**Step 1: Setting Up MongoDB**

* Make sure you have MongoDB installed on your system and that the MongoDB server is running.
* Use mongo shell to interact with MongoDB.

**Step 2: Create Database DYPIT**

Code:

use DYPIT

This command switches to the DYPIT database. If it doesn’t exist, it will be created.

**Step 3: Create Teachers and Students Collections**

1. **Inserting documents into Teachers collection**:

Code:

db.Teachers.insertMany([

{ Tname: "Praveen", dno: 1, dname: "COMP", experience: "8 years", salary: 12000, date\_of\_joining: new Date("2015-06-20") },

{ Tname: "Rajesh", dno: 2, dname: "IT", experience: "5 years", salary: 15000, date\_of\_joining: new Date("2018-07-15") },

{ Tname: "Suresh", dno: 3, dname: "E&TC", experience: "6 years", salary: 9000, date\_of\_joining: new Date("2017-03-10") },

{ Tname: "Anita", dno: 4, dname: "COMP", experience: "4 years", salary: 11000, date\_of\_joining: new Date("2019-08-01") }

])

1. **Inserting documents into Students collection**:

Code:

db.Students.insertMany([

{ Sname: "John", roll\_no: 1, class: "FY" },

{ Sname: "xyz", roll\_no: 2, class: "SY" },

{ Sname: "Alice", roll\_no: 3, class: "TY" }

])

**Step 4: Queries to Fetch Information**

1. **Find information about all teachers**:

Code:

db.Teachers.find().pretty()

1. **Find information about all teachers of the Computer department**:

Code:

db.Teachers.find({ dname: "COMP" }).pretty()

1. **Find information about all teachers of the Computer, IT, and E&TC departments**:

Code:

db.Teachers.find({ dname: { $in: ["COMP", "IT", "E&TC"] } }).pretty()

1. **Find teachers of Computer, IT, and E&TC departments with salary >= 10000**:

Code:

db.Teachers.find({

dname: { $in: ["COMP", "IT", "E&TC"] },

salary: { $gte: 10000 }

}).pretty()

1. **Find student information having roll\_no = 2 or Sname = "xyz"**:

Code:

db.Students.find({

$or: [{ roll\_no: 2 }, { Sname: "xyz" }]

}).pretty()

**Step 5: Updating Documents**

1. **Update experience of teacher "Praveen" to 10 years (or insert if not present)**:

Code:

db.Teachers.updateOne(

{ Tname: "Praveen" },

{ $set: { experience: "10 years" } },

{ upsert: true }

)

1. **Update the department of all teachers in the IT department to COMP**:

Code:

db.Teachers.updateMany(

{ dname: "IT" },

{ $set: { dname: "COMP" } }

)

**Step 6: Projection and Inserting with save() Method**

1. **Find teachers' names and experience**:

Code:

db.Teachers.find({}, { Tname: 1, experience: 1, \_id: 0 }).pretty()

1. **Insert one entry into the Teachers collection using save() method**:

Code:

db.Teachers.save({

Tname: "NewTeacher", dno: 5, dname: "Math", experience: "2 years", salary: 8000, date\_of\_joining: new Date("2023-01-01")

})

1. **Change the department of teacher Rajesh to IT using save() method**:

Code:

let teacher = db.Teachers.findOne({ Tname: "Rajesh" });

if (teacher) {

teacher.dname = "IT";

db.Teachers.save(teacher);

}

**Step 7: Deleting Documents**

1. **Delete all documents from Teachers collection having IT department**:

Code:

db.Teachers.deleteMany({ dname: "IT" })

**Step 8: Display Documents Using pretty() Method**

1. **Display the first 3 documents in ascending order using pretty()**:

Code:

db.Teachers.find().sort({ Tname: 1 }).limit(3).pretty()

**How to Run and Check Output**

1. **Start MongoDB Server**:
   * Run mongod in your terminal to start the MongoDB server.
2. **Open MongoDB Shell**:
   * Run mongo in a new terminal to open the MongoDB shell.
3. **Run the above commands step by step**.
   * Use .find().pretty() to get neatly formatted output.
4. **Check the Output**:
   * After running each query, you should see the expected results printed in the shell.