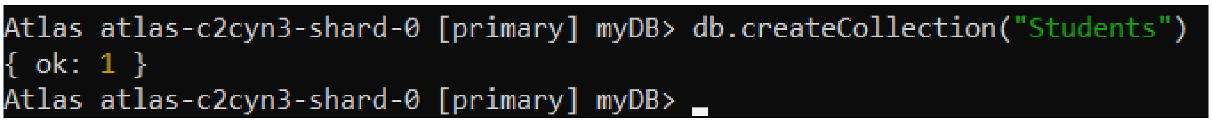
NOSQL LAB1 and LAB2

LAB-1 Perform the following DB operations using MongoDB.

1. Create a database “Student” with the following attributesRollno, Age, ContactNo, Email-Id.

use myDB;
db.createCollection(“Student”);

Output:

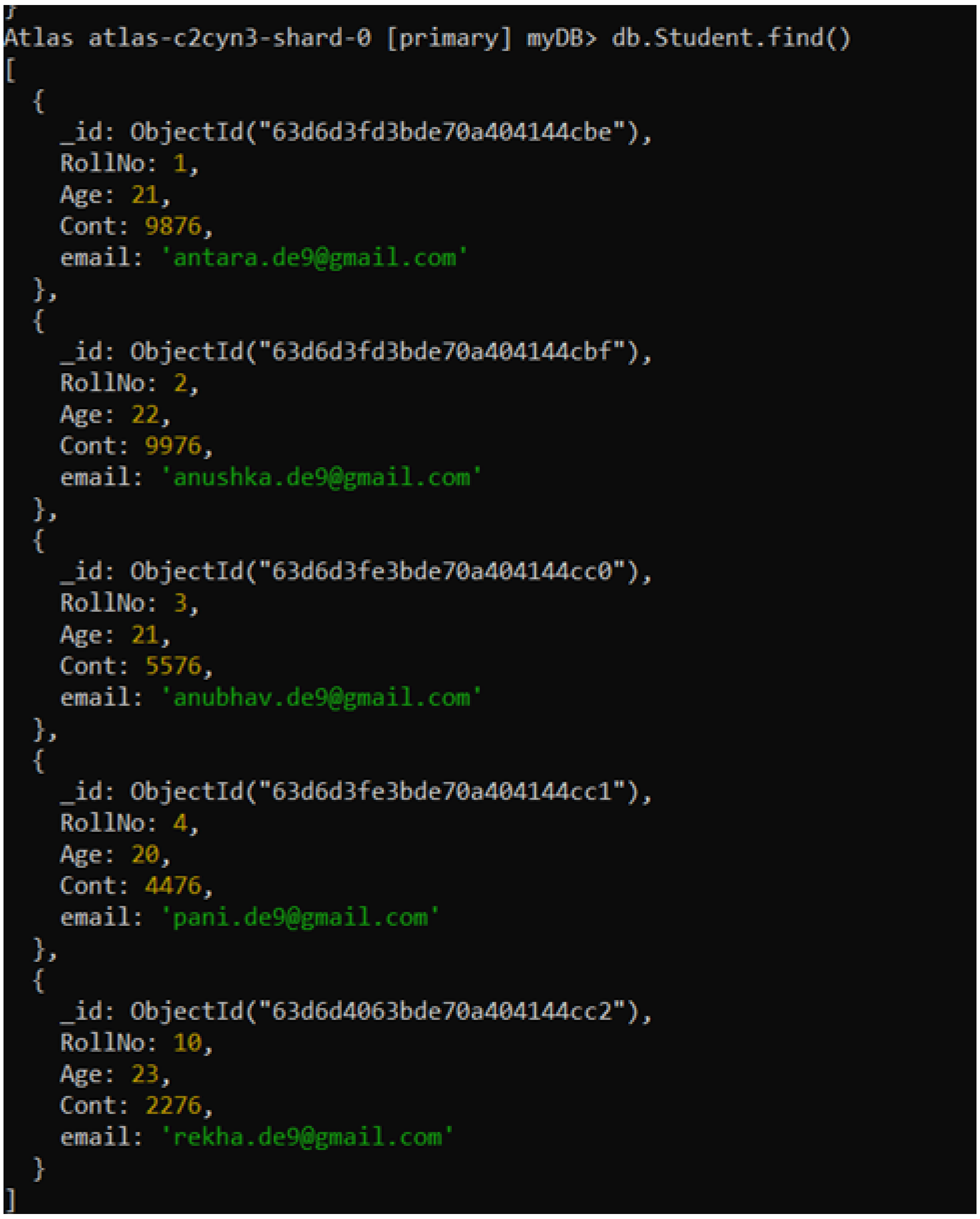


```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.createCollection("Students")
{ ok: 1 }
Atlas atlas-c2cyn3-shard-0 [primary] myDB> _
```

