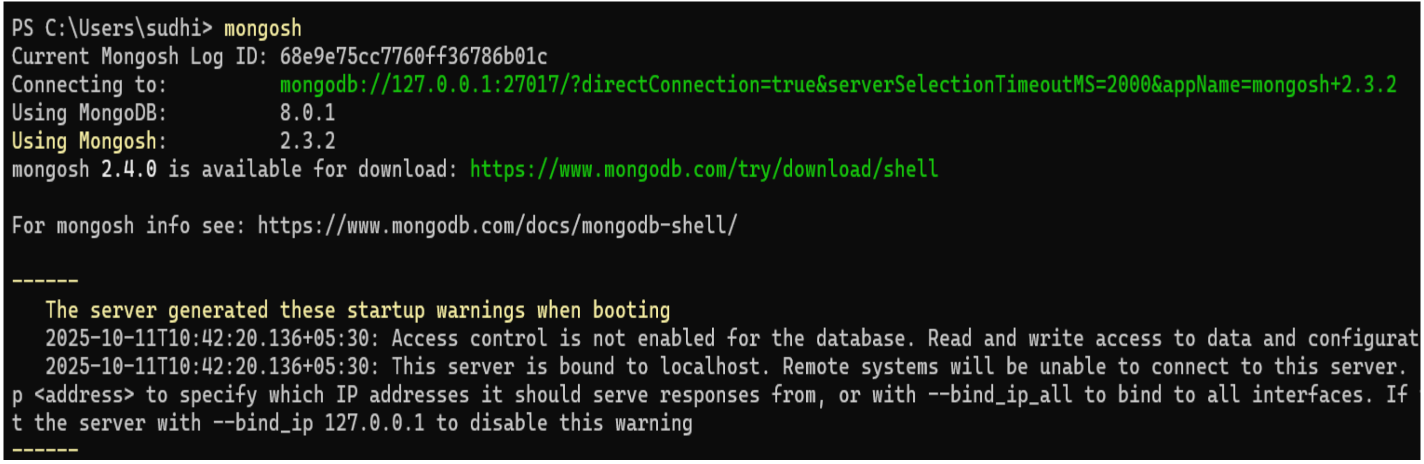
**Name: Sudhir Singh**

**Roll No: TEAD23152**
**Batch: A3**

# Practical no.3

**Title:** MongoDB Queries using CRUD operations. (Use CRUDoperations, SAVE method, logical operators etc)

## MongoDB start and enable commands:

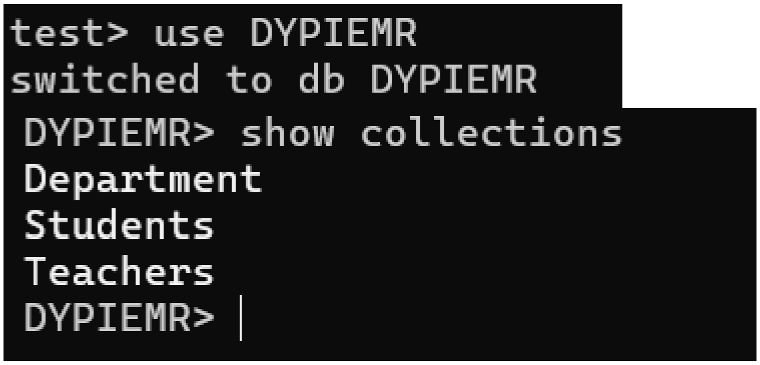


```
PS C:\Users\sudhi> mongosh
Current Mongosh Log ID: 68e9e75cc7760ff36786b01c
Connecting to:          mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.3.2
Using MongoDB:          8.0.1
Using Mongosh:          2.3.2
mongosh 2.4.0 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://www.mongodb.com/docs/mongodb-shell/

------
   The server generated these startup warnings when booting
   2025-10-11T10:42:20.136+05:30: Access control is not enabled for the database. Read and write access to data and configurat
   2025-10-11T10:42:20.136+05:30: This server is bound to localhost. Remote systems will be unable to connect to this server.
p <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If
t the server with --bind_ip 127.0.0.1 to disable this warning
------
```

1. **Create Database DYPIEMR**

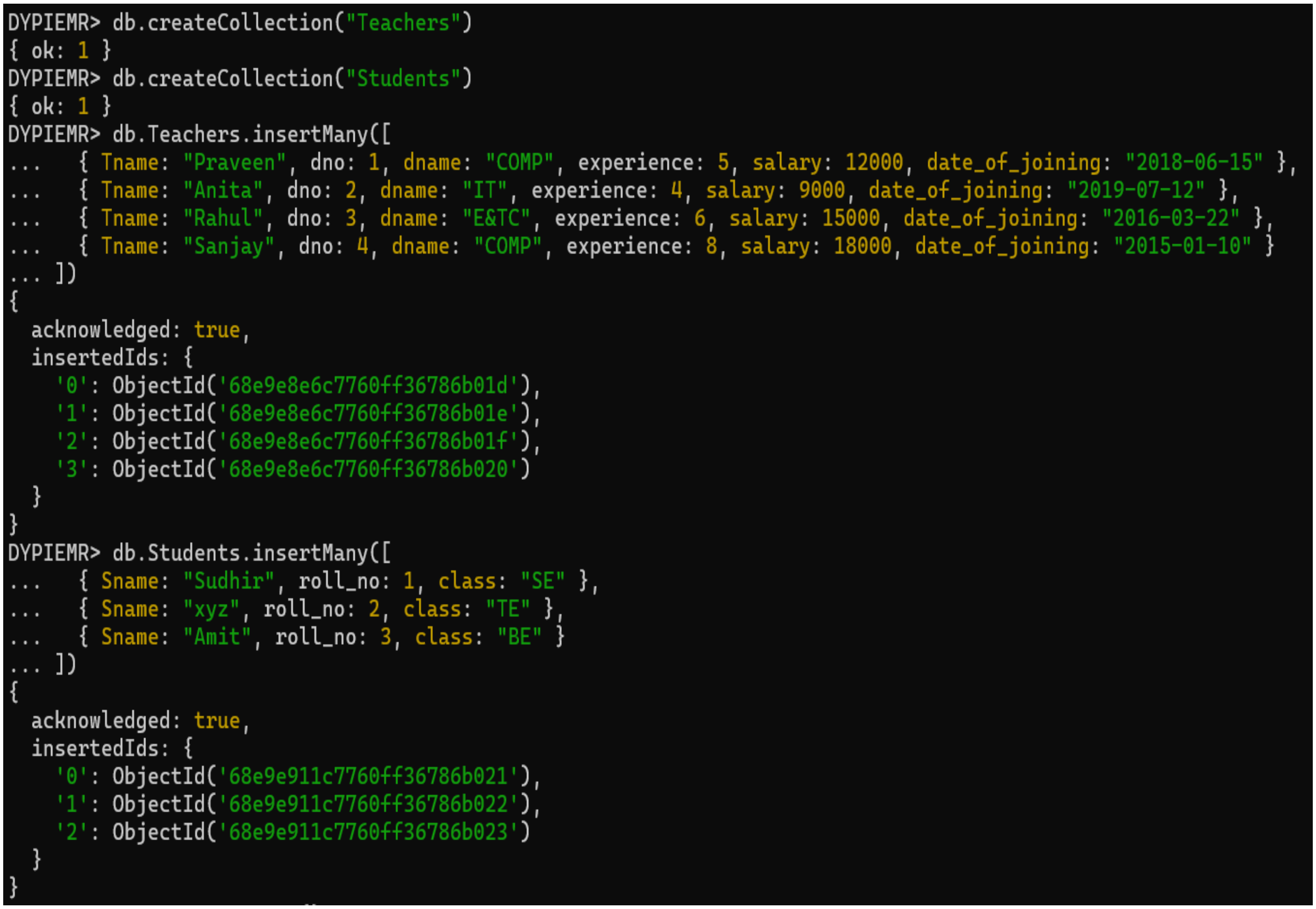


```
test> use DYPIEMR
switched to db DYPIEMR
DYPIEMR> show collections
Department
Students
Teachers
DYPIEMR>
```

## 2. Create following Collections

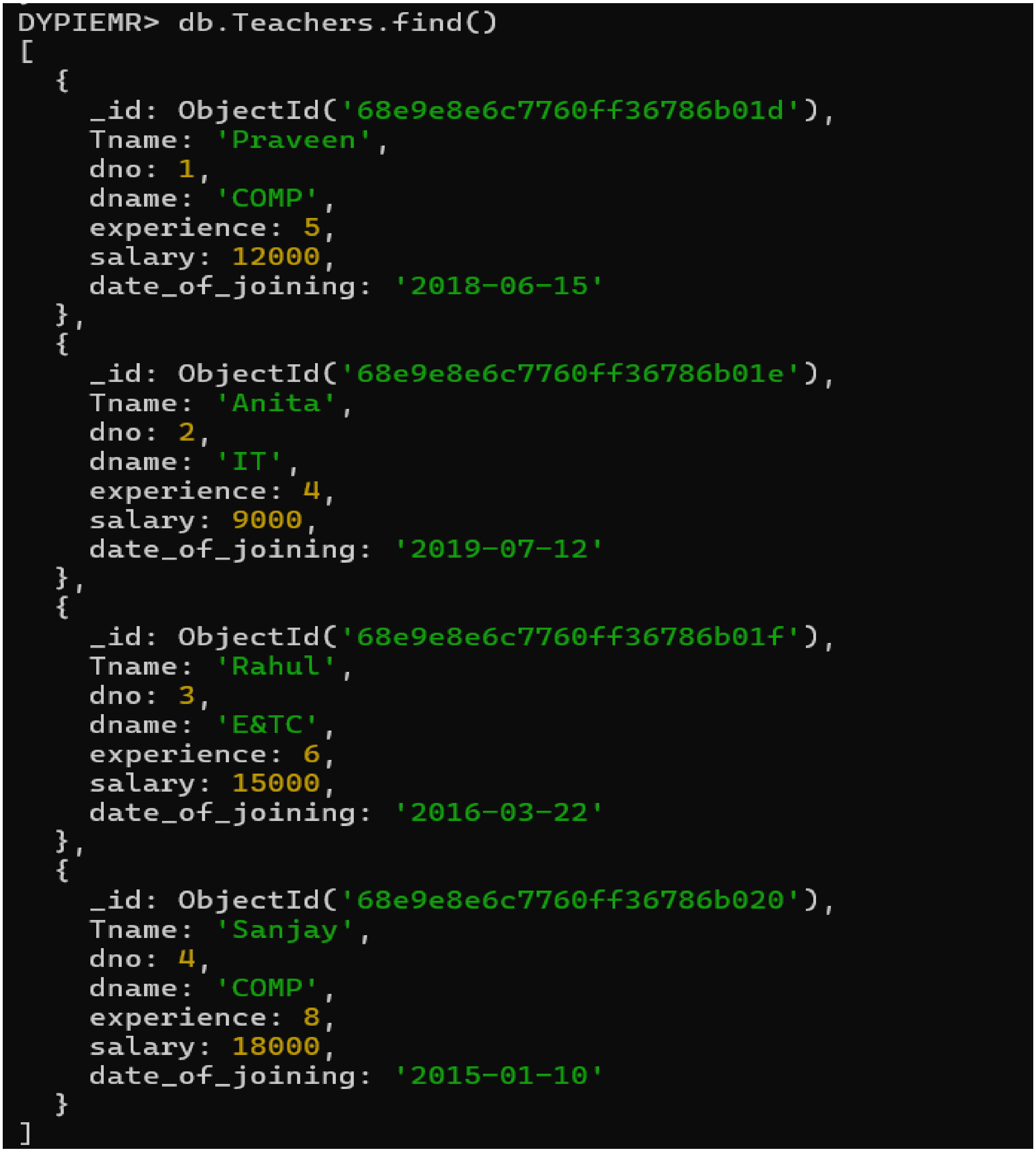
**Teachers(Tname,dno,dname,experience,salary,date_of_joining )**

