**Day 1 Lab Assignments**

**Use MongoDB to do the following assignments:**

1. Create database named: FacultySystemDB.

use FacultySystemDB

1. 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.student.insert**({**

"FirstName"**:**"Nouran"**,**

"LastName"**:**"El Mohamady"**,**

"Age"**:**23**,**

"Faculty"**:{**

"Name"**:**"Faculty of Engineering"**,**

"Address"**:**"Alexandria"

**},**

"Grades"**:[**

**{**"CourseName"**:**"Math"**,**"Grade"**:**50**,**"Pass"**:**true**},**

**{**"CourseName"**:**"electronics"**,**"Grade"**:**30**,**"Pass"**:**false **},**

**{**"CourseName"**:**"programming"**,**"Grade"**:**70**,**"Pass"**:** true **}],**

"IsFired"**:**true

**})**

1. Insert 3 (at least) documents in Student collection with different values.
   * Try inserting one record each time

db.student.insertOne**(**

**{**

"FirstName"**:**"Nadeen"**,**

"LastName"**:**"El Mohamady"**,**

"Age"**:**21**,**

"Faculty"**:{**

"Name"**:**"Faculty of Science"**,**

"Address"**:**"Alexandria"

**},**

"Grades"**:[**

**{**"CourseName"**:**"physics"**,**"Grade"**:**80**,**"Pass"**:**true**},**

**{**"CourseName"**:**"geo"**,**"Grade"**:**50**,**"Pass"**:**false **},**

**{**"CourseName"**:**"programming"**,**"Grade"**:**70**,**"Pass"**:** true **}],**

"IsFired"**:**false

**})**

* + Try inserting collection using on insert statement.

db.student.insertMany**([**

**{**

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

"LastName"**:**"El Mohamady"**,**

"Age"**:**20**,**

"Faculty"**:{**

"Name"**:**"Faculty of Enfineering"**,**

"Address"**:**"Alexandria"

**},**

"Grades"**:[**

**{**"CourseName"**:**"Math"**,**"Grade"**:**80**,**"Pass"**:**true**},**

**{**"CourseName"**:**"php"**,**"Grade"**:**95**,**"Pass"**:**true **},**

**{**"CourseName"**:**"c"**,**"Grade"**:**70**,**"Pass"**:** true **}],**

"IsFired"**:**true

**},**

**{**

"FirstName"**:**"Mohamed"**,**

"LastName"**:**"salah"**,**

"Age"**:**26**,**

"Faculty"**:{**

"Name"**:**"Faculty of archticture"**,**

"Address"**:**"Kafer el shakh"

**},**

"Grades"**:[**

**{**"CourseName"**:**"archticture"**,**"Grade"**:**100**,**"Pass"**:**true**},**

**{**"CourseName"**:**"Math"**,**"Grade"**:**60**,**"Pass"**:**true **},**

**{**"CourseName"**:**"programming"**,**"Grade"**:**70**,**"Pass"**:** true **}],**

"IsFired"**:**false

**}**

**])**

1. Retrieve the following data:
   * All Students.

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

* + Student with specific First Name.

**db.student.find({},{\_id:0,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: { $eq: true } } )**

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

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

1. Update the student with specific FirstName, and change his LastName.
   * Try Update() statement.

**db.student.updateOne({FirstName:"Nouran"},{$set:{FirstName:"Nour"}})**

* + Try Update() with Mulit option.

**db.student.updateMany({FirstName:"Ahmed"},{$set:{FirstName:"Aya","Faculty.Name":"Medecin"}})**

* + Try Save().

**db.student.save({**

**"FirstName": "Nermeen",**

**"LastName": "Khaled",**

**"age": 19,**

**"Faculty": {**

**"Name": "Faculty of Pharmacy",**

**"Address": "Alexandria"**

**},**

**"Grades": [**

**{ "CourseName": "technical report", "Grade": 80, "Pass": true },**

**{ "CourseName": "math2", "Grade": 95, "Pass": true },**

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

**],**

**"IsFired": false**

**})**

1. Delete Fired students.

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

1. Delete all students collection.

db.student.drop**()**

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.student.insert([{

"FirstName":"Nouran",

"LastName":"El Mohamady",

"FacultyID":238,

"cources":[

{"CourseID":2001,"Grade":50},

{"CourseID":2002,"Grade":30,},

{"CourseID":2003,"Grade":70,}

],

"IsFired":true

},

{

"FirstName":"nadeen",

"LastName":"El Mohamady",

"FacultyID":300,

"cources":[

{"CourseID":100,"Grade":100},

{"CourseID":101,"Grade":50,},

{"CourseID":102,"Grade":70,}

],

"IsFired":false

}

])

* + Create Faculty collection that has (Faculty Name, Address).
  + Create Course collection, which has (Course Name, Final Mark).
  + Insert some data in previous collections.

db.Faculty.insert([

{

"Name": "Faculty of Engineering",

"Address": "Alexandria"

},

{

"Name": "Faculty of archticture",

"Address": "Kafer el shakh"

},

])

db.course.insert([

{

"courseName":"English",

"FinalMark":100

},

{

"courseName":"arabic",

"FinalMark":70

}

])

**Bonus:**

1. Install Redis database, and try to insert and select data from it.
2. Retrieve and insert to MongoDB using a nodeJs
3. Display each student along with his grades in course he studied from FacultySystemV2.
4. Add your own key to any collection.

**<Script>document.write(“Thank YOU”) </Script>**