// Step 1: Create Database and Insert Dummy Data

use myDatabase

// Insert sample student documents

db.students.insertMany([

{ name: "Raj", age: 22, subjects: ["Math", "Science"], graduated: false },

{ name: "Priya", age: 24, subjects: ["History", "Literature"], graduated: true },

{ name: "Anil", age: 23, subjects: ["Math", "Literature"], graduated: false },

{ name: "Sita", age: 25, subjects: ["Science", "Math"], graduated: true }

]);

// Insert a new student document (Reema)

db.students.insertOne({ name: "Reema", age: 21, subjects: ["Math", "Science"], graduated: false });

// Step 2: Remove Documents

// Remove the document for the student named "Raj"

db.students.deleteOne({ name: "Raj" });

// Remove all students who have not graduated

db.students.deleteMany({ graduated: false });

// Step 3: Update Documents

// Replace Priya's document with new subjects

db.students.updateOne(

{ name: "Priya" },

{ $set: { subjects: ["History", "Philosophy"] } }

);

// Update Anil's graduation status to true without replacing the document

db.students.updateOne(

{ name: "Anil" },

{ $set: { graduated: true } }

);

// Perform an upsert operation for Reema

db.students.updateOne(

{ name: "Reema" },

{ $set: { age: 21, subjects: ["Math", "Science"], graduated: false } },

{ upsert: true }

);

// Update all documents to add a field hasMath for students with "Math" in their subjects

db.students.updateMany(

{ subjects: { $in: ["Math"] } },

{ $set: { hasMath: true } }

);

// Update Sita's graduation status to false

db.students.updateOne(

{ name: "Sita" },

{ $set: { graduated: false } }

);

// Step 4: Execute Queries

// Find and display all students

db.students.find();

// Find the document for "Anil"

db.students.find({ name: "Anil" });

// Retrieve all students older than 22

db.students.find({ age: { $gt: 22 } });

// Retrieve students who are either "Sita" or 23 years old

db.students.find({ $or: [{ name: "Sita" }, { age: 23 }] });

// Retrieve students who are not older than 23

db.students.find({ age: { $lte: 23 } });

// Retrieve students whose age is not 22

db.students.find({ age: { $ne: 22 } });

// Retrieve documents where the nickname field does not exist

db.students.find({ nickname: { $exists: false } });

// Find students whose names end with the letter "a"

db.students.find({ name: { $regex: /a$/, $options: 'i' } });

// Find students who have "Math" as one of their subjects

db.students.find({ subjects: { $in: ["Math"] } });

// Use $where to find students older than 22

db.students.find({ $where: "this.age > 22" });

// Limit the number of results to 2

db.students.find().limit(2);

// Skip the first document and retrieve the rest

db.students.find().skip(1);

// Sort the documents by age in ascending order

db.students.find().sort({ age: 1 });

// Advanced query: Find students older than 21, limit to 2, skip the first, and sort by name

db.students.find({ age: { $gt: 21 } }).skip(1).limit(2).sort({ name: 1 });