TE IT

Name : Omkar Gurav

Roll No : 8048

Implement aggregation with customer collection using MongoDB. Customer collection consist of following data

> db.customer.find();

{ "_id" : ObjectId("54265694517b30434f6a2bdc"), "custID" : "A123", "Amount" : 500, "status" :

"A" }

{ "_id" : ObjectId("542656eb517b30434f6a2bdd"), "custID" : "A123", "Amount" : 250, "status" :

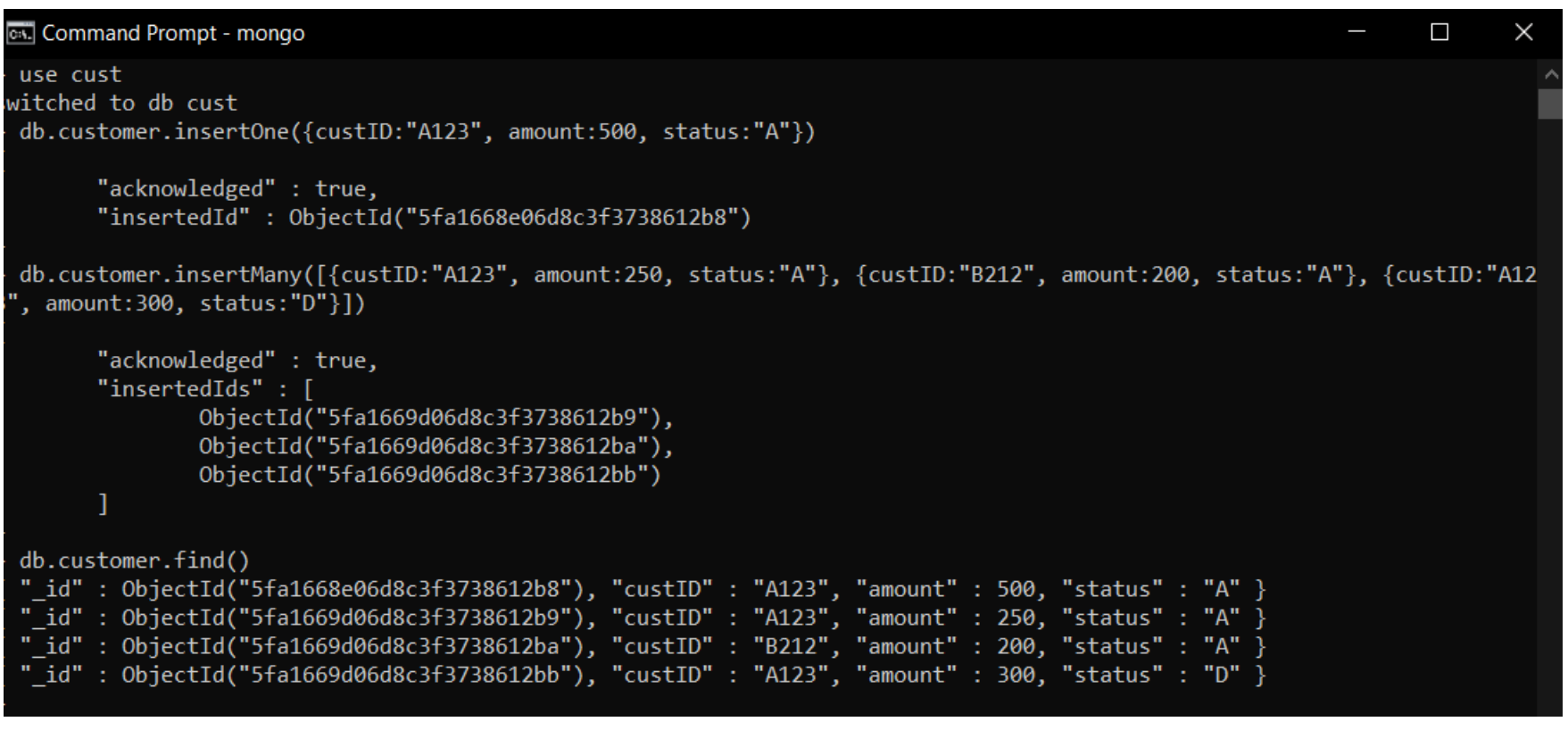
"A" }

{ "_id" : ObjectId("54265726517b30434f6a2bde"), "custID" : "B212", "Amount" : 200, "status" :

"A" }

{ "_id" : ObjectId("54265757517b30434f6a2bdf"), "custID" : "A123", "Amount" : 300, "status" :

"D" }



Execute following queries on customer collection.

PART-A

a) Find the total amount of each customer.

```
> db.customer.aggregate([{$group:{_id:"$custID", Total_amount:{$sum:"$amount"}}}])
{ "_id" : "B212", "Total_amount" : 200 }
{ "_id" : "A123", "Total_amount" : 1050 }
>
```

b) Find the total amount of each customer whose status is A.

```
> db.customer.aggregate({$match:{status:"A"}}, {$group:{_id:"_id", Total_amount:{$sum:"$amount"}}})
{ "_id" : "_id", "Total_amount" : 950 }
```

c) Find the minimum amount of each customer whose Status is A.

```
{ "_id" : "_id", "Total_amount" : 950 }
> db.customer.aggregate({$match:{status:"A"}}, {$group:{_id:"$custID", Min_amount:{$min:"$amount"}}})
{ "_id" : "B212", "Min_amount" : 200 }
{ "_id" : "A123", "Min_amount" : 250 }
>
```

d) Find the maximum amount of each customer whose Status is A.

```
> db.customer.aggregate({$match:{status:"A"}}, {$group:{_id:"$custID", Max_amount:{$max:"$amount"}}})
{ "_id" : "B212", "Max_amount" : 200 }
{ "_id" : "A123", "Max_amount" : 500 }
>
```

e) Find the average total amount of each customer whose Status is A.

```
> db.customer.aggregate({$match:{status:"A"}}, {$group:{_id:"$custID", Avg_amount:{$avg:"$amount"}}})
{ "_id" : "B212", "Avg_amount" : 200 }
{ "_id" : "A123", "Avg_amount" : 375 }
>
```

PART-B

f) Create index on custID.

```
{ "_id" : "A123", "Avg_amount" : 375 }
> db.customer.createIndex({custID:1})
{
        "createdCollectionAutomatically" : false,
        "numIndexesBefore" : 1,
        "numIndexesAfter" : 2,
        "ok" : 1
}
```

g) Execute getIndexes.

```
> db.customer.getIndexes()
[
        {
                "v" : 2,
                "key" : {
                        "_id" : 1
                },
                "name" : "_id_"
        },
        {
                "v" : 2,
                "key" : {
                        "custID" : 1
                },
                "name" : "custID_1"
        }
]
```

h) Drop the index created.

```
> db.customer.dropIndexes()
{
        "nIndexesWas" : 2,
        "msg" : "non-_id indexes dropped for collection",
        "ok" : 1
}
```