VANDAN N KOTHARI

1BM21CS236

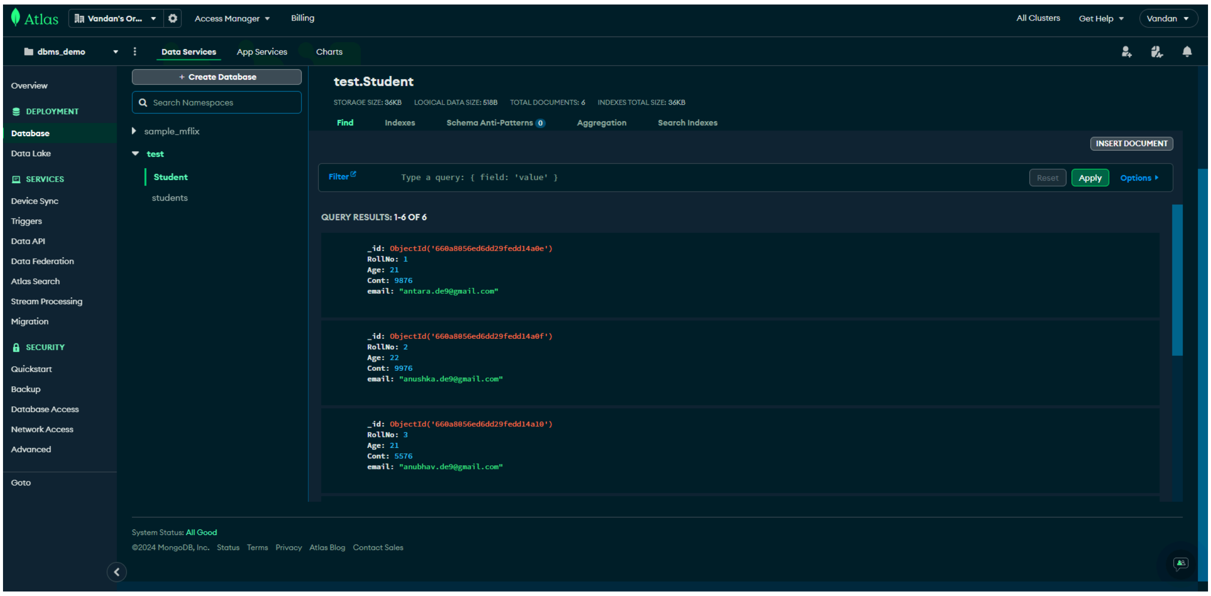
I. Perform the following DB operations using MongoDB.

