

Date: 02-01-2025

Day 5 - MongoDB Assignment

1. CREATING DATABASE

```
test> use InsuranceDB
switched to db InsuranceDB
```

2. INSERTIONS INTO THE InsuranceDB DATABASE

- Customers

```
InsuranceDB> db.customers.insertMany([
...   { customerId: 1, firstName: "Ajay", lastName: "Kumar", age: 21, phone: "9876543210", city: "Hyderabad" },
...   { customerId: 2, firstName: "Ravi", lastName: "Sharma", age: 30, phone: "9123456780", city: "Delhi" },
...   { customerId: 3, firstName: "Neha", lastName: "Singh", age: 27, phone: "9988776655", city: "Mumbai" },
...   { customerId: 4, firstName: "Anita", lastName: "Verma", age: 35, phone: "8899001122", city: "Pune" },
...   { customerId: 5, firstName: "Rahul", lastName: "Mehta", age: 40, phone: "7766554433", city: "Ahmedabad" }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('695750c1651abec0271e2621'),
    '1': ObjectId('695750c1651abec0271e2622'),
    '2': ObjectId('695750c1651abec0271e2623'),
    '3': ObjectId('695750c1651abec0271e2624'),
    '4': ObjectId('695750c1651abec0271e2625')
  }
}
```

- Agents

```
InsuranceDB> db.agents.insertMany([
...   { agentId: 1, agentName: "Suresh Rao", phone: "9000011111", city: "Hyderabad", experience: 5 },
...   { agentId: 2, agentName: "Kiran Patel", phone: "9000022222", city: "Ahmedabad", experience: 8 },
...   { agentId: 3, agentName: "Priya Nair", phone: "9000033333", city: "Kochi", experience: 4 },
...   { agentId: 4, agentName: "Amit Verma", phone: "9000044444", city: "Delhi", experience: 10 },
...   { agentId: 5, agentName: "Sneha Joshi", phone: "9000055555", city: "Pune", experience: 6 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('695750ec651abec0271e2626'),
    '1': ObjectId('695750ec651abec0271e2627'),
    '2': ObjectId('695750ec651abec0271e2628'),
    '3': ObjectId('695750ec651abec0271e2629'),
    '4': ObjectId('695750ec651abec0271e262a')
  }
}
```

- Policies

```
InsuranceDB> db.policies.insertMany([
...   { policyId: 101, policyName: "Life Secure", policyType: "Life", premium: 15000, duration: 20 },
...   { policyId: 102, policyName: "Health Plus", policyType: "Health", premium: 12000, duration: 10 },
...   { policyId: 103, policyName: "Car Protect", policyType: "Vehicle", premium: 8000, duration: 5 },
...   { policyId: 104, policyName: "Home Shield", policyType: "Property", premium: 10000, duration: 15 },
...   { policyId: 105, policyName: "Travel Safe", policyType: "Travel", premium: 5000, duration: 2 }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('695750fa651abec0271e262b'),
    '1': ObjectId('695750fa651abec0271e262c'),
    '2': ObjectId('695750fa651abec0271e262d'),
    '3': ObjectId('695750fa651abec0271e262e'),
    '4': ObjectId('695750fa651abec0271e262f')
  }
}
```

- Policy Assignments

```
InsuranceDB> db.policyAssignments.insertMany([
...   { assignmentId: 1, customerId: 1, policyId: 101, agentId: 1, startDate: "2023-01-10" },
...   { assignmentId: 2, customerId: 2, policyId: 102, agentId: 2, startDate: "2023-03-15" },
...   { assignmentId: 3, customerId: 3, policyId: 103, agentId: 3, startDate: "2022-06-20" },
...   { assignmentId: 4, customerId: 4, policyId: 104, agentId: 4, startDate: "2021-09-05" },
...   { assignmentId: 5, customerId: 5, policyId: 105, agentId: 5, startDate: "2024-02-01" }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('69575112651abec0271e2630'),
    '1': ObjectId('69575112651abec0271e2631'),
    '2': ObjectId('69575112651abec0271e2632'),
    '3': ObjectId('69575112651abec0271e2633'),
    '4': ObjectId('69575112651abec0271e2634')
  }
}
```

- Claims

```
InsuranceDB> db.claims.insertMany([
...   { claimId: 201, assignmentId: 1, claimAmount: 50000, status: "Approved" },
...   { claimId: 202, assignmentId: 2, claimAmount: 20000, status: "Pending" },
...   { claimId: 203, assignmentId: 3, claimAmount: 15000, status: "Rejected" },
...   { claimId: 204, assignmentId: 4, claimAmount: 80000, status: "Approved" },
...   { claimId: 205, assignmentId: 5, claimAmount: 10000, status: "Pending" }
... ])
...
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('6957511e651abec0271e2635'),
    '1': ObjectId('6957511e651abec0271e2636'),
    '2': ObjectId('6957511e651abec0271e2637'),
    '3': ObjectId('6957511e651abec0271e2638'),
    '4': ObjectId('6957511e651abec0271e2639')
  }
}
```

