Day-5 MongoDB Assignment

1.Creating database

```
 use insuranceDB;
```

2. Create collection

```
db.createCollection("customers");
```

3. Insert documents

```
db.customers.insertOne({  customer_id: 101,first_name: "Nani", last_name: "Rao",
created_date: new Date(),
   phone: 949239230,
   email: "mail@gmail.com"
 });
```

4.Select customers

```
db.customers.find({customer_id:101});
```

5. Display only few fields

```
db.customers.find({customer_id:101},{_id:0,first_name:1});
```

6.Update the document

```
db.customers.updateOne({_id:ObjectId("695753959aa93f3dad7e47d2")},{$set:{customer_id:103}});
```

7. delete the document

```
db.posts.deleteOne({customer_id:103});
```

8. Using gt

```
db.customers.find({customer_id:{$gt:101}});
```

9. Using in

```
db.customers.find({customer_id:{$in:[101,102]}});
```

10. Using and

```
db.customers.find({$and:[{first_name:'Nani'},{last_name:'Rao'}]});
```

11. Using aggregate

```
db.policies.aggregate([{$group:{_id:"$policy_type",total:{$sum:"$premiumamount"}}},{$limit:1}
])
```

12. Creating index

```
db.createIndex({customer_id:1});
```

13. Selecting indexes

```
db.customers.getIndexes();

[

  { v: 2, key: { _id: 1 }, name: '_id_' },

  { v: 2, key: { customer_id: 1 }, name: 'customer_id_1' }

]
```

14. Dropping index

```
db.customers.dropIndex({ email: 1 });
```

15. Grouping using aggregate

```
db.policies.aggregate([{$group:{_id:"$policy_type",total:{$sum:"$premiumamount"}}},{$limit:1}]);
```

16. Sorting

```
 Db.policies.sort({premiumamount:1});
```

17. Number of policies per city

```
 db.policies.aggregate([ { $group: { _id: "$city", count: { $sum: 1 } } } ])
```

18. Highest premium per policyType

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

19. Find policyType having avg premium > 15000

```
db.policies.aggregate([ { $group: { _id: "$policyType", avgPremium: { $avg: "$premium" } } },

 { $match: { avgPremium: { $gt: 15000 } } } ])
```

20. Skip first 2 records

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

21. Top 2 highest premium policies

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

22. Total premium of each policy_type

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

23. Count of each policy_type

```
db.policies.aggregate([{$group:{_id:"$policy_type",count:{$sum:1}}}]);
```