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

2. Insert appropriate values

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

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



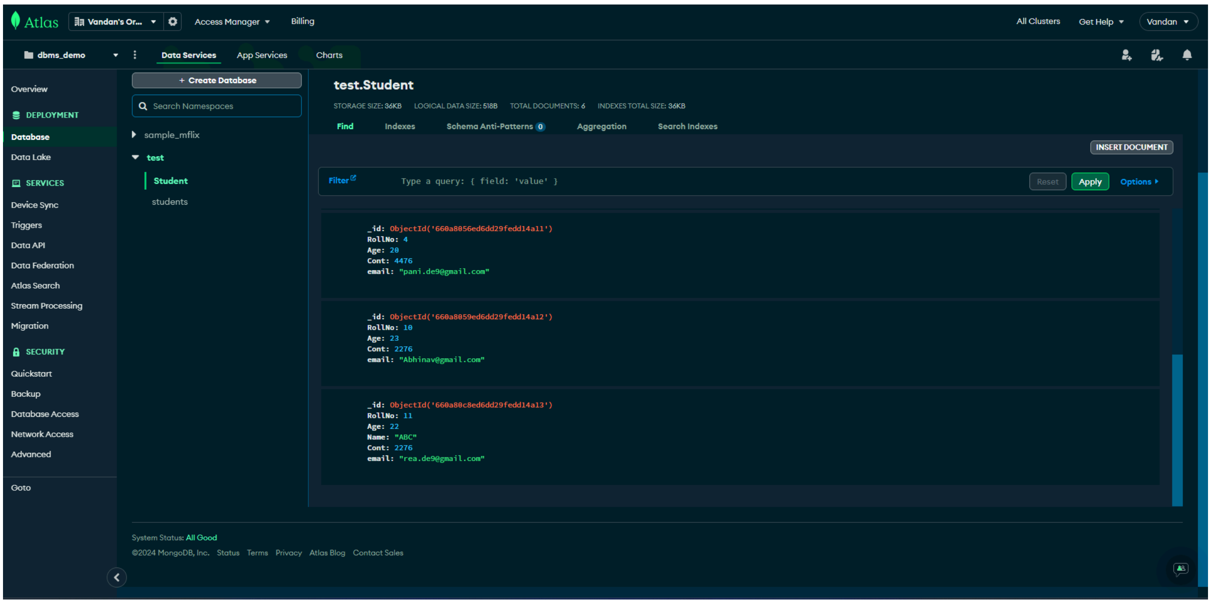
test.Student

STORAGE SIZE: 36KB LOGICAL DATA SIZE: 518B TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Type a query: { field: 'value' }

QUERY RESULTS: 1-6 OF 6

```
_id: ObjectId('660a8056ed6dd29fedd14a0e')
RollNo: 1
Age: 21
Cont: 9876
email: "antara.de9@gmail.com"

_id: ObjectId('660a8056ed6dd29fedd14a0f')
RollNo: 2
Age: 22
Cont: 9976
email: "anushka.de9@gmail.com"

_id: ObjectId('660a8056ed6dd29fedd14a10')
RollNo: 3
Age: 21
Cont: 5576
email: "anubhav.de9@gmail.com"
```



test.Student

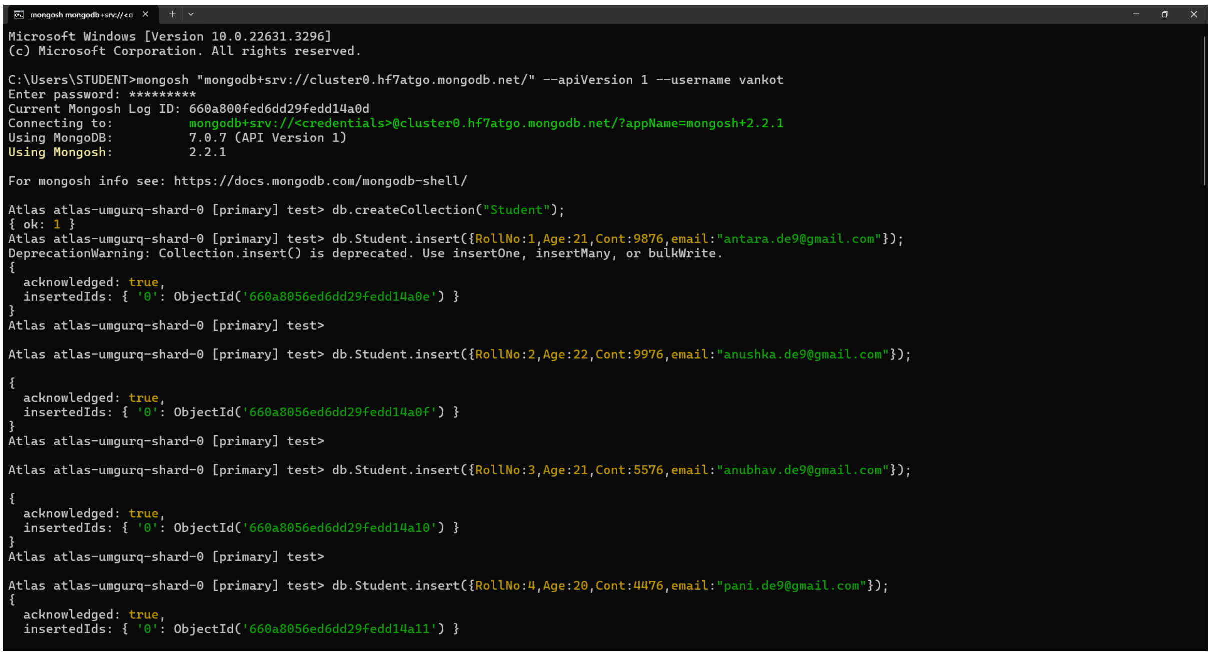
STORAGE SIZE: 36KB LOGICAL DATA SIZE: 518B TOTAL DOCUMENTS: 6 INDEXES TOTAL SIZE: 36KB

