1.Create a database “Student” with the following attributesRollno, Age, ContactNo, Email-Id. db.createCollection("Student");

```
Atlas atlas-n519bd-shard-0 [primary] myDB> db.createCollection("Student");
{ ok: 1 }
```

2.Insert appropriate values

Var mystudent=[{RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"},{RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"},{RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"},{RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"},{RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"}]

db.Student.insert(mystudent)

3.Write query to update Email-Id of a student with rollno 10.

db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})

```
Atlas atlas-n519bd-shard-0 [primary] myDB> db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-n519bd-shard-0 [primary] myDB>
```

4.Replace the student name from “ABC” to “FEM” of rollno 11.

db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2276,email:"rea.de9@gmail.com"});

```
{
  _id: ObjectId("63cfdf8e77c9e0d6024ebe61"),
  RollNo: 11,
  Age: 22,
  Name: 'ABC',
  Cont: 2276,
  email: 'rea.de9@gmail.com'
}
```

db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})

```
Atlas atlas-n519bd-shard-0 [primary] myDB> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
{
  _id: ObjectId("63cfdf8e77c9e0d6024ebe61"),
  RollNo: 11,
  Age: 22,
  Name: 'FEM',
  Cont: 2276,
  email: 'rea.de9@gmail.com'
}
```

1. Create a collection by name Customers with the following attributes. Cust_id, Acc_Bal, Acc_Type

db.createCollection("Customers");

```
Atlas atlas-n519bd-shard-0 [primary] myDB> db.createCollection("Customers");
{ ok: 1 }
```

2.Insert at least 5 values into the table

var mycustomer=[{cust_id:1,Balance:200, Type:"S"},{cust_id:1,Balance:1000, Type:"Z"},{cust_id:2,Balance:100, Type:"Z"},{cust_id:2,Balance:1000, Type:"C"},{cust_id:2,Balance:500, Type:"C"},{cust_id:2,Balance:50, Type:"S"},{cust_id:3,Balance:500, Type:"Z"}]

db.Customers.insert(mycustomer)

```
Atlas atlas-n519bd-shard-0 [primary] myDB> db.Customers.find()
[
  {
    _id: ObjectId("63cff09f2969884310682f91"),
    cust_id: 1,
    Balance: 200,
    Type: 'S'
  },
  {
    _id: ObjectId("63cff09f2969884310682f92"),
    cust_id: 1,
    Balance: 1000,
    Type: 'Z'
  },
  {
    _id: ObjectId("63cff09f2969884310682f93"),
    cust_id: 2,
    Balance: 100,
    Type: 'Z'
  },
  {
    _id: ObjectId("63cff09f2969884310682f94"),
    cust_id: 2,
    Balance: 1000,
    Type: 'C'
  },
  {
    _id: ObjectId("63cff09f2969884310682f95"),
    cust_id: 2,
    Balance: 500,
    Type: 'C'
  },
  {
    _id: ObjectId("63cff09f2969884310682f96"),
    cust_id: 2,
    Balance: 50,
    Type: 'S'
  },
  {
    _id: ObjectId("63cff09f2969884310682f97"),
```

3. Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.

```
db.Customers.aggregate({$match:{Type:"Z"}},{$group:{_id:"$cust_id",TotAccBal:{$sum:"
$Balance"}}},{$match:{TotAccBTotAccBal:{$gt:1200}}});
```

```
[ { _id: 3, TotAccBal: 2000 } ]
```

4. Determine Minimum and Maximum account balance for each customer_id.

db.Customers.aggregate ({$group : { _id :"$cust_id",minAccBal:{$min:"$Balance"},maxAccBal :{$max:"$Balance"} }});

```
[
  { _id: 1, minAccBal: 200, maxAccBal: 1000 },
  { _id: 2, minAccBal: 50, maxAccBal: 1000 },
  { _id: 3, minAccBal: 500, maxAccBal: 1500 }
]
```