**Students(Sname,roll_no,class)**



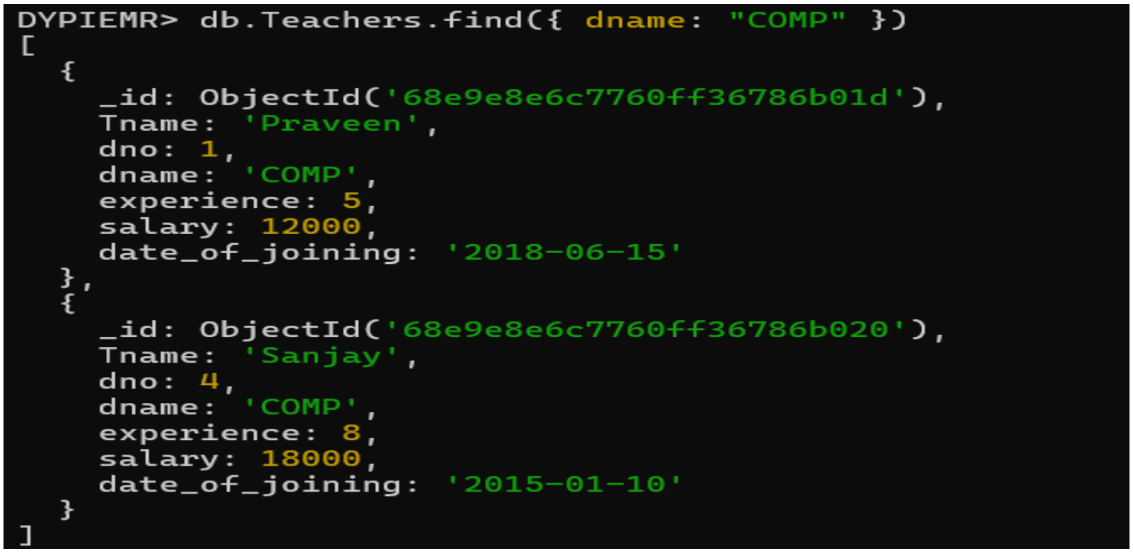
```
DYPIEMR> db.createCollection("Teachers")
{ ok: 1 }
DYPIEMR> db.createCollection("Students")
{ ok: 1 }
DYPIEMR> db.Teachers.insertMany([
...   { Tname: "Praveen", dno: 1, dname: "COMP", experience: 5, salary: 12000, date_of_joining: "2018-06-15" },
...   { Tname: "Anita", dno: 2, dname: "IT", experience: 4, salary: 9000, date_of_joining: "2019-07-12" },
...   { Tname: "Rahul", dno: 3, dname: "E&TC", experience: 6, salary: 15000, date_of_joining: "2016-03-22" },
...   { Tname: "Sanjay", dno: 4, dname: "COMP", experience: 8, salary: 18000, date_of_joining: "2015-01-10" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68e9e8e6c7760ff36786b01d'),
    '1': ObjectId('68e9e8e6c7760ff36786b01e'),
    '2': ObjectId('68e9e8e6c7760ff36786b01f'),
    '3': ObjectId('68e9e8e6c7760ff36786b020')
  }
}
DYPIEMR> db.Students.insertMany([
...   { Sname: "Sudhir", roll_no: 1, class: "SE" },
...   { Sname: "xyz", roll_no: 2, class: "TE" },
...   { Sname: "Amit", roll_no: 3, class: "BE" }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('68e9e911c7760ff36786b021'),
    '1': ObjectId('68e9e911c7760ff36786b022'),
    '2': ObjectId('68e9e911c7760ff36786b023')
  }
}
```