Type a query: { field: 'value' }

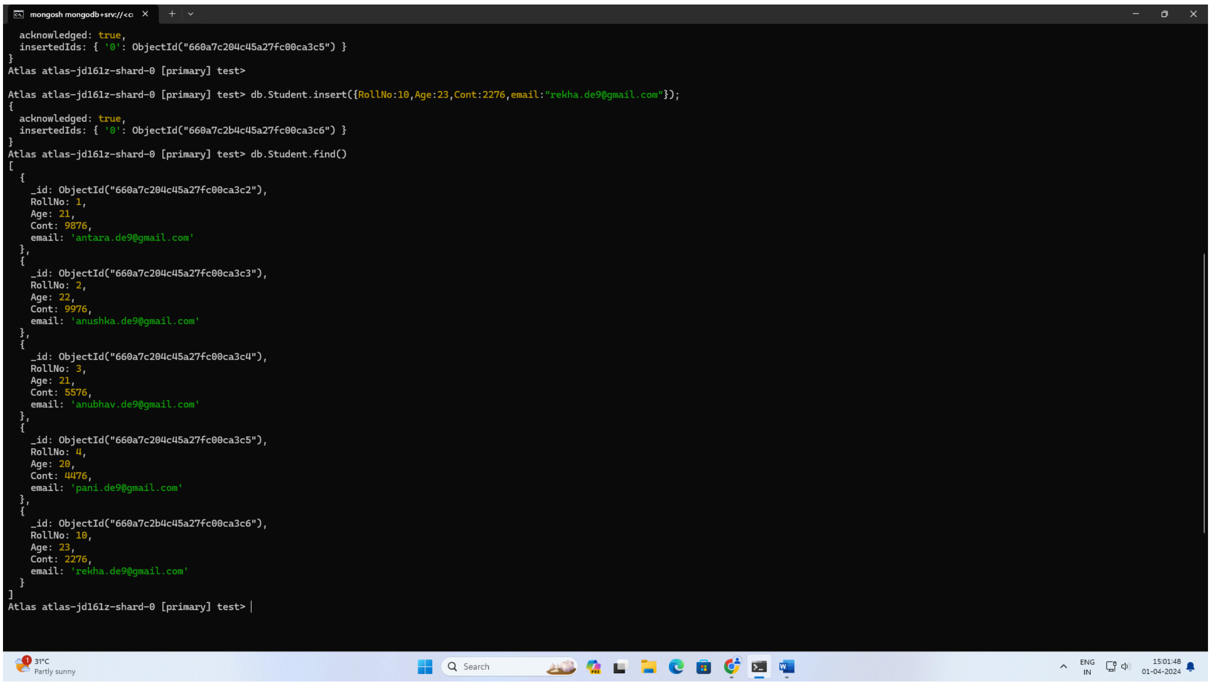
```
_id: ObjectId('660a8056ed6dd29fedd14a11')
RollNo: 4
Age: 20
Cont: 4476
email: "pani.de9@gmail.com"

_id: ObjectId('660a8059ed6dd29fedd14a12')
RollNo: 10
Age: 23
Cont: 2276
email: "Abhinav@gmail.com"

_id: ObjectId('660a80c8ed6dd29fedd14a13')
RollNo: 11
Age: 22
Name: "ABC"
Cont: 2276
email: "rea.de9@gmail.com"
```

