**NOSQL LAB1 and LAB2**

**-Tanisha Gotadke (1BM21CS229)**

**LAB-1 Perform the following DB operations using MongoDB.**

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

Id.

use myDB;

db.createCollection(“Student”);

Output:



2. Insert appropriate values

db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});

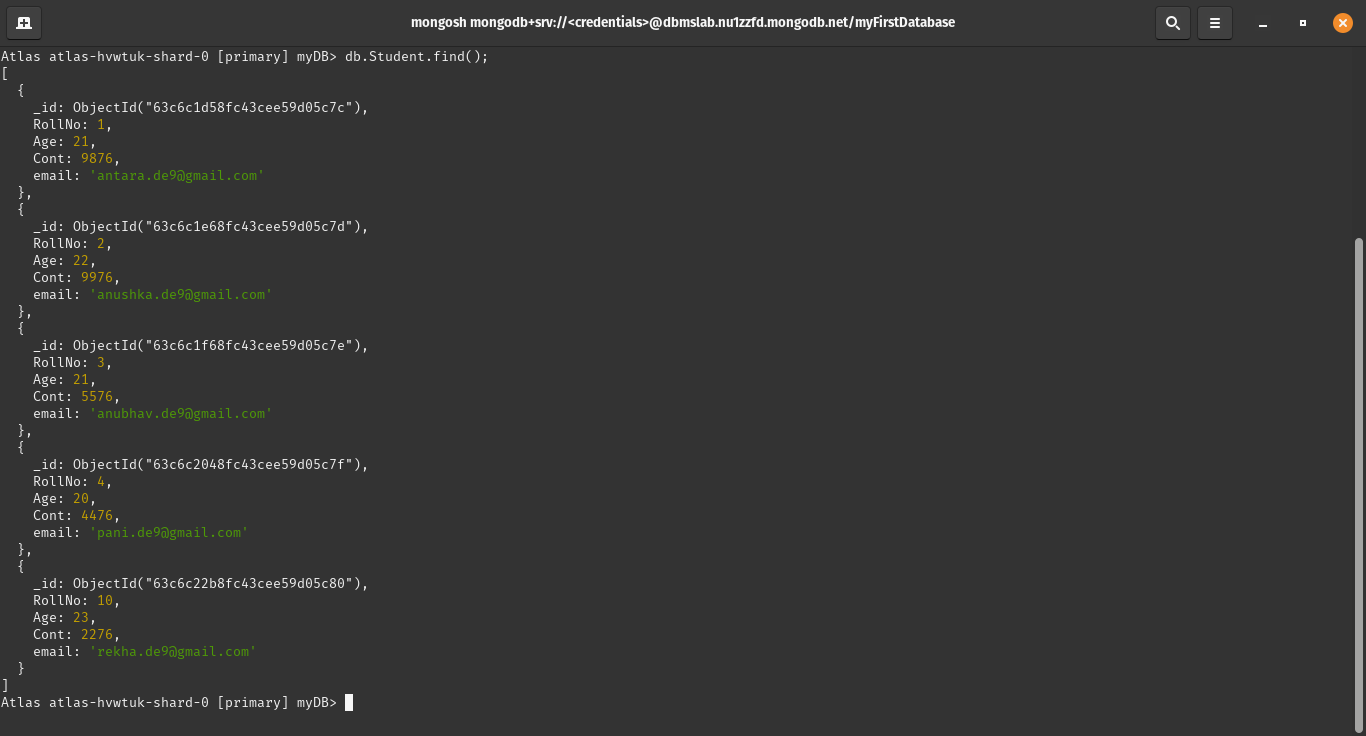
db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

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](mailto:rekha.de9@gmail.com)"});