3. **Find the information about all teachers**



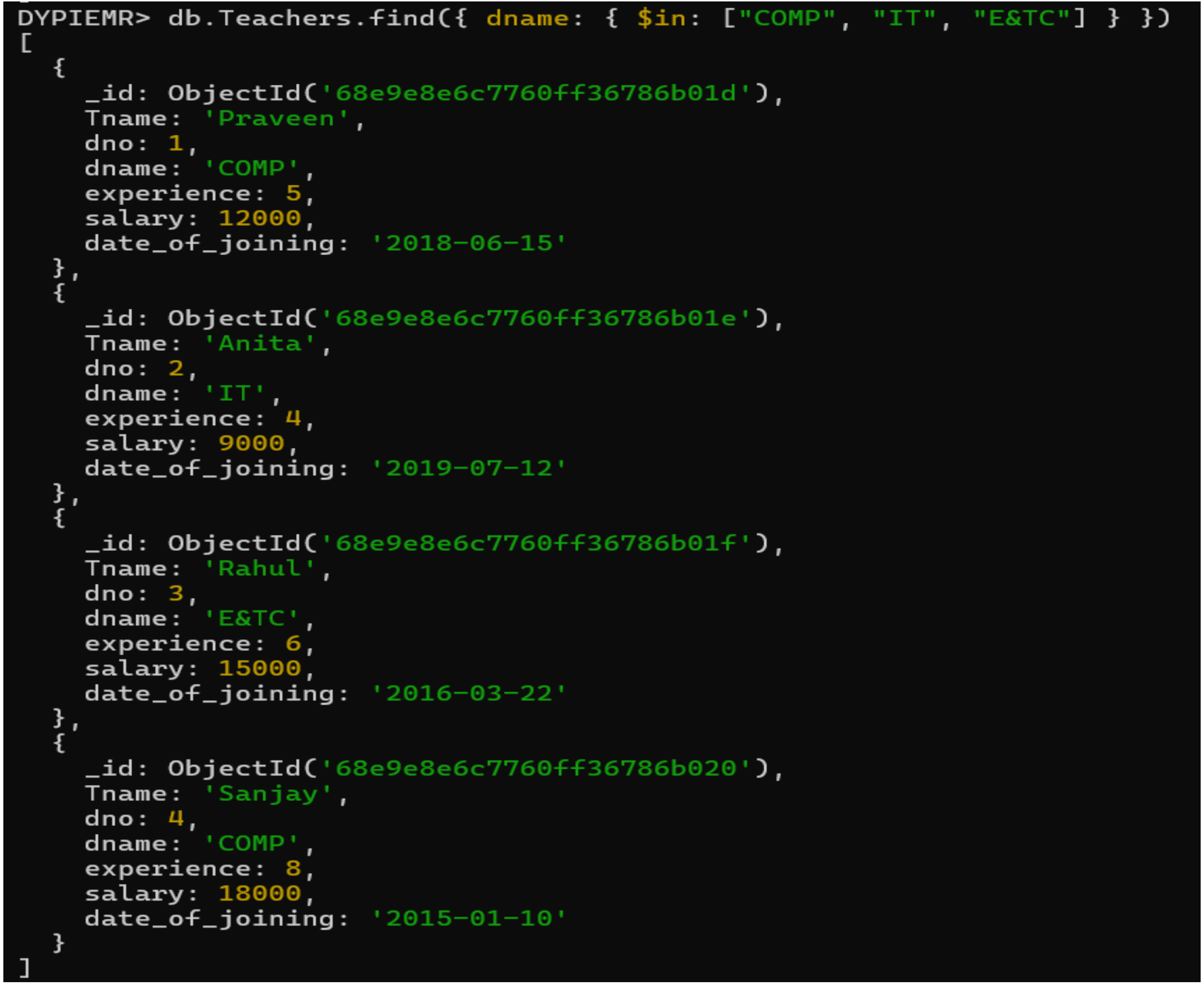
```
DYPIEMR> db.Teachers.find()
[
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01d'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'COMP',
    experience: 5,
    salary: 12000,
    date_of_joining: '2018-06-15'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01e'),
    Tname: 'Anita',
    dno: 2,
    dname: 'IT',
    experience: 4,
    salary: 9000,
    date_of_joining: '2019-07-12'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01f'),
    Tname: 'Rahul',
    dno: 3,
    dname: 'E&TC',
    experience: 6,
    salary: 15000,
    date_of_joining: '2016-03-22'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b020'),
    Tname: 'Sanjay',
    dno: 4,
    dname: 'COMP',
    experience: 8,
    salary: 18000,
    date_of_joining: '2015-01-10'
  }
]
```

