1. Create Database BSIOTR
2. Create following Collections **Teachers(Tname,dno,dname,experience,salary,date\_of\_joining ) Students(Sname,roll\_no,class)**
3. Find the information about two teachers
4. Find the information about all teachers of computer department
5. Find the information about all teachers of computer,IT,and e&TC department
6. Find the information about all teachers of computer,IT,and E&TC department having salary greate than or equl to 25000/-
7. Find the student information having roll\_no = 25 or Sname=xyz
8. Update the experience of teacher-praveen to 10years, if the entry is not available in database consider the entry as new entry.
9. Update the deparment of all the teachers working in IT deprtment to COMP
10. find the teachers name and their experience from teachers collection
11. Using Save() method insert one entry in department collection
12. Delete all the doccuments from teachers collection having IT dept.
13. display with pretty() method, the first 5 documents in teachers collection in ascending order

>>>Here are the steps to create the "BSIOTR" database and perform the requested operations in MongoDB:

1. Create the "BSIOTR" database:

```javascript

use BSIOTR

```

2. Create the "Teachers" and "Students" collections:

```javascript

db.createCollection("Teachers")

db.createCollection("Students")

```

3. Insert sample data into the "Teachers" collection:

```javascript

db.Teachers.insertMany([

{

Tname: "John",

dno: 1,

dname: "Computer",

experience: 5,

salary: 26000,

date\_of\_joining: ISODate("2018-08-15T00:00:00Z")

},

{

Tname: "Mary",

dno: 2,

dname: "IT",

experience: 7,

salary: 28000,

date\_of\_joining: ISODate("2016-07-20T00:00:00Z")

},

{

Tname: "Praveen",

dno: 3,

dname: "E&TC",

experience: 3,

salary: 24000,

date\_of\_joining: ISODate("2020-03-10T00:00:00Z")

},

{

Tname: "Rajesh",

dno: 2,

dname: "IT",

experience: 6,

salary: 27000,

date\_of\_joining: ISODate("2019-02-12T00:00:00Z")

}

])

```

4. Insert sample data into the "Students" collection:

```javascript

db.Students.insertMany([

{

Sname: "Alice",

roll\_no: 1,

class: "10A"

},

{

Sname: "Bob",

roll\_no: 2,

class: "11B"

},

{

Sname: "Charlie",

roll\_no: 3,

class: "12C"

}

])

```

Now, you can perform the requested operations:

3. Find information about two teachers:

```javascript

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

```

4. Find information about all teachers of the Computer department:

```javascript

db.Teachers.find({ dname: "Computer" })

```

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

```javascript

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

```

6. Find information about all teachers of Computer, IT, and E&TC departments with a salary greater than or equal to 25,000:

```javascript

db.Teachers.find({

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

salary: { $gte: 25000 }

})

```

7. Find student information with roll\_no = 25 or Sname = "xyz":

```javascript

db.Students.find({ $or: [{ roll\_no: 25 }, { Sname: "xyz" }] })

```

8. Update the experience of teacher Praveen to 10 years or insert a new entry if not found:

```javascript

db.Teachers.update(

{ Tname: "Praveen" },

{

$set: {

experience: 10,

salary: 28000

}

},

{ upsert: true }

)

```

9. Update the department of all teachers working in the IT department to "COMP":

```javascript

db.Teachers.updateMany({ dname: "IT" }, { $set: { dname: "COMP" } })

```

10. Find teachers' names and their experience:

```javascript

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

```

11. Using `save()` method, insert one entry in the "Department" collection (assumes the "Department" collection exists):

```javascript

db.Department.save({ dept\_name: "NewDept", location: "LocationXYZ" })

```

13. Delete all documents from the "Teachers" collection with the IT department:

```javascript

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

```

14. Display the first 5 documents in the "Teachers" collection in ascending order:

```javascript

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

```

These are the MongoDB shell commands to create the database, collections, insert data, and perform the requested operations in the "BSIOTR" database.