Week 10 - LAQ's

Instructions

Explain the CRUD methods in MongoDB with an example

CRUD stands for Create, Read, Update, and Delete, which are the four basic operations you can perform on a database. In MongoDB, these operations are performed using specific methods. Below is an explanation of each CRUD operation along with examples.

1. Create

The Create operation is used to insert new documents into a collection.

Method: insertOne() or insertMany()

Example:

```
const { MongoClient } = require('mongodb');

async function createDocument() {
    const client = new MongoClient('mongodb://localhost:27017');
    await client.connect();
    const database = client.db('myDatabase');
    const collection = database.collection('users');

    // Insert a single document
    const newUser  = { name: 'Alice', age: 25, email: 'alice@example.com' };
    const result = await collection.insertOne(newUser );
    console.log(`New user created with the following id: ${result.insertedId}`);
```

```
  // Insert multiple documents
  const newUsers = [
    { name: 'Bob', age: 30, email: 'bob@example.com' },
    { name: 'Charlie', age: 35, email: 'charlie@example.com' }
  ];
  const resultMany = await collection.insertMany(newUsers);
  console.log(`${resultMany.insertedCount} new users created.`);

  await client.close();
}

createDocument();
```

2. Read

The Read operation is used to retrieve documents from a collection.

Method: find() or findOne()

Example:

```
async function readDocuments() {
  const client = new MongoClient('mongodb://localhost:27017');
  await client.connect();
  const database = client.db('myDatabase');
  const collection = database.collection('users');

  // Find all users
  const users = await collection.find({}).toArray();
```

```
    console.log('All users:', users);

    // Find a single user
    const user = await collection.findOne({ name: 'Alice' });
    console.log('User  found:', user);

    await client.close();
}

readDocuments();
```

3. Update

The Update operation is used to modify existing documents in a collection.

Method: updateOne(), updateMany(), or replaceOne()

Example:

```
async function updateDocument() {
    const client = new MongoClient('mongodb://localhost:27017');
    await client.connect();
    const database = client.db('myDatabase');
    const collection = database.collection('users');

   // Update a single document
    const result = await collection.updateOne(
        { name: 'Alice' }, // Filter
        { $set: { age: 26 } } // Update
    );
    console.log(`${result.modifiedCount} document(s) updated.`);

    // Update multiple documents
   const resultMany = await collection.updateMany(
        { age: { $gte: 30 } }, // Filter
        { $set: { status: 'senior' } } // Update
    );
    console.log(`${resultMany.modifiedCount} documents updated to senior status.`);

    await client.close();
}

updateDocument();
```

4. Delete

The Delete operation is used to remove documents from a collection.

Method: deleteOne() or deleteMany()

Example:

```
async function deleteDocument() {
    const client = new MongoClient('mongodb://localhost:27017');
    await client.connect();
    const database = client.db('myDatabase');
    const collection = database.collection('users');
```

```
  // Delete a single document
  const result = await collection.deleteOne({ name: 'Alice' });
  console.log(`${result.deletedCount} document(s) deleted.`);
  // Delete multiple documents
  const resultMany = await collection.deleteMany({ age: { $lt: 30 } });
  console.log(`${resultMany.deletedCount} documents deleted with age less than 30.`);
  await client.close();
}

deleteDocument();
```

Summary

- Create: Use insertOne() or insertMany() to add new documents.
- Read: Use find() or findOne() to retrieve documents.
- Update: Use updateOne(), updateMany(), or replaceOne() to modify existing documents.
- Delete: Use deleteOne() or deleteMany() to remove documents.

These CRUD operations are fundamental for managing data in a MongoDB database.