

Day-5-MongoDB Assignment

1. Create database

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

2. Show Databases

```
InsuranceDB> show dbs  
InsuranceDB 40.00 KiB  
admin      40.00 KiB  
config     104.00 KiB  
local      72.00 KiB
```

3. Create Collections

```
InsuranceDB> db.createCollection("Customers")  
{ ok: 1 }  
InsuranceDB> db.createCollection("Policies")  
{ ok: 1 }  
InsuranceDB> db.createCollection("Agents")  
{ ok: 1 }  
InsuranceDB> db.createCollection("Claims")  
{ ok: 1 }  
InsuranceDB> db.createCollection("PolicyAssignments")  
{ ok: 1 }
```

4. Insert into Collections

a) Customers collection

```
InsuranceDB> db.Customers.insertOne({  
...   CustomerId: 101,  
...   FirstName: "Manvitha",  
...   LastName: "Reddy",  
...   DateOfBirth: "2000-09-23",  
...   Phone: "8999389997",  
...   Email: "manvithareddy23@gmail.com"  
... });  
{  
  acknowledged: true,  
  insertedId: ObjectId('6957561e7990c1917e46b799')  
}  
InsuranceDB> db.Customers.insertOne({  
...   CustomerId: 102,  
...   FirstName: "Rajesh",  
...   LastName: "Sharma",  
...   DateOfBirth: "2000-04-15",  
...   Phone: "895938967",  
...   Email: "rajeshsharma13@gmail.com"  
... });  
{
```

```

    acknowledged: true,
    insertedId: ObjectId('695756847990c1917e46b79a')
}
InsuranceDB> db.Customers.insertOne({
...   CustomerId: 103,
...   FirstName: "Mohit",
...   LastName: "Oberoi",
...   DateOfBirth: "2001-04-17",
...   Phone: "8229389007",
...   Email: "mohittob@gmail.com"
... });
{
  acknowledged: true,
  insertedId: ObjectId('695756917990c1917e46b79b')
}
InsuranceDB> db.Customers.insertOne({
...   CustomerId: 104,
...   FirstName: "Priya",
...   LastName: "Singh",
...   DateOfBirth: "2000-09-29",
...   Phone: "8922380017",
...   Email: "priyas0929@gmail.com"
... });
{
  acknowledged: true,
  insertedId: ObjectId('6957569c7990c1917e46b79c')
}

```

b) Agents collection

```

InsuranceDB> db.Agents.insertMany([
...   {
...     AgentId: 1001,
...     AgentName: "Manoj",
...     Phone: "8897100567",
...     City: "Hyderabad"
...   },
...   {
...     AgentId: 1002,
...     AgentName: "Samanvi",
...     Phone: "9907100567",
...     City: "Mumbai"
...   },
...   {
...     AgentId: 1003,
...     AgentName: "Krishna",
...     Phone: "8894600710",
...     City: "Pune"
...   },
... ])

```

```

... {
...   AgentId: 1004,
...   AgentName: "Bhavani",
...   Phone: "9907132467",
...   City: "Mumbai"
... },
... {
...   AgentId: 1005,
...   AgentName: "Arjun",
...   Phone: "9802278570",
...   City: "Patna"
... }
... ]);

```

c) Policies collection

InsuranceDB> db.Policies.insertMany([

```

... {
...   PolicyId: 1,
...   PolicyName: "Jeevan Sathi",
...   PolicyType: "Life Insurance",
...   PremiumAmount: 500000,
...   DurationYears: 5
... },
... {
...   PolicyId: 2,
...   PolicyName: "Acko",
...   PolicyType: "Car Insurance",
...   PremiumAmount: 100000,
...   DurationYears: 5
... },
... {
...   PolicyId: 3,
...   PolicyName: "Star Health",
...   PolicyType: "Health Insurance",
...   PremiumAmount: 250000,
...   DurationYears: 5
... },
... {
...   PolicyId: 4,
...   PolicyName: "Safar Yojana",
...   PolicyType: "Travel Insurance",
...   PremiumAmount: 100000,
...   DurationYears: 1
... }
... ]);

```

{
 acknowledged: true,

```
insertedIds: {
  '0': ObjectId('695757597990c1917e46b79d'),
  '1': ObjectId('695757597990c1917e46b79e'),
  '2': ObjectId('695757597990c1917e46b79f'),
  '3': ObjectId('695757597990c1917e46b7a0'),
  '4': ObjectId('695757597990c1917e46b7a1')
}
}
```

d) PolicyAssignments collection

```
InsuranceDB> db.PolicyAssignments.insertMany([
```

```
...  {
...   AssignmentId: 2001,
...   CustomerId: 101,
...   PolicyId: 1,
...   AgentId: 1001,
...   StartDate: "2024-10-08",
...   EndDate: "2029-10-08"
... },
...  {
...   AssignmentId: 2002,
...   CustomerId: 102,
...   PolicyId: 2,
...   AgentId: 1002,
...   StartDate: "2024-12-06",
...   EndDate: "2029-12-06"
... },
...  {
...   AssignmentId: 2003,
...   CustomerId: 103,
...   PolicyId: 3,
...   AgentId: 1003,
...   StartDate: "2024-07-07",
...   EndDate: "2029-07-07"
... },
...  {
...   AssignmentId: 2004,
...   CustomerId: 104,
...   PolicyId: 4,
...   AgentId: 1004,
...   StartDate: "2024-02-26",
...   EndDate: "2025-02-26"
... }
```

```
... ]);
{
  acknowledged: true,
  insertedIds: {
```

```
'0': ObjectId('69575f7b7990c1917e46b7a2'),
'1': ObjectId('69575f7b7990c1917e46b7a3'),
'2': ObjectId('69575f7b7990c1917e46b7a4'),
'3': ObjectId('69575f7b7990c1917e46b7a5')
}
}
```

e) Claims collection

```
InsuranceDB> db.Claims.insertMany([
...   {
...     ClaimId: 1501,
...     AssignmentId: 2001,
...     ClaimDate: "2024-10-08",
...     ClaimAmount: 50000,
...     ClaimStatus: "Approved"
...   },
...   {
...     ClaimId: 1502,
...     AssignmentId: 2002,
...     ClaimDate: "2024-12-06",
...     ClaimAmount: 10000,
...     ClaimStatus: "Rejected"
...   },
...   {
...     ClaimId: 1503,
...     AssignmentId: 2003,
...     ClaimDate: "2024-07-07",
...     ClaimAmount: 25000,
...     ClaimStatus: "Rejected"
...   },
...   {
...     ClaimId: 1504,
...     AssignmentId: 2004,
...     ClaimDate: "2024-02-26",
...     ClaimAmount: 9000,
...     ClaimStatus: "Approved"
...   }
... ]);
{
  acknowledged: true,
  insertedIds: [
    '0': ObjectId('69575ffe7990c1917e46b7aa'),
    '1': ObjectId('69575ffe7990c1917e46b7ab'),
    '2': ObjectId('69575ffe7990c1917e46b7ac'),
    '3': ObjectId('69575ffe7990c1917e46b7ad')
  ]
}
```

5) MongoDB Find

a) find()

```
InsuranceDB> db.Customers.find()
[
  {
    _id: ObjectId('6957561e7990c1917e46b799'),
    CustomerId: 101,
    FirstName: 'Manvitha',
    LastName: 'Reddy',
    DateOfBirth: '2000-09-23',
    Phone: '8999389997',
    Email: 'manvithareddy23@gmail.com'
  },
  {
    _id: ObjectId('695756847990c1917e46b79a'),
    CustomerId: 102,
    FirstName: 'Rajesh',
    LastName: 'Sharma',
    DateOfBirth: '2000-04-15',
    Phone: '895938967',
    Email: 'rajeshsharma13@gmail.com'
  },
  {
    _id: ObjectId('695756917990c1917e46b79b'),
    CustomerId: 103,
    FirstName: 'Mohit',
    LastName: 'Oberoi',
    DateOfBirth: '2001-04-17',
    Phone: '8229389007',
    Email: 'mohittob@gmail.com'
  },
  {
    _id: ObjectId('6957569c7990c1917e46b79c'),
    CustomerId: 104,
    FirstName: 'Priya',
    LastName: 'Singh',
    DateOfBirth: '2000-09-29',
    Phone: '8922380017',
    Email: 'priyas0929@gmail.com'
  }
]
```

```
InsuranceDB> db.PolicyAssignments.find(
...   {},
...   {
...     _id: 0,
...     CustomerId: 1,
```

```

...   PolicyId: 1,
...   StartDate: 1,
...   EndDate: 1
... }
... );
[
{
  CustomerId: 101,
  PolicyId: 1,
  StartDate: '2024-10-08',
  EndDate: '2029-10-08'
},
{
  CustomerId: 102,
  PolicyId: 2,
  StartDate: '2024-12-06',
  EndDate: '2029-12-06'
},
{
  CustomerId: 103,
  PolicyId: 3,
  StartDate: '2024-07-07',
  EndDate: '2029-07-07'
},
{
  CustomerId: 104,
  PolicyId: 4,
  StartDate: '2024-02-26',
  EndDate: '2025-02-26'
}
]

```

b) **findOne()**

```

InsuranceDB> db.Agents.findOne()
{
  _id: ObjectId('695757597990c1917e46b79d'),
  AgentId: 1001,
  AgentName: 'Manoj',
  Phone: '8897100567',
  City: 'Hyderabad'
}

```

c) **find() - Querying Data**

```

InsuranceDB> db.Policies.find( {PolicyName: "Jeevan Sathi"} )
[
{
  _id: ObjectId('69575f7b7990c1917e46b7a2'),
  PolicyId: 1,

```

```
    PolicyName: 'Jeevan Sathi',
    PolicyType: 'Life Insurance',
    PremiumAmount: 500000,
    DurationYears: 5
}
]
```

6) Update

a) updateOne()

```
InsuranceDB> db.Customers.updateOne(
...  { CustomerId: 101 },
...  { $set: { Email: "manvitha.new@gmail.com" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
InsuranceDB> db.Claims.updateOne(
...  { ClaimId: 1502 },
...  { $set: { ClaimStatus: "Approved" } }
... );
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

b) updateMany()

i) \$set

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

ii) \$inc

```
InsuranceDB> db.Policies.updateMany(  
... { PolicyType: "Health Insurance" },  
... { $inc: { PremiumAmount: 25000 } }  
...  
... );  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

iii) \$rename

```
InsuranceDB> db.Policies.updateOne(  
... { PolicyId: 2 },  
... { $rename: { PolicyType: "Allianz Premier" } }  
... );  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

7) Query Operator

a) Comparison Operator

i) \$eq: Values are equal

```
InsuranceDB> db.Customers.find({ CustomerId: { $eq: 101 } });  
[  
 {  
   _id: ObjectId('6957561e7990c1917e46b799'),  
   CustomerId: 101,  
   FirstName: 'Manvitha',  
   LastName: 'Reddy',  
   DateOfBirth: '2000-09-23',  
   Phone: '8999389997',  
   Email: 'manvitha.new@gmail.com'  
 }
```

ii) \$ne — Values are NOT equal

```
InsuranceDB> db.Claims.find({ ClaimStatus: { $ne: "Approved" } });  
[
```

```
{
  _id: ObjectId('69575ffe7990c1917e46b7ac'),
  ClaimId: 1503,
  AssignmentId: 2003,
  ClaimDate: '2024-07-07',
  ClaimAmount: 25000,
  ClaimStatus: 'Under Review'
}
]
```

iii) \$gt : Value is greater than another value

```
InsuranceDB> db.Policies.find({ PremiumAmount: { $gt: 200000 } });
```

```
[
  {
    _id: ObjectId('695773aa7990c1917e46b7ae'),
    PolicyId: 1,
    PolicyName: 'Jeevan Sathi',
    PolicyType: 'Life Insurance',
    PremiumAmount: 500000,
    DurationYears: 5
  },
  {
    _id: ObjectId('695773aa7990c1917e46b7b0'),
    PolicyId: 3,
    PolicyName: 'Star Health',
    PolicyType: 'Health Insurance',
    PremiumAmount: 275000,
    DurationYears: 5
  }
]
```

iv) \$lte: Value is less than or equal to another value

```
InsuranceDB> db.Claims.find({ ClaimAmount: { $lte: 10000 } });
```

```
[
  {
    _id: ObjectId('69575ffe7990c1917e46b7ab'),
    ClaimId: 1502,
    AssignmentId: 2002,
    ClaimDate: '2024-12-06',
    ClaimAmount: 10000,
    ClaimStatus: 'Approved'
  },
  {
    _id: ObjectId('69575ffe7990c1917e46b7ad'),
    ClaimId: 1504,
    AssignmentId: 2004,
    ClaimDate: '2024-02-26',
    ClaimAmount: 9000,
  }
]
```

```
    ClaimStatus: 'Approved'  
  }  
]
```

v) Find policies with PremiumAmount between 100000 and 400000

```
InsuranceDB> db.Policies.find({  
...   PremiumAmount: { $gte: 100000, $lt: 400000 }  
... });  
[  
  {  
    _id: ObjectId('695773aa7990c1917e46b7af'),  
    PolicyId: 2,  
    PolicyName: 'Acko',  
    PremiumAmount: 100000,  
    DurationYears: 5,  
    'Allianz Premier': 'Car Insurance'  
  },  
  {  
    _id: ObjectId('695773aa7990c1917e46b7b0'),  
    PolicyId: 3,  
    PolicyName: 'Star Health',  
    PolicyType: 'Health Insurance',  
    PremiumAmount: 275000,  
    DurationYears: 5  
  },  
  {  
    _id: ObjectId('695773aa7990c1917e46b7b1'),  
    PolicyId: 4,  
    PolicyName: 'Safar Yojana',  
    PolicyType: 'Travel Insurance',  
    PremiumAmount: 100000,  
    DurationYears: 1  
  }  
]
```

b)Logical Operators

i) \$or: Either condition matches

```
InsuranceDB> db.Policies.find({  
...   $or: [  
...     { PolicyType: "Life Insurance" },  
...     { PolicyType: "Health Insurance" }  
...   ]  
... });  
[  
  {  
    _id: ObjectId('695773aa7990c1917e46b7ae'),
```

```

PolicyId: 1,
PolicyName: 'Jeevan Sathi',
PolicyType: 'Life Insurance',
PremiumAmount: 500000,
DurationYears: 5
},
{
  _id: ObjectId('695773aa7990c1917e46b7b0'),
  PolicyId: 3,
  PolicyName: 'Star Health',
  PolicyType: 'Health Insurance',
  PremiumAmount: 275000,
  DurationYears: 5
}
]

```

ii) \$nor: Either condition matches

```

InsuranceDB> db.Policies.find({
...   $nor: [
...     { PolicyType: "Life Insurance" },
...     { PolicyType: "Health Insurance" }
...   ]
... });
[
{
  _id: ObjectId('695773aa7990c1917e46b7af'),
  PolicyId: 2,
  PolicyName: 'Acko',
  PremiumAmount: 100000,
  DurationYears: 5,
  'Allianz Premier': 'Car Insurance'
},
{
  _id: ObjectId('695773aa7990c1917e46b7b1'),
  PolicyId: 4,
  PolicyName: 'Safar Yojana',
  PolicyType: 'Travel Insurance',
  PremiumAmount: 100000,
  DurationYears: 1
}
]

```

iii)\$not: Condition does matches

```

InsuranceDB> db.Claims.find({
...   ClaimAmount: { $not: { $gt: 20000 } }
... });
[

```

```
{
  _id: ObjectId('69575ffe7990c1917e46b7ab'),
  ClaimId: 1502,
  AssignmentId: 2002,
  ClaimDate: '2024-12-06',
  ClaimAmount: 10000,
  ClaimStatus: 'Approved'
},
{
  _id: ObjectId('69575ffe7990c1917e46b7ad'),
  ClaimId: 1504,
  AssignmentId: 2004,
  ClaimDate: '2024-02-26',
  ClaimAmount: 9000,
  ClaimStatus: 'Approved'
}
]
```

8) Aggregations

a) \$match — Filter documents

```
InsuranceDB> db.Claims.aggregate([
... { $match: { ClaimStatus: "Approved" } }
... ]);
[
  {
    _id: ObjectId('69575ffe7990c1917e46b7aa'),
    ClaimId: 1501,
    AssignmentId: 2001,
    ClaimDate: '2024-10-08',
    ClaimAmount: 50000,
    ClaimStatus: 'Approved'
  },
  {
    _id: ObjectId('69575ffe7990c1917e46b7ab'),
    ClaimId: 1502,
    AssignmentId: 2002,
    ClaimDate: '2024-12-06',
    ClaimAmount: 10000,
    ClaimStatus: 'Approved'
  },
  {
    _id: ObjectId('69575ffe7990c1917e46b7ad'),
    ClaimId: 1504,
    AssignmentId: 2004,
    ClaimDate: '2024-02-26',
    ClaimAmount: 9000,
    ClaimStatus: 'Approved'
  }
]
```

b) Group By and \$sum

```
InsuranceDB> db.Claims.aggregate([
...   {
...     $group: {
...       _id: "$ClaimStatus",
...       TotalAmount: { $sum: "$ClaimAmount" }
...     }
...   }
... ]);
[
  { _id: 'Approved', TotalAmount: 69000 },
  { _id: 'Under Review', TotalAmount: 25000 }
]
```

c) Show top 2 highest premium policies

sort() + limit()

```
InsuranceDB> db.Policies.aggregate([
...   { $sort: { PremiumAmount: -1 } },
...   { $limit: 2 }
... ]);
[
  {
    _id: ObjectId('695773aa7990c1917e46b7ae'),
    PolicyId: 1,
    PolicyName: 'Jeevan Sathi',
    PolicyType: 'Life Insurance',
    PremiumAmount: 500000,
    DurationYears: 5
  },
  {
    _id: ObjectId('695773aa7990c1917e46b7b0'),
    PolicyId: 3,
    PolicyName: 'Star Health',
    PolicyType: 'Health Insurance',
    PremiumAmount: 275000,
    DurationYears: 5
  }
]
```

d) combine data from PolicyAssignments and Customers collections.

\$lookup and \$project

```
InsuranceDB> db.PolicyAssignments.aggregate([
...   {
...     $lookup: {
...       from: "Customers",
...       localField: "CustomerId",
...       foreignField: "Id",
...       as: "Customer"
...     }
...   }
... ])
```

```
...     foreignField: "CustomerId",
...     as: "CustomerDetails"
...   }
... },
... {
...   $project: {
...     _id: 0,
...     AssignmentId: 1,
...     PolicyId: 1,
...     StartDate: 1,
...     EndDate: 1,
...     "CustomerDetails.FirstName": 1,
...     "CustomerDetails.LastName": 1
...   }
... }
... ]);
[
{
  AssignmentId: 2001,
  PolicyId: 1,
  StartDate: '2024-10-08',
  EndDate: '2029-10-08',
  CustomerDetails: [ { FirstName: 'Manvitha', LastName: 'Reddy' } ]
},
{
  AssignmentId: 2002,
  PolicyId: 2,
  StartDate: '2024-12-06',
  EndDate: '2029-12-06',
  CustomerDetails: [ { FirstName: 'Rajesh', LastName: 'Sharma' } ]
},
{
  AssignmentId: 2003,
  PolicyId: 3,
  StartDate: '2024-07-07',
  EndDate: '2029-07-07',
  CustomerDetails: [ { FirstName: 'Mohit', LastName: 'Oberoi' } ]
},
{
  AssignmentId: 2004,
  PolicyId: 4,
  StartDate: '2024-02-26',
  EndDate: '2025-02-26',
  CustomerDetails: [ { FirstName: 'Priya', LastName: 'Singh' } ]
}
]
```

9) Count approved claims

```
InsuranceDB> db.Claims.countDocuments({ ClaimStatus: "Approved" });
3
```

10) Delete

a) deleteOne()

```
InsuranceDB> db.PolicyAssignments.deleteOne({ AssignmentId: 2004 });
{ acknowledged: true, deletedCount: 1 }
```

b) deleteMany()

```
InsuranceDB> db.Policies.deleteMany({ DurationYears: 5 });
{ acknowledged: true, deletedCount: 3 }
```