Day 5 MongoDB Assignments

1. Create Database insuranceDB

```
> use insuranceDB;
< switched to db insuranceDB
```

2. Create Collection commands

db.createCollection("Customers")

db.createCollection("Policies");

db.createCollection("Agents")

db.createCollection("PolicyAssignments");

db.createCollection("Claims");

db.createCollection("Payments");

3. Insert Data Commands

#Customers Data

db.customers.insertMany([

{ customerid: 1, firstname: "Rahul", lastname: "Sharma", dob: ISODate("1995-12-12"), phone: "9704873975", city: "Mumbai"},

{ customerid: 2, firstname: "Neha", lastname: "Shetty", dob: ISODate("1999-07-12"), phone: "9014486579", city: "Bangalore"},

{ customerid: 3, firstname: "Amit", lastname: "Patel", dob: ISODate("1993-03-25"), phone: "9823456789", city: "Ahmedabad"},

{ customerid: 4, firstname: "Sneha", lastname: "Reddy", dob: ISODate("1997-11-05"), phone: "9345678123", city: "Hyderabad"},

{ customerid: 5, firstname: "Arjun", lastname: "Verma", dob: ISODate("1994-06-18"), phone: "9876543210", city: "Delhi"}

]);

#Policies Data

db.policies.insertMany([

{ policyid: 101, policyname: "Life Secure", policytype: "Life", premiumamount: 15000, durationyears: 20 },

{ policyid: 102, policyname: "Health Plus", policytype: "Health", premiumamount: 12000, durationyears: 12 },

{ policyid: 103, policyname: "Car Protect", policytype: "Motor", premiumamount: 8000, durationyears: 5 },

```
  { policyid: 104, policyname: "Home Shield", policytype: "Property", premiumamount: 20000, durationyears: 25 },
  { policyid: 105, policyname: "Travel Safe", policytype: "Travel", premiumamount: 3000, durationyears: 1 }
]);
```

#Agents data

```
db.agents.insertMany([
  { agentid: 1, agentname: "Suresh Kumar", phone: "9876501234", city: "Hyderabad" },
  { agentid: 2, agentname: "Neha Singh", phone: "9123409876", city: "Bangalore" },
  { agentid: 3, agentname: "Virat", phone: "9871201234", city: "Delhi" },
  { agentid: 4, agentname: "Vijay", phone: "9127809876", city: "Chennai" },
  { agentid: 5, agentname: "Mahesh", phone: "9812345678", city: "Mumbai" }
]);
```

#PolicyAssignments Data

```
db.policyAssignments.insertMany([
  { assignmentid: 1001, customerid: 1, policyid: 101, agentid: 1, startdate: ISODate("2022-01-01"), enddate: ISODate("2042-01-01") },
  { assignmentid: 1002, customerid: 2, policyid: 102, agentid: 2, startdate: ISODate("2023-06-15"), enddate: ISODate("2035-06-15") },
  { assignmentid: 1003, customerid: 3, policyid: 103, agentid: 3, startdate: ISODate("2021-03-10"), enddate: ISODate("2026-03-10") },
  { assignmentid: 1004, customerid: 4, policyid: 104, agentid: 4, startdate: ISODate("2020-09-01"), enddate: ISODate("2045-09-01") },
  { assignmentid: 1005, customerid: 5, policyid: 105, agentid: 5, startdate: ISODate("2024-01-01"), enddate: ISODate("2025-01-01") }
]);
```

#Claims Data

```
db.claims.insertMany([
  { claimid: 5001, assignmentid: 1001, claimdate: ISODate("2024-02-10"), claimamount: 75000, claimstatus: "Approved" },
  { claimid: 5002, assignmentid: 1002, claimdate: ISODate("2024-05-20"), claimamount: 30000, claimstatus: "Pending" },
  { claimid: 5003, assignmentid: 1003, claimdate: ISODate("2023-11-15"), claimamount: 50000, claimstatus: "Rejected" },
```

{ claimid: 5004, assignmentid: 1004, claimdate: ISODate("2022-08-05"), claimamount: 120000, claimstatus: "Approved" },

{ claimid: 5005, assignmentid: 1005, claimdate: ISODate("2024-03-18"), claimamount: 10000, claimstatus: "Pending" }

]);

#Payments Data

db.payments.insertMany([

{ paymentid: 9001, policyid: 101, paymentdate: ISODate("2023-01-15"), amount: 15000 },

{ paymentid: 9002, policyid: 102, paymentdate: ISODate("2023-06-20"), amount: 12000 },

{ paymentid: 9003, policyid: 103, paymentdate: ISODate("2022-03-10"), amount: 8000 },

