

MongoDB Practical Assignment

1. Create an Employee collection with mentioned fields Employee
(eno,ename,salary,desig,dept: {deptno,deptname,location}, project: {pname,hrs})
→

```
db.createCollection('employee');
```

2. Insert 10 documents in Employee collection
→
Query

```
db.employee.insertMany([
  {
    eno:1,
    ename:"Prajakta Kamble",
    salary:20000,
    desig:"Developer",
    dept:{
      deptno:111,
      deptname:'IT',
      location:'Pune'
    },
    project:{
      pname:'artmantis',
      hrs:20
    }
  },
  {
    eno:2,
    ename:"Manjusha Nagpure",
    salary:35000,
    desig:"Developer",
    dept:{
      deptno:111,
      deptname:'IT',
      location:'Pune'
    },
    project:{
      pname:'artmantis',
      hrs:20
    }
  },
  {
    eno:3,
    ename:"Ankita Karanje",
    salary:20000,
    desig:"Civil Engineer",
    dept:{
      deptno:112,
```

```
    deptname:'Civil',
    location:'Pune'
  },
  project:{
    pname:'Construction',
    hrs:100
  }
},
{
  eno:4,
  ename:"Uma Bidkar",
  salary:20000,
  desig:"Nurse",
  dept:{
    deptno:113,
    deptname:'Dector',
    location:'Pune'
  },
  project:{
    pname:'Gynac',
    hrs:20
  }
},
{
  eno:5,
  ename:"Pournima Khengare",
  salary:20000,
  desig:"Project manager",
  dept:{
    deptno:111,
    deptname:'IT',
    location:'Pune'
  },
  project:{
    pname:'artmantis',
    hrs:20
  }
},
{
  eno:6,
  ename:"Shubhangi Ghorpade",
  salary:20000,
  desig:"Civil Engineer",
  dept:{
    deptno:112,
    deptname:'Civil',
    location:'Pune'
  },
}
```

```
project:{
  pname:'Construction',
  hrs:100
},
{
  eno:7,
  ename:"Arpita Yenpure",
  salary:20000,
  desig:"Civil Engineer",
  dept:{
    deptno:112,
    deptname:'Civil',
    location:'Pune'
  },
  project:{
    pname:'Construction',
    hrs:100
  }
},
{
  eno:8,
  ename:"Prajakta Kamble",
  salary:20000,
  desig:"Developer",
  dept:{
    deptno:111,
    deptname:'IT',
    location:'Pune'
  },
  project:{
    pname:'artmantis',
    hrs:20
  }
},
{
  eno:9,
  ename:"Akash Sunthaval",
  salary:70000,
  desig:"Developer",
  dept:{
    deptno:111,
    deptname:'IT',
    location:'Pune'
  },
  project:{
    pname:'artmantis',
    hrs:20
  }
}
```

```

    }
  },
  {
    eno:10,
    ename:"Vikas Wani",
    salary:60000,
    desig:"Developer",
    dept:{
      deptno:111,
      deptname:'IT',
      location:'Pune'
    },
    project:{
      pname:'artmantis',
      hrs:20
    }
  }
]

```

Output:

```

< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("657462e74660983211b252e4"),
    '1': ObjectId("657462e74660983211b252e5"),
    '2': ObjectId("657462e74660983211b252e6"),
    '3': ObjectId("657462e74660983211b252e7"),
    '4': ObjectId("657462e74660983211b252e8"),
    '5': ObjectId("657462e74660983211b252e9"),
    '6': ObjectId("657462e74660983211b252ea"),
    '7': ObjectId("657462e74660983211b252eb"),
    '8': ObjectId("657462e74660983211b252ec"),
    '9': ObjectId("657462e74660983211b252ed")
  }
}

