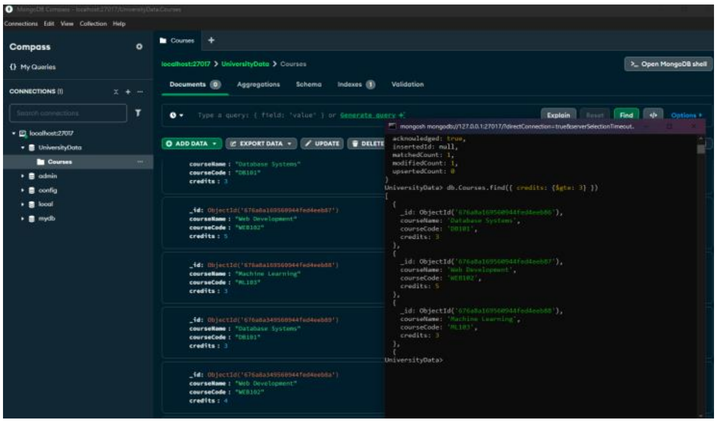
CRUD Operations:

- ### Update:



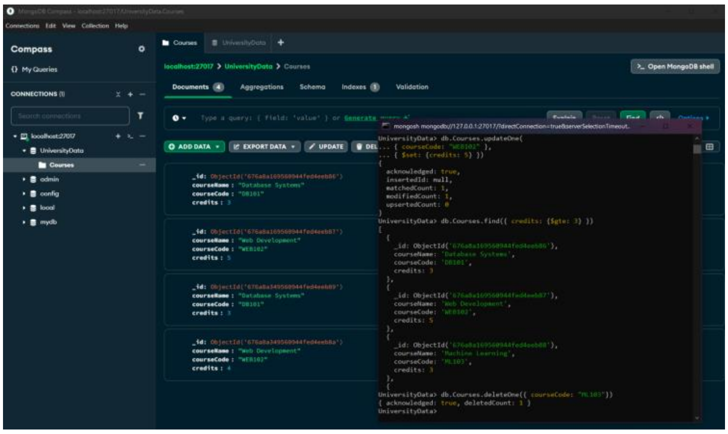
```
  insertedIds: {
    '0': ObjectId('676a8a349560944fed4eeb89'),
    '1': ObjectId('676a8a349560944fed4eeb8a'),
    '2': ObjectId('676a8a349560944fed4eeb8b')
  }
}
UniversityData> db.Courses.updateOne(
... { courseCode: "WEB102" },
... { $set: {credits: 5} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
UniversityData>
```

- ### Query:



```
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
UniversityData> db.Courses.find({ credits: {$gte: 3} })
[
  {
    _id: ObjectId('676a8a169560944fed4eeb86'),
    courseName: 'Database Systems',
    courseCode: 'DB101',
    credits: 3
  },
  {
    _id: ObjectId('676a8a169560944fed4eeb87'),
    courseName: 'Web Development',
    courseCode: 'WEB102',
    credits: 5
  },
  {
    _id: ObjectId('676a8a169560944fed4eeb88'),
    courseName: 'Machine Learning',
    courseCode: 'ML103',
    credits: 3
  },
  {
UniversityData>
```

Delete:



```
UniversityData> db.Courses.updateOne(
... { courseCode: "WEB102" },
... { $set: {credits: 5} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
UniversityData> db.Courses.find({ credits: {$gte: 3} })
[
  {
    _id: ObjectId('676a8a169560944fed4eeb86'),
    courseName: 'Database Systems',
    courseCode: 'DB101',
    credits: 3
  },
  {
    _id: ObjectId('676a8a169560944fed4eeb87'),
    courseName: 'Web Development',
    courseCode: 'WEB102',
    credits: 5
  },
  {
    _id: ObjectId('676a8a169560944fed4eeb88'),
    courseName: 'Machine Learning',
    courseCode: 'ML103',
    credits: 3
  },
  {
UniversityData> db.Courses.deleteOne({ courseCode: "ML103"})
{ acknowledged: true, deletedCount: 1 }
UniversityData>
```