

Day 5-MongoDB Assignment

```
use InsuranceDB

db.createCollection("customers")
db.createCollection("agents")
db.createCollection("policies")
db.createCollection("policyAssignments")
db.createCollection("claims")

db.customers.insertMany([
  {_id:1,FirstName:"Ram",LastName:"Kumar",DateOfBirth:"2004-05-02",Phone:8976543210,Email:"ram@gmail.com"},

  {_id:2,FirstName:"Sita",LastName:"Reddy",DateOfBirth:"2003-06-12",Phone:9123456780,Email:"sita@gmail.com"},

  {_id:3,FirstName:"Arjun",LastName:"Verma",DateOfBirth:"2002-03-22",Phone:9012345678,Email:"arjun@gmail.com"},

  {_id:4,FirstName:"Neha",LastName:"Sharma",DateOfBirth:"2004-09-10",Phone:9988776655,Email:"neha@gmail.com"},

  {_id:5,FirstName:"Rahul",LastName:"Mehta",DateOfBirth:"2001-01-15",Phone:9876543211,Email:"rahul@gmail.com"},

  {_id:6,FirstName:"Anita",LastName:"Singh",DateOfBirth:"2003-07-19",Phone:9765432109,Email:"anita@gmail.com"},

  {_id:7,FirstName:"Kiran",LastName:"Patel",DateOfBirth:"2002-11-30",Phone:9654321098,Email:"kiran@gmail.com"},

  {_id:8,FirstName:"Pooja",LastName:"Gupta",DateOfBirth:"2004-04-08",Phone:9543210987,Email:"pooja@gmail.com"},

  {_id:9,FirstName:"Vikas",LastName:"Nair",DateOfBirth:"2001-12-25",Phone:9432109876,Email:"vikas@gmail.com"},

  {_id:10,FirstName:"Sneha",LastName:"Iyer",DateOfBirth:"2003-10-05",Phone:9321098765,Email:"sneha@gmail.com"}])

]);
```

```
db.agents.insertMany([
  {_id:1,AgentName:"Ramesh",Phone:9000011111,City:"Hyderabad"},
  {_id:2,AgentName:"Suresh",Phone:9000022222,City:"Chennai"},
  {_id:3,AgentName:"Mahesh",Phone:9000033333,City:"Bangalore"},
  {_id:4,AgentName:"Naresh",Phone:9000044444,City:"Pune"},
  {_id:5,AgentName:"Rajesh",Phone:9000055555,City:"Mumbai"},
  {_id:6,AgentName:"Karthik",Phone:9000066666,City:"Coimbatore"},
  {_id:7,AgentName:"Venkatesh",Phone:9000077777,City:"Vijayawada"},
  {_id:8,AgentName:"Praveen",Phone:9000088888,City:"Delhi"},
  {_id:9,AgentName:"Anil",Phone:9000099999,City:"Kolkata"},
  {_id:10,AgentName:"Sunil",Phone:9000000000,City:"Jaipur"}
]);
```

```
db.policies.insertMany([
  {_id:1,PolicyName:"Life Secure",PolicyType:"Life",PremiumAmount:6000,DurationYears:15},
  {_id:2,PolicyName:"Health Plus",PolicyType:"Health",PremiumAmount:4500,DurationYears:10},
  {_id:3,PolicyName:"Car Protect",PolicyType:"Vehicle",PremiumAmount:3000,DurationYears:5},
  {_id:4,PolicyName:"Home Shield",PolicyType:"Home",PremiumAmount:5500,DurationYears:20},
  {_id:5,PolicyName:"Travel Safe",PolicyType:"Travel",PremiumAmount:2000,DurationYears:2},
  {_id:6,PolicyName:"Life Premium",PolicyType:"Life",PremiumAmount:8000,DurationYears:25},
  {_id:7,PolicyName:"Health Gold",PolicyType:"Health",PremiumAmount:7000,DurationYears:15},
  {_id:8,PolicyName:"Bike Cover",PolicyType:"Vehicle",PremiumAmount:1500,DurationYears:3},
  {_id:9,PolicyName:"Senior Life",PolicyType:"Life",PremiumAmount:5000,DurationYears:10},
```

```
{_id:10,PolicyName:"Student  
Health",PolicyType:"Health",PremiumAmount:2500,DurationYears:5}  
]);  
  
db.policyAssignments.insertMany([  
{_id:101,customerId:1,policyId:1,agentId:1,startDate:"2023-01-01",endDate:"2038-01-01"},  
{_id:102,customerId:1,policyId:2,agentId:2,startDate:"2023-02-01",endDate:"2033-02-01"},  
{_id:103,customerId:1,policyId:3,agentId:3,startDate:"2023-03-01",endDate:"2028-03-01"},  
{_id:104,customerId:2,policyId:4,agentId:4,startDate:"2023-01-15",endDate:"2043-01-15"},  
{_id:105,customerId:2,policyId:5,agentId:5,startDate:"2023-02-15",endDate:"2025-02-15"},  
{_id:106,customerId:3,policyId:6,agentId:6,startDate:"2023-03-10",endDate:"2048-03-10"},  
{_id:107,customerId:3,policyId:7,agentId:7,startDate:"2023-04-10",endDate:"2038-04-10"},  
{_id:108,customerId:4,policyId:8,agentId:8,startDate:"2023-05-05",endDate:"2026-05-05"},  
{_id:109,customerId:4,policyId:9,agentId:9,startDate:"2023-06-05",endDate:"2033-06-05"},  
{_id:110,customerId:5,policyId:10,agentId:10,startDate:"2023-07-01",endDate:"2028-07-01"}  
]);
```

```
db.claims.insertMany([  
{ _id:1001, assignmentId:101, claimDate:"2024-01-10", claimAmount:50000,  
claimStatus:"Approved" },  
  
{ _id:1002, assignmentId:101, claimDate:"2024-03-15", claimAmount:30000,  
claimStatus:"Pending" },  
  
{ _id:1003, assignmentId:102, claimDate:"2024-02-20", claimAmount:45000,  
claimStatus:"Approved" },  
  
{ _id:1004, assignmentId:103, claimDate:"2024-04-05", claimAmount:25000,  
claimStatus:"Rejected" },  
  
{ _id:1005, assignmentId:103, claimDate:"2024-06-18", claimAmount:40000,  
claimStatus:"Approved" },
```

```
{ _id:1006, assignmentId:104, claimDate:"2024-05-10", claimAmount:120000,  
claimStatus:"Approved" },  
  
{ _id:1007, assignmentId:105, claimDate:"2024-07-22", claimAmount:15000,  
claimStatus:"Pending" },  
  
{ _id:1008, assignmentId:106, claimDate:"2024-08-30", claimAmount:300000,  
claimStatus:"Approved" },  
  
{ _id:1009, assignmentId:107, claimDate:"2024-09-14", claimAmount:80000,  
claimStatus:"Approved" },  
  
{ _id:1010, assignmentId:108, claimDate:"2024-10-25", claimAmount:20000,  
claimStatus:"Rejected" }  
  
]);
```

1. Display one Agent

```
db.agents.findOne()
```

```
InsuranceDB> db.agents.findOne()  
{ _id: 1, AgentName: 'Ramesh', Phone: 9000011111, City: 'Hyderabad' }
```

2. Show only FirstName and Email of Customers

```
db.customers.find({}, { FirstName: 1, Email: 1, _id: 0 })
```

```
InsuranceDB> db.customers.find({}, { FirstName: 1, Email: 1, _id: 0 })  
[  
  { FirstName: 'Ram', Email: 'ram@gmail.com' },  
  { FirstName: 'Sita', Email: 'sita@gmail.com' },  
  { FirstName: 'Arjun', Email: 'arjun@gmail.com' },  
  { FirstName: 'Neha', Email: 'neha@gmail.com' },  
  { FirstName: 'Rahul', Email: 'rahul@gmail.com' },  
  { FirstName: 'Anita', Email: 'anita@gmail.com' },  
  { FirstName: 'Kiran', Email: 'kiran@gmail.com' },  
  { FirstName: 'Pooja', Email: 'pooja@gmail.com' },  
  { FirstName: 'Vikas', Email: 'vikas@gmail.com' },  
  { FirstName: 'Sneha', Email: 'sneha@gmail.com' }  
]
```

3.4Find claims with amount greater than 50,000

```
db.claims.find({ claimAmount: { $gt: 50000 } })
```

```
InsuranceDB> db.claims.find({ claimAmount: { $gt: 50000 } })
[
  {
    _id: 1006,
    assignmentId: 104,
    claimDate: '2024-05-10',
    claimAmount: 120000,
    claimStatus: 'Approved'
  },
  {
    _id: 1008,
    assignmentId: 106,
    claimDate: '2024-08-30',
    claimAmount: 300000,
    claimStatus: 'Approved'
  },
  {
    _id: 1009,
    assignmentId: 107,
    claimDate: '2024-09-14',
    claimAmount: 80000,
    claimStatus: 'Approved'
  }
]
```