```

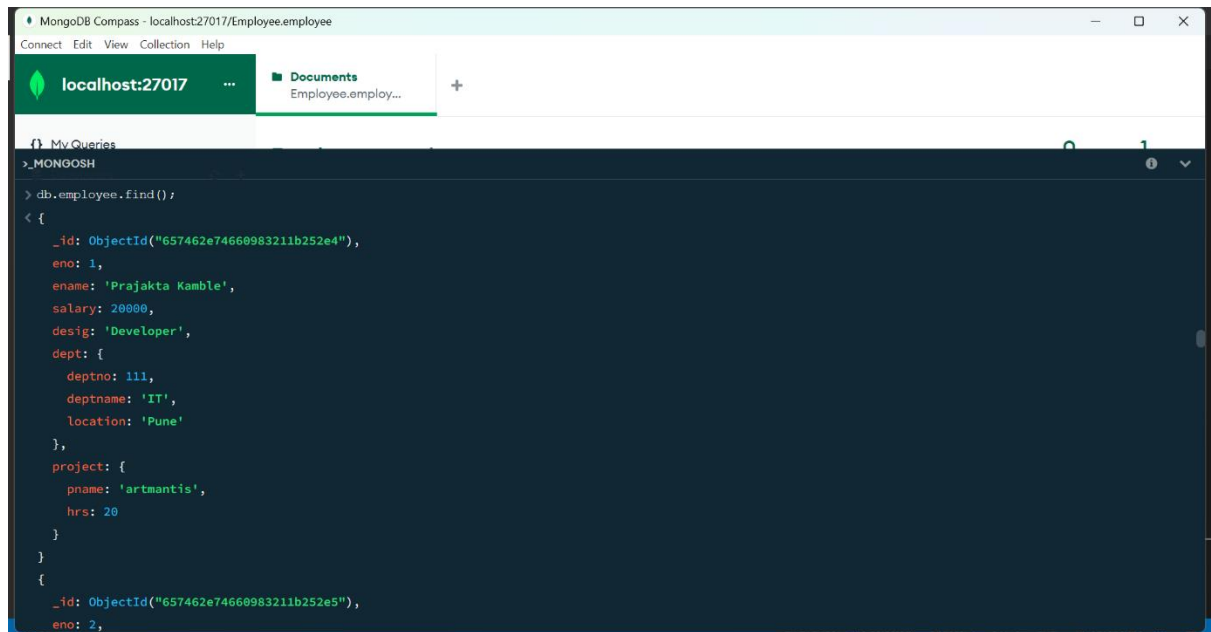
3. Display all the documents from Employee collection

→

Query:

```
db.employee.find();
```

Output:

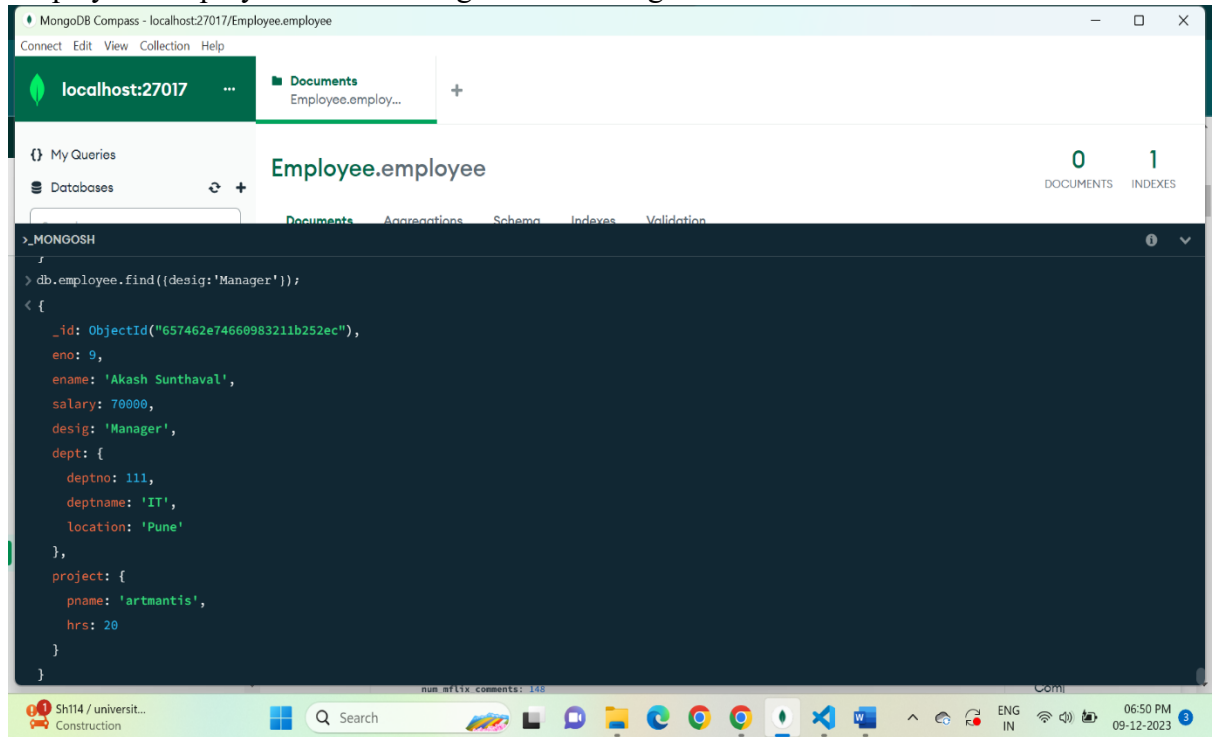


```
> db.employee.find();
< {
  _id: ObjectId("657462e74660983211b252e4"),
  eno: 1,
  ename: 'Prajakta Kamble',
  salary: 20000,
  desig: 'Developer',
  dept: {
    deptno: 111,
    deptname: 'IT',
    location: 'Pune'
  },
  project: {
    pname: 'artmantis',
    hrs: 20
  }
}
{
  _id: ObjectId("657462e74660983211b252e5"),
  eno: 2,
```

4. Display all employees whose name starts with “S”

```
> db.employee.find({ename:/^S/});
< {
  _id: ObjectId("657462e74660983211b252e9"),
  eno: 6,
  ename: 'Shubhangi Ghorpade',
  salary: 20000,
  desig: 'Civil Engineer',
  dept: {
    deptno: 112,
    deptname: 'Civil',
    location: 'Pune'
  },
  project: {
    pname: 'Construction',
    hrs: 100
  }
}
Employee>
```

5. Display all Employee with the designation “Manager”



6. Display all employees with salary >50000 and salary

```
db.employee.find({
  salary: {
    $gt: 50000,
    $lt: 80000
  }
})
```

Output:

```
[
  {
    "_id": {
      "$oid": "657463d5882bb59f49af932e"
    },
    "eno": 9,
    "ename": "Akash Sunthaval",
    "salary": 70000,
    "desig": "Developer",
    "dept": {
      "deptno": 111,
      "deptname": "IT",
      "location": "Pune"
    },
    "project": {
```

```

    "pname": "artmantis",
    "hrs": 20
  },
  {
    "_id": {
      "$oid": "657463d5882bb59f49af932f"
    },
    "eno": 10,
    "ename": "Vikas Wani",
    "salary": 60000,
    "desig": "Developer",
    "dept": {
      "deptno": 111,
      "deptname": "IT",
      "location": "Pune"
    },
    "project": {
      "pname": "artmantis",
      "hrs": 20
    }
  }
]

```

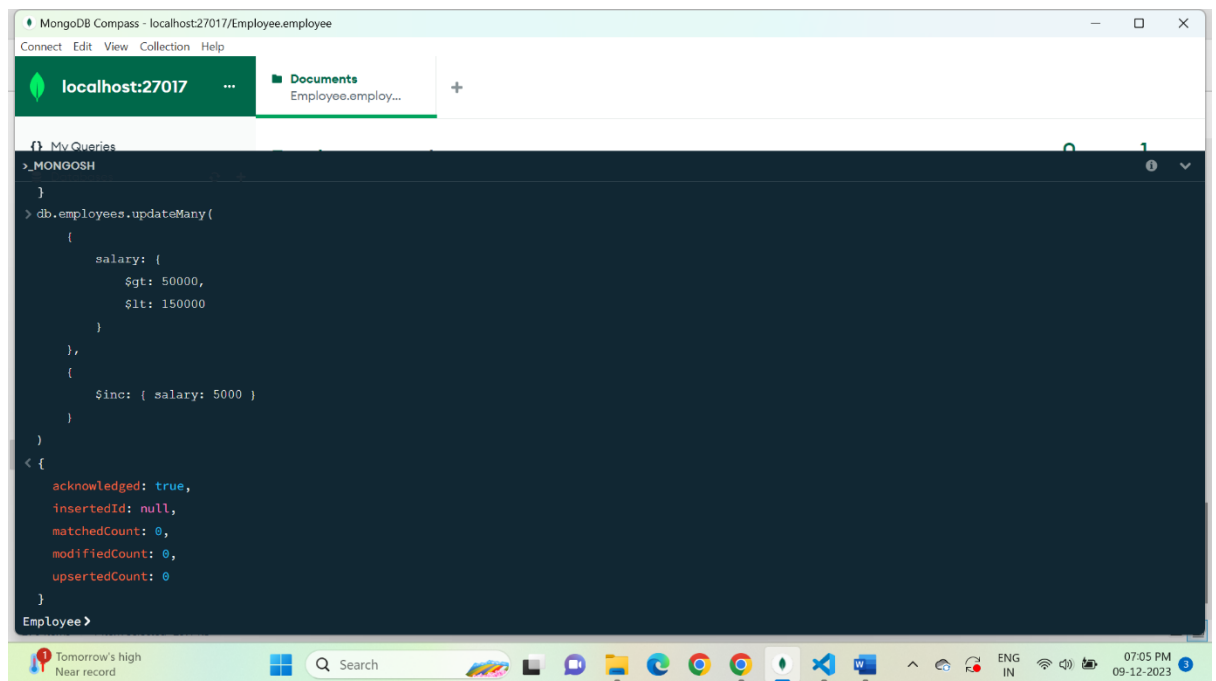
7. Update no. of hrs to 7 for pname=_____

```

> db.employee.updateOne(
  { pname:'Vikas Wani'},
  { $set:{ project:{ hrs:7 }}}
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
Employee >

```

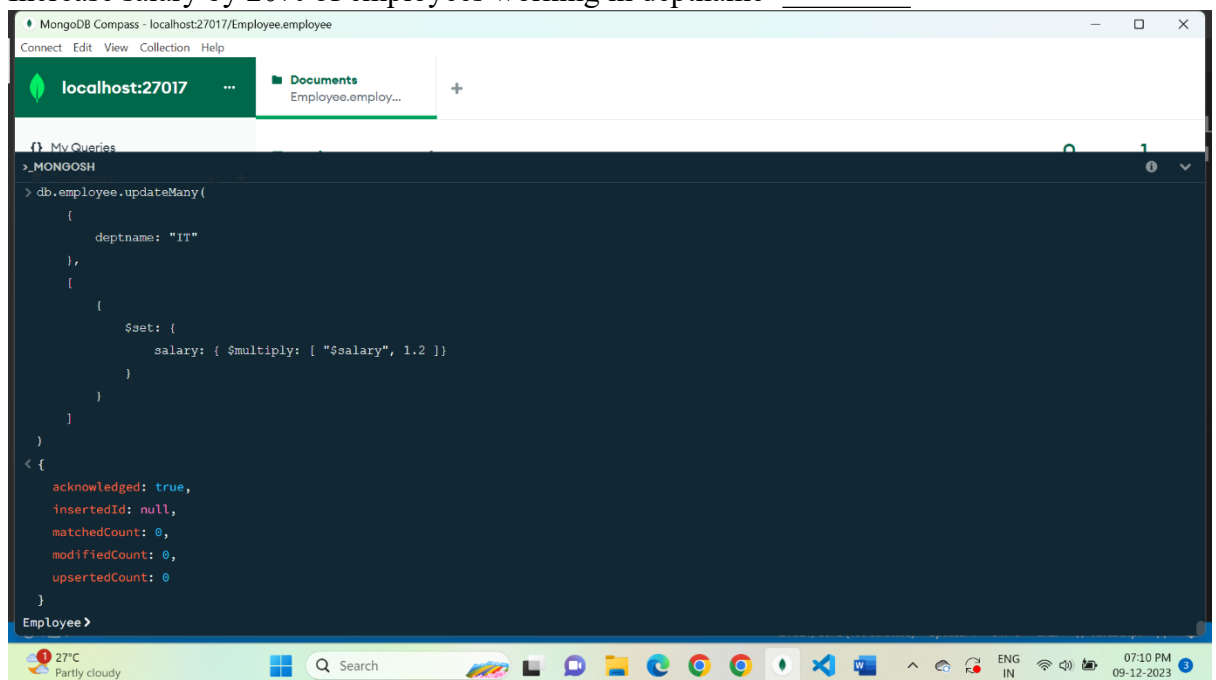
8. Add bonus Rs. 5000 for all employees with salary >50000 and salary <150000



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the 'Employee' collection. The 'My Queries' tab is active, displaying a JavaScript query in the 'MONGOSH' console. The query is a `updateMany` operation that targets employees with a salary between 50,000 and 150,000 and adds a 5,000 bonus. The response shows that 0 documents were matched and 0 were modified.

```
> db.employee.updateMany(
  {
    salary: {
      $gt: 50000,
      $lt: 150000
    }
  },
  {
    $inc: { salary: 5000 }
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

9. Increase salary by 20% of employees working in deptname=_____



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the 'Employee' collection. The 'My Queries' tab is active, displaying a JavaScript query in the 'MONGOSH' console. The query is a `updateMany` operation that targets employees in the 'IT' department and increases their salary by 20% using the `$multiply` operator. The response shows that 0 documents were matched and 0 were modified.

```
> db.employee.updateMany(
  {
    deptname: "IT"
  },
  {
    $set: {
      salary: { $multiply: [ "$salary", 1.2 ] }
    }
  }
)
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```


10. Remove all employees working on pname=_____

```
> db.employee.deleteMany({pname:"Uma Bidkar"})  
< {  
  acknowledged: true,  
  deletedCount: 0  
}  
Employee >
```