VANDAN N KOTHARI

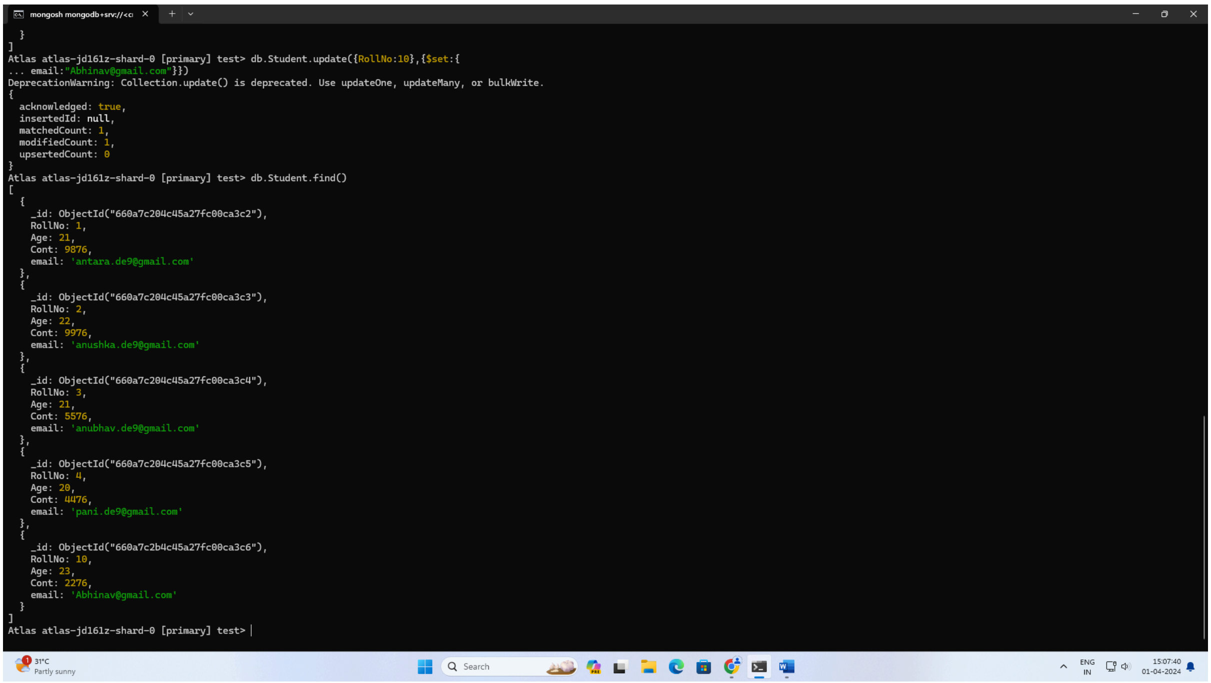
1BM21CS236



```
Microsoft Windows [Version 10.0.22631.3296]
(c) Microsoft Corporation. All rights reserved.

C:\Users\STUDENT>mongosh "mongodb+srv://cluster0.hf7atgo.mongodb.net/" --apiVersion 1 --username vankot
Enter password: *********
Current Mongosh Log ID: 660a800fed6dd29fedd14a0d
Connecting to:          mongodb+srv://<credentials>@cluster0.hf7atgo.mongodb.net/?appName=mongosh+2.2.1
Using MongoDB:          7.0.7 (API Version 1)
Using Mongosh:          2.2.1

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

Atlas atlas-umgurq-shard-0 [primary] test> db.createCollection("Student");
{ ok: 1 }
Atlas atlas-umgurq-shard-0 [primary] test> 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('660a8056ed6dd29fedd14a0e') }
}
Atlas atlas-umgurq-shard-0 [primary] test>

Atlas atlas-umgurq-shard-0 [primary] test> db.Student.insert({RollNo:2,Age:22,Cont:9976,email:"anushka.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a8056ed6dd29fedd14a0f') }
}
Atlas atlas-umgurq-shard-0 [primary] test>

Atlas atlas-umgurq-shard-0 [primary] test> db.Student.insert({RollNo:3,Age:21,Cont:5576,email:"anubhav.de9@gmail.com"});

{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a8056ed6dd29fedd14a10') }
}
Atlas atlas-umgurq-shard-0 [primary] test>

Atlas atlas-umgurq-shard-0 [primary] test> db.Student.insert({RollNo:4,Age:20,Cont:4476,email:"pani.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId('660a8056ed6dd29fedd14a11') }
```



```
  acknowledged: true,
  insertedIds: { '0': ObjectId("660a7c204c45a27fc00ca3c5") }
}
Atlas atlas-jd161z-shard-0 [primary] test>

Atlas atlas-jd161z-shard-0 [primary] test> db.Student.insert({RollNo:10,Age:23,Cont:2276,email:"rekha.de9@gmail.com"});
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("660a7c2b4c45a27fc00ca3c6") }
}
Atlas atlas-jd161z-shard-0 [primary] test> db.Student.find()
[
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c2"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c3"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c4"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c5"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c2b4c45a27fc00ca3c6"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'rekha.de9@gmail.com'
  }
]
Atlas atlas-jd161z-shard-0 [primary] test> |
```



```
  }
]
Atlas atlas-jd161z-shard-0 [primary] test> 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-jd161z-shard-0 [primary] test> db.Student.find()
[
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c2"),
    RollNo: 1,
    Age: 21,
    Cont: 9876,
    email: 'antara.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c3"),
    RollNo: 2,
    Age: 22,
    Cont: 9976,
    email: 'anushka.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c4"),
    RollNo: 3,
    Age: 21,
    Cont: 5576,
    email: 'anubhav.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c204c45a27fc00ca3c5"),
    RollNo: 4,
    Age: 20,
    Cont: 4476,
    email: 'pani.de9@gmail.com'
  },
  {
    _id: ObjectId("660a7c2b4c45a27fc00ca3c6"),
    RollNo: 10,
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  }
]
Atlas atlas-jd161z-shard-0 [primary] test> |
```

II. Perform the following DB operations using MongoDB.

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

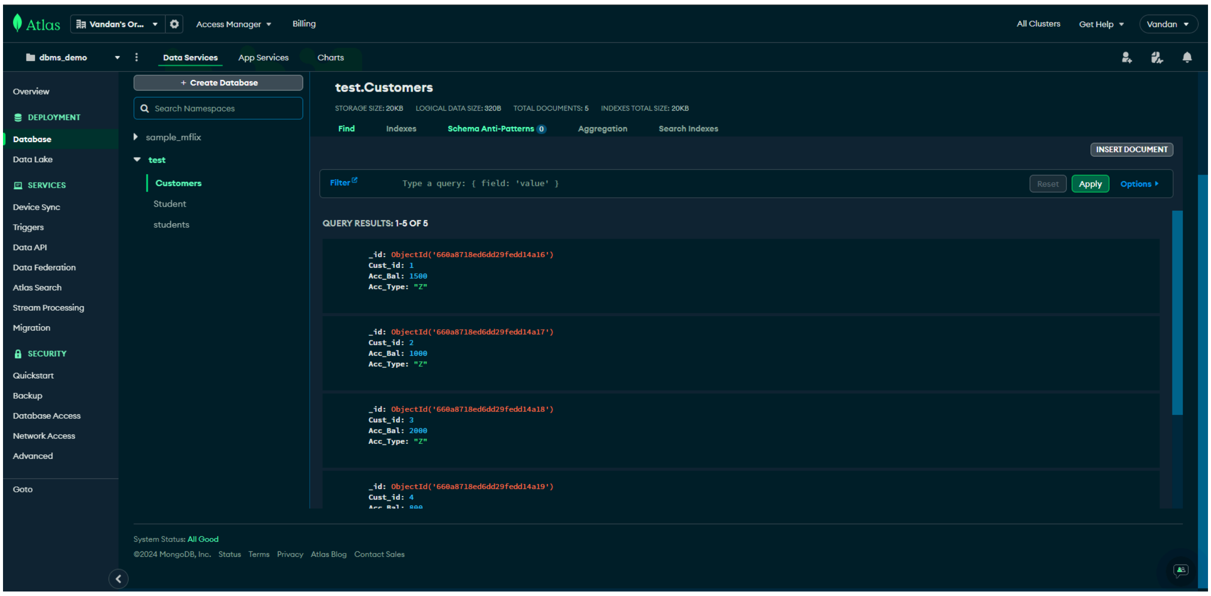
Cust_id, Acc_Bal, Acc_Type

2. Insert at least 5 values into the table

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

1200 of account type ‘Z’ for each customer_id.

4. Determine Minimum and Maximum account balance for each customer_i



test.Customers

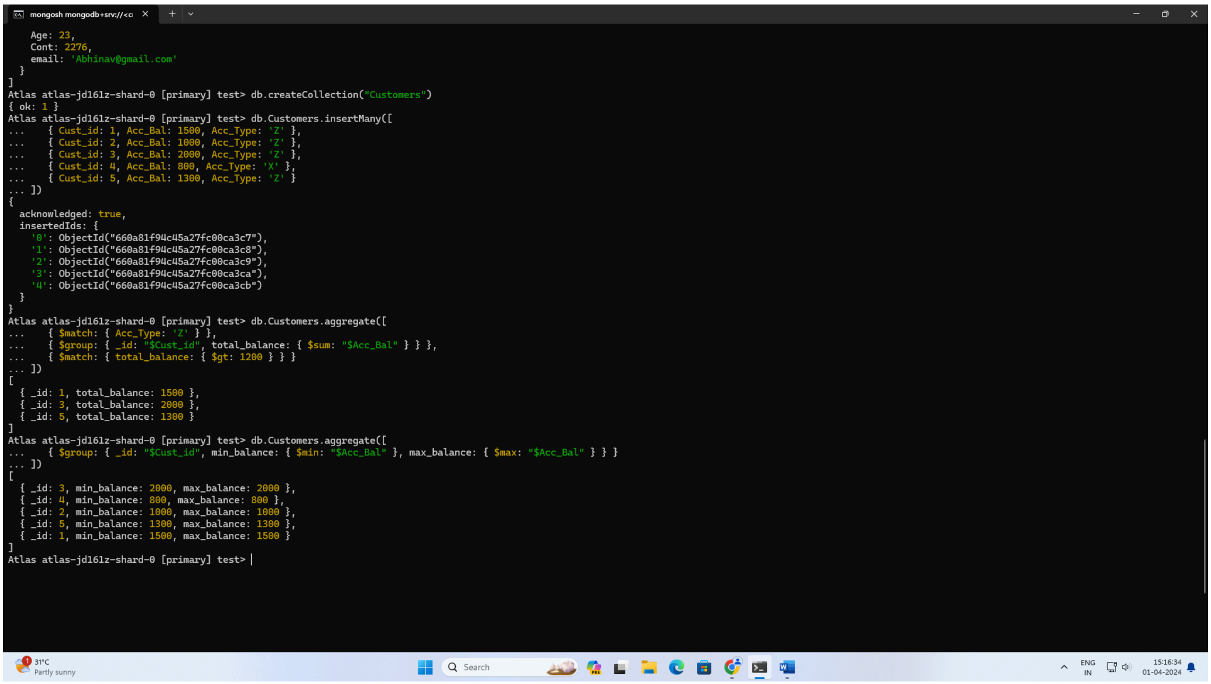
STORAGE SIZE: 20KB   LOGICAL DATA SIZE: 320B   TOTAL DOCUMENTS: 5   INDEXES TOTAL SIZE: 20KB

QUERY RESULTS: 1-5 OF 5

```
_id: ObjectId('660a8718ed6dd29fedd14a16')
Cust_id: 1
Acc_Bal: 1500
Acc_Type: "Z"

_id: ObjectId('660a8718ed6dd29fedd14a17')
Cust_id: 2
Acc_Bal: 1000
Acc_Type: "Z"

_id: ObjectId('660a8718ed6dd29fedd14a18')
Cust_id: 3
Acc_Bal: 2000
Acc_Type: "Z"

_id: ObjectId('660a8718ed6dd29fedd14a19')
Cust_id: 4
Acc_Bal: 800
```

VANDAN N KOTHARI

1BM21CS236



```
    Age: 23,
    Cont: 2276,
    email: 'Abhinav@gmail.com'
  }
]
Atlas atlas-jd161z-shard-0 [primary] test> db.createCollection("Customers")
{ ok: 1 }
Atlas atlas-jd161z-shard-0 [primary] test> db.Customers.insertMany([
...     { Cust_id: 1, Acc_Bal: 1500, Acc_Type: 'Z' },
...     { Cust_id: 2, Acc_Bal: 1000, Acc_Type: 'Z' },
...     { Cust_id: 3, Acc_Bal: 2000, Acc_Type: 'Z' },
...     { Cust_id: 4, Acc_Bal: 800, Acc_Type: 'X' },
...     { Cust_id: 5, Acc_Bal: 1300, Acc_Type: 'Z' }
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("660a81f94c45a27fc00ca3c7"),
    '1': ObjectId("660a81f94c45a27fc00ca3c8"),
    '2': ObjectId("660a81f94c45a27fc00ca3c9"),
    '3': ObjectId("660a81f94c45a27fc00ca3ca"),
    '4': ObjectId("660a81f94c45a27fc00ca3cb")
  }
}
Atlas atlas-jd161z-shard-0 [primary] test> db.Customers.aggregate([
...     { $match: { Acc_Type: 'Z' } },
...     { $group: { _id: "$Cust_id", total_balance: { $sum: "$Acc_Bal" } } },
...     { $match: { total_balance: { $gt: 1200 } } }
... ])
[
  { _id: 1, total_balance: 1500 },
  { _id: 3, total_balance: 2000 },
  { _id: 5, total_balance: 1300 }
]
Atlas atlas-jd161z-shard-0 [primary] test> db.Customers.aggregate([
...     { $group: { _id: "$Cust_id", min_balance: { $min: "$Acc_Bal" }, max_balance: { $max: "$Acc_Bal" } } }
... ])
[
  { _id: 3, min_balance: 2000, max_balance: 2000 },
  { _id: 4, min_balance: 800, max_balance: 800 },
  { _id: 2, min_balance: 1000, max_balance: 1000 },
  { _id: 5, min_balance: 1300, max_balance: 1300 },
  { _id: 1, min_balance: 1500, max_balance: 1500 }
]
Atlas atlas-jd161z-shard-0 [primary] test> |
```