4.Get all policies of a customer where customerId = 1

```
db.policyAssignments.find({ customerId: 1 })
```

```
InsuranceDB> db.policyAssignments.find({customerId: 1 })
[
  {
    _id: 101,
    customerId: 1,
    policyId: 1,
    agentId: 1,
    startDate: '2023-01-01',
    endDate: '2038-01-01'
  },
  {
    _id: 102,
    customerId: 1,
    policyId: 2,
    agentId: 2,
    startDate: '2023-02-01',
    endDate: '2033-02-01'
  },
  {
    _id: 103,
    customerId: 1,
    policyId: 3,
    agentId: 3,
    startDate: '2023-03-01',
    endDate: '2028-03-01'
  }
]
```

5.Update agentid to 6 for a policy assignment with id 120

```
db.policyAssignments.updateOne({ _id: 120 }, { $set: { agentId: 6 } })
```

```
InsuranceDB> db.policyAssignments.updateOne( { _id: 120 }, { $set: { agentId: 6 } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

6.Update claim status where claimed is 1020

```
db.claims.updateOne({ _id: 1020 }, { $set: { claimStatus: "Approved" } })
```

```
InsuranceDB> db.claims.updateOne({ _id: 1020 }, { $set: { claimStatus: "Approved" } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 0,
  modifiedCount: 0,
  upsertedCount: 0
}
```

7.Display Policies of type Life

```
db.policies.find({ PolicyType: { $regex: "^Life$" } })
```

```
InsuranceDB> db.policies.find({ PolicyType: { $regex: "^Life$" } })
[
  {
    _id: 1,
    PolicyName: 'Life Secure',
    PolicyType: 'Life',
    PremiumAmount: 6000,
    DurationYears: 15
  },
  {
    _id: 6,
    PolicyName: 'Life Premium',
    PolicyType: 'Life',
    PremiumAmount: 8000,
    DurationYears: 25
  },
  {
    _id: 9,
    PolicyName: 'Senior Life',
    PolicyType: 'Life',
    PremiumAmount: 5000,
    DurationYears: 10
  }
]
```

8.Customer FirstNames starting with R

```
db.customers.find({ FirstName: { $regex: "^R" } })
```

```
InsuranceDB> db.customers.find({ FirstName: { $regex: "^R" } })
[
  {
    _id: 1,
    FirstName: 'Ram',
    LastName: 'Kumar',
    DateOfBirth: '2004-05-02',
    Phone: 8976543210,
    Email: 'ram@gmail.com'
  },
  {
    _id: 5,
    FirstName: 'Rahul',
    LastName: 'Mehta',
    DateOfBirth: '2001-01-15',
    Phone: 9876543211,
    Email: 'rahul@gmail.com'
  }
]
```

9.Find policies whose premium amount is between 3000 and 7000.

```
db.policies.find({PremiumAmount: { $gte: 3000, $lte: 7000 } })
```

```
InsuranceDB> db.policies.find({PremiumAmount: { $gte: 3000, $lte: 7000 } })
[
  {
    _id: 1,
    PolicyName: 'Life Secure',
    PolicyType: 'Life',
    PremiumAmount: 6000,
    DurationYears: 15
  },
  {
    _id: 2,
    PolicyName: 'Health Plus',
    PolicyType: 'Health',
    PremiumAmount: 4500,
    DurationYears: 10
  },
  {
    _id: 3,
    PolicyName: 'Car Protect',
    PolicyType: 'Vehicle',
    PremiumAmount: 3000,
    DurationYears: 5
  },
  {
    _id: 4,
    PolicyName: 'Home Shield',
    PolicyType: 'Home',
    PremiumAmount: 5500,
    DurationYears: 20
  },
  {
    _id: 7,
    PolicyName: 'Health Gold',
    PolicyType: 'Health',
    PremiumAmount: 7000,
    DurationYears: 15
  },
  {
    _id: 9,
    PolicyName: 'Senior Life',
    PolicyType: 'Life',
    PremiumAmount: 5000,
    DurationYears: 10
  }
]
```

10. Display claims that are either Pending or Rejected.

```
db.claims.find({claimStatus: { $in: ["Pending", "Rejected"] }})
```

```
InsuranceDB> db.claims.find({claimStatus: { $in: ["Pending", "Rejected"] }})
[
  {
    _id: 1002,
    assignmentId: 101,
    claimDate: '2024-03-15',
    claimAmount: 30000,
    claimStatus: 'Pending'
  },
  {
    _id: 1004,
    assignmentId: 103,
    claimDate: '2024-04-05',
    claimAmount: 25000,
    claimStatus: 'Rejected'
  },
  {
    _id: 1007,
    assignmentId: 105,
    claimDate: '2024-07-22',
    claimAmount: 15000,
    claimStatus: 'Pending'
  },
  {
    _id: 1010,
    assignmentId: 108,
    claimDate: '2024-10-25',
    claimAmount: 20000,
    claimStatus: 'Rejected'
  }
]
```

11. Display policies sorted by premium amount in ascending order.

```
db.policies.find().sort({ PremiumAmount: 1 })
```

```
InsuranceDB> db.policies.find().sort({ PremiumAmount: 1 })
[
  {
    _id: 8,
    PolicyName: 'Bike Cover',
    PolicyType: 'Vehicle',
    PremiumAmount: 1500,
    DurationYears: 3
  },
  {
    _id: 5,
    PolicyName: 'Travel Safe',
    PolicyType: 'Travel',
    PremiumAmount: 2000,
    DurationYears: 2
  },
  {
    _id: 10,
    PolicyName: 'Student Health',
    PolicyType: 'Health',
    PremiumAmount: 2500,
    DurationYears: 5
  },
  {
    _id: 3,
    PolicyName: 'Car Protect',
    PolicyType: 'Vehicle',
    PremiumAmount: 3000,
    DurationYears: 5
  },
  {
    _id: 2,
    PolicyName: 'Health Plus',
    PolicyType: 'Health',
    PremiumAmount: 4500,
    DurationYears: 10
  },
  {
    _id: 9,
    PolicyName: 'Senior Life',
    PolicyType: 'Life',
    PremiumAmount: 5000,
    DurationYears: 10
  }
]
```

12. Display the second highest claim amount.

```
db.claims.find().sort({ claimAmount: -1 }).skip(1).limit(1)
```

```
InsuranceDB> db.claims.find().sort({ claimAmount: -1 }).skip(1).limit(1)
[
  {
    _id: 1006,
    assignmentId: 104,
    claimDate: '2024-05-10',
    claimAmount: 120000,
    claimStatus: 'Approved'
  }
]
```

13. Delete all rejected claims.

```
db.claims.deleteMany({ claimStatus: "Rejected" })
```

```
InsuranceDB> db.claims.deleteMany({ claimStatus: "Rejected" })
{ acknowledged: true, deletedCount: 2 }
```

14. Update the end date of all policies of customer ID 1.

```
db.policyAssignments.updateMany({ customerId: 1 }, { $set: { endDate: "2035-12-31" } })
```

```
InsuranceDB> db.policyAssignments.updateMany({ customerId: 1 }, { $set: { endDate: "2035-12-31" } })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

15. Find the average claim Amount

```
db.claims.aggregate([
  { $group: { _id: null, averageClaimAmount: { $avg: "$claimAmount" } } }
])
```

```
[ { _id: null, averageClaimAmount: 85000 } ]
```

16. Count claims per status

```
db.claims.aggregate([
  { $group: { _id: "$claimStatus", totalClaims: { $sum: 1 } } }
])
```

```
InsuranceDB> db.claims.aggregate([
...   { $group: { _id: "$claimStatus", totalClaims: { $sum: 1 } } }
... ])
[
  { _id: 'Approved', totalClaims: 6 },
  { _id: 'Pending', totalClaims: 2 }
]
```

17. Add GST (18%) to premium (calculated field)

```
db.policies.aggregate([ { $project: { customerName: 1, totalPremium: { $multiply: ["$premium", 1.18] } } } ])
```

```
[ { PolicyName: 'Life Secure', totalPremium: 7080 }, { PolicyName: 'Health Plus', totalPremium: 5310 }, { PolicyName: 'Car Protect', totalPremium: 3540 }, { PolicyName: 'Home Shield', totalPremium: 6490 }, { PolicyName: 'Travel Safe', totalPremium: 2360 }, { PolicyName: 'Life Premium', totalPremium: 9440 }, { PolicyName: 'Health Gold', totalPremium: 8260 }, { PolicyName: 'Bike Cover', totalPremium: 1770 }, { PolicyName: 'Senior Life', totalPremium: 5900 }, { PolicyName: 'Student Health', totalPremium: 2950 } ]
```

18. Highest premium per policyType

```
db.policies.aggregate([ { $group: { _id: "$policyType", maxPremium: { $max: "$premium" } } } ])
```

```
InsuranceDB> db.policies.aggregate([ { $group: { _id: "$policyType", maxPremium: { $max: "$PremiumAmount" } } } ]) [ { _id: null, maxPremium: 8000 } ]
```

19. Count number of agents in each city.

```
db.agents.aggregate([ { $group: { _id: "$City", totalAgents: { $sum: 1 } } } ])
```

```
InsuranceDB> db.agents.aggregate([ { $group: { _id: "$City", totalAgents: { $sum: 1 } } } ]) [ { _id: 'Delhi', totalAgents: 1 }, { _id: 'Pune', totalAgents: 1 }, { _id: 'Bangalore', totalAgents: 1 }, { _id: 'Hyderabad', totalAgents: 1 }, { _id: 'Chennai', totalAgents: 1 }, { _id: 'Mumbai', totalAgents: 1 }, { _id: 'Coimbatore', totalAgents: 1 }, { _id: 'Vijayawada', totalAgents: 1 }, { _id: 'Kolkata', totalAgents: 1 }, { _id: 'Jaipur', totalAgents: 1 } ]
```

20. Find claims with claim amount greater than 50,000.

```
db.claims.find({ claimAmount: { $gt: 50000 } })
```

```
InsuranceDB> db.claims.find({ claimAmount: { $gt: 50000 } }) [ { _id: 1006, assignmentId: 104, claimDate: '2024-05-10', claimAmount: 120000, claimStatus: 'Approved' }, { _id: 1008, assignmentId: 106, claimDate: '2024-08-30', claimAmount: 300000, claimStatus: 'Approved' }, { _id: 1009, assignmentId: 107, claimDate: '2024-09-14', claimAmount: 80000, claimStatus: 'Approved' } ]
```

21. Count number of agents in each city.

```
db.agents.aggregate([{"$group": {"_id": "$City", "agentCount": {"$sum": 1}}})
```

```
[  
  { _id: 'Delhi', agentCount: 1 },  
  { _id: 'Pune', agentCount: 1 },  
  { _id: 'Bangalore', agentCount: 1 },  
  { _id: 'Hyderabad', agentCount: 1 },  
  { _id: 'Chennai', agentCount: 1 },  
  { _id: 'Mumbai', agentCount: 1 },  
  { _id: 'Coimbatore', agentCount: 1 },  
  { _id: 'Vijayawada', agentCount: 1 },  
  { _id: 'Kolkata', agentCount: 1 },  
  { _id: 'Jaipur', agentCount: 1 }  
]
```

22. Display all customers who have taken more than one policy.

```
db.policyAssignments.aggregate([{"$group": {"_id": "$customerId", "policyCount": {"$sum": 1}}},  
  {"$match": {"policyCount": {"$gt": 1}}}] )
```

```
[  
  { _id: 4, policyCount: 2 },  
  { _id: 2, policyCount: 2 },  
  { _id: 1, policyCount: 3 },  
  { _id: 3, policyCount: 2 }  
]
```

23. Find the policy that has the highest premium amount.

```
db.policies.find().sort({ PremiumAmount: -1 }).limit(1)
```

```
InsuranceDB> db.policies.find().sort({ PremiumAmount: -1 }).limit(1)  
[  
  {  
    _id: 6,  
    PolicyName: 'Life Premium',  
    PolicyType: 'Life',  
    PremiumAmount: 8000,  
    DurationYears: 25  
  }  
]
```

24. Display policies that have received at least one claim.

```
db.policyAssignments.aggregate([{$lookup: {  
  from: "claims",  
  localField: "_id",  
  foreignField: "assignmentId",  
  as: "claims"}},  
  {"$match": {"claims": {"$exists": true}}},  
  {"$group": {"_id": "$_id", "count": {"$sum": 1}}},  
  {"$sort": {"count": -1}}])
```

```
as: "claims"}},{$match: {
  claims: { $ne: [] }}

},{$project: {policyId: 1, _id: 0}}])
```

```
[
  { policyId: 1 },
  { policyId: 2 },
  { policyId: 3 },
  { policyId: 4 },
  { policyId: 5 },
  { policyId: 6 },
  { policyId: 7 }
```

25. Display the total premium amount collected for each policy type.

```
db.policies.aggregate([{$group: { _id: "$PolicyType",totalPremiumCollected: { $sum: "$PremiumAmount" }}}])
```

```
[
  { _id: 'Vehicle', totalPremiumCollected: 4500 },
  { _id: 'Life', totalPremiumCollected: 19000 },
  { _id: 'Health', totalPremiumCollected: 14000 },
  { _id: 'Travel', totalPremiumCollected: 2000 },
  { _id: 'Home', totalPremiumCollected: 5500 }
```