**Step 1: Create Database DYPIT**

javascript

Copy code

use DYPIT

* **Explanation**: This command switches to the DYPIT database. If it does not exist, it will be created.

**Step 2: Create Teachers and Students Collections and Insert Sample Data**

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

Code:

db.Teachers.insertMany([

{ Tname: "Praveen", dno: 1, dname: "COMP", experience: "8 years", salary: 30000, date\_of\_joining: new Date("2015-06-20") },

{ Tname: "Rajesh", dno: 2, dname: "IT", experience: "5 years", salary: 27000, date\_of\_joining: new Date("2018-07-15") },

{ Tname: "Suresh", dno: 3, dname: "E&TC", experience: "6 years", salary: 25000, date\_of\_joining: new Date("2017-03-10") },

{ Tname: "Anita", dno: 4, dname: "COMP", experience: "4 years", salary: 28000, date\_of\_joining: new Date("2019-08-01") }

])

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

Code:

db.Students.insertMany([

{ Sname: "John", roll\_no: 20, class: "FY" },

{ Sname: "xyz", roll\_no: 25, class: "SY" },

{ Sname: "Alice", roll\_no: 30, class: "TY" }

])

**Step 3: Queries to Fetch Information**

1. **Find the information about two teachers**:

Code:

db.Teachers.find().limit(2).pretty()

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

Code:

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

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

Code:

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

1. **Find the information about teachers of Computer, IT, and E&TC departments with salary >= 25000**:

Code:

db.Teachers.find({

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

salary: { $gte: 25000 }

}).pretty()

1. **Find student information having roll\_no = 25 or Sname = "xyz"**:

Code:

db.Students.find({

$or: [{ roll\_no: 25 }, { Sname: "xyz" }]

}).pretty()

**Step 4: Updating Documents**

1. **Update the 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 working in the IT department to COMP**:

Code:

db.Teachers.updateMany(

{ dname: "IT" },

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

)

**Step 5: Projection and Using save() Method**

1. **Find teachers' names and their 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: "Physics", experience: "3 years", salary: 22000, date\_of\_joining: new Date("2022-09-15")

})

**Step 6: Deleting Documents**

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

Code:

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

**Step 7: Display Documents in Ascending Order Using pretty()**

1. **Display the first 5 documents in ascending order**:

code:

db.Teachers.find().sort({ Tname: 1 }).limit(5).pretty()

**How to Run and Check Output**

1. **Start the MongoDB Server**:
   * Make sure the MongoDB server is running. Use mongod to start the server.
2. **Open MongoDB Shell**:
   * Open a new terminal and type mongo to start the MongoDB shell.
3. **Run the commands**:
   * Execute the above commands step by step. Use .find().pretty() to format the output nicely.
4. **Verify Output**:
   * The data should be displayed as expected, and updates should reflect correctly in the collections.