**Assignment -2**

**Ques:1 Create a database named university and a collection named students. Insert multiple student documents with fields: name, age, department, and grades.**

**latest\_db1> use University**

**switched to db University**

University> db.createCollection("Students")

{ ok: 1 }

University> db.Students.insertOne({name:"Alice",age:20,department:"Computer Science",grades:{math:85,english:92}})

{

acknowledged: true,

insertedId: ObjectId('672f087b50b6eeaf4a2710bc')

}

University> db.Students.find({})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 20,

department: 'Computer Science',

grades: { math: 85, english: 92 }

}

]

University> db.Students.insertOne({name:"Bob",age:21,department:"Physics",grades:{math:88,physics:90}})

{

acknowledged: true,

insertedId: ObjectId('672f08fd50b6eeaf4a2710bd')

}

University> db.Students.insertOne({name:"Charlie",age:22,department:"Mathematics",grades:{math:95,statistics:89}})

{

acknowledged: true,

insertedId: ObjectId('672f095550b6eeaf4a2710be')

}

University> db.Students.find({})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 20,

department: 'Computer Science',

grades: { math: 85, english: 92 }

},

{

\_id: ObjectId('672f08fd50b6eeaf4a2710bd'),

name: 'Bob',

age: 21,

department: 'Physics',

grades: { math: 88, physics: 90 }

},

{

\_id: ObjectId('672f095550b6eeaf4a2710be'),

name: 'Charlie',

age: 22,

department: 'Mathematics',

grades: { math: 95, statistics: 89 }

}

]

**Ques.2 Write a query to display all students who are in the Computer Science department.**

Ans: University> db.Students.find({department:"Computer Science"})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 20,

department: 'Computer Science',

grades: { math: 85, english: 92 }

}

]

**Ques:3 Write a query to update the grades of a student named Alice by adding a new subject programming with a grade of 93.**

Ans: University> db.Students.updateOne({name:"Alice"},{$set:{grades:{programming:93}}})

University> db.Students.find({})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 20,

department: 'Computer Science',

grades: { programming: 93 }

}

**Ques:4 Write a query to increment the age of all students by 1.**

Ans: University> db.Students.updateMany({},{$inc:{age:1}})

{

acknowledged: true,

insertedId: null,

matchedCount: 3,

modifiedCount: 3,

upsertedCount: 0

}

University> db.Students.find({})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 21,

department: 'Computer Science',

grades: { programming: 93 }

},

{

\_id: ObjectId('672f08fd50b6eeaf4a2710bd'),

name: 'Bob',

age: 22,

department: 'Physics',

grades: { math: 88, physics: 90 }

},

{

\_id: ObjectId('672f095550b6eeaf4a2710be'),

name: 'Charlie',

age: 23,

department: 'Mathematics',

grades: { math: 95, statistics: 89 }

}

]

**Ques:5 Write a query to delete all students who are 23 years old.**

Ans: University> db.Students.deleteMany({age:23})

{ acknowledged: true, deletedCount: 1 }

University> db.Students.find({})

[

{

\_id: ObjectId('672f087b50b6eeaf4a2710bc'),

name: 'Alice',

age: 21,

department: 'Computer Science',

grades: { programming: 93 }

},

{

\_id: ObjectId('672f08fd50b6eeaf4a2710bd'),

name: 'Bob',

age: 22,

department: 'Physics',

grades: { math: 88, physics: 90 }

}

]

**Ques:6 Write a query to create an index on the name field of the students collection.**

Ans: University> db.Students.createIndex({name:1})

name\_1

**Ques:7 Write an aggregation query to group students by their department and calculate the average age in each department.**

Ans: University> db.Students.aggregate([{$group:{\_id:"$department",averageAge:{$avg:"$age"}}}])

[

{ \_id: 'Computer Science', averageAge: 21 },

{ \_id: 'Physics', averageAge: 22 }

]

**Ques:8 Write a query to find all students who have scored more than 90 in any subject.**

Ans: University> db.Students.find({grades:{$gt:90}})

**Ques:9 Write a query to add a new field graduated set to false for all students who are in the Mathematics department.**

Ans: University> db.Students.updateMany({department:"Mathematics"},{$set:{graduated:false}})

**Ques:10 How can you retrieve only the name and department fields for all students, excluding the \_id field?**

Ans: University> db.Students.find({},{name:1,department:1,\_id:0})

[

{ name: 'Alice', department: 'Computer Science' },

{ name: 'Bob', department: 'Physics' }

]