Date: 02-01-2026

Day 5 MongoDB Assignment

1. **Start the 'mongod' engine**
   C:\Users\Admin>mongod

2. **Start the shell interactive interface**
   mongosh
   Current Mongosh Log ID: 6957515f7ee08e48ae1e2620
   Connecting to:
   mongodb://127.0.0.1:27017/?directConnection=true&serverSelectio
   nTimeoutMS=2000&appName=mongosh+2.5.10
   Using MongoDB: 8.2.3
   Using Mongosh: 2.5.10

3. show dbs
   test> show dbs
   PracticeDB 48.00 KiB
   admin 40.00 KiB
   config 84.00 KiB
   local 72.00 KiB

4. **Create Database**: 'insuranceDB'
   use insurance dbs

5. use insurance dbs
   switched to db insuranceDB
   insuranceDB> show dbs
   PracticeDB 48.00 KiB
   admin 40.00 KiB
   config 84.00 KiB
   local 72.00 KiB

6. **Drop** PracticeDB **database**
   insuranceDB> use PracticeDB

```
switched to db PracticeDB
PracticeDB> db.dropDatabase()
{ ok: 1, dropped: 'PracticeDB' }
```

7. **Create Collections and Insert Documents:**
   'agents'

```
insuranceDB> db.agents.insertMany([
...   {
...     "AgentName": "Ravi Kumar",
...     "Phone": "8888888888",
...     "City": "Bangalore"
...   },
...   {
...     "AgentName": "Anita Sharma",
...     "Phone": "7777777777",
...     "City": "Hyderabad"
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6957574b7ee08e48ae1e2621'),
    '1': ObjectId('6957574b7ee08e48ae1e2622')
  }
}

insuranceDB> db.agents.find()
[
  {
    _id: ObjectId('6957574b7ee08e48ae1e2621'),
    AgentName: 'Ravi Kumar',
    Phone: '8888888888',
    City: 'Bangalore'
  },
  {
    _id: ObjectId('6957574b7ee08e48ae1e2622'),
    AgentName: 'Anita Sharma',
```

```
    Phone: '7777777777',
    City: 'Hyderabad'
  }
]
```

‘customers’:

```
insuranceDB> db.customers.insertMany([
...   {
...     "FirstName": "Srujana",
...     "LastName": "GL",
...     "DateOfBirth": ISODate("2003-05-12"),
...     "Phone": "9876543210",
...     "Email": "srujana@gmail.com"
...   },
...   {
...     "FirstName": "Harsh",
...     "LastName": "K",
...     "DateOfBirth": ISODate("2005-03-18"),
...     "Phone": "9000023456",
...     "Email": "harsh@gmail.com"
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('695766df7ee08e48ae1e2623'),
    '1': ObjectId('695766df7ee08e48ae1e2624')
  }
}
```

‘policies’

```
insuranceDB> db.policies.insertMany([
...   {
...     "PolicyName": "Life Shield",
...     "PolicyType": "Life",
...     "PremiumAmount": 15000,
...     "DurationYears": 10
```

```
...   },
...   {
...     "PolicyName": "Health Secure",
...     "PolicyType": "Health",
...     "PremiumAmount": 8000,
...     "DurationYears": 5
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6957686a7ee08e48ae1e2625'),
    '1': ObjectId('6957686a7ee08e48ae1e2626')
  }
}
```

‘policyassignments’

```
insuranceDB> db.policyAssignments.insertMany([
...   {
...     "CustomerID": 1, // Linking to Customers collection
...     "PolicyID": 1,   // Linking to Policies collection
...     "AgentID": 1,    // Linking to Agents collection
...     "StartDate": ISODate("2024-01-01"),
...     "EndDate": ISODate("2034-01-01")
...   },
...   {
...     "CustomerID": 2,
...     "PolicyID": 2,
...     "AgentID": 2,
...     "StartDate": ISODate("2024-02-01"),
...     "EndDate": ISODate("2029-02-01")
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6957694d7ee08e48ae1e2627'),
```

```
    '1': ObjectId('6957694d7ee08e48ae1e2628')
  }
}
```

‘claims’

```
insuranceDB> db.claims.insertMany([
...   {
...     "AssignmentID": 1,
...     "ClaimDate": ISODate("2024-06-10"),
...     "ClaimAmount": 50000,
...     "ClaimStatus": "Approved"
...   },
...   {
...     "AssignmentID": 2,
...     "ClaimDate": ISODate("2024-07-15"),
...     "ClaimAmount": 20000,
...     "ClaimStatus": "Pending"
...   }
... ]);
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('695769ba7ee08e48ae1e2629'),
    '1': ObjectId('695769ba7ee08e48ae1e262a')
  }
}
```

8. **MongoDB commands**

UpdateOne and Create Index

```
insuranceDB> db.agents.updateOne({ AgentName: "Ravi Kumar" }, {
$set: { agentId: 1 } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
```

```
  modifiedCount: 1,
  upsertedCount: 0
}
insuranceDB> db.agents.updateOne({ AgentName: "Anita Sharma" },
{ $set: { agentId: 2 } });
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
insuranceDB> db.agents.createIndex({ agentId: 1 }, { unique: true });
agentId_1
```

UpdateMany

```
insuranceDB> db.claims.updateMany(
...   { "ClaimStatus": "Pending" },
...   { $set: { "ClaimStatus": "Under Review" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

Using Update Operators

`$inc operator:`

```
insuranceDB> db.policies.updateMany(
...   { PolicyType: "Life" },
...   { $inc: { PremiumAmount: 500 } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
```

```
  modifiedCount: 1,
  upsertedCount: 0
}

$set operator:
insuranceDB> db.claims.updateMany(
...   { "ClaimStatus": "Pending" },
...   { $set: { "ClaimStatus": "Under Review" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}

$gt operator:
insuranceDB> db.policies.find({ "PremiumAmount": { $gt: 10000 } })
[
  {
    _id: ObjectId('6957686a7ee08e48ae1e2625'),
    PolicyName: 'Life Shield',
    PolicyType: 'Life',
    PremiumAmount: 15500,
    DurationYears: 10
  }
]

$rename:
insuranceDB> db.agents.updateMany(
...   {}, // Update all agents
...   { $rename: { "AgentName": "FullName" } }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 2,
```

```
  modifiedCount: 2,
  upsertedCount: 0
}
```

ReplaceOne()

```
insuranceDB> db.policies.replaceOne(
...   { "PolicyName": "Health Secure" },
...   {
...     "PolicyName": "Health Platinum",
...     "Type": "Medical",
...     "Price": 12000,
...     "Active": true
...   }
... )
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

DeleteOne()

```
insuranceDB> db.claims.deleteOne({ "AssignmentID": 1 })
{ acknowledged: true, deletedCount: 1 }
```

DeleteMany()

```
insuranceDB> db.claims.deleteMany({ "ClaimStatus": "Denied" })
{ acknowledged: true, deletedCount: 0 }
```

Query Operators

$eq

```
insuranceDB> db.customers.find({ age: { $eq: 25 } })
```

$lte

```
insuranceDB> db.claims.find({ claimAmount: { $lte: 50000 } })
```

$nor

```
insuranceDB> db.customers.find({
...   $nor: [
...     { city: "Delhi" },
...     { city: "Mumbai" }
...   ]
... })
...
[
  {
    _id: ObjectId('695766df7ee08e48ae1e2623'),
    FirstName: 'Srujana',
    LastName: 'GL',
    DateOfBirth: ISODate('2003-05-12T00:00:00.000Z'),
    Phone: '9876543210',
    Email: 'srujana@gmail.com'
  },
  {
    _id: ObjectId('695766df7ee08e48ae1e2624'),
    FirstName: 'Harsh',
    LastName: 'K',
    DateOfBirth: ISODate('2005-03-18T00:00:00.000Z'),
    Phone: '9000023456',
    Email: 'harsh@gmail.com'
  }
]
```

$regex

```
insuranceDB> db.customers.find({ name: { $regex: "^A" } })
db.customers.find({ name: { $regex: "del", $options: "i" } })
```

$text

```
insuranceDB> db.articles.createIndex({ content: "text" })
content_text
```

Aggregation Operators

$match

```
insuranceDB> db.policies.aggregate([
```

```
...   { $match: { policyType: "Health" } }
... ])
...
```

$group

```
insuranceDB> db.claims.aggregate([
...   {
...     $group: {
...       _id: "$policyId",
...       totalClaim: { $sum: "$claimAmount" }
...     }
...   }
... ])
...
[ { _id: null, totalClaim: 0 } ]
```

$skip

```
insuranceDB> db.policies.aggregate([
...   { $skip: 10 }
... ])
...
```

$count

```
insuranceDB> db.claims.aggregate([
...   { $count: "totalClaims" }
... ])
...
[ { totalClaims: 1 } ]
```