2. Insert appropriate values

db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});
db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});
db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});

Output:

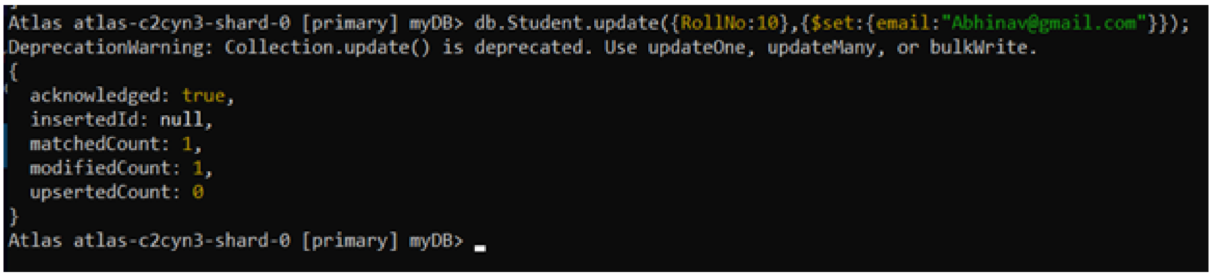


```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.find()
[
  {
    _id: ObjectId("63d6d3fd3bde70a404144cbe"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("63d6d3fd3bde70a404144cbf"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("63d6d3fe3bde70a404144cc0"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("63d6d3fe3bde70a404144cc1"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("63d6d4063bde70a404144cc2"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
```

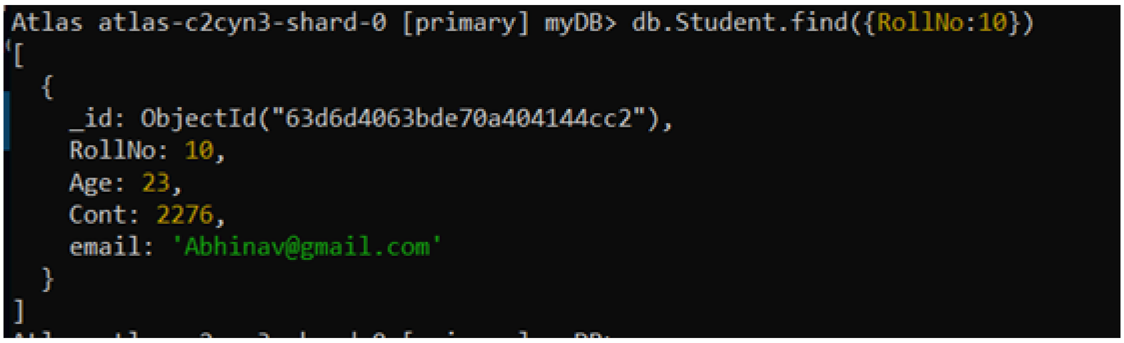
3. Write query to update Email-Id of a student with rollno 10.

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

Output:



```
Atlas atlas-c2cyn3-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-c2cyn3-shard-0 [primary] myDB> _
```