4. Find the information about all teachers of computer department



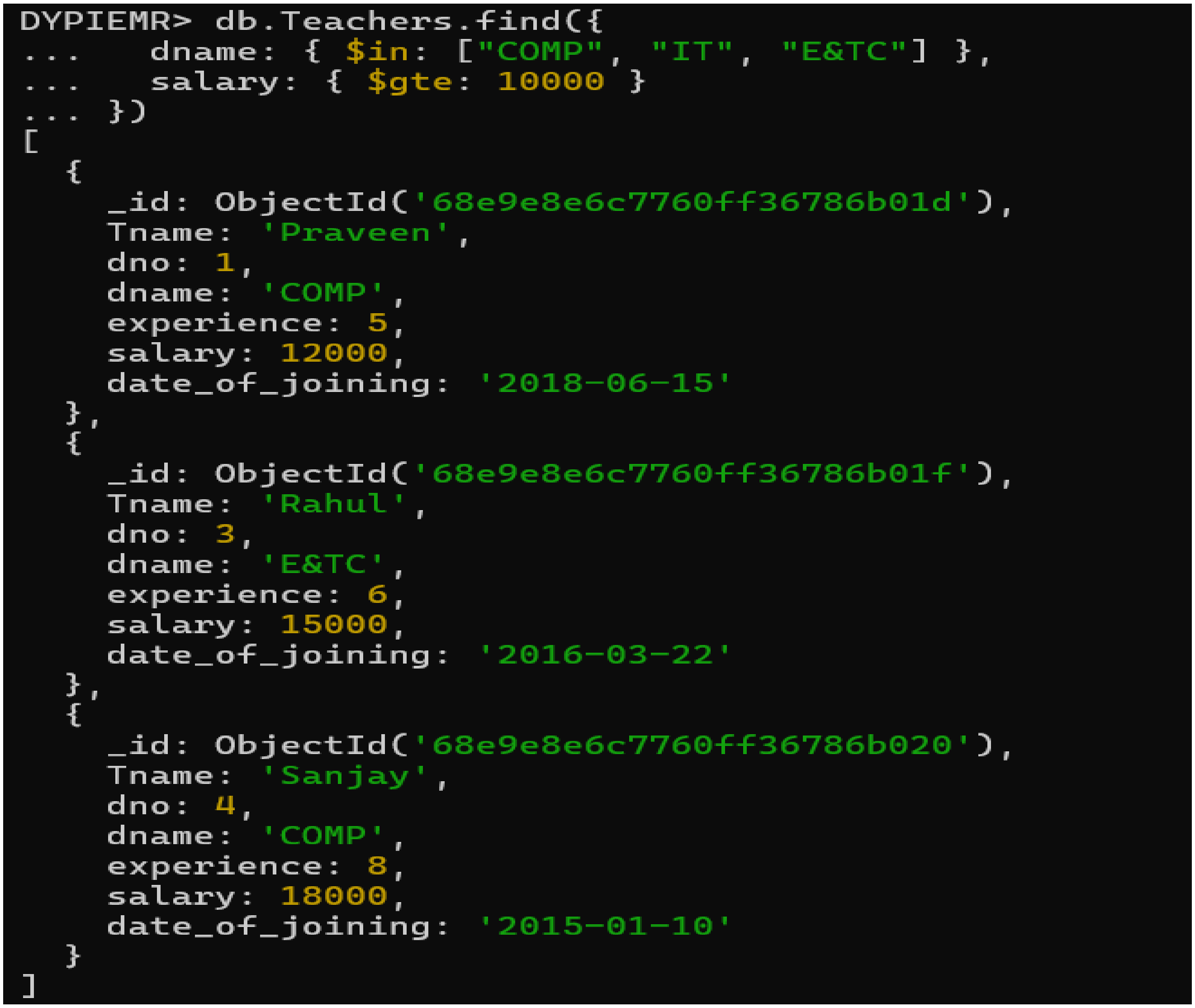
```
DYPIEMR> db.Teachers.find({ dname: "COMP" })
[
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01d'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'COMP',
    experience: 5,
    salary: 12000,
    date_of_joining: '2018-06-15'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b020'),
    Tname: 'Sanjay',
    dno: 4,
    dname: 'COMP',
    experience: 8,
    salary: 18000,
    date_of_joining: '2015-01-10'
  }
]
```

5. Find the information about all teachers of computer,IT,and e&TC department



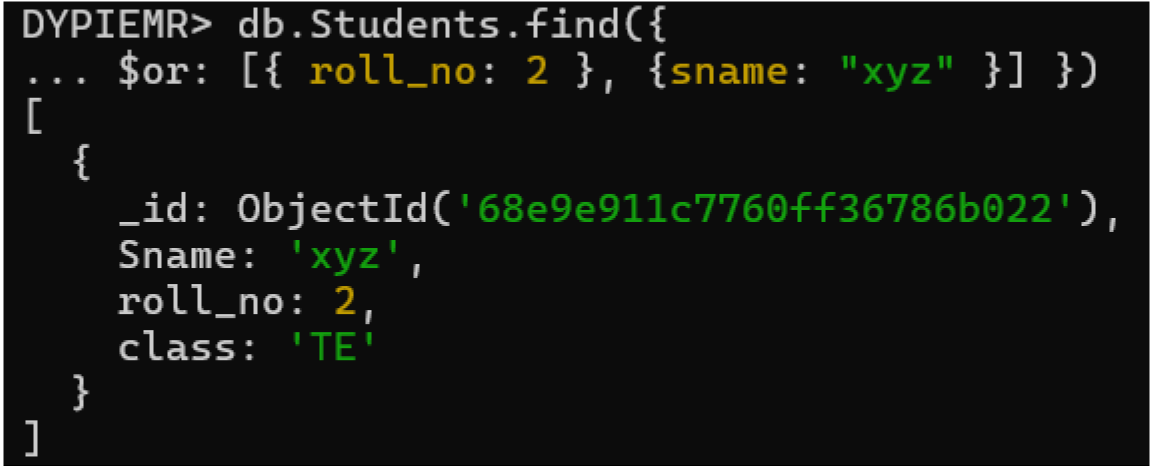
```
DYPIEMR> db.Teachers.find({ dname: { $in: ["COMP", "IT", "E&TC"] } })
[
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01d'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'COMP',
    experience: 5,
    salary: 12000,
    date_of_joining: '2018-06-15'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01e'),
    Tname: 'Anita',
    dno: 2,
    dname: 'IT',
    experience: 4,
    salary: 9000,
    date_of_joining: '2019-07-12'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01f'),
    Tname: 'Rahul',
    dno: 3,
    dname: 'E&TC',
    experience: 6,
    salary: 15000,
    date_of_joining: '2016-03-22'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b020'),
    Tname: 'Sanjay',
    dno: 4,
    dname: 'COMP',
    experience: 8,
    salary: 18000,
    date_of_joining: '2015-01-10'
  }
]
```

**6. Find the information about all teachers of computer,IT,and E&TC department having salary greater than or equal to 10000/-**



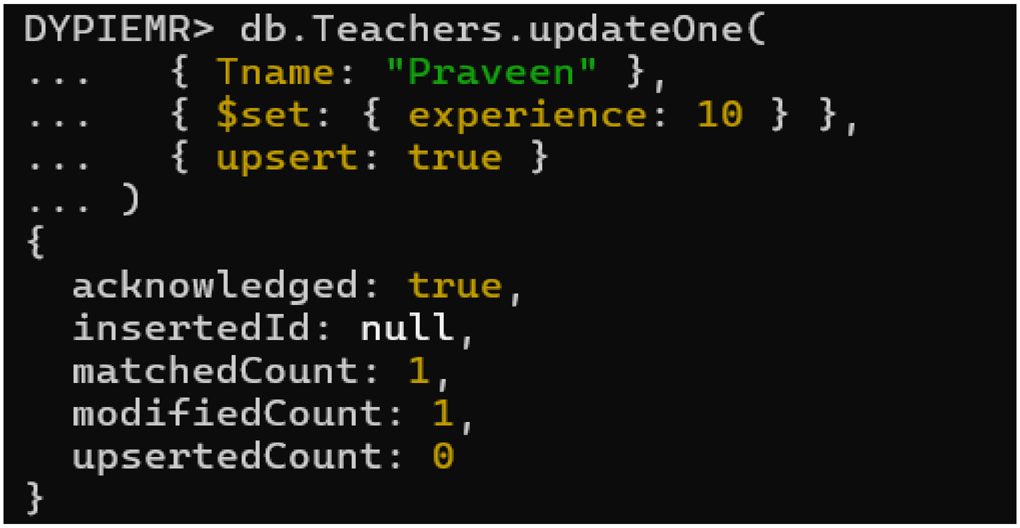
```
DYPIEMR> db.Teachers.find({
...     dname: { $in: ["COMP", "IT", "E&TC"] },
...     salary: { $gte: 10000 }
... })
[
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01d'),
    Tname: 'Praveen',
    dno: 1,
    dname: 'COMP',
    experience: 5,
    salary: 12000,
    date_of_joining: '2018-06-15'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01f'),
    Tname: 'Rahul',
    dno: 3,
    dname: 'E&TC',
    experience: 6,
    salary: 15000,
    date_of_joining: '2016-03-22'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b020'),
    Tname: 'Sanjay',
    dno: 4,
    dname: 'COMP',
    experience: 8,
    salary: 18000,
    date_of_joining: '2015-01-10'
  }
]
```

**7. Find the student information having roll_no = 2 or Sname=xyz**



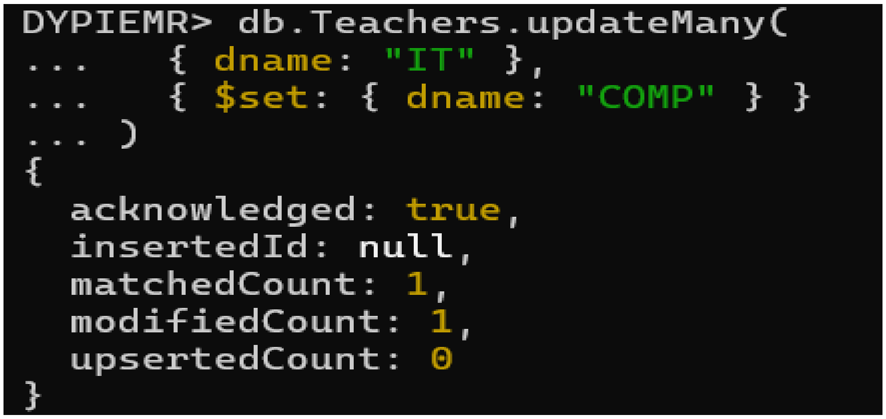
```
DYPIEMR> db.Students.find({
... $or: [{ roll_no: 2 }, {sname: "xyz" }] })
[
  {
    _id: ObjectId('68e9e911c7760ff36786b022'),
    Sname: 'xyz',
    roll_no: 2,
    class: 'TE'
  }
]
```

**8. Update the experience of teacher-praveen to 10years, if the entry is not available in database consider the entry as new entry.**



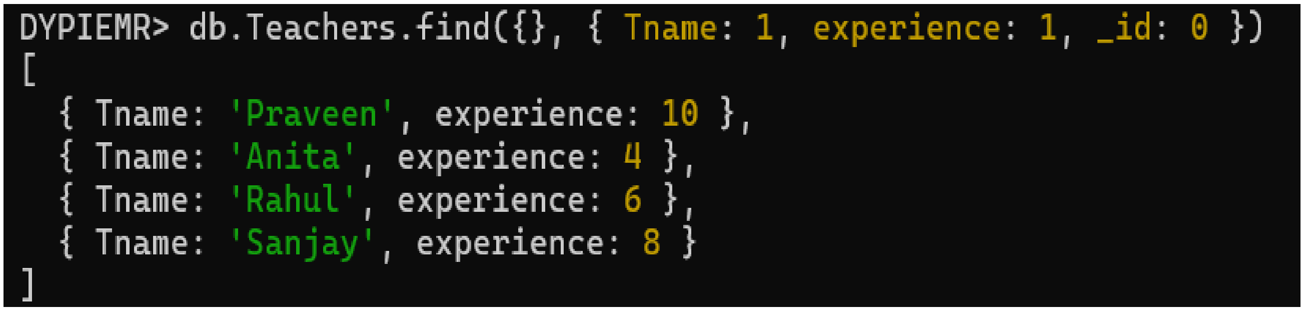
```
DYPIEMR> db.Teachers.updateOne(
...    { Tname: "Praveen" },
...    { $set: { experience: 10 } },
...    { upsert: true }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

**9. Update the deparment of all the teachers working in IT department to COMP**



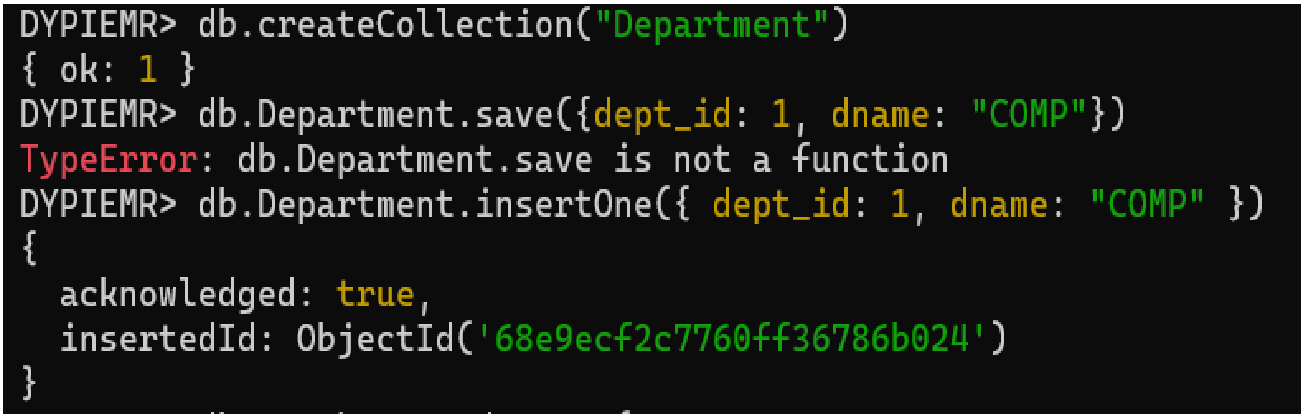
```
DYPIEMR> db.Teachers.updateMany(
...    { dname: "IT" },
...    { $set: { dname: "COMP" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

**10. find the teachers name and their experience from teachers collection**



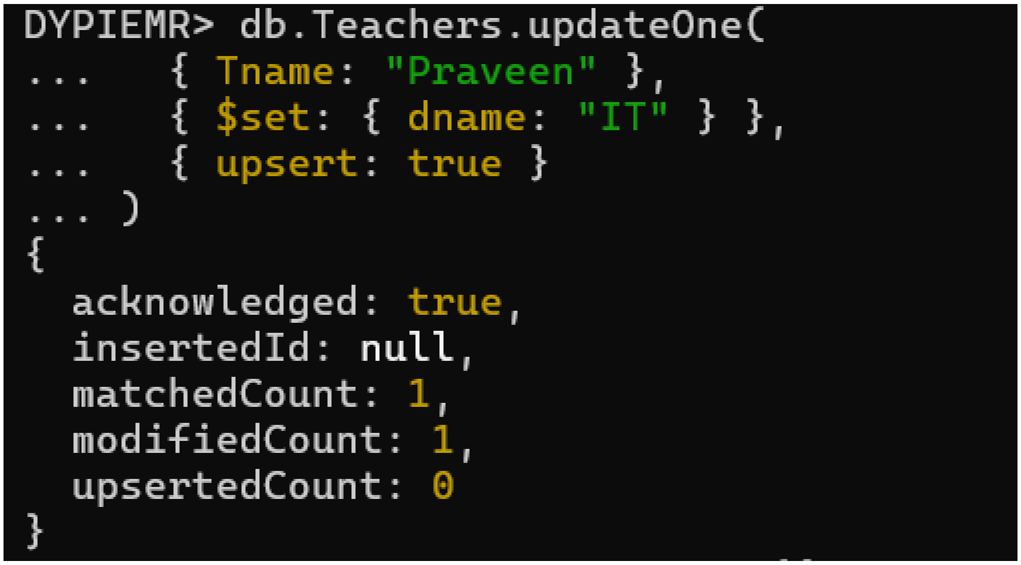
```
DYPIEMR> db.Teachers.find({}, { Tname: 1, experience: 1, _id: 0 })
[
  { Tname: 'Praveen', experience: 10 },
  { Tname: 'Anita', experience: 4 },
  { Tname: 'Rahul', experience: 6 },
  { Tname: 'Sanjay', experience: 8 }
]
```

11. **Using Save() method insert one entry in department collection**



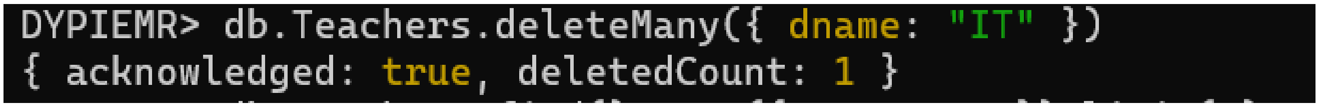
```
DYPIEMR> db.createCollection("Department")
{ ok: 1 }
DYPIEMR> db.Department.save({dept_id: 1, dname: "COMP"})
TypeError: db.Department.save is not a function
DYPIEMR> db.Department.insertOne({ dept_id: 1, dname: "COMP" })
{
  acknowledged: true,
  insertedId: ObjectId('68e9ecf2c7760ff36786b024')
}
```