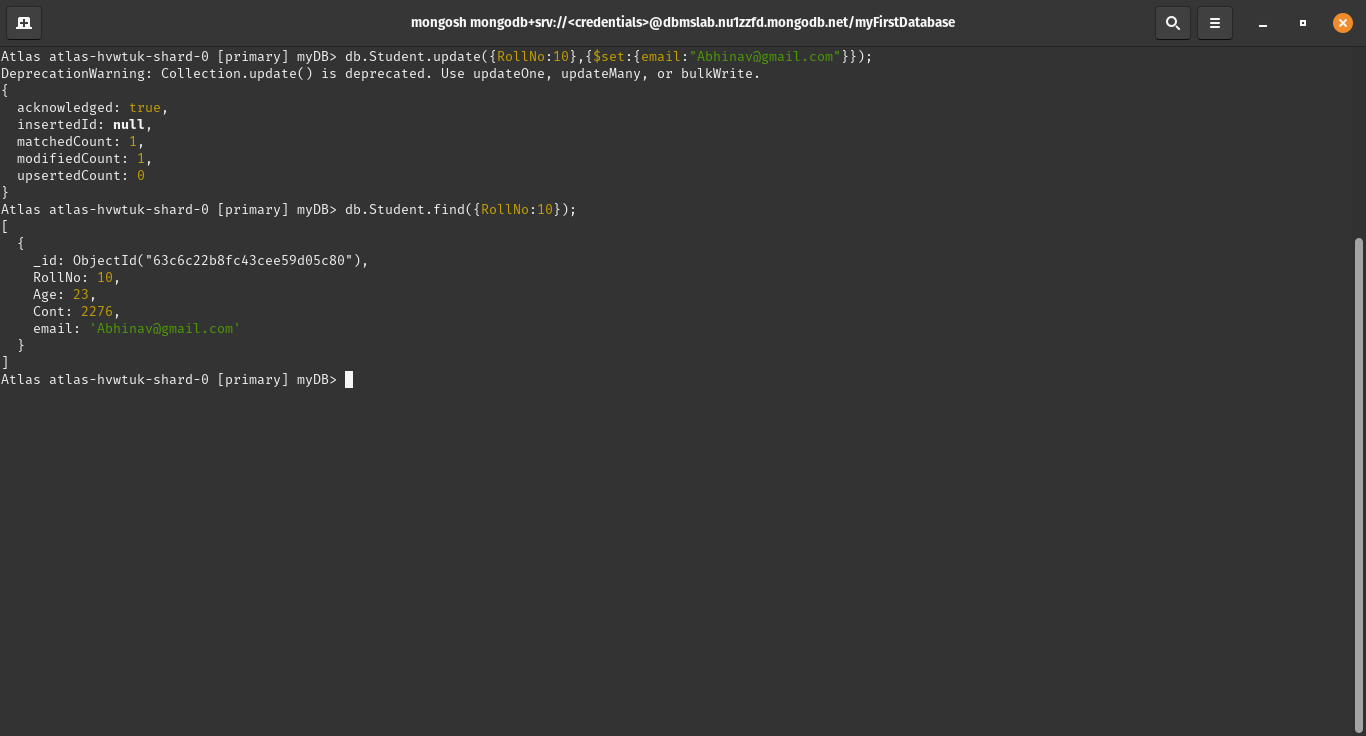
Output:



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

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

Output:

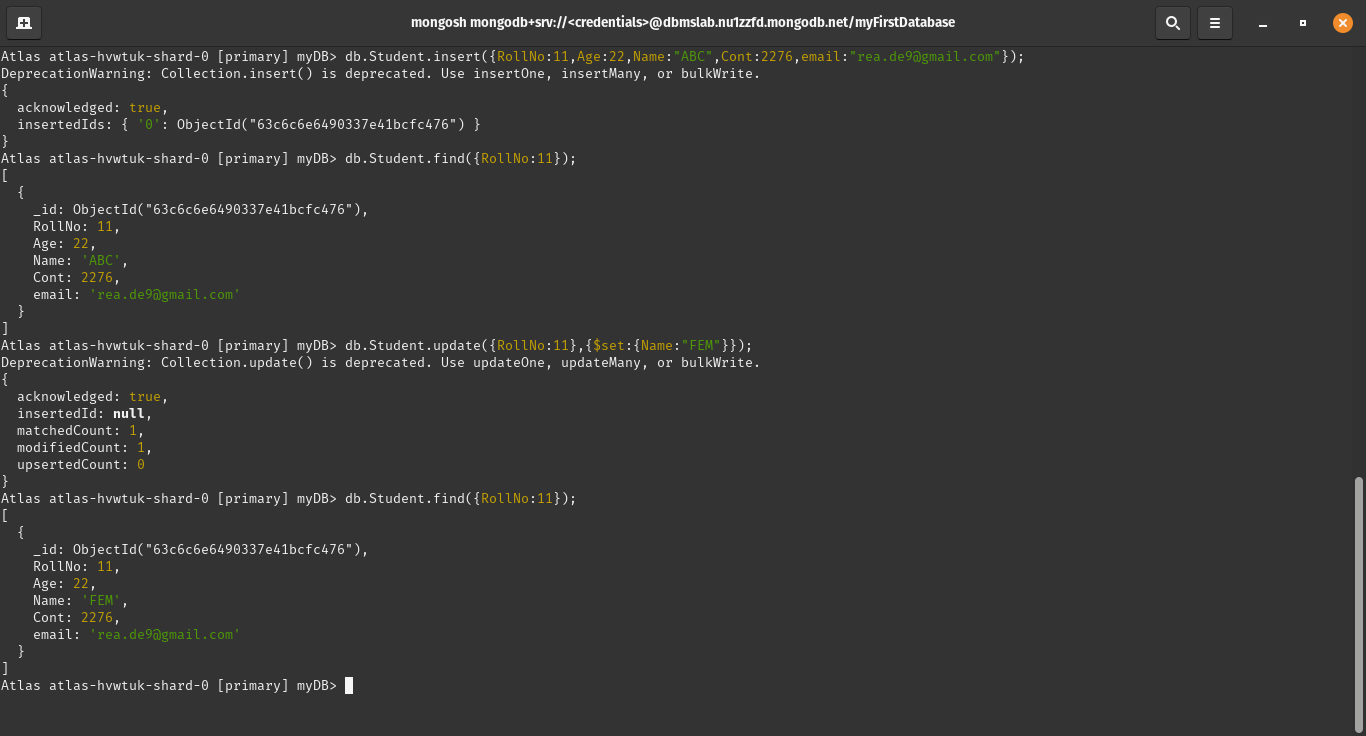


4. Replace the student name from “ABC” to “FEM” of rollno 11.

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

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

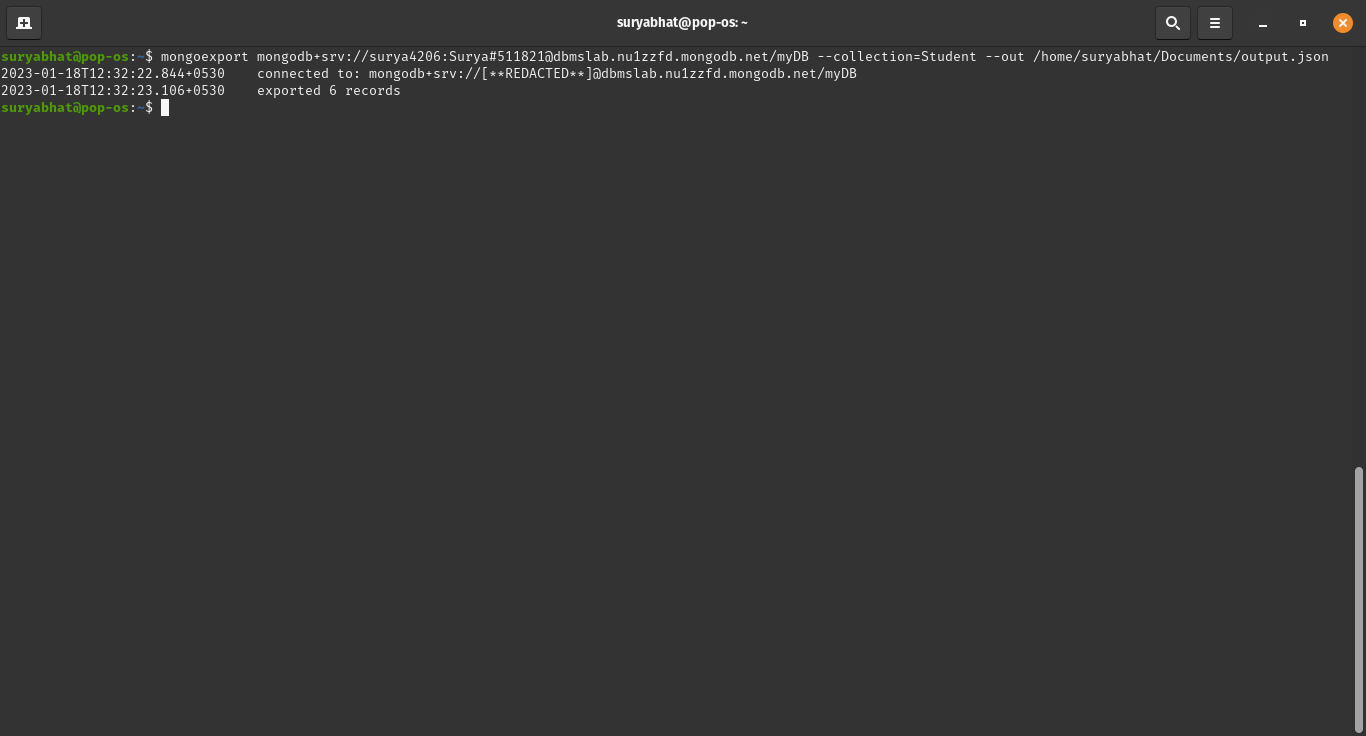
Output:



5. Export the created table into local file system

Mongoexport mongodb+srv://:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=Student --out /home/Tanishagotadke/Documents/output.json

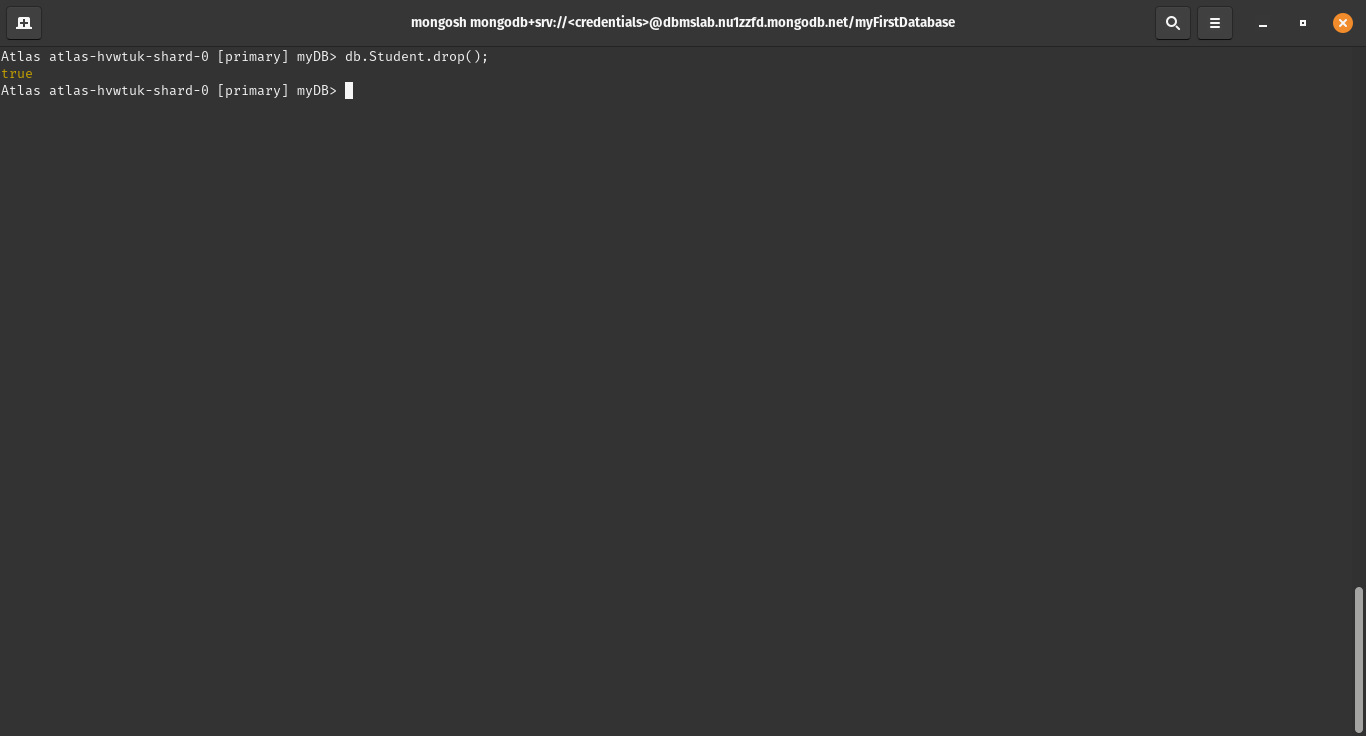
Output:



6. Drop the table

db.Student.drop();

Output:



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

mongoimport mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=New\_Student --type json --file /home/Tanishagotadke/Documents/output.json

Output:



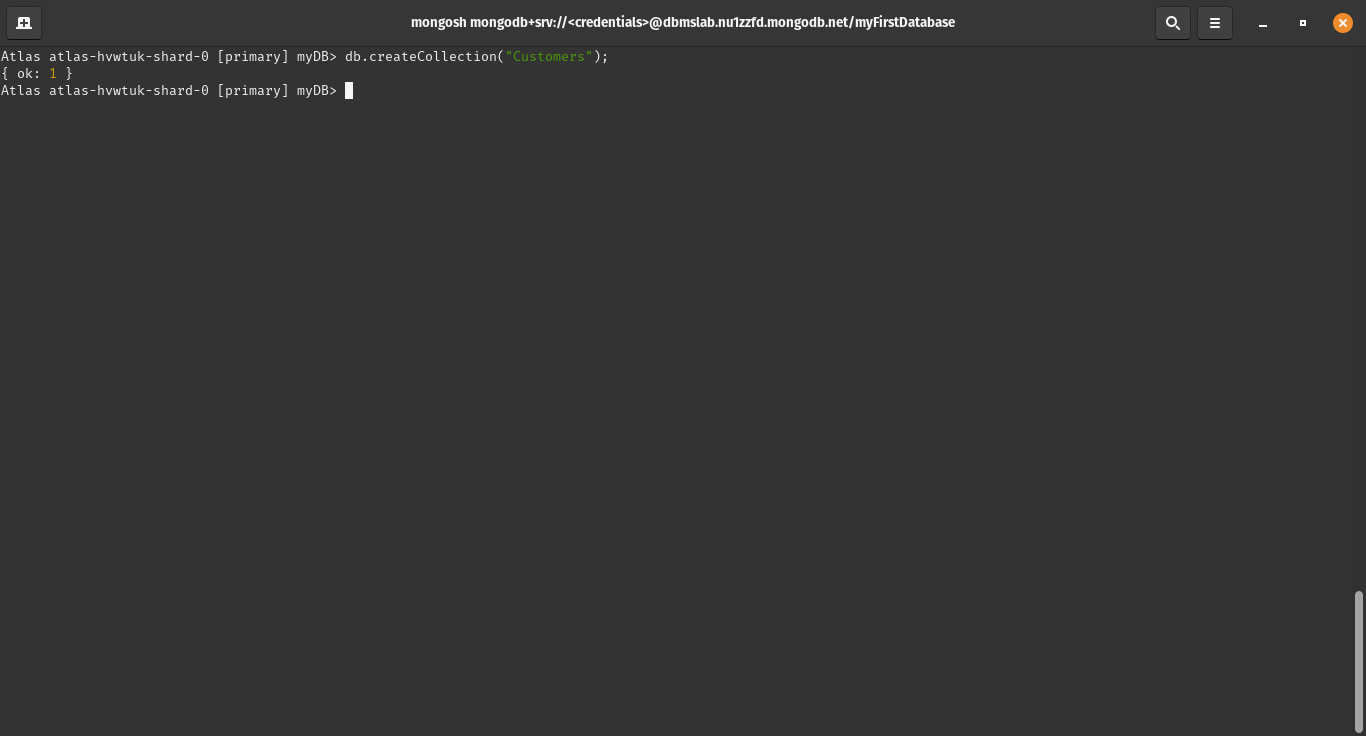
**LAB2: Perform the following DB operations using MongoDB.**

1. Create a collection by name Customers with the following attributes.

Cust\_id, Acc\_Bal, Acc\_Type

db.createCollection(“Customers”);

Output:



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"});

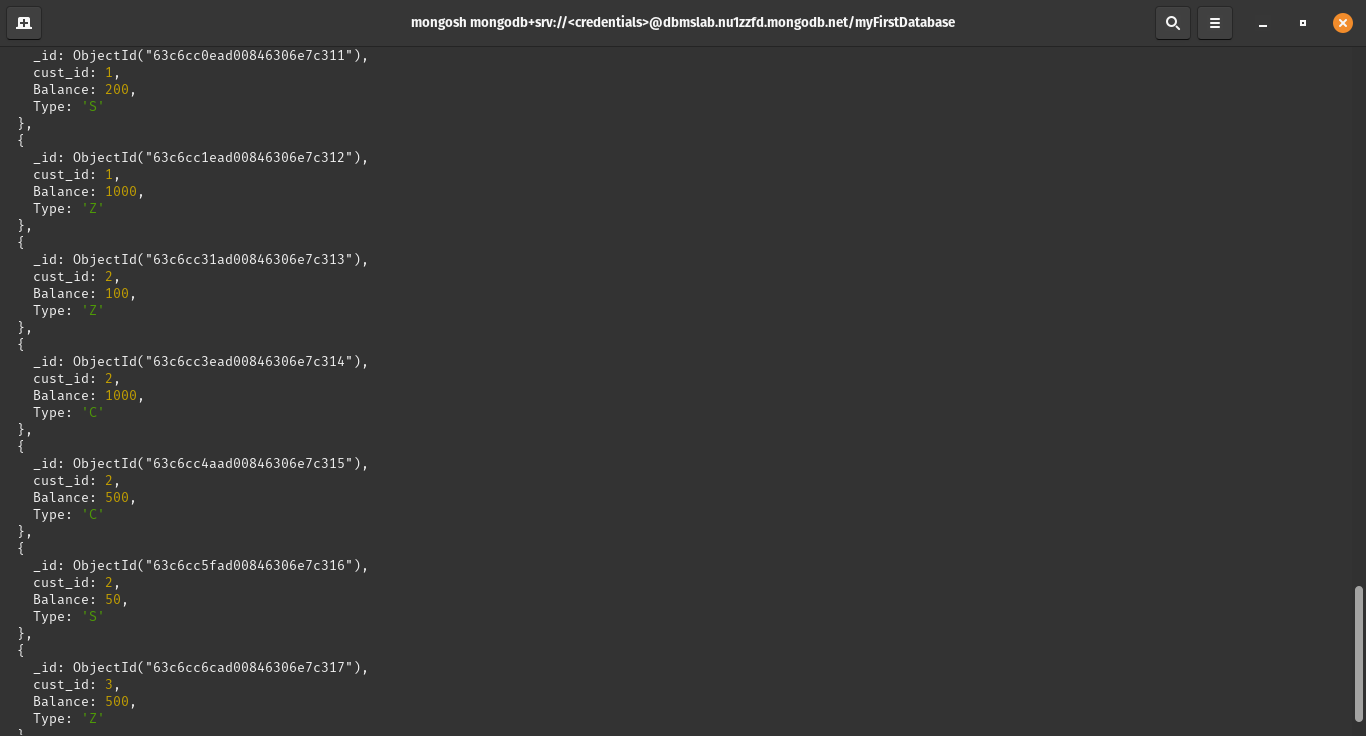
db.Customers.insert({cust\_id:2,Balance:1000, Type:"C"});

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:

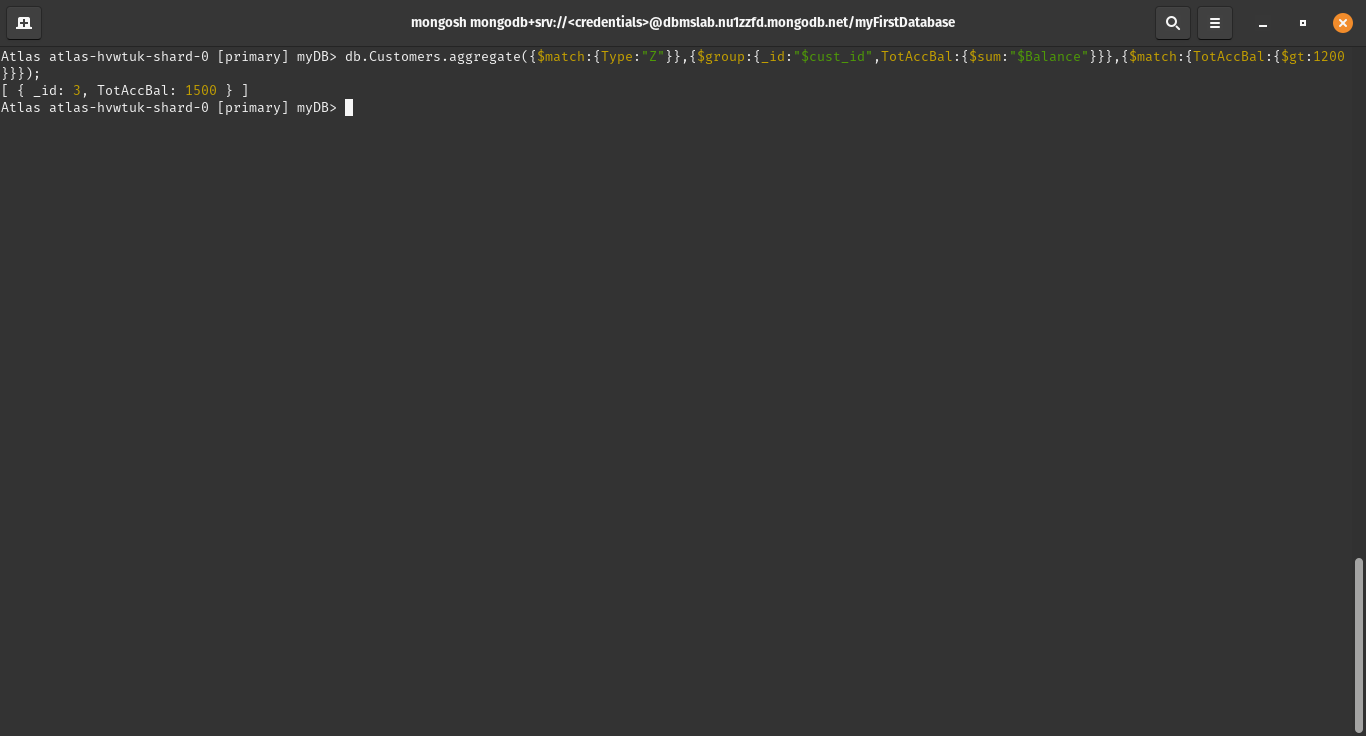


3. Write a query to display those records whose total account balance is greater than

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}}});

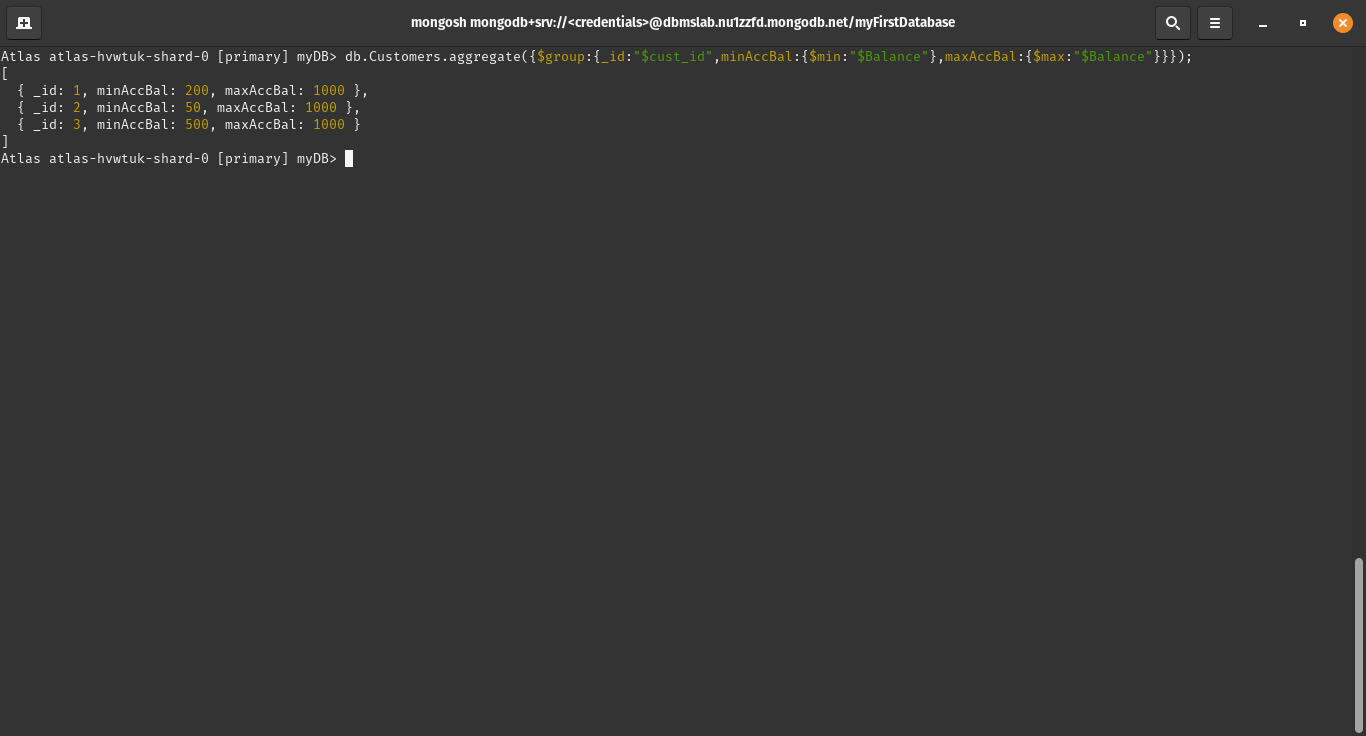
Output:



4. Determine Minimum and Maximum account balance for each customer\_id.

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

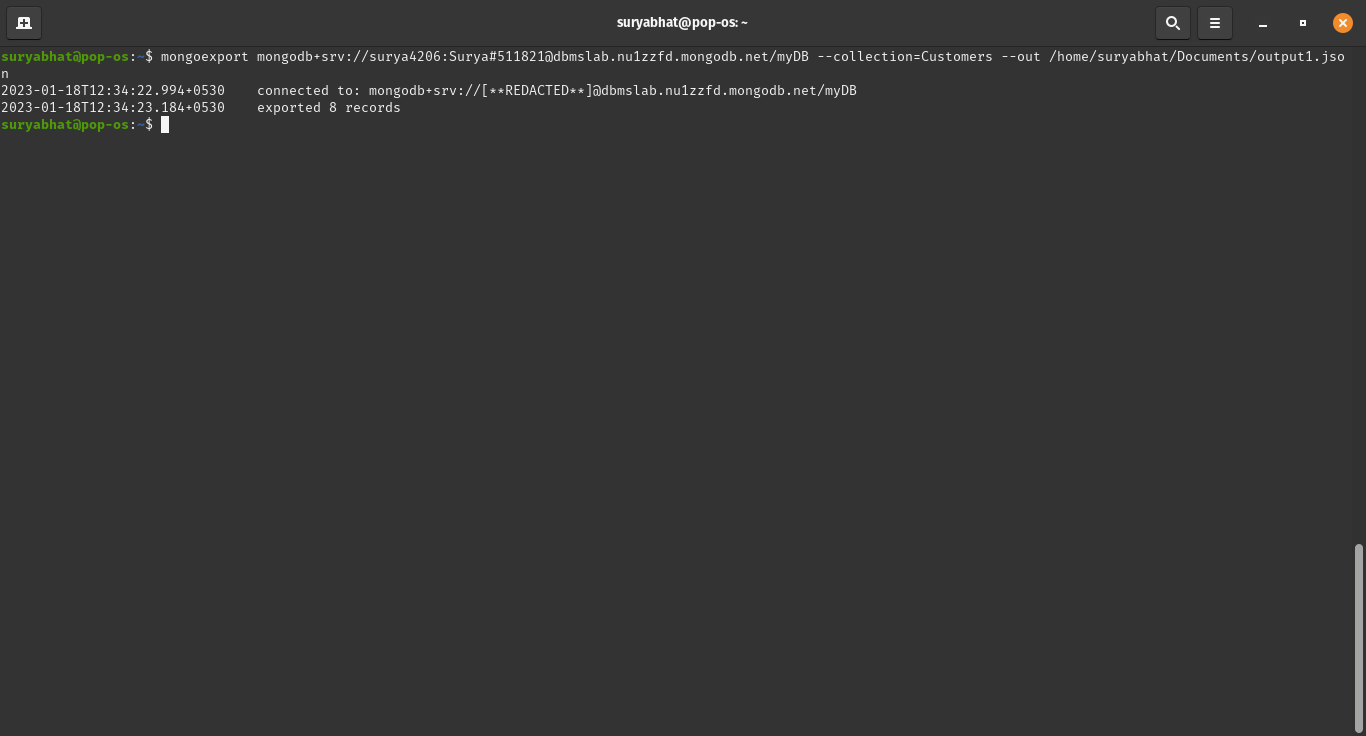
Output:



5. Export the created collection into local file system

mongoexport mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=Customers --out /home/Tanishagotadke/Documents/output1.json

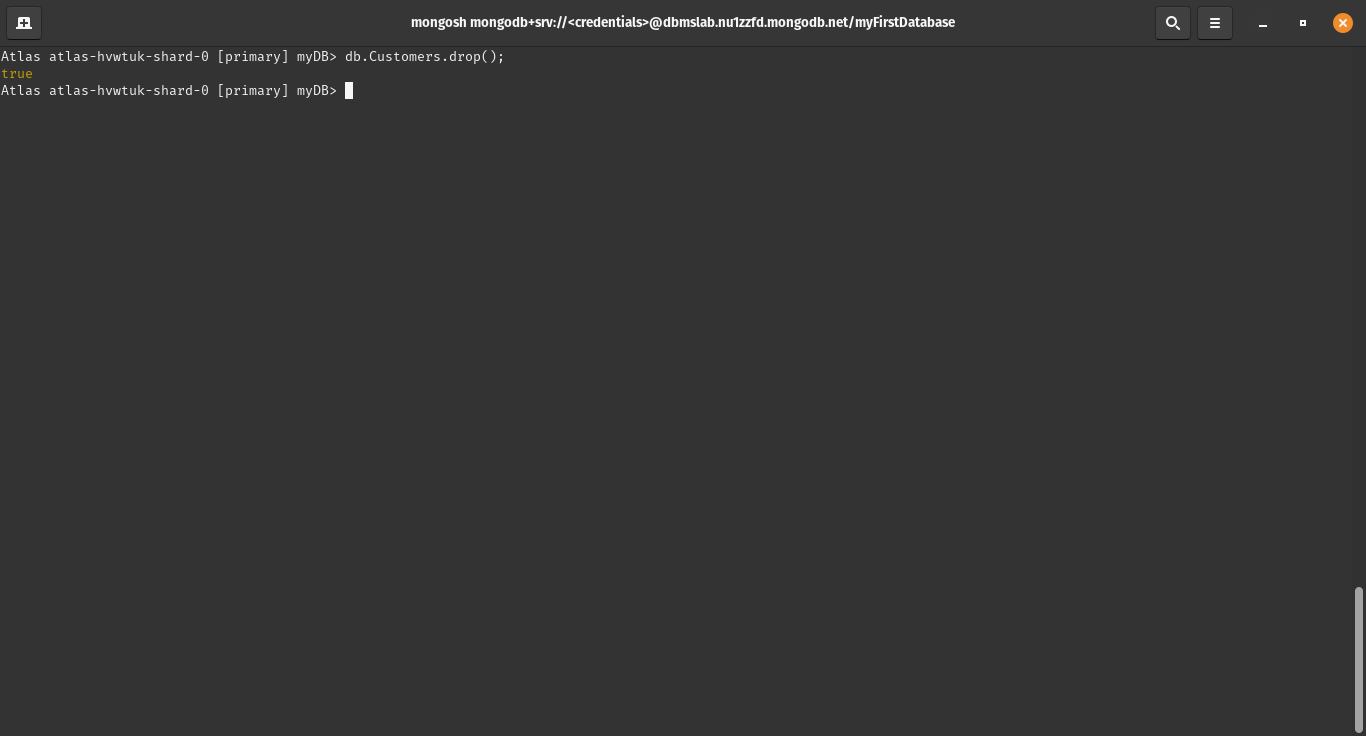
Output:



6. Drop the table

db.Customers.drop();

Output:



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

mongoimport mongodb+srv://Tanisha4206:Tanisha#511821@dbmslab.nu1zzfd.mongodb.net/myDB --collection=New\_Collection --type json --file /home/Tanishagotadke/Documents/output1.json

Output:

