MongoDB Practical Assignment

1. Create a Employee collection with mentioned fields

Employee (eno,ename,salary,desig,dept: {deptno,deptname,location}, project:{pname,hrs})

Solution:

```
db.createCollection("Employee")
db.Employee.insert({
  eno: 1,
  ename: "John Doe",
  salary: 60000,
  desig: "Software Engineer",
  dept: {
    deptno: 101,
    deptname: "Engineering",
    location: "New York"
  },
  project: {
    pname: "Project A",
    hrs: 40
  }
})
db.Employee.insert({
  eno: 2,
  ename: "Jane Smith",
  salary: 70000,
  desig: "Project Manager",
  dept: {
```

```
    deptno: 102,
```

MongoDB Practical Assignment

1. Create a Employee collection with mentioned fields

Employee (eno,ename,salary,desig,dept: {deptno,deptname,location}, project:{pname,hrs})

Solution:

```
db.createCollection("Employee")
db.Employee.insert({
  eno: 1,
  ename: "John Doe",
  salary: 60000,
  desig: "Software Engineer",
  dept: {
    deptno: 101,
    deptname: "Engineering",
    location: "New York"
  },
  project: {
    pname: "Project A",
    hrs: 40
  }
})
db.Employee.insert({
  eno: 2,
  ename: "Jane Smith",
  salary: 70000,
  desig: "Project Manager",
```

dept: {
deptno: 102,

Course- BTech	Type- Core
Course Code- CSET201	Course Name- Information Management System (Lab)
Year- 2025	Semester- Odd
Date- 03/11/2025 – 07/11/2025	Batch- 2024-2028

Lab Assignment No. (11)

```
test> db.createCollection("Employee")
...
... db.Employee.insertMany([
...   {
...     eno: 1,
...     ename: "John Doe",
...     salary: 60000,
...     desig: "Software Engineer",
...     dept: {
...       deptno: 101,
...       deptname: "Engineering",
...       location: "New York"
...     },
...     project: {
...       pname: "Project A",
...       hrs: 40
...     }
...   },
...   {
...     eno: 2,
...     ename: "Jane Smith",
...     salary: 70000,
...     desig: "Project Manager",
...     dept: {
...       deptno: 102,
...       deptname: "Management",
...       location: "Los Angeles"
...     },
...     project: {
...       pname: "Project B",
...       hrs: 35
...     }
...   },
...   {
...     eno: 3,
...     ename: "Michael Brown",
...     salary: 55000,
...     desig: "QA Analyst",
...     dept: {
...       deptno: 101,
...       deptname: "Engineering",
```

```
...   {
...     eno: 3,
...     ename: "Michael Brown",
...     salary: 55000,
...     desig: "QA Analyst",
...     dept: {
...       deptno: 101,
...       deptname: "Engineering",
...       location: "New York"
...     },
...     project: {
...       pname: "Project A",
...       hrs: 38
...     }
...   }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6909772f49f79f0963cebea4'),
    '1': ObjectId('6909772f49f79f0963cebea5'),
    '2': ObjectId('6909772f49f79f0963cebea6')
  }
}
test> db.Employee.find({ desig: "Project Manager" })
[
  {
    _id: ObjectId('6909772f49f79f0963cebea5'),
    eno: 2,
    ename: 'Jane Smith',
    salary: 70000,
    desig: 'Project Manager',
    dept: { deptno: 102, deptname: 'Management', location: 'Los Angeles' },
    project: { pname: 'Project B', hrs: 35 }
  }
]
test> db.Employee.find({ ename: { $regex: /^M/ } })
[
  {
    _id: ObjectId('6909772f49f79f0963cebea6'),
```

```
test> db.Employee.find({ ename: { $regex: /^M/ } })
[
  {
    _id: ObjectId('6909772f49f79f0963cebea6'),
    eno: 3,
    ename: 'Michael Brown',
    salary: 55000,
    desig: 'QA Analyst',
    dept: { deptno: 101, deptname: 'Engineering', location: 'New York' },
    project: { pname: 'Project A', hrs: 38 }
  }
]
test> db.Employee.updateMany(
...   { "project.pname": "Project A" },
...   { $set: { "project.hrs": 70 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
test> db.Employee.updateMany(
...   { salary: { $gt: 50000, $lt: 150000 } },
...   { $inc: { salary: 5000 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
test> db.Employee.updateMany(
...   { "dept.deptname": "Engineering" },
...   { $mul: { salary: 1.20 } }
... )
...
```

\

```
test> db.Employee.updateMany(
...   { "dept.deptname": "Engineering" },
...   { $mul: { salary: 1.20 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
  modifiedCount: 2,
  upsertedCount: 0
}
test> db.Employee.countDocuments({ "dept.deptno": 102 })
1
test> db.Employee.aggregate([
...   {
...     $group: {
...       _id: "$dept.deptname",
...       totalSalary: { $sum: "$salary" },
...       totalEmployees: { $sum: 1 }
...     }
...   }
... ])
...
[
  { _id: 'Engineering', totalSalary: 150000, totalEmployees: 2 },
  { _id: 'Management', totalSalary: 75000, totalEmployees: 1 }
]
test> var avgSalary = db.Employee.aggregate([
...   { $group: { _id: null, avgSal: { $avg: "$salary" } } }
... ]).toArray()[0].avgSal;
...
... db.Employee.find({ salary: { $gt: avgSalary } })
...
[
  {
    _id: ObjectId('6909772f49f79f0963cebea4'),
    eno: 1,
    ename: 'John Doe',
    salary: 78000,
    desig: 'Software Engineer',
```

```
test> var avgSalary = db.Employee.aggregate([
...   { $group: { _id: null, avgSal: { $avg: "$salary" } } }
... ]).toArray()[0].avgSal;
...
... db.Employee.find({ salary: { $gt: avgSalary } })
...
[
  {
    _id: ObjectId('6909772f49f79f0963cebea4'),
    eno: 1,
    ename: 'John Doe',
    salary: 78000,
    desig: 'Software Engineer',
    dept: { deptno: 101, deptname: 'Engineering', location: 'New York' },
    project: { pname: 'Project A', hrs: 70 }
  }
]
```