1.Create database named: FacultySystemDB.

use FacultySystemDB

2.Create collection (student) that has:

•FirstName: string

•LastName: string

•Age: Number

•Faculty: An object that has Name and Address

•Grades: An array of objects, each object has: CourseName, Grade, Pass (Boolean).

•IsFired: Boolean

db.createCollection**(**"student"**)**

db.student.insertOne**({**

FirstName**:**"Ali"**,**

LastName**:**"Galal"**,**

age**:**25**,**

Faculty**:{**Name**:**"Engineering"**,**Address**:**"cairo"**},**

Grades**:[{**CourseName**:**"Programming"**,**Grade**:**55**,**Pass**:**true**},**

**{**CourseName**:**"oop"**,**Grade**:**60**,**Pass**:**true**}],**

IsFired**:**false

**})**

3.Insert 3 (at least) documents in Student collection with different values.

* + Try inserting one record each time (((using insertOne)))

db.student.insertOne**({**

FirstName**:**"Ahmed"**,**

LastName**:**"Selim"**,**

age**:**24**,**

Faculty**:{**Name**:**"Education"**,**Address**:**"alex"**},**

Grades**:[{**CourseName**:**"Teaching Ways"**,**Grade**:**30**,**Pass**:**false**},**

**{**CourseName**:**"Learning Sycology"**,**Grade**:**70**,**Pass**:**true**},**

**{**CourseName**:**"Anatomy"**,**Grade**:**35**,**Pass**:**false**}],**

IsFired**:**true

**})**

* + Try inserting collection using on insert statement.(((using insertMany)))

db.student.insertMany**([{**

FirstName**:**"Ola"**,**

LastName**:**"Allam"**,**

age**:**24**,**

Faculty**:{**Name**:**"Science"**,**Address**:**"alex"**},**

Grades**:[{**CourseName**:**"Physics"**,**Grade**:**50**,**Pass**:**true**},**

**{**CourseName**:**"plants"**,**Grade**:**70**,**Pass**:**true**},**

**{**CourseName**:**"Geology of Rocks"**,**Grade**:**90**,**Pass**:**true**}],**

IsFired**:**true

**},**

**{**

FirstName**:**"Mai"**,**

LastName**:**"Mohamed"**,**

age**:**24**,**

Faculty**:{**Name**:**"Medicine"**,**Address**:**"menofia"**},**

Grades**:[{**CourseName**:**"MBBS"**,**Grade**:**30**,**Pass**:**false**},**

**{**CourseName**:**"BAMS"**,**Grade**:**65**,**Pass**:**true**},**

**{**CourseName**:**"BHMS"**,**Grade**:**55**,**Pass**:**true**}],**

IsFired**:**false

**},**

**{**

FirstName**:** "Ali"**,**

LastName**:** "Emad"**,**

age**:** 19**,**

Faculty**:** **{** Name**:** "Engineering"**,** Address**:** "cairo" **},**

Grades**:** **[{** CourseName**:** "Programming"**,** Grade**:** 85**,** Pass**:** true **},**

**{** CourseName**:** "oop"**,** Grade**:** 90**,** Pass**:** true **}],**

IsFired**:** false

**}])**

4. Retrieve the following data:

* + All Students.

**db**.**student**.**find()**.**pretty()**

* + Student with specific First Name.

**db**.**student**.**find({},{**FirstName**:**1**})**

* + Students who his First Name=Ahmed, or Last Name= Ahmed.

**db**.**student**.**find({**$or**:[{**FirstName**:**"Ahmed"**},{**LastName**:**"Ahmed"**}]})**

* + Students that their First name isn't "Ahmed".

**db**.**student**.**find({**FirstName**:{**$ne**:**"Ahmed"**}})**

* + Students with Age less than 21.

**db**.**student**.**find({**age**:{**$lt**:**21**}})**

* + All fired students.

**db**.**student**.**find({**IsFired**:**true**})**

* + Students with Age more than or equal to 21, and their faculty isn't NULL.

**db**.**student**.**find({**$and**:[{**age**:{**$gte**:**21**}},{**Faculty**:{**$ne**:**null**}}]})**

* + Display student with specific First Name, and display his First Name, Last name, IsFired fields only.

**db**.**student**.**find({**FirstName**:**"Ola"**},{**FirstName**:**1**,**LastName**:**1**,**IsFired**:**1**,**\_id**:**0**})**

5. Update the student with specific FirstName, and change his LastName.

* + Try Update() statement.

db.student.updateOne**({**FirstName**:**"Mai"**},{**$set**:{**LastName**:**"Omran"**}})**

* + Try Update() with Mulit option.

db.student.updateMany**({**FirstName**:**"Ali"**},**

**{**$set**:[{**LastName**:**"Mostafa"**},{**"Grades.1.Grade"**:**69**}]})**

* + Try Save().

db.student.save**({**

FirstName**:** "Nada"**,**

LastName**:** "Mourad"**,**

age**:** 19**,**

Faculty**:** **{** Name**:** "Pharmacy"**,** Address**:** "alex" **},**

Grades**:** **[{** CourseName**:** "Research science"**,** Grade**:** 45**,** Pass**:** false **},**

**{** CourseName**:** "Clinical research associate"**,** Grade**:** 70**,** Pass**:** true **},**

**{** CourseName**:** "Pharm.D"**,** Grade**:** 60**,** Pass**:** true **}],**

IsFired**:** false

**},{** writeConcern**:** **{** w**:** "majority"**,** wtimeout**:** 5000 **}** **})**

1. Delete Fired students.

db.student.deleteMany**({**IsFired**:**true**})**

1. Delete all students collection.

db.student.deleteMany**({})**

1. Delete the whole DB.

db.dropDatabase**()**

1. Create new database named: FacultySystemV2.

use FacultySystemV2

* + Create student collection that has (FirstName, lastName, IsFired, FacultyID, array of objects, each object has CourseID, grade).

db.createCollection**(**"student"**)**

* + Create Faculty collection that has (Faculty Name, Address).

db.createCollection**(**"faculty"**)**

* + Create Course collection, which has (Course Name, Final Mark).

db.createCollection**(**"course"**)**

* + Insert some data in previous collections.

**//insert data in student collection**

db.student.insertMany**([**

**{**FirstName**:**"Omar"**,**

LastName**:**"Ahmed"**,**

IsFired**:**false**,**

FacultyId**:**707**,**

Courses**:[{**CourseId**:**8899**,**Grade**:**75**},**

**{**CourseId**:**5566**,**Grade**:**43**},**

**{**CourseId**:**2233**,**Grade**:**69**}]**

**},**

**{**FirstName**:**"Malak"**,**

LastName**:**"Salah"**,**

IsFired**:**true**,**

FacultyId**:**606**,**

Courses**:[{**CourseId**:**5566**,**Grade**:**54**},**

**{**CourseId**:**1122**,**Grade**:**73**},**

**{**CourseId**:**7744**,**Grade**:**84**}]**

**},**

**{**FirstName**:**"Yassin"**,**

LastName**:**"Elsayed"**,**

IsFired**:**false**,**

FacultyId**:**808**,**

Courses**:[{**CourseId**:**2244**,**Grade**:**95**},**

**{**CourseId**:**8899**,**Grade**:**83**},**

**{**CourseId**:**7744**,**Grade**:**79**}]}])**

**//insert data in faculty collection**

db.faculty.insertMany**([**

**{**FacultyId**:**707**,**FacultyName**:**"Education"**,**Address**:**"Assuit"**},**

**{** FacultyId**:**606**,**FacultyName**:**"Science"**,**Address**:**"Aswan"**},**

**{** FacultyId**:**808**,**FacultyName**:**"Engineering"**,**Address**:**"Cairo"**},**

**{** FacultyId**:**909**,**FacultyName**:**"Medicine"**,**Address**:**"Ain Shams"**}**

**])**

**//insert some data in Course collection**

db.course.insertMany**([**

**{**CourseId**:** 2244**,**Coursename**:**"OOP"**,**FinalMark**:**100**},**

**{**CourseId**:**5566**,**Coursename**:**"Linq"**,**FinalMark**:**150**},**

**{**CourseId**:**1122**,**Coursename**:**"Angular"**,**FinalMark**:**150**},**

**{**CourseId**:**2233**,**Coursename**:**"JS"**,**FinalMark**:**100**},**

**{**CourseId**:**8899**,**Coursename**:**"MongoDB"**,**FinalMark**:**100**},**

**{**CourseId**:**7744**,**Coursename**:**"SQL"**,**FinalMark**:**100**}])**

**Bonus:**

1. Install Redis database, and try to insert and select data from it.
2. Retrieve and insert to MongoDB using a nodeJs

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

var url = "mongodb://localhost:27017/";

MongoClient.connect(url, function(err, db) {

  if (err) throw err;

  var dbo = db.db("FacultySystemV2");

  var obj = { FirstName:"Huda", LastName:"Ahmed",age:21 };

  dbo.collection("student").insertOne(obj, function(err, res) {

    if (err) throw err;

    console.log("1 document inserted");

    db.close();

  });

});

1. Display each student along with his grades in course he studied from FacultySystemV2.

**db**.**student**.**aggregate([**

**{**

$lookup**:**

**{**

from**:** 'course'**,**

localField**:** 'Courses.CourseId'**,**

foreignField**:** 'CourseId'**,**

as**:** 'CourseInfo'

**}**

**},**

**{**

$project**:** **{**

FirstName**:** 1**,**

LastName**:**1**,**

'Courses.Grade'**:** 1**,**

CourseInfo**:**1**,**

\_id**:**0

**}**

**}**

**])**.**pretty();**

1. Add your own key to any collection.

db.faculty.createIndex**({**FacultyId**:**1**},{**unique**:**true**})**