

## Program 2

Poorna Vishwanath  
1BM17CS061

Create Company Database with following Collections  
(i) Employee (ii) Department

Use Company

db.createCollection("Employee")

db.createCollection("Department")

Mongo DB operations

1. Insert at least five documents in each collection using all three methods of insertion { show update method using upsert set to true and set to false }

```
db.Employee.insert({  
  _id: 1  
  Ename: 'Ajay',  
  salary: '12000',  
  dept: 'manager',  
  Age: '35',  
  dnum: '1001'  
})
```

```
db.Employee.insertMany([  
  { _id: 2, Ename: 'Arha', salary: '13000', dept: 'Supervisor', Age: '24',  
    dnum: '1002' },  
  { _id: 3, Ename: 'Akash', salary: '12500', dept: 'Developer', Age: '30',  
    dnum: '1003' }  
])
```

```
db.Employee.update(  
  { _id: 4, Ename: 'Tilak', salary: '12500', Age: '32', dnum: '1003' },  
  { $set: { dept: 'Developer' } }, { upsert: true });
```



db.Employee.update (

```
{_id:10, Ename: 'Akhil', salary: '12500'}, { $set: { dept: 'Developer' },  
  { upsert: false } );
```

db.Employee.save (

```
{_id: 5, Ename: 'Poorna', salary: '12600', dept: 'supervisor', Age: '23',  
  dnum: '1002' } );
```

db.Department.insert ( {

```
-id: 10,  
  Dname: 'Manager',  
  number: '1001',  
  location: 'Basavanagudi' } )
```

db.Department.update (

```
{_id: 12, Dname: 'Supervisor', number: '1002'}, { $set: { location: 'Ginnesa',  
  { upsert: 'true' } );
```

db.Department.save (

```
{_id: 14, Dname: 'Programmer', number: '1003', location: 'Tayyanagar' } );
```

db.Department.find (14)

2. Update 'Employee' collection to add new field to an existing document.

db.Employee.update (

```
{_id: 5}, { $set: { Phonenum: '9980678080' } } );
```

db.Employee.find ( {\_id: 5 } ).pretty ();

3. Remove a field from an existing document

db.Department.find ( {\_id: 12 } ).pretty ();

db.Department.update ( {\_id: 12}, { \$unset: { location: 'Ginnesa' } } );



4. Select all documents from both collections

db.Employee.find({})

db.Department.find({})

5. Select only employee named & department number whose department number falls between 1001 to 1005

db.Employee.find({dnum: {"\$gt": '1001', "\$lt": '1005'}},  
{Ename: true, dnum: true});

6. Select employee document whose name begins with 'A'

db.Employee.find({Ename: {"\$regex": '^A'}})

7. Select employee documents whose age is greater than 30

db.Employee.find({Age: {"\$gt": '30'}}).pretty();