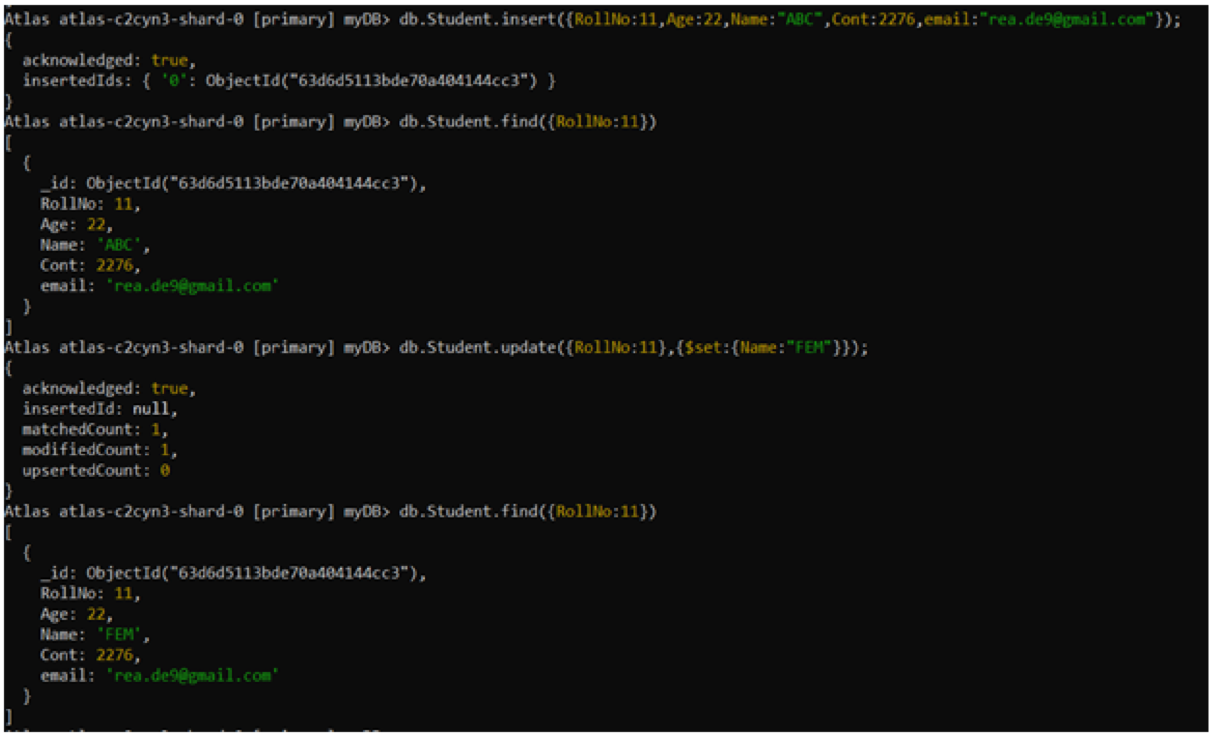


```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.find({RollNo:10})
[
  {
    _id: ObjectId("63d6d4063bde70a404144cc2"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  }
]
```

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

Output:



```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.insert({RollNo:11,Age:22,Name:"ABC",Cont:2276,email:"rea.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63d6d5113bde70a404144cc3") }
}
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.find({RollNo:11})
[
  {
    _id: ObjectId("63d6d5113bde70a404144cc3"),
    RollNo: 11,
    Age: 22,
    Name: 'ABC',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.update({RollNo:11},{$set:{Name:"FEM"}});
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.find({RollNo:11})
[
  {
    _id: ObjectId("63d6d5113bde70a404144cc3"),
    RollNo: 11,
    Age: 22,
    Name: 'FEM',
    Cont: 2276,
    email: 'rea.de9@gmail.com'
  }
]
```

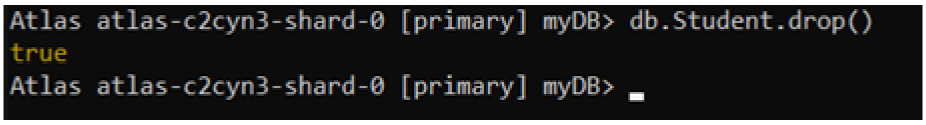
5. Export the created table into local file system

Mongoexport
mongodb+srv://Shamil:asdQWE123@cluster0.pgvw4kr.mongodb.net/myDB
--collection=Student --out C:\Users\shami\Downloads\NOSQL/output.json
Output:

6. Drop the table

db.Student.drop();

Output:



```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Student.drop()
true
Atlas atlas-c2cyn3-shard-0 [primary] myDB> _
```

7. Import a given csv dataset from local file system into mongodb collection.

Mongoimport
mongodb+srv://Shamil:asdQWE123@cluster0.pgvw4kr.mongodb.net/myDB --collection= New_Student --type json --file C:\Users\shami\Downloads\NOSQL/output.json

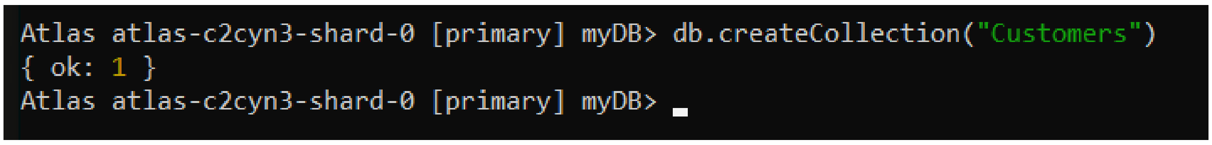
Output:

LAB2: Perform the following DB operations using MongoDB.

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

db.createCollection("Customers");

Output:

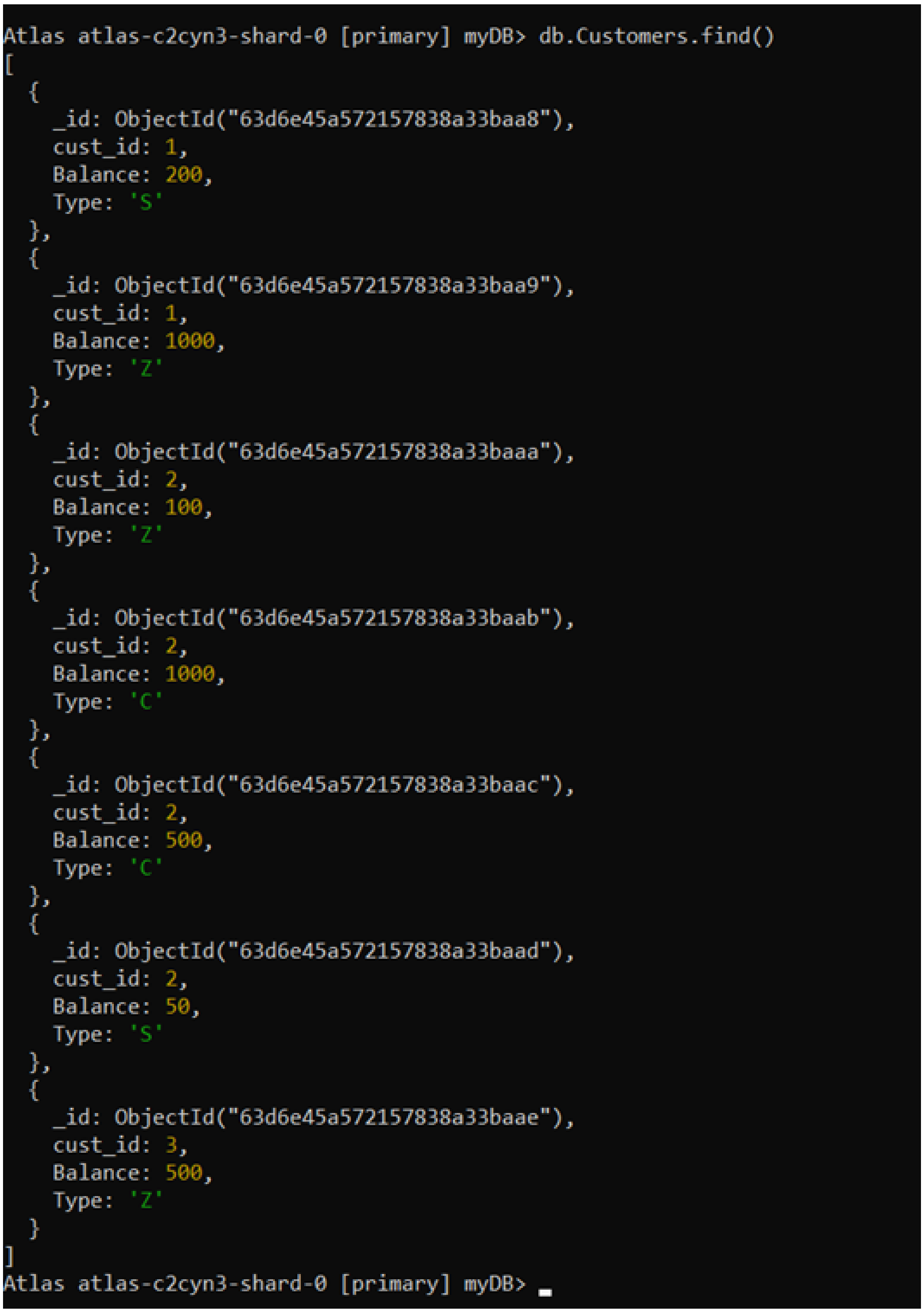


```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.createCollection("Customers")
{ ok: 1 }
Atlas atlas-c2cyn3-shard-0 [primary] myDB> _
```

2. Insert at least 5 values into the table

```
db.Customers.insert({cust_id:1,Balance:200, Type:"S"});
db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
db.Customers.insert({cust_id:2,Balance:50, Type:"S"});
db.Customers.insert({cust_id:3,Balance:500, Type:"Z"});
```

Output:

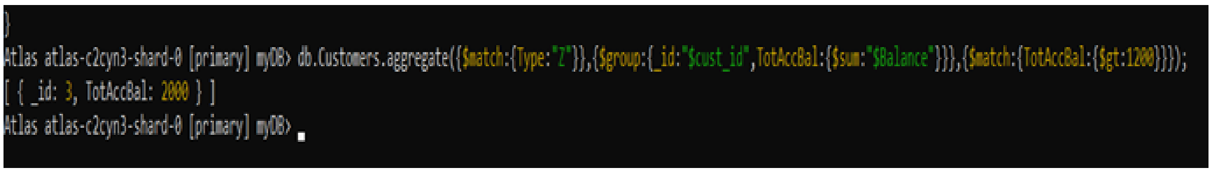


```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Customers.find()
[
  {
    _id: ObjectId("63d6e45a572157838a33baa8"),
    cust_id: 1,
    Balance: 200,
    Type: 'S'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baa9"),
    cust_id: 1,
    Balance: 1000,
    Type: 'Z'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baaa"),
    cust_id: 2,
    Balance: 100,
    Type: 'Z'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baab"),
    cust_id: 2,
    Balance: 1000,
    Type: 'C'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baac"),
    cust_id: 2,
    Balance: 500,
    Type: 'C'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baad"),
    cust_id: 2,
    Balance: 50,
    Type: 'S'
  },
  {
    _id: ObjectId("63d6e45a572157838a33baae"),
    cust_id: 3,
    Balance: 500,
    Type: 'Z'
  }
]
Atlas atlas-c2cyn3-shard-0 [primary] myDB> _
```

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"}}},{$match:{TotAccBal:{$gt:1200}}});
```

Output:

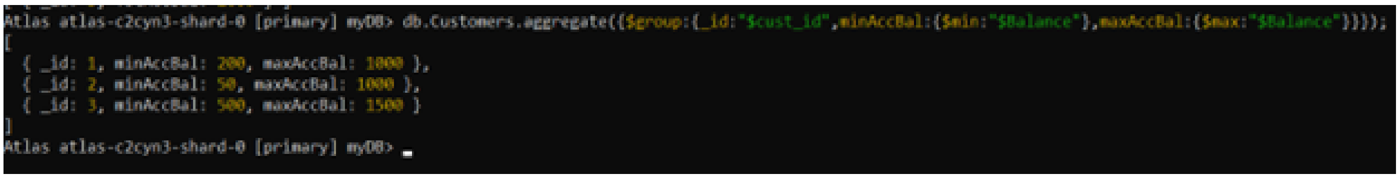


```
}
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Customers.aggregate({$match:{Type:"Z"}},{$group:{_id:"$cust_id",TotAccBal:{$sum:"$Balance"}}},{$match:{TotAccBal:{$gt:1200}}});
[ { _id: 3, TotAccBal: 2000 } ]
Atlas atlas-c2cyn3-shard-0 [primary] myDB>
```

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

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

Output:



```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> 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 }
]
Atlas atlas-c2cyn3-shard-0 [primary] myDB>
```

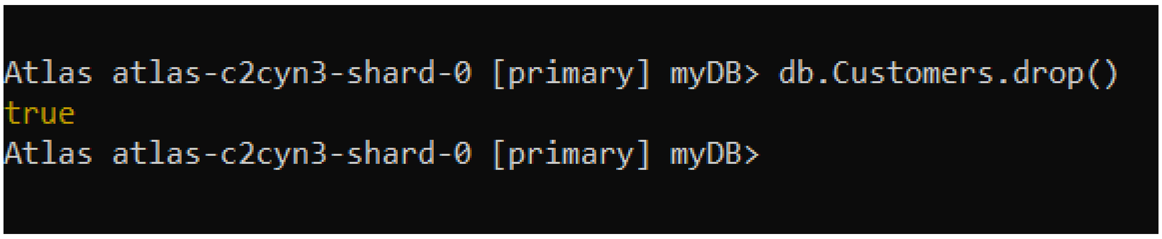
5. Export the created collection into local file system

Mongoexport
mongodb+srv://Shamil:asdQWE123@cluster0.pgvw4kr.mongodb.net/myDB
--collection=Customers --out C:\Users\shami\Downloads\NOSQL/output.json
Output:

6. Drop the table

db.Customers.drop();

Output:



```
Atlas atlas-c2cyn3-shard-0 [primary] myDB> db.Customers.drop()
true
Atlas atlas-c2cyn3-shard-0 [primary] myDB>
```

7. Import a given csv dataset from local file system into mongodb collection.

Mongoimport
mongodb+srv://Shamil:asdQWE123@cluster0.pgvw4kr.mongodb.net/myDB --collection=
New_Collection --type json --file C:\Users\shami\Downloads\NOSQL/output.json

Output: