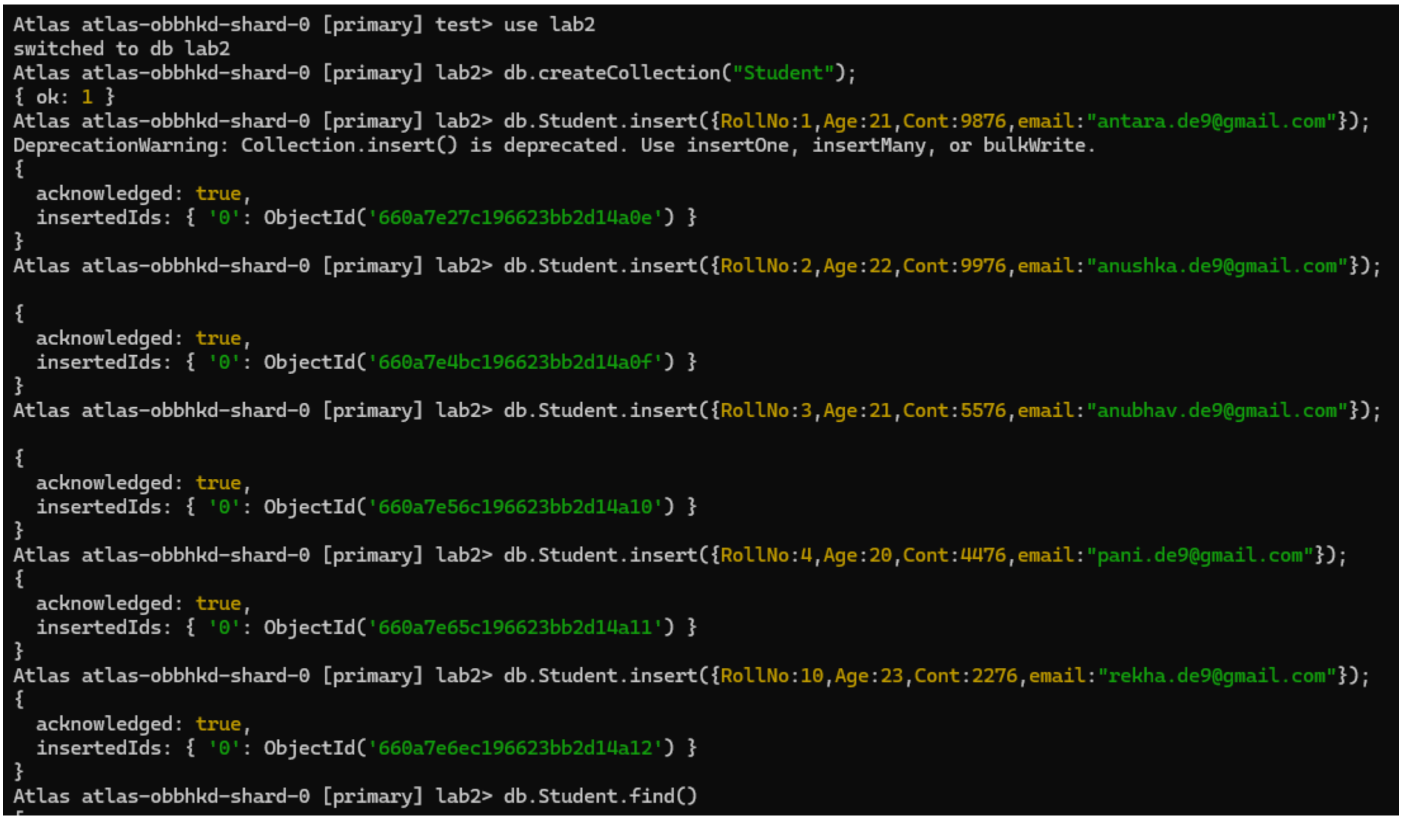
LAB 2

I. Perform the following DB operations using MongoDB.

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



```
Atlas atlas-obbhkd-shard-0 [primary] test> use lab2
switched to db lab2
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e27c196623bb2d14a0e') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e4bc196623bb2d14a0f') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e56c196623bb2d14a10') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e65c196623bb2d14a11') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e6ec196623bb2d14a12') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.find()
```

use lab2

db.createCollection("Student");

2. Insert appropriate values

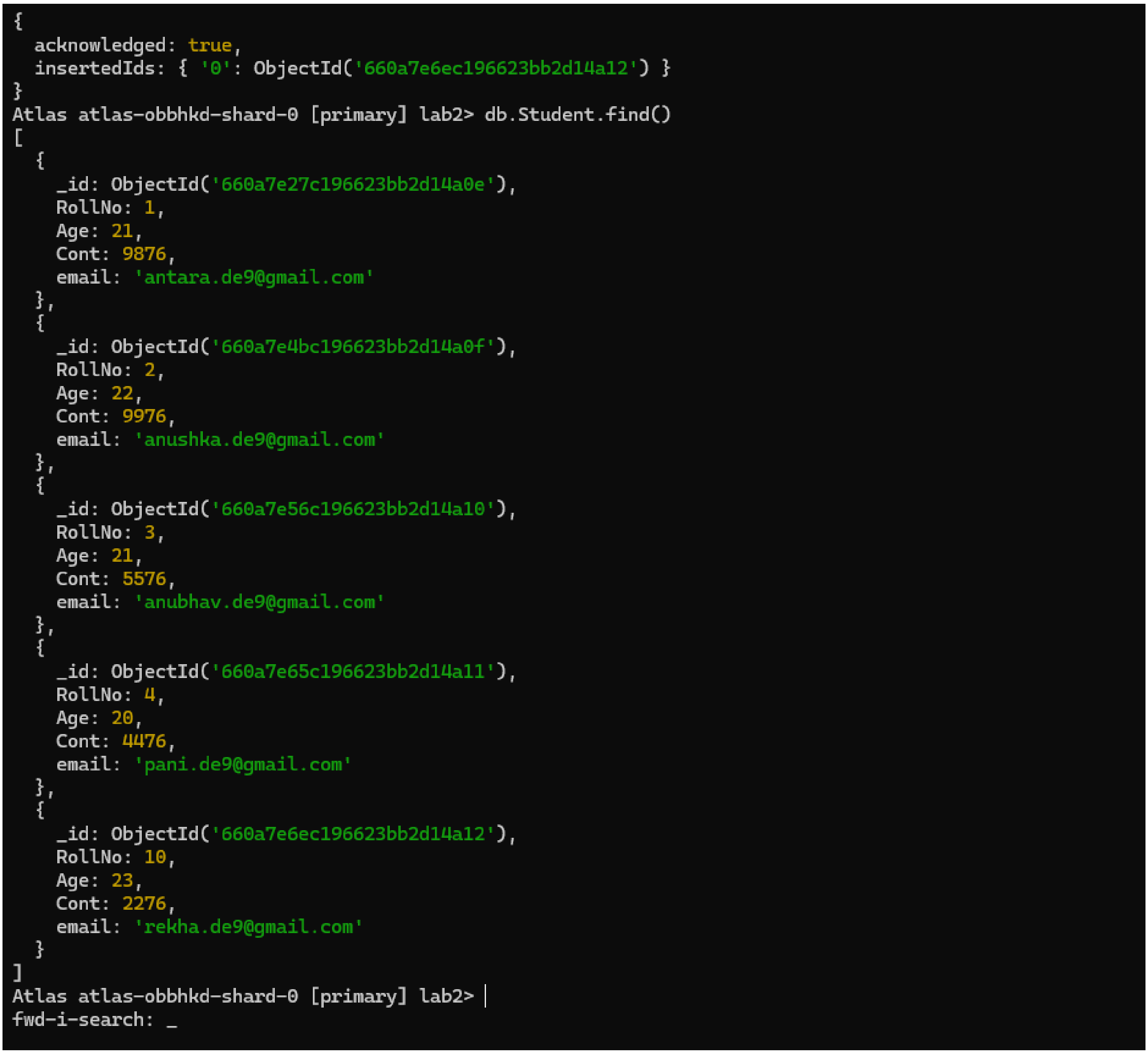
db.Student.insert({RollNo:1,Age:21,Cont:9876,email:"antara.de9@gmail.com"});

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"});

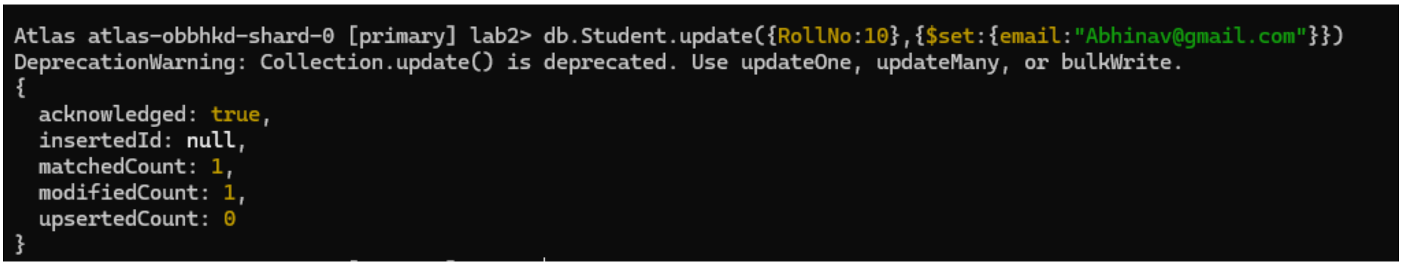
db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});

db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});



```
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a7e6ec196623bb2d14a12') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.find()
[
  {
    _id: ObjectId('660a7e27c196623bb2d14a0e'),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId('660a7e4bc196623bb2d14a0f'),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId('660a7e56c196623bb2d14a10'),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId('660a7e65c196623bb2d14a11'),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId('660a7e6ec196623bb2d14a12'),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
Atlas atlas-obbhkd-shard-0 [primary] lab2> |
fwd-i-search: _
```

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

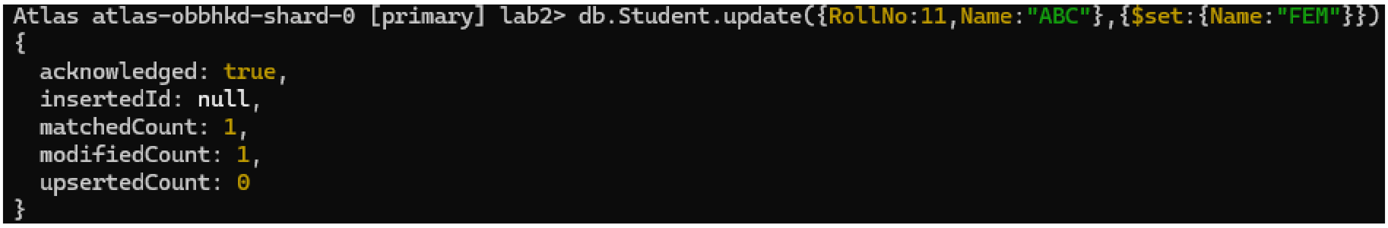


```
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.update({RollNo:10},{$set:{email:"Abhinav@gmail.com"}})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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

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

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

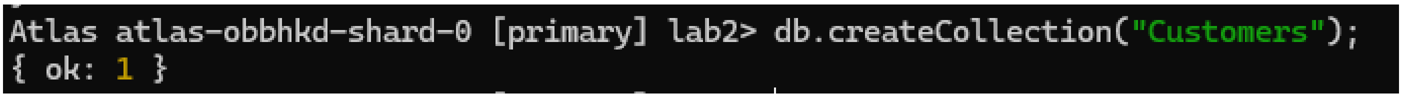


```
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Student.update({RollNo:11,Name:"ABC"},{$set:{Name:"FEM"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

II. Perform the following DB operations using MongoDB.

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

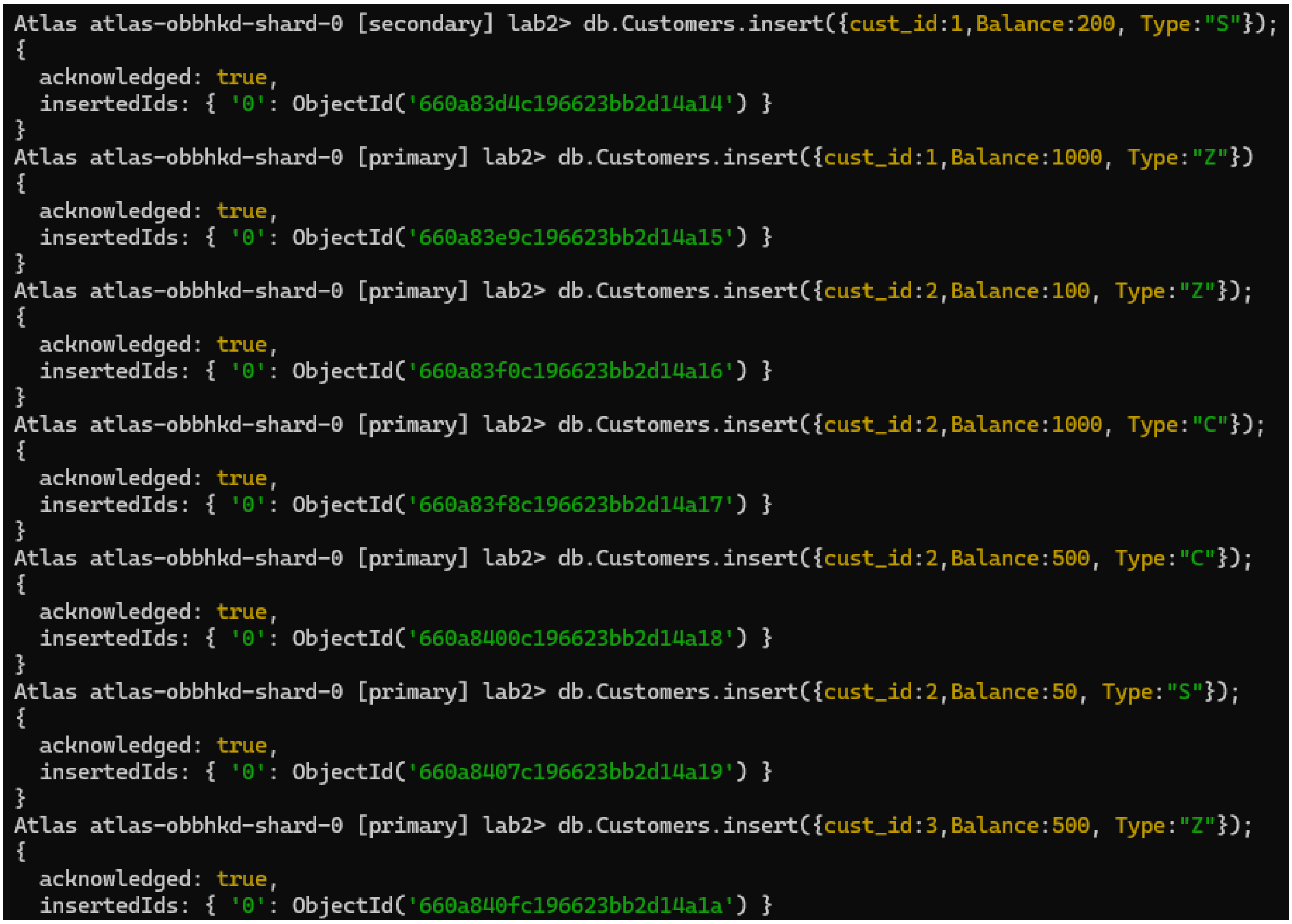
Cust_id, Acc_Bal, Acc_Type



```
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.createCollection("Customers");
{ ok: 1 }
```

db.createCollection("Customers");

2. Insert at least 5 values into the table



```
Atlas atlas-obbhkd-shard-0 [secondary] lab2> db.Customers.insert({cust_id:1,Balance:200, Type:"S"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a83d4c196623bb2d14a14') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:1,Balance:1000, Type:"Z"})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a83e9c196623bb2d14a15') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:2,Balance:100, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a83f0c196623bb2d14a16') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:2,Balance:1000, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a83f8c196623bb2d14a17') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:2,Balance:500, Type:"C"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a8400c196623bb2d14a18') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:2,Balance:50, Type:"S"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a8407c196623bb2d14a19') }
}
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.insert({cust_id:3,Balance:500, Type:"Z"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a840fc196623bb2d14a1a') }
```

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"});

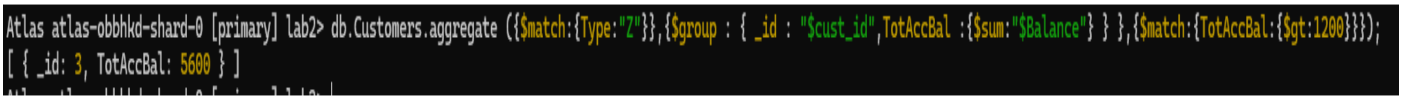
db.Customers.insert({cust_id:2,Balance:500, Type:"C"});

db.Customers.insert({cust_id:2,Balance:50, Type:"S"});

db.Customers.insert({cust_id:3,Balance:500, Type:"Z"});

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

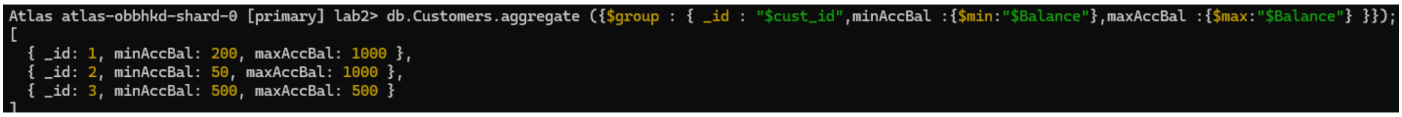
1200 of account type ‘Z’ for each customer_id.



```
Atlas atlas-obbhkd-shard-0 [primary] lab2> db.Customers.aggregate ({$match:{Type:"Z"}},{$group : { _id : "$cust_id",TotAccBal :{$sum:"$Balance"} } },{$match:{TotAccBal:{$gt:1200}}});
[ { _id: 3, TotAccBal: 5600 } ]
```

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

4. Determine Minimum and Maximum account balance for each customer_i



```
Atlas atlas-obbhkd-shard-0 [primary] lab2> 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: 500 }
]
```

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