12. **Using Save() method change the dept of teacher praveen to IT**



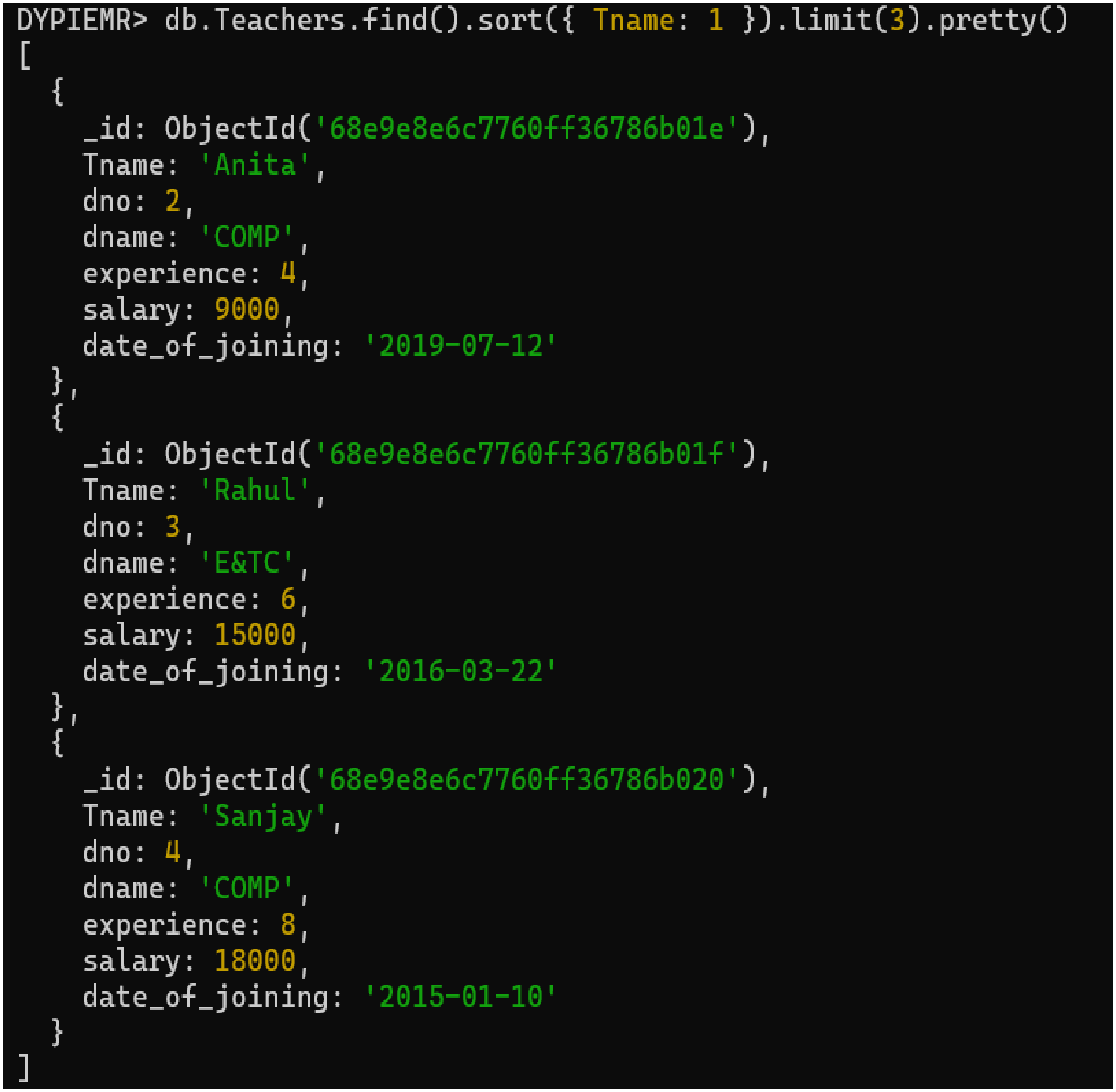
```
DYPIEMR> db.Teachers.updateOne(
...   { Tname: "Praveen" },
...   { $set: { dname: "IT" } },
...   { upsert: true }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

13. **Delete all the documents from teachers collection having IT dept.**



```
DYPIEMR> db.Teachers.deleteMany({ dname: "IT" })
{ acknowledged: true, deletedCount: 1 }
```

**14. display with pretty() method, the first 3 documents in teachers collection in ascending order.**



```
DYPIEMR> db.Teachers.find().sort({ Tname: 1 }).limit(3).pretty()
[
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01e'),
    Tname: 'Anita',
    dno: 2,
    dname: 'COMP',
    experience: 4,
    salary: 9000,
    date_of_joining: '2019-07-12'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b01f'),
    Tname: 'Rahul',
    dno: 3,
    dname: 'E&TC',
    experience: 6,
    salary: 15000,
    date_of_joining: '2016-03-22'
  },
  {
    _id: ObjectId('68e9e8e6c7760ff36786b020'),
    Tname: 'Sanjay',
    dno: 4,
    dname: 'COMP',
    experience: 8,
    salary: 18000,
    date_of_joining: '2015-01-10'
  }
]
```