{ paymentid: 9004, policyid: 104, paymentdate: ISODate("2021-09-01"), amount: 20000 },

{ paymentid: 9005, policyid: 105, paymentdate: ISODate("2024-01-05"), amount: 3000 }

]);

4. Practice Queries

a. MongoDB Comparison Operators

i. Find policies with premium greater than 10,000.

db.policies.find({ premiumamount: { $gt: 10000 } })

```
> db.policies.find({ premiumamount: { $gt: 10000 } })
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'Life Secure',
    policytype: 'Life',
    premiumamount: 15000,
    durationyears: 20
  }
```

ii. Find claims with claim amount less than 50,000.

db.claims.find({ claimamount: { $lt: 50000 } })

```
> db.claims.find({ claimamount: { $lt: 50000 } })
< {
    _id: ObjectId('695755e08adc8916580ddfae'),
    claimid: 5002,
    assignmentid: 1002,
    claimdate: 2024-05-20T00:00:00.000Z,
    claimamount: 30000,
    claimstatus: 'Pending'
  }
```

iii. Find customers who are not from Mumbai.

db.customers.find({ city: { $ne: "Mumbai" } }

```
> db.customers.find({city : {$ne: "Mumbai"}})
< {
    _id: ObjectId('6957559a8adc8916580ddf9a'),
    customerid: 2,
    firstname: 'Neha',
    lastname: 'Shetty',
    dob: 1999-07-12T00:00:00.000Z,
    phone: '9014486579',
    city: 'Bangalore'
  }
```

b. MongoDB Logical Operators

i. Find Life policies with premium greater than 12,000.

db.policies.find({ $and: [{ policytype: "Life" }, { premiumamount: {$gt: 15000 } }] })

```
> db.policies.find({ $and: [{ policytype: "Life" }, { premiumamount: { $gt: 12000 } }] }
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'Life Secure',
    policytype: 'Life',
    premiumamount: 15000,
    durationyears: 20
  }
```

ii. Find customers from Delhi or Bangalore.

db.customers.find({ $or: [{ city: "Delhi" }, { city: "Bangalore" }] })

```
> db.customers.find({ $or: [{ city: "Delhi" }, { city: "Bangalore" }] })
< {
    _id: ObjectId('6957559a8adc8916580ddf9a'),
    customerid: 2,
    firstname: 'Neha',
    lastname: 'Shetty',
    dob: 1999-07-12T00:00:00.000Z,
    phone: '9014486579',
    city: 'Bangalore'
  }
```

iii. Find claims whose amount is NOT greater than 1,00,000.

db.claims.find({ claimamount: { $not: { $gt: 100000 } } })

```
> db.claims.find({ claimamount: { $not: { $gt: 100000 } } })
< {
    _id: ObjectId('695755e08adc8916580ddfad'),
    claimid: 5001,
    assignmentid: 1001,
    claimdate: 2024-02-10T00:00:00.000Z,
    claimamount: 75000,
    claimstatus: 'Approved'
  }
```

c. MongoDB $in Operator

i. Find policies whose type is Life or Health.
db.policies.find({ policytype: { $in: ["Life", "Health"] } })

```
> db.policies.find({ policytype: { $in: ["Life", "Health"] } })
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'Life Secure',
    policytype: 'Life',
    premiumamount: 15000,
    durationyears: 20
  }
```

ii. Find agents working in Mumbai, Delhi, or Chennai.
db.agents.find({ city: { $in: ["Mumbai", "Delhi", "Chennai"] }})

```
> db.agents.find({ city: { $in: ["Mumbai", "Delhi", "Chennai"] } })
< {
    _id: ObjectId('695755b58adc8916580ddfa5'),
    agentid: 3,
    agentname: 'Virat',
    phone: '9871201234',
    city: 'Delhi'
  }
```

iii. Find customers with customerid 1, 3, or 5.
db.customers.find({ customerid: { $in: [1, 3, 5]}})

```
> db.customers.find({ customerid: { $in: [1, 3, 5] } })
< {
    _id: ObjectId('6957559a8adc8916580ddf99'),
    customerid: 1,
    firstname: 'Rahul',
    lastname: 'Sharma',
    dob: 1995-12-12T00:00:00.000Z,
    phone: '9704873975',
    city: 'Mumbai'
  }
```

d. MongoDB Range Queries

i. Find policies with premium between 5,000 and 20,000.
db.policies.find({ premiumamount: { $gte: 5000, $lte: 20000 } })

```
> db.policies.find({ premiumamount: { $gte: 5000, $lte: 20000 } })
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'Life Secure',
    policytype: 'Life',
    premiumamount: 15000,
    durationyears: 20
  }
```

ii. Find claims with amount between 10,000 and 80,000.
db.claims.find({ claimamount: { $gte: 10000, $lte: 80000 } })

```
> db.claims.find({ claimamount: { $gte: 10000, $lte: 80000 } })
< {
    _id: ObjectId('695755e08adc8916580ddfad'),
    claimid: 5001,
    assignmentid: 1001,
    claimdate: 2024-02-10T00:00:00.000Z,
    claimamount: 75000,
    claimstatus: 'Approved'
  }
```

iii. Find payments with amount between 3,000 and 15,000.

```
> db.payments.find({ amount: { $gte: 3000, $lte: 15000 } })
< {
    _id: ObjectId('695756a88adc8916580ddfb2'),
    paymentid: 9001,
    policyid: 101,
    paymentdate: 2023-01-15T00:00:00.000Z,
    amount: 15000
  }
```

e. Sorting commands

i. Ascending order
db.policies.find().sort({ premiumamount: 1 })

```
db.policies.find().sort({ premiumamount: 1 })
< {
    _id: ObjectId('695755a98adc8916580ddfa2'),
    policyid: 105,
    policyname: 'Travel Safe',
    policytype: 'Travel',
    premiumamount: 3000,
    durationyears: 1
  }
  {
    _id: ObjectId('695755a98adc8916580ddfa0'),
    policyid: 103,
    policyname: 'Car Protect',
    policytype: 'Motor',
    premiumamount: 8000,
    durationyears: 5
  }
```

ii. Descending order
db.policies.find().sort({ premiumamount: -1 })

```
db.policies.find().sort({ premiumamount: -1 })
{
  _id: ObjectId('695755a98adc8916580ddfa1'),
  policyid: 104,
  policyname: 'Home Shield',
  policytype: 'Property',
  premiumamount: 20000,
  durationyears: 25
}
{
  _id: ObjectId('695755a98adc8916580ddf9e'),
  policyid: 101,
  policyname: 'Life Secure',
  policytype: 'Life',
  premiumamount: 15000,
  durationyears: 20
}
```

f. Limit and Skip

i. Limit command

db.policies.find().limit(3)

```
db.policies.find().limit(3)
{
  _id: ObjectId('695755a98adc8916580ddf9e'),
  policyid: 101,
  policyname: 'Life Secure',
  policytype: 'Life',
  premiumamount: 15000,
  durationyears: 20
}
{
  _id: ObjectId('695755a98adc8916580ddf9f'),
  policyid: 102,
  policyname: 'Health Plus',
  policytype: 'Health',
  premiumamount: 12000,
  durationyears: 12
}
{
  _id: ObjectId('695755a98adc8916580ddfa0'),
  policyid: 103,
  policyname: 'Car Protect',
  policytype: 'Motor',
  premiumamount: 8000,
  durationyears: 5
}
```

ii. Limit and Skip command

db.policies.find().skip(2).limit(2)

```
> db.policies.find().skip(2).limit(2)
< {
    _id: ObjectId('695755a98adc8916580ddfa0'),
    policyid: 103,
    policyname: 'Car Protect',
    policytype: 'Motor',
    premiumamount: 8000,
    durationyears: 5
  }
  {
    _id: ObjectId('695755a98adc8916580ddfa1'),
    policyid: 104,
    policyname: 'Home Shield',
    policytype: 'Property',
    premiumamount: 20000,
    durationyears: 25
  }
```

g. Update Commands

i. Set city to pune for customerid with 1

db.customers.updateOne({ customerid: 1 }, { $set: { city: "pune" } })

```
> db.customers.updateOne({ customerid: 1 }, { $set: { city: "pune" } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.customers.find({customerid: 1})
< {
    _id: ObjectId('6957559a8adc8916580ddf99'),
    customerid: 1,
    firstname: 'Rahul',
    lastname: 'Sharma',
    dob: 1995-12-12T00:00:00.000Z,
    phone: '9704873975',
    city: 'pune'
  }
```

ii. Increment 1000 to to all insurances of type ‘Life’

```
> db.policies.updateMany({ policytype: "Life" }, { $inc: { premiumamount: 1000 } })
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.policies.find({policytype:"Life"})
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'Life Secure',
    policytype: 'Life',
    premiumamount: 16000,
    durationyears: 20
  }
```

h. Replace Document

i. Update Policy ‘Life Plus’

db.policies.replaceOne(

{ policyid: 101 },

{ policyid: 101, policyname: "life plus", policytype: "life", premiumamount: 18000 })

```
> db.policies.replaceOne(
    { policyid: 101 },
    { policyid: 101, policyname: "life plus", policytype: "life", premiumamount: 18000 }
  )
< {
    acknowledged: true,
    insertedId: null,
    matchedCount: 1,
    modifiedCount: 1,
    upsertedCount: 0
  }
> db.policies.find({policytype:"life"})
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyid: 101,
    policyname: 'life plus',
    policytype: 'life',
    premiumamount: 18000
  }
```

i. Delete command

i. Delete customer details with id 5 from customers collection

db.customers.deleteOne({ customerid: 5 })

```
> db.customers.deleteOne({ customerid: 5 })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

```
> db.customers.find({customerid: 5})
<
```

ii. Delete all claims with status as ‘Rejected’

```
> db.claims.deleteMany({ claimstatus: "Rejected" })
< {
    acknowledged: true,
    deletedCount: 1
  }
```

```
> db.claims.find({claimstatus: "Rejected"})
<
```

j. Aggregation Functions

i. Total premium collected per PolicyType

```
db.policies.aggregate([
  { $group: { _id: "$policytype", totalPremium: { $sum: "$premiumamount" } } }
])
```

```
> db.policies.aggregate([
    { $group: { _id: "$policytype", totalPremium: { $sum: "$premiumamount" } } }
  ])
< {
    _id: 'Property',
    totalPremium: 20000
  }
  {
    _id: 'Motor',
    totalPremium: 8000
  }
  {
    _id: 'life',
    totalPremium: 18000
  }
  {
    _id: 'Health',
    totalPremium: 12000
  }
  {
    _id: 'Travel',
    totalPremium: 3000
  }
```

ii. Count number of claims per ClaimStatus

```
db.claims.aggregate([
  { $group: { id: "$claimstatus", totalClaims: { $count: {} } } }
])
```

```
> db.claims.aggregate([
    { $group: { _id: "$claimstatus", totalClaims: { $count: {} } } }
  ])
< {
    _id: 'Approved',
    totalClaims: 2
  }
  {
    _id: 'Pending',
    totalClaims: 2
  }
```

k. Projections and Expressions Queries

i. Show only customer name and city

```
db.customers.find(
  {},
  { _id: 0, firstname: 1, lastname: 1, city: 1 }
)
```

```
> db.customers.find(
    {},
    { _id: 0, firstname: 1, lastname: 1, city: 1 }
  )
< {
    firstname: 'Rahul',
    lastname: 'Sharma',
    city: 'pune'
  }
  {
    firstname: 'Neha',
    lastname: 'Shetty',
    city: 'Bangalore'
  }
```

ii. Calculate premium including 18% GST

```
> db.policies.aggregate([
    { $project: { policyname: 1, premiumWithGST: { $multiply: ["$premiumamount", 1.18] } } }
  ])
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyname: 'life plus',
    premiumWithGST: 21240
  }
  {
    _id: ObjectId('695755a98adc8916580ddf9f'),
    policyname: 'Health Plus',
    premiumWithGST: 14160
  }
```

iii. Convert premium to monthly premium

```
db.policies.aggregate([
  { $project: { policyname: 1, monthlyPremium: { $divide: ["$premiumamount", 12] } } }
])
```

```
> db.policies.aggregate([
    { $project: { policyname: 1, monthlyPremium: { $divide: ["$premiumamount", 12] } } }
  ])
< {
    _id: ObjectId('695755a98adc8916580ddf9e'),
    policyname: 'life plus',
    monthlyPremium: 1500
  }
  {
```

iv. Total premium received per policy

db.payments.aggregate([

{ $group: { _id: "$policyid", totalPremiumReceived: { $sum: "$amount" } } }

])

```
> db.payments.aggregate([
    { $group: { _id: "$policyid", totalPremiumReceived: { $sum: "$amount" } } }
  ])
< {
    _id: 103,
    totalPremiumReceived: 8000
  }
  {
    _id: 101,
    totalPremiumReceived: 15000
  }
  {
    _id: 102,
    totalPremiumReceived: 12000
  }
```

l. **MongoDB Join Like Commands**.

i. List Customers with Their Policies (INNER JOIN)

```
db.policyAssignments.aggregate([
 {
  $lookup: {
   from: "customers",
   localField: "customerid",
   foreignField: "customerid",
   as: "customer"
  }
 },
 { $unwind: "$customer" },
 {
  $lookup: {
   from: "policies",
   localField: "policyid",
   foreignField: "policyid",
   as: "policy"
  }
 },
 { $unwind: "$policy" },
 {
  $project: {
```

```
    _id: 0,
    customername: { $concat: ["$customer.firstname", " ", "$customer.lastname"] },
    city: "$customer.city",
    policytype: "$policy.policytype",
    premium: "$policy.premiumamount"
   }
  }
])
```

```
{
  customername: 'Rahul Sharma',
  city: 'pune',
  policytype: 'life',
  premium: 18000
}
{
  customername: 'Neha Shetty',
  city: 'Bangalore',
  policytype: 'Health',
  premium: 12000
}
{
  customername: 'Amit Patel',
  city: 'Ahmedabad',
  policytype: 'Motor',
  premium: 8000
}
{
  customername: 'Sneha Reddy',
  city: 'Hyderabad',
  policytype: 'Property',
  premium: 20000
}
```

ii. Customers Having Health Policies Only Display customers who have Health policy.

```
db.policyAssignments.aggregate([
  {
   $lookup: {
     from: "policies",
     localField: "policyid",
     foreignField: "policyid",
     as: "policy"
   }
  },
  { $unwind: "$policy" },
  { $match: { "policy.policytype": "Health" } },
  {
   $lookup: {
     from: "customers",
     localField: "customerid",
     foreignField: "customerid",
     as: "customer"
   }
```

```
  },
  { $unwind: "$customer" },
  {
   $project: {
    _id: 0,
    customername: { $concat: ["$customer.firstname", " ", "$customer.lastname"] },
    city: "$customer.city",
    policytype: "$policy.policytype"
   }
  }
])
```

```
< {
    customername: 'Neha Shetty',
    city: 'Bangalore',
    policytype: 'Health'
  }
```

iii. List ALL Customers Including Those Without Policies (LEFT OUTER JOIN)

```
db.customers.aggregate([
  {
   $lookup: {
    from: "policyAssignments",
    localField: "customerid",
    foreignField: "customerid",
    as: "assignments"
   }
  },
  {
   $lookup: {
    from: "policies",
    localField: "assignments.policyid",
    foreignField: "policyid",
    as: "policy"
   }
  },
  {
   $project: {
    _id: 0,
    customername: { $concat: ["$firstname", " ", "$lastname"] },
    city: 1,
    policytype: { $ifNull: [{ $arrayElemAt: ["$policy.policytype", 0] }, "No Policy"] },
    premium: { $ifNull: [{ $arrayElemAt: ["$policy.premiumamount", 0] }, 0] }
   }
  }
])
```

```
{
  city: 'pune',
  customername: 'Rahul Sharma',
  policytype: 'life',
  premium: 18000
}
{
  city: 'Bangalore',
  customername: 'Neha Shetty',
  policytype: 'Health',
  premium: 12000
}
{
  city: 'Ahmedabad',
  customername: 'Amit Patel',
  policytype: 'Motor',
  premium: 8000
}
{
  city: 'Hyderabad',
  customername: 'Sneha Reddy',
  policytype: 'Property',
  premium: 20000
}
```


iv. Count Policies Per Customer

```
db.customers.aggregate([
  {
   $lookup: {
     from: "policyAssignments",
     localField: "customerid",
     foreignField: "customerid",
     as: "policies"
   }
  },
  {
   $project: {
     _id: 0,
     customername: { $concat: ["$firstname", " ", "$lastname"] },
     totalPolicies: { $size: "$policies" }
   }
  }
])
```


```
{
  customername: 'Rahul Sharma',
  totalPolicies: 1
}
{
  customername: 'Neha Shetty',
  totalPolicies: 1
}
{
  customername: 'Amit Patel',
  totalPolicies: 1
}
{
  customername: 'Sneha Reddy',
  totalPolicies: 1
}
```

v. Policies Without Claims

```
db.policies.aggregate([
  {
   $lookup: {
    from: "policyAssignments",
    localField: "policyid",
    foreignField: "policyid",
    as: "assignments"
   }
  },
  {
   $lookup: {
    from: "claims",
    localField: "assignments.assignmentid",
    foreignField: "assignmentid",
    as: "claims"
   }
  },
  {
   $match: {
    claims: { $eq: [] }
   }
  }
])
```

```
< {
    _id: ObjectId('695755a98adc8916580ddfa0'),
    policyid: 103,
    policyname: 'Car Protect',
    policytype: 'Motor',
    premiumamount: 8000,
    durationyears: 5,
    assignments: [
      {
        _id: ObjectId('695755c68adc8916580ddfaa'),
        assignmentid: 1003,
        customerid: 3,
        policyid: 103,
        agentid: 3,
        startdate: 2021-03-10T00:00:00.000Z,
        enddate: 2026-03-10T00:00:00.000Z
      }
    ],
    claims: []
  }
```