Day - 5 MongoDB Assignment

```
use insuranceDB
```

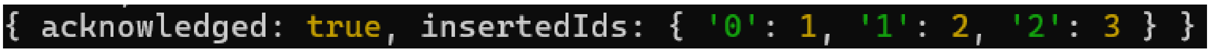
Creating Collections

```
db.createCollection("customers")

db.createCollection("policies")

db.createCollection("agents")

db.createCollection("policyAssignments")

db.createCollection("claims")
```

Inserting Data

1.Inserting into customers collection

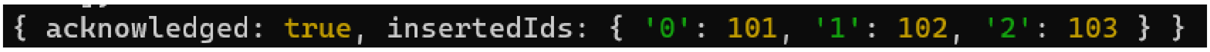
```
db.customers.insertMany([
 {
   _id: 1,
   firstName: "Sita",
   lastName: "Rao",
   dateOfBirth: ISODate("1998-05-10"),
   phone: 9876543210,
   email: "sita@gmail.com"
 },
 {
   _id: 2,
   firstName: "Rahul",
   lastName: "Sharma",
   dateOfBirth: ISODate("1995-03-22"),
   phone: 9123456780,
   email: "rahul@gmail.com"
 },
 {
   _id: 3,
   firstName: "Ananya",
   lastName: "Iyer",
   dateOfBirth: ISODate("2000-11-01"),
   phone: 9988776655,
   email: "ananya@gmail.com"
 }
])
```



```
{ acknowledged: true, insertedIds: { '0': 1, '1': 2, '2': 3 } }
```

2.Inserting into policies collection

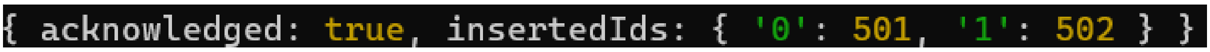
```
db.policies.insertMany([
 {
  _id: 101,
  policyName: "Health Plus",
  policyType: "Health",
  premiumAmount: 25000,
  durationYears: 5
 },
 {
  _id: 102,
  policyName: "Life Secure",
  policyType: "Life",
  premiumAmount: 30000,
  durationYears: 10
 },
 {
  _id: 103,
  policyName: "Car Protect",
  policyType: "Vehicle",
  premiumAmount: 18000,
  durationYears: 3
 }
])
```



```
{ acknowledged: true, insertedIds: { '0': 101, '1': 102, '2': 103 } }
```

3.Inserting into agents collection

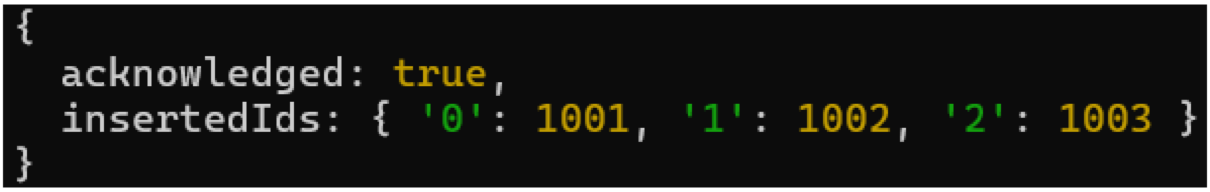
```
db.agents.insertMany([
 {
   _id: 501,
   agentName: "Ramesh",
   phone: "9876501234",
   city: "Hyderabad"
 },
 {
   _id: 502,
   agentName: "Suresh",
   phone: "9123409876",
   city: "Bangalore"
 }
])
```



```
{ acknowledged: true, insertedIds: { '0': 501, '1': 502 } }
```

4.Inserting into policyAssignments collection

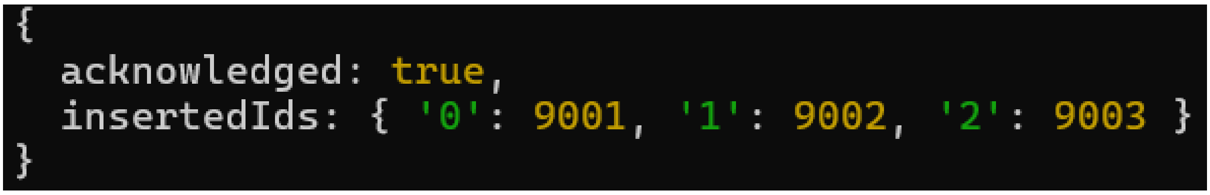
```
db.policyAssignments.insertMany([
 {
   _id: 1001,
   customerId: 1,
   policyId: 101,
   agentId: 501,
   startDate: ISODate("2024-01-01"),
   endDate: ISODate("2029-01-01")
 },
 {
   _id: 1002,
   customerId: 2,
   policyId: 102,
   agentId: 502,
   startDate: ISODate("2023-06-15"),
   endDate: ISODate("2033-06-15")
 },
 {
   _id: 1003,
   customerId: 3,
   policyId: 103,
   agentId: 501,
   startDate: ISODate("2025-02-01"),
   endDate: ISODate("2028-02-01")
 }
])
```



```
{
  acknowledged: true,
  insertedIds: { '0': 1001, '1': 1002, '2': 1003 }
}
```

5.Inserting into claims collections

```
db.claims.insertMany([
  {
    _id: 9001,
    assignmentId: 1001,
    claimDate: ISODate("2025-02-10"),
    claimAmount: 15000,
    claimStatus: "Approved"
  },
  {
    _id: 9002,
    assignmentId: 1002,
    claimDate: ISODate("2024-12-05"),
    claimAmount: 20000,
    claimStatus: "Pending"
  },
  {
    _id: 9003,
    assignmentId: 1003,
    claimDate: ISODate("2025-03-18"),
    claimAmount: 8000,
    claimStatus: "Rejected"
  }
])
```

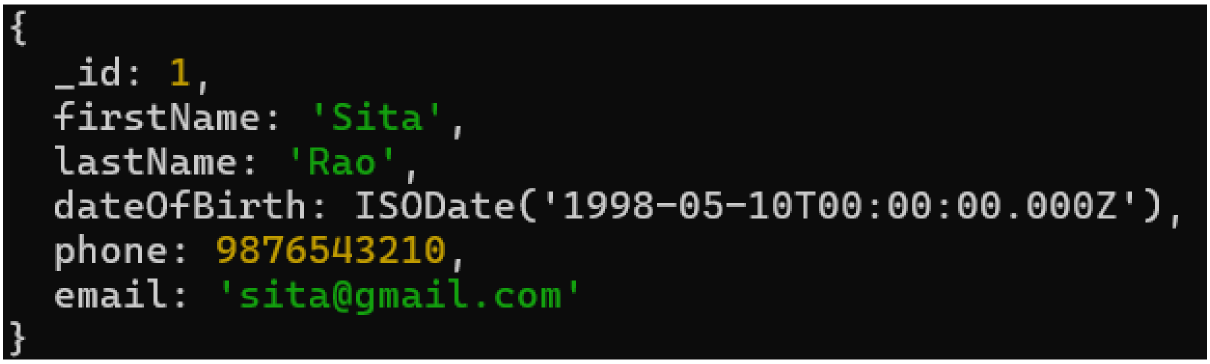


```
{
  acknowledged: true,
  insertedIds: { '0': 9001, '1': 9002, '2': 9003 }
}
```

QUERYING

1.Find customer details having id as 1.

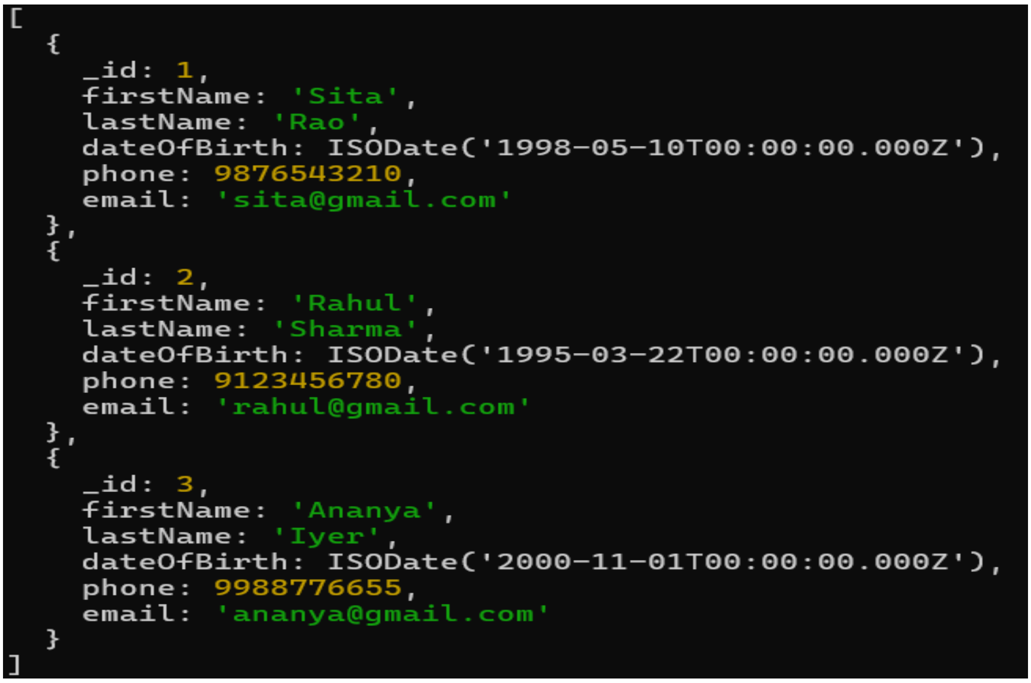
```
db.customers.findOne({_id:1})
```



```
{
  _id: 1,
  firstName: 'Sita',
  lastName: 'Rao',
  dateOfBirth: ISODate('1998-05-10T00:00:00.000Z'),
  phone: 9876543210,
  email: 'sita@gmail.com'
}
```

2.Retrieve all customer details.

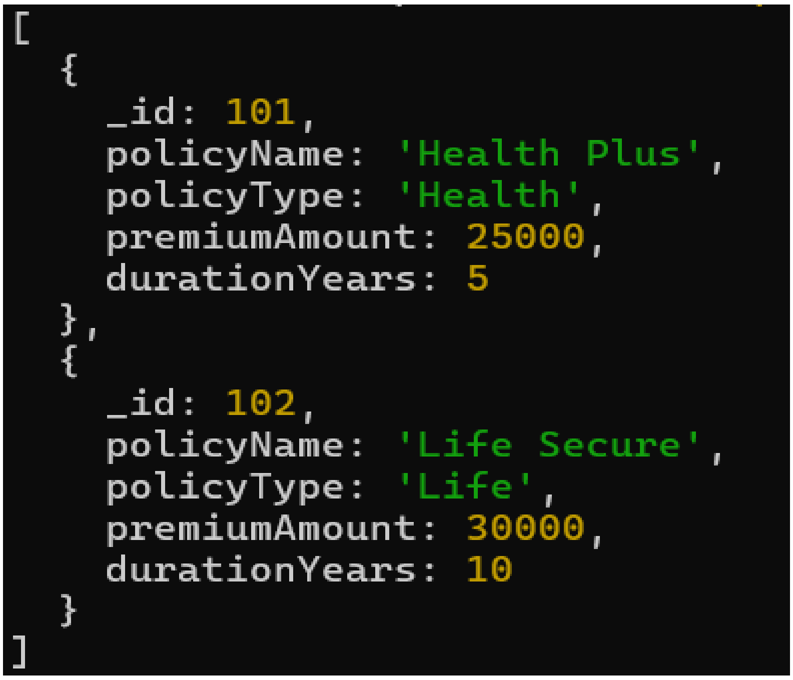
db.customers.find()



```
[
  {
    _id: 1,
    firstName: 'Sita',
    lastName: 'Rao',
    dateOfBirth: ISODate('1998-05-10T00:00:00.000Z'),
    phone: 9876543210,
    email: 'sita@gmail.com'
  },
  {
    _id: 2,
    firstName: 'Rahul',
    lastName: 'Sharma',
    dateOfBirth: ISODate('1995-03-22T00:00:00.000Z'),
    phone: 9123456780,
    email: 'rahul@gmail.com'
  },
  {
    _id: 3,
    firstName: 'Ananya',
    lastName: 'Iyer',
    dateOfBirth: ISODate('2000-11-01T00:00:00.000Z'),
    phone: 9988776655,
    email: 'ananya@gmail.com'
  }
]
```

3.Find Policies having premium amount greater than 20000.

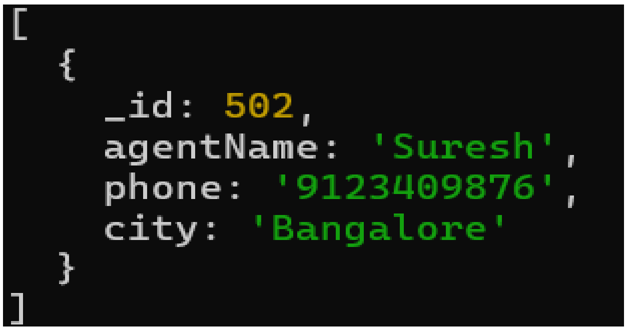
db.policies.find({premiumAmount:{$gt:20000}})



```
[
  {
    _id: 101,
    policyName: 'Health Plus',
    policyType: 'Health',
    premiumAmount: 25000,
    durationYears: 5
  },
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  }
]
```

4.Find agents whose city is Bangalore.

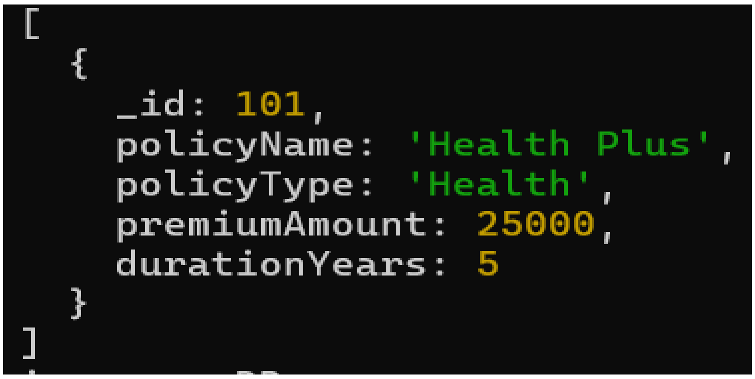
db.agents.find({city:"Bangalore"})



```
[
  {
    _id: 502,
    agentName: 'Suresh',
    phone: '9123409876',
    city: 'Bangalore'
  }
]
```

5.Find policies having policy type as Health.

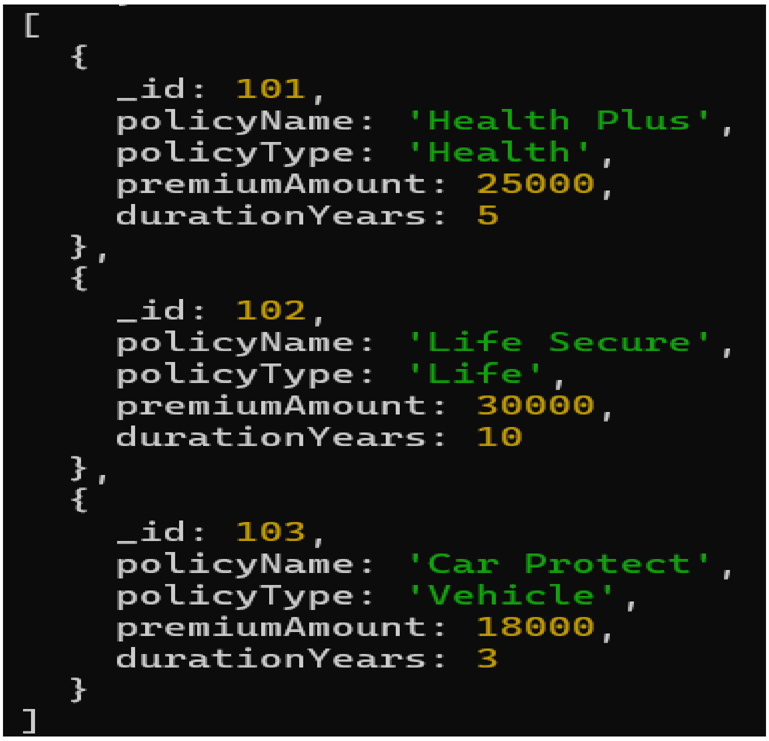
db.policies.find({policyType:"Health"})



```
[
  {
    _id: 101,
    policyName: 'Health Plus',
    policyType: 'Health',
    premiumAmount: 25000,
    durationYears: 5
  }
]
```

6.List policies of type Life, Health, vehicle use OR clause.

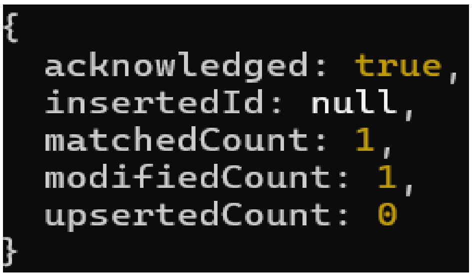
db.policies.find({$or: [{ policyType: "Life" },{ policyType: "Health" },{ policyType: "Vehicle" }]})



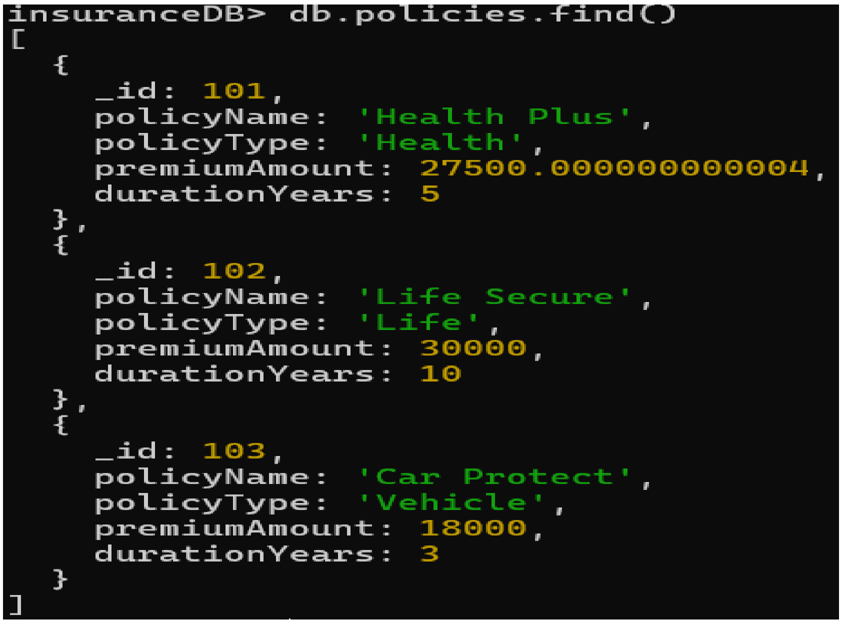
```
[
  {
    _id: 101,
    policyName: 'Health Plus',
    policyType: 'Health',
    premiumAmount: 25000,
    durationYears: 5
  },
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  },
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

7. Increase premium amount to 10% for all health insurance policies.

db.policies.updateMany({ policyType: "Health" },{ $mul: { premiumAmount: 1.10 } })



```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



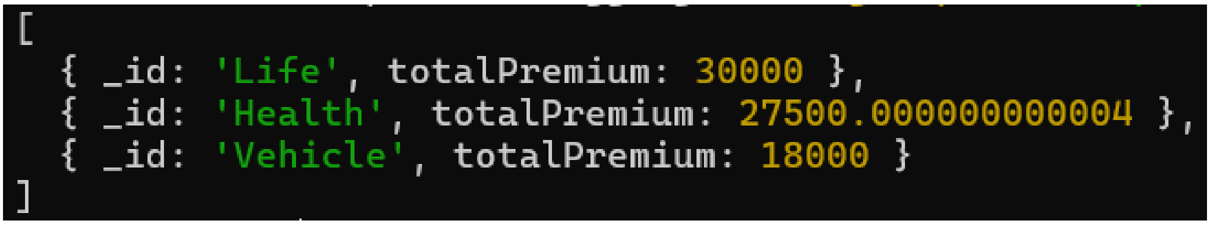
```
insuranceDB> db.policies.find()
[
  {
    _id: 101,
    policyName: 'Health Plus',
    policyType: 'Health',
    premiumAmount: 27500.000000000004,
    durationYears: 5
  },
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  },
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

8. Total premium collected per policy type

db.policies.aggregate([

{$group:{_id:"$policyType",totalPremium:{$sum:"$premiumAmount"}}}

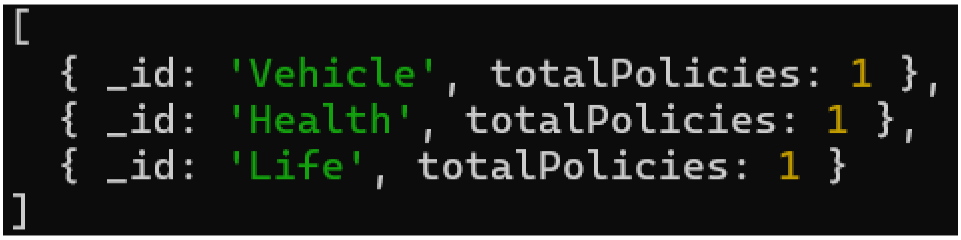
])



```
[
  { _id: 'Life', totalPremium: 30000 },
  { _id: 'Health', totalPremium: 27500.000000000004 },
  { _id: 'Vehicle', totalPremium: 18000 }
]
```

9. Count policies by type.

db.policies.aggregate([{$group: { _id: "$policyType",totalPolicies: { $sum: 1 }}}])



```
[
  { _id: 'Vehicle', totalPolicies: 1 },
  { _id: 'Health', totalPolicies: 1 },
  { _id: 'Life', totalPolicies: 1 }
]
```

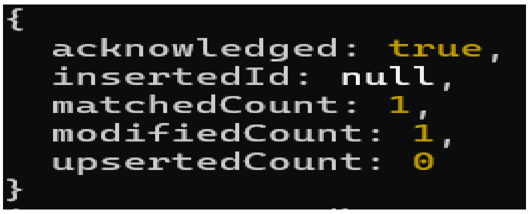
10. Update agent city to Chennai where agent name is Suresh.

db.agents.updateOne(.

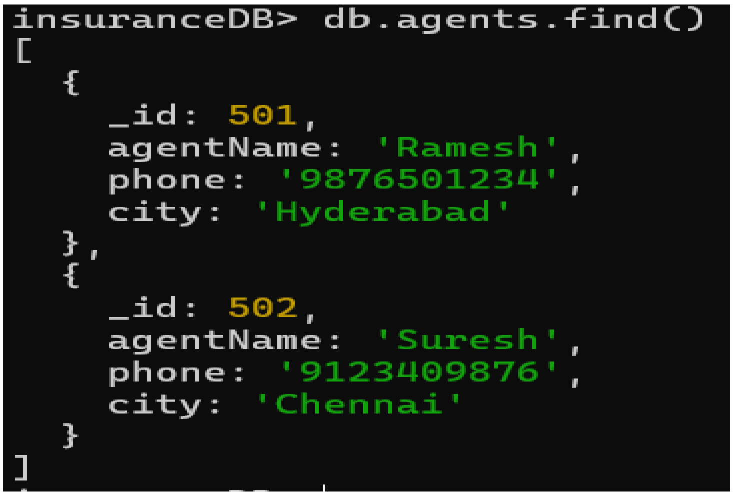
{ agentName: "Suresh" },

{ $set: { city: "Chennai" } }

)



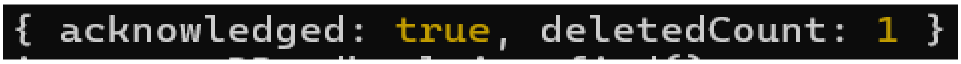
```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



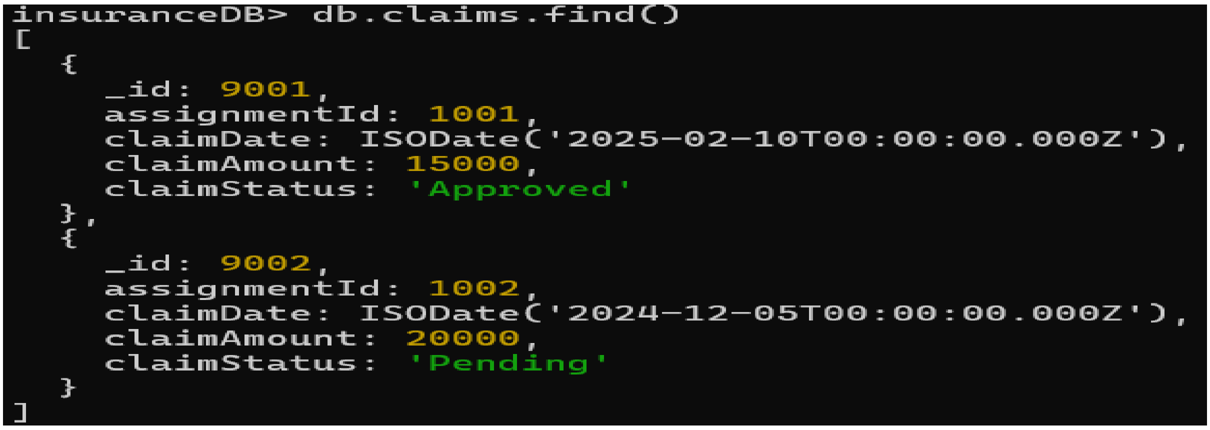
```
insuranceDB> db.agents.find()
[
  {
    _id: 501,
    agentName: 'Ramesh',
    phone: '9876501234',
    city: 'Hyderabad'
  },
  {
    _id: 502,
    agentName: 'Suresh',
    phone: '9123409876',
    city: 'Chennai'
  }
]
```

11. Delete all claims with status "Rejected.

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



```
{ acknowledged: true, deletedCount: 1 }
```

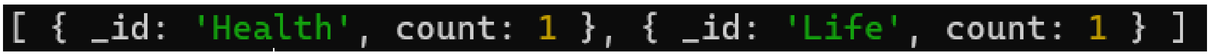


```
insuranceDB> db.claims.find()
[
  {
    _id: 9001,
    assignmentId: 1001,
    claimDate: ISODate('2025-02-10T00:00:00.000Z'),
    claimAmount: 15000,
    claimStatus: 'Approved'
  },
  {
    _id: 9002,
    assignmentId: 1002,
    claimDate: ISODate('2024-12-05T00:00:00.000Z'),
    claimAmount: 20000,
    claimStatus: 'Pending'
  }
]
```

12. Display the number of policies for each policy type where the premium amount is greater than 20,000.

db.policies.aggregate([{ $match: { premiumAmount: { $gt: 20000 } } },

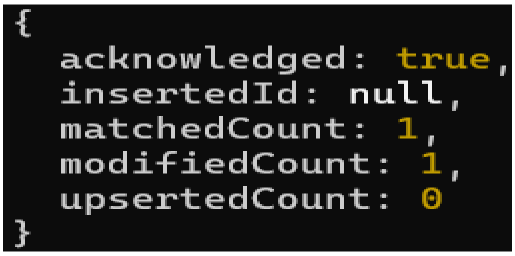
{$group: { _id: "$policyType",count: { $sum: 1 }}}])



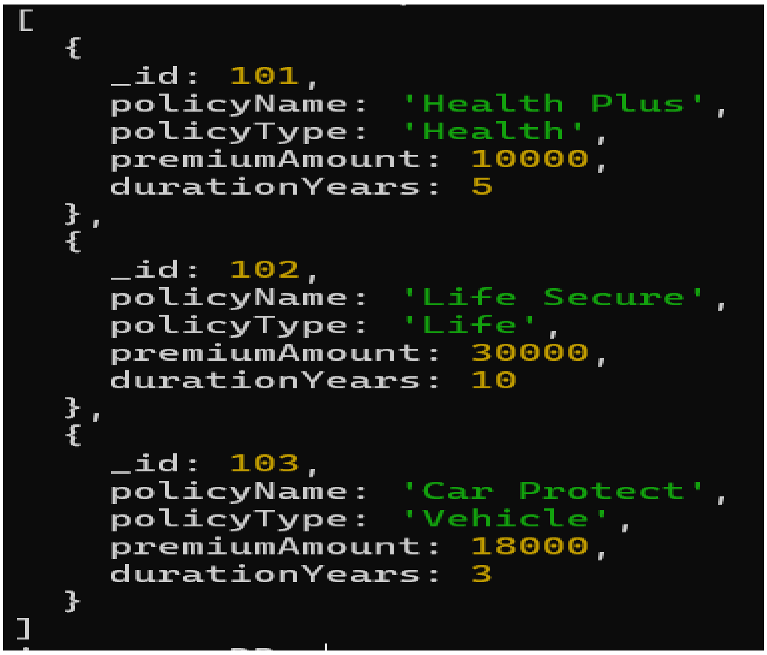
```
[ { _id: 'Health', count: 1 }, { _id: 'Life', count: 1 } ]
```

13. Update premium of policyNo 101.

db.policies.updateOne({ _id: 101 }, { $set: { premiumAmount: 10000 } })



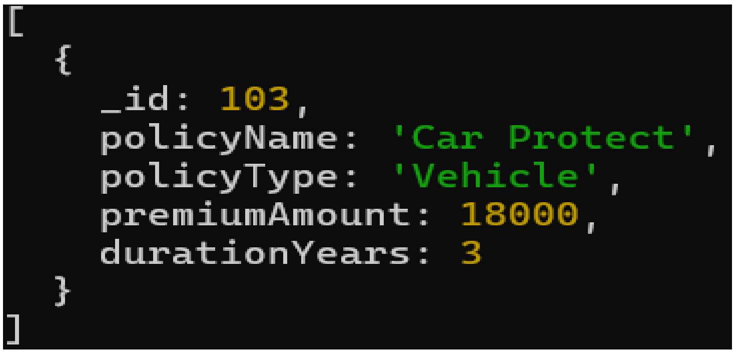
```
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```



```
[
  {
    _id: 101,
    policyName: 'Health Plus',
    policyType: 'Health',
    premiumAmount: 10000,
    durationYears: 5
  },
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  },
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

14. Find Policies with premium between 15000 and 20000.

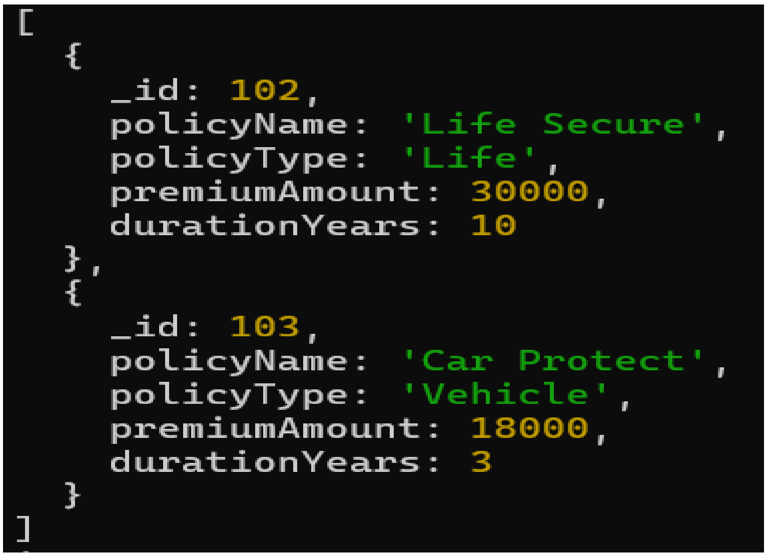
db.policies.find({ premiumAmount: { $gte: 15000, $lte: 20000 } })



```
[
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

15.Find top 2 highest policies based on premium amount.

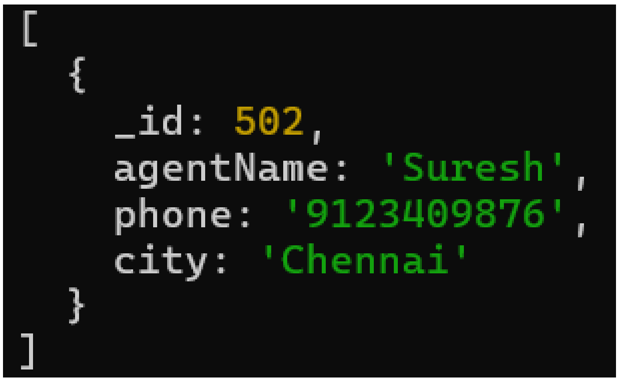
db.policies.find().sort({ premiumAmount: -1 }).limit(2)



```
[
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  },
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

16. Skip first 1 record from agents.

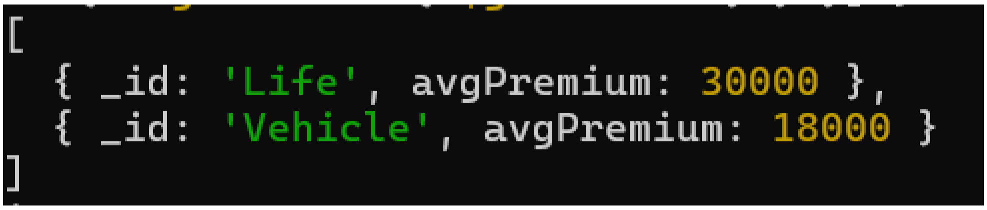
db.agents.find().skip(1)



```
[
  {
    _id: 502,
    agentName: 'Suresh',
    phone: '9123409876',
    city: 'Chennai'
  }
]
```

17. Find policyType having avg premium > 15000

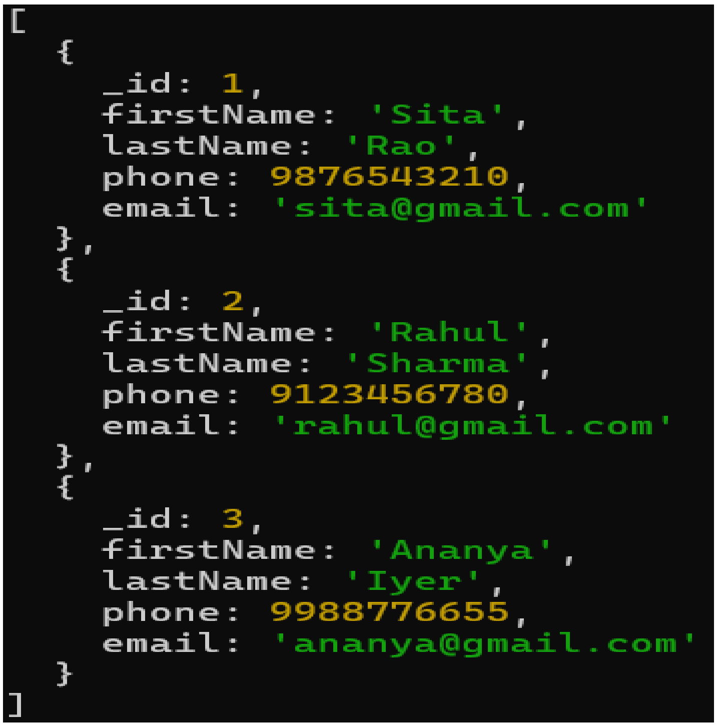
db.policies.aggregate([{ $group: { _id: "$policyType", avgPremium: { $avg: "$premiumAmount" } } }, { $match: { avgPremium: { $gt: 15000 } } }])



```
[
  { _id: 'Life', avgPremium: 30000 },
  { _id: 'Vehicle', avgPremium: 18000 }
]
```

18.Show all fields except date of birth from customers.

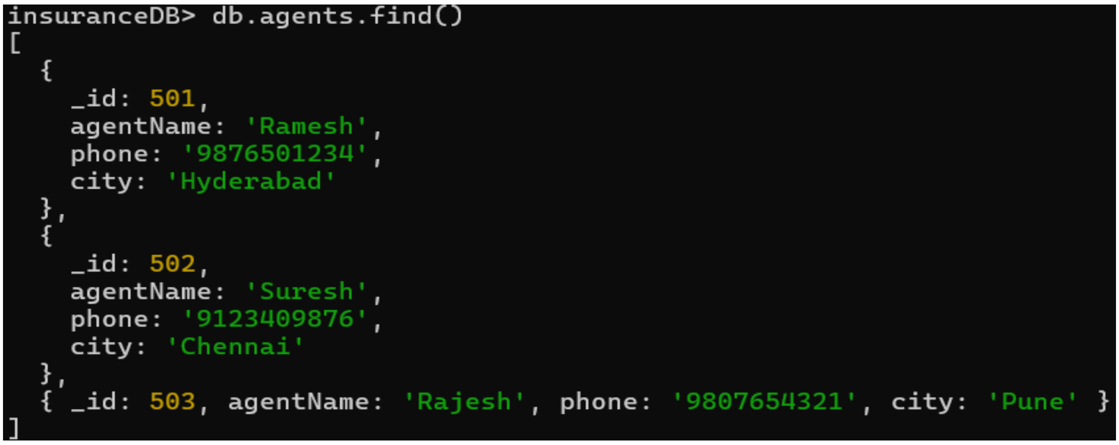
db.customers.find({},{dateOfBirth:0})



```
[
  {
    _id: 1,
    firstName: 'Sita',
    lastName: 'Rao',
    phone: 9876543210,
    email: 'sita@gmail.com'
  },
  {
    _id: 2,
    firstName: 'Rahul',
    lastName: 'Sharma',
    phone: 9123456780,
    email: 'rahul@gmail.com'
  },
  {
    _id: 3,
    firstName: 'Ananya',
    lastName: 'Iyer',
    phone: 9988776655,
    email: 'ananya@gmail.com'
  }
]
```

19.Insert new agent

db.agents.insertOne({_id:503,agentName:'Rajesh',phone:'9807654321',city:'Pune'})



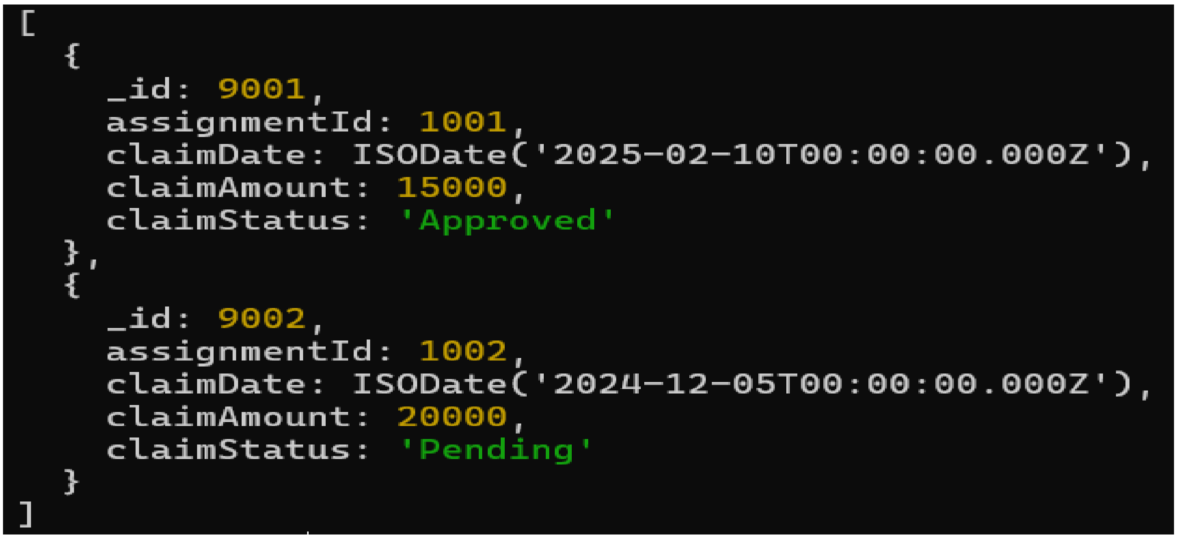
```
insuranceDB> db.agents.find()
[
  {
    _id: 501,
    agentName: 'Ramesh',
    phone: '9876501234',
    city: 'Hyderabad'
  },
  {
    _id: 502,
    agentName: 'Suresh',
    phone: '9123409876',
    city: 'Chennai'
  },
  { _id: 503, agentName: 'Rajesh', phone: '9807654321', city: 'Pune' }
]
```

20. Find claims with claim amount greater than 10000

db.claims.find({

claimAmount: { $gt: 10000 }

})



```
[
  {
    _id: 9001,
    assignmentId: 1001,
    claimDate: ISODate('2025-02-10T00:00:00.000Z'),
    claimAmount: 15000,
    claimStatus: 'Approved'
  },
  {
    _id: 9002,
    assignmentId: 1002,
    claimDate: ISODate('2024-12-05T00:00:00.000Z'),
    claimAmount: 20000,
    claimStatus: 'Pending'
  }
]
```

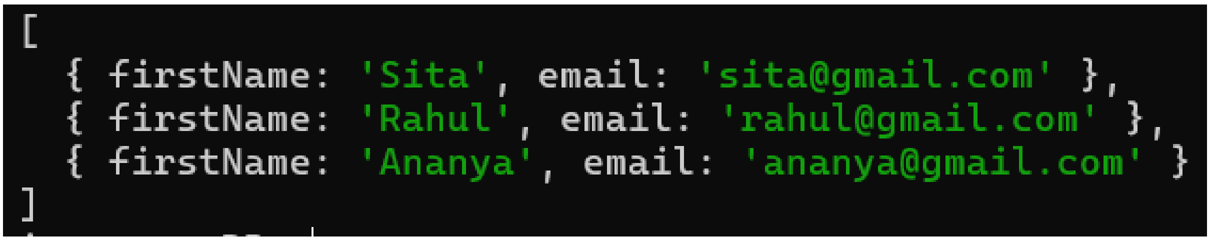
21. Display customers with only first name and email.

db.customers.find(

{},

{ firstName: 1, email: 1, _id: 0 }

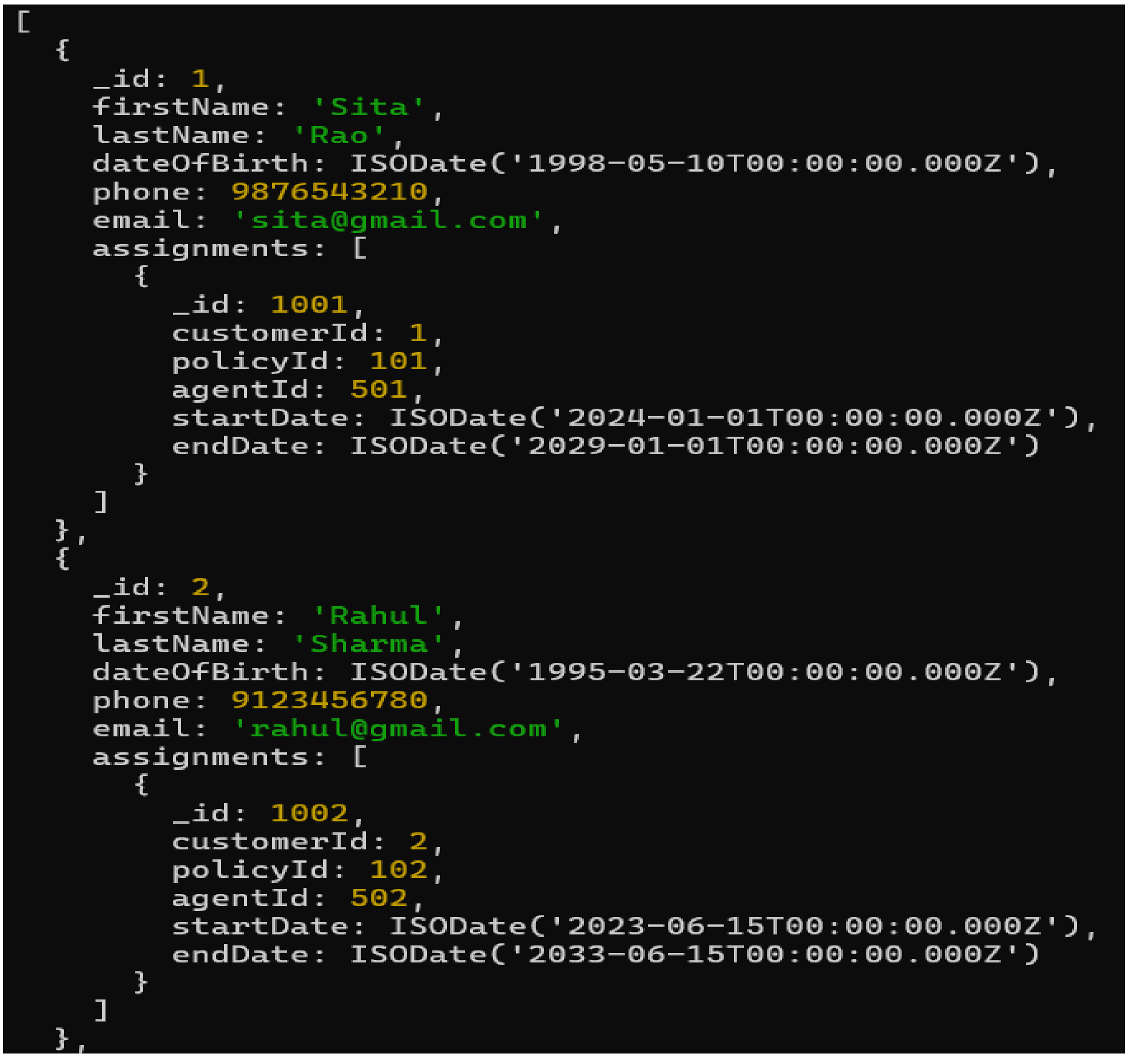
)



```
[
  { firstName: 'Sita', email: 'sita@gmail.com' },
  { firstName: 'Rahul', email: 'rahul@gmail.com' },
  { firstName: 'Ananya', email: 'ananya@gmail.com' }
]
```

22. Join customers with policyAssignments.

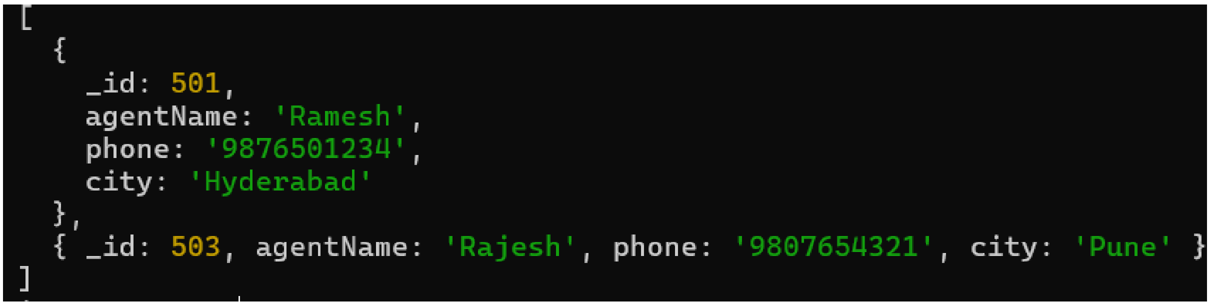
db.customers.aggregate([{$lookup: {from: "policyAssignments", localField: "_id", foreignField: "customerId", as: "assignments"}}])



```
[
  {
    _id: 1,
    firstName: 'Sita',
    lastName: 'Rao',
    dateOfBirth: ISODate('1998-05-10T00:00:00.000Z'),
    phone: 9876543210,
    email: 'sita@gmail.com',
    assignments: [
      {
        _id: 1001,
        customerId: 1,
        policyId: 101,
        agentId: 501,
        startDate: ISODate('2024-01-01T00:00:00.000Z'),
        endDate: ISODate('2029-01-01T00:00:00.000Z')
      }
    ]
  },
  {
    _id: 2,
    firstName: 'Rahul',
    lastName: 'Sharma',
    dateOfBirth: ISODate('1995-03-22T00:00:00.000Z'),
    phone: 9123456780,
    email: 'rahul@gmail.com',
    assignments: [
      {
        _id: 1002,
        customerId: 2,
        policyId: 102,
        agentId: 502,
        startDate: ISODate('2023-06-15T00:00:00.000Z'),
        endDate: ISODate('2033-06-15T00:00:00.000Z')
      }
    ]
  },
```

23. Find agents whose name starts with "R".

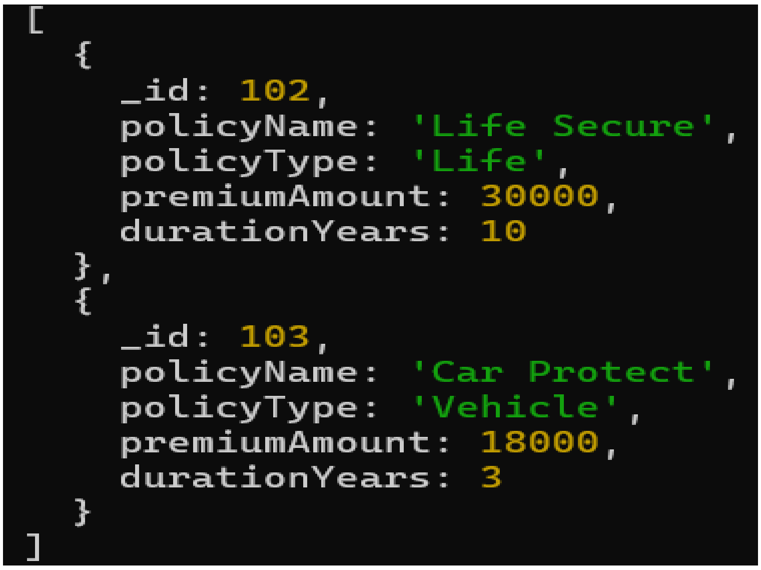
db.agents.find({agentName: { $regex: "^R" }})



```
[
  {
    _id: 501,
    agentName: 'Ramesh',
    phone: '9876501234',
    city: 'Hyderabad'
  },
  { _id: 503, agentName: 'Rajesh', phone: '9807654321', city: 'Pune' }
]
```

24. Find policies NOT of type Health.

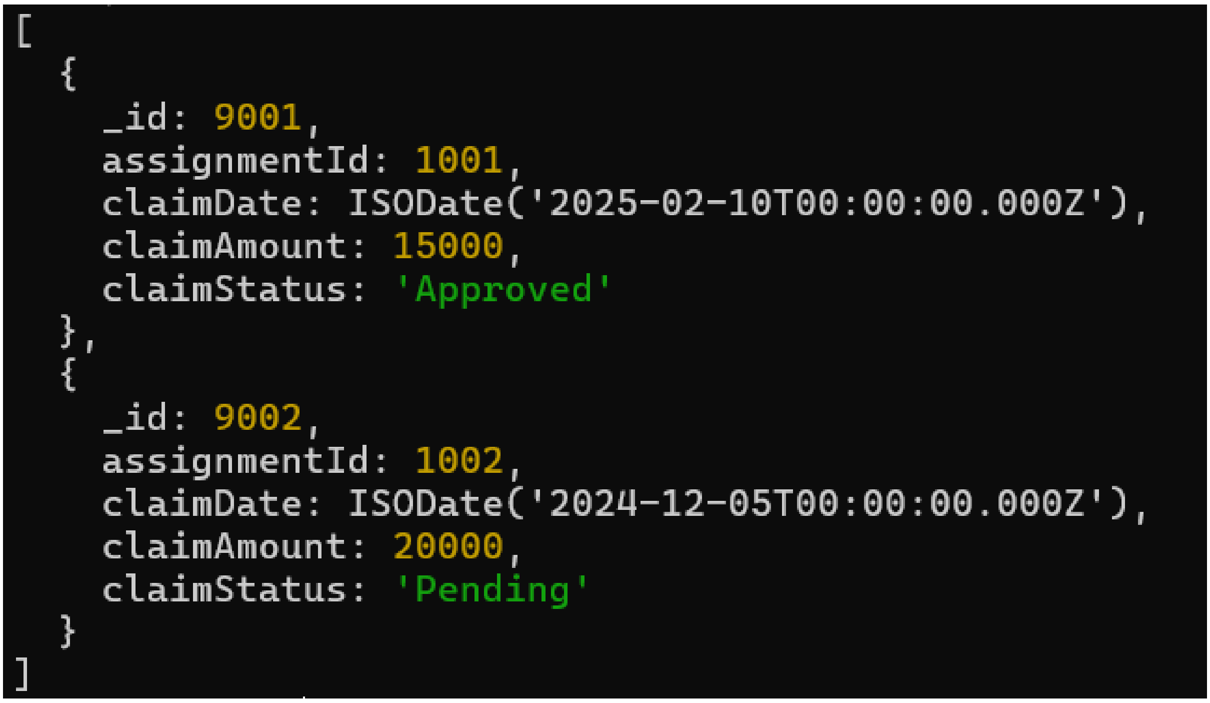
db.policies.find({policyType: { $ne: "Health" }})



```
[
  {
    _id: 102,
    policyName: 'Life Secure',
    policyType: 'Life',
    premiumAmount: 30000,
    durationYears: 10
  },
  {
    _id: 103,
    policyName: 'Car Protect',
    policyType: 'Vehicle',
    premiumAmount: 18000,
    durationYears: 3
  }
]
```

25. Find claims between two dates.

```
db.claims.find({
 claimDate: {
  $gte: ISODate("2024-01-01"),
  $lte: ISODate("2025-12-31")
 }
})
```



```
[
  {
    _id: 9001,
    assignmentId: 1001,
    claimDate: ISODate('2025-02-10T00:00:00.000Z'),
    claimAmount: 15000,
    claimStatus: 'Approved'
  },
  {
    _id: 9002,
    assignmentId: 1002,
    claimDate: ISODate('2024-12-05T00:00:00.000Z'),
    claimAmount: 20000,
    claimStatus: 'Pending'
  }
]
```