**1. Creating the MongoDB Collection:**

JavaScript

// Assuming you're using the MongoDB Node.js driver (mongodb)

const MongoClient = require('mongodb').MongoClient;

const uri = "mongodb://localhost:27017"; // Replace with your MongoDB connection string

async function createCollection() {

const client = new MongoClient(uri);

try {

await client.connect();

const database = client.db("your\_database\_name");

const collection = database.collection("students");

// Insert sample data

const data = [

{ student\_id: 1, first\_name: "John", last\_name: "Doe", age: 20, grade: "A", major: "Computer Science" },

// ... other student data

];

const result = await collection.insertMany(data);

console.log(`${result.insertedCount} documents were inserted`);

} finally {

await client.close();

}

}

createCollection();

Use code [with caution.](https://d.docs.live.net/faq#coding)

**2. MongoDB Queries:**

JavaScript

// Assuming you have a MongoDB client connected to the database

**// 1. Find All Students**

const allStudents = await collection.find().toArray();

**// 2. Find Students in a Specific Major (Computer Science)**

const computerScienceStudents = await collection.find({ major: "Computer Science" }).toArray();

**// 3. Count the Number of Students in Each Major**

const majorCounts = await collection.aggregate([

{ $group: { \_id: "$major", count: { $sum: 1 } } }

]).toArray();

**// 4. Find Students with a Specific Grade (A)**

const gradeAStudents = await collection.find({ grade: "A" }).toArray();

**// 5. Count the Number of Students in Each Grade Having More Than 2 Students**

const gradesWithManyStudents = await collection.aggregate([

{ $group: { \_id: "$grade", count: { $sum: 1 } } },

{ $match: { count: { $gt: 2 } } }

]).toArray();

**// 6. List Students Ordered by Age**

const studentsByAge = await collection.find().sort({ age: 1 }).toArray();

**// 7. Update a Student's Major (Emily to Mathematics)**

const updateMajor = await collection.updateOne({ first\_name: "Emily" }, { $set: { major: "Mathematics" } });

**// 8. Find the Oldest Student**

const oldestStudent = await collection.find().sort({ age: -1 }).limit(1).toArray();

**// 9. Find the Youngest Student**

const youngestStudent = await collection.find().sort({ age: 1 }).limit(1).toArray();

**// 10. Delete a Student Record by student\_id=6**

const deleteStudent = await collection.deleteOne({ student\_id: 6 });