BDA LAB

WEEK 2

USN :- 1BM21CS247

I. Perform the following DB operations using MongoDB.

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

Ans. use Student

```
Atlas atlas-u03yqp-shard-0 [primary] test> use Student
switched to db Student
```

2. Insert appropriate values

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.students.insertMany([
...   {Rollno: 10, Age: 20, ContactNo: "1234567890", Email_Id: "student10@example.com"},
...   {Rollno: 11, Age: 22, ContactNo: "9876543210", Email_Id: "student11@example.com"}
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("660a7c497d2bf3fa60a45598"),
    '1': ObjectId("660a7c497d2bf3fa60a45599")
  }
}
```

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

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.students.updateOne({Rollno: 10}, {$set: {Email_Id: "updated_student10@example.com"}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

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

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.students.updateOne({Rollno: 11}, {$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

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.createCollection("Customers")
{ ok: 1 }
```

2. Insert at least 5 values into the table

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.Customers.insertMany([
...    {Cust_id: 1, Acc_Bal: 1000, Acc_Type: 'Z'},
...    {Cust_id: 2, Acc_Bal: 1500, Acc_Type: 'Z'},
...    {Cust_id: 3, Acc_Bal: 800, Acