**MongoDB Assignment 1**

**Ques:1 Create a new database called student\_management.**

Ans: use student\_management

switched to db student\_management

**Ques:2 Create a collection called students in the student\_management database.**

**Query:** student\_management> db.createCollection("students")

{ ok: 1 }

**Ques:3 Insert at least five student records into the students collection. Each record should have the following fields:**

**student\_id (integer)**

**name (string)**

**age (integer)**

**department (string)**

**courses (array of strings)**

**grade (string)**

**Query:** db.students.insertMany([

{

student\_id: 1,

name: "Alice Johnson",

age: 20,

department: "Computer Science",

courses: ["Data Structures", "Algorithms", "Databases"],

grade: "A"

},

{

student\_id: 2,

name: "Bob Smith",

age: 22,

department: "Mathematics",

courses: ["Calculus", "Linear Algebra", "Statistics"],

grade: "B+"

},

{

student\_id: 3,

name: "Charlie Brown",

age: 19,

department: "Physics",

courses: ["Quantum Mechanics", "Classical Mechanics", "Electromagnetism"],

grade: "A-"

},

{

student\_id: 4,

name: "Diana Lopez",

age: 21,

department: "Biology",

courses: ["Genetics", "Microbiology", "Biochemistry"],

grade: "B"

},

{

student\_id: 5,

name: "Ethan White",

age: 23,

department: "Chemistry",

courses: ["Organic Chemistry", "Analytical Chemistry", "Physical Chemistry"],

grade: "B+"

}

]);

**Ques:4 Query the Collection:**

**Write queries to perform the following tasks:**

**1. Retrieve all students who are in the "Computer Science" department.**

**Qurey:** student\_management> db.students.find({department:"Computer Science"})

**Ans:** [

{

\_id: ObjectId('672fa91a54a23b23972710bd'),

student\_id: 1,

name: 'Alice johnson',

age: 20,

department: 'Computer Science',

courses: [ 'Data Structures', 'Algorithms', 'Databases' ],

grade: 'A'

}

]

**2.Retrieve students who have an age greater than 21.**

**Query:** student\_management> db.students.find({age:{$gt:21}})

**Ans:** [

{

\_id: ObjectId('672fa9f854a23b23972710be'),

student\_id: 2,

name: 'Bob Smith',

age: 22,

department: 'Mathematics',

courses: [ 'Calculus', 'Linear Algebra', 'Statistics' ],

grade: 'B+'

},

{

\_id: ObjectId('672faea454a23b23972710c2'),

student\_id: 5,

name: 'Ethan White',

age: 23,

department: 'Chemistry',

courses: [

'Organic Chemistry',

'Analytical Chemistry',

'Physical Chemistry'

],

grade: 'B+'

}

]

**3.Retrieve students who are taking the "Data Structures" course.**

**Query:** student\_management> db.students.find({courses:"Data Structures"})

**Ans: [** {

\_id: ObjectId('672fa91a54a23b23972710bd'),

student\_id: 1,

name: 'Alice johnson',

age: 20,

department: 'Computer Science',

courses: [ 'Data Structures', 'Algorithms', 'Databases' ],

grade: 'A'

}

]

**4.Retrieve students with a grade of "A".**

Ans:student\_management> db.students.find({grade:"A"})

[

{

\_id: ObjectId('672fa91a54a23b23972710bd'),

student\_id: 1,

name: 'Alice johnson',

age: 20,

department: 'Computer Science',

courses: [ 'Data Structures', 'Algorithms', 'Databases' ],

grade: 'A'

}

]

**Ques:5 Update Documents:**

**1.Update the age of a student with student\_id 101 to 21.**

Ans:student\_management> db.students.updateOne({student\_id:1},{$set:{age:21}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

student\_management> db.students.find({})

[

{

\_id: ObjectId('672fa91a54a23b23972710bd'),

student\_id: 1,

name: 'Alice johnson',

age: 21,

department: 'Computer Science',

courses: [ 'Data Structures', 'Algorithms', 'Databases' ],

grade: 'A'

},

**2.Add a new course, "Machine Learning", to the courses array for students in the "Computer Science" department.**

Ans: student\_management> db.students.updateMany({department:"Computer Science"},{$addToSet:{courses:"Machine Learning"}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

student\_management> db.students.find({})

[

{

\_id: ObjectId('672fa91a54a23b23972710bd'),

student\_id: 1,

name: 'Alice johnson',

age: 21,

department: 'Computer Science',

courses: [

'Data Structures',

'Algorithms',

'Databases',

'Machine Learning'

],

grade: 'A'

}

**Ques:6 Delete Documents:**

**1.Delete a student record with student\_id 105.**

Ans:student\_management> db.students.deleteOne({student\_id:5})

{ acknowledged: true, deletedCount: 1 }

**2. Delete all students who have a grade lower than "C".**

Ans: student\_management> db.students.deleteMany({grade:{$lt:"C"}})

{ acknowledged: true, deletedCount: 4 }