3.DISPLAYING COLLECTIONS IN InsuranceDB

```
InsuranceDB> show collections
agents
claims
customers
policies
policyAssignments
```

4.find()

```
InsuranceDB> db.customers.find()
[
  {
    _id: ObjectId('695750c1651abec0271e2621'),
    customerId: 1,
    firstName: 'Ajay',
    lastName: 'Kumar',
    age: 21,
    phone: '9876543210',
    city: 'Hyderabad'
  },
  .....
]
```

5.find() with condition

```
InsuranceDB> db.customers.find({ city: "Hyderabad" })
[
  {
    _id: ObjectId('695750c1651abec0271e2621'),
    customerId: 1,
    firstName: 'Ajay',
    lastName: 'Kumar',
    age: 21,
    phone: '9876543210',
    city: 'Hyderabad'
  }
]
```

5. findone()

```
InsuranceDB> db.policies.findOne({ policyType: "Life" })
{
  _id: ObjectId('695750fa651abec0271e262b'),
  policyId: 101,
  policyName: 'Life Secure',
  policyType: 'Life',
  premium: 15000,
  duration: 20
}
```

6. UpdateOne()

```
InsuranceDB> db.agents.updateOne(
...   { agentId: 1 },
...   { $set: { experience: 6 } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

7.UpdateMany()

```
InsuranceDB> db.customers.updateMany(
...   { city: "Mumbai" },
...   { $set: { city: "Navi Mumbai" } }
... )
...
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

8. deleteOne()

```
InsuranceDB> db.claims.deleteOne({ claimId: 203 })
{ acknowledged: true, deletedCount: 1 }
```

9. deleteMany()

```
InsuranceDB> db.customers.deleteMany({ city: "Chennai" })
{ acknowledged: true, deletedCount: 0 }
```

10. Sort()

```
InsuranceDB> db.agents.find().sort({ experience: -1 })
[
  {
    _id: ObjectId('695750ec651abec0271e2629'),
    agentId: 4,
    agentName: 'Amit Verma',
    phone: '9000044444',
    city: 'Delhi',
    experience: 10
  },
  {
    _id: ObjectId('695750ec651abec0271e2627'),
    agentId: 2,
    agentName: 'Kiran Patel',
    phone: '9000022222',
    city: 'Ahmedabad',
    experience: 8
  },
  .....
]
```

11.Comparison Operators

- Greater than

```
InsuranceDB> db.agents.find({ experience: { $gt: 5 } })
[
  {
    _id: ObjectId('695750ec651abec0271e2626'),
    agentId: 1,
    agentName: 'Suresh Rao',
    phone: '9000011111',
    city: 'Hyderabad',
    experience: 6
  },
  {
    _id: ObjectId('695750ec651abec0271e2627'),
    agentId: 2,
    agentName: 'Kiran Patel',
    phone: '9000022222',
    city: 'Ahmedabad',
    experience: 8
  }
].....
```

- Less than

```
InsuranceDB> db.policies.find({ premium: { $lt: 10000 } })
[
  {
    _id: ObjectId('695750fa651abec0271e262d'),
    policyId: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premium: 8000,
    duration: 5
  },
  {
  }
].....
```

- Between

```
InsuranceDB> db.customers.find({ age: { $gte: 25, $lte: 35 } })
[
  {
    _id: ObjectId('695750c1651abec0271e2622'),
    customerId: 2,
    firstName: 'Ravi',
    lastName: 'Sharma',
    age: 30,
    phone: '9123456780',
    city: 'Delhi'
  },
  {
  }
].....
```

12. Limit() & Skip()

- limit()

```
InsuranceDB> db.customers.find().limit(1)
[
  {
    _id: ObjectId('695750c1651abec0271e2621'),
    customerId: 1,
    firstName: 'Ajay',
    lastName: 'Kumar',
    age: 21,
    phone: '9876543210',
    city: 'Hyderabad'
  }
]
```

- Skip()

```
InsuranceDB> db.customers.find().skip(2).limit(2)
[
  {
    _id: ObjectId('695750c1651abec0271e2623'),
    customerId: 3,
    firstName: 'Neha',
    lastName: 'Singh',
    age: 27,
    phone: '9988776655',
    city: 'Navi Mumbai'
  },
  {
    _id: ObjectId('695750c1651abec0271e2624'),
    customerId: 4,
    firstName: 'Anita',
    lastName: 'Verma',
    age: 35,
    phone: '8899001122',
    city: 'Pune'
  }
]
```

13. \$lookup

```
InsuranceDB> db.policyAssignments.aggregate([
...   {
...     $lookup: {
...       from: "customers",
...       localField: "customerId",
...       foreignField: "customerId",
...       as: "customerDetails"
...     }
...   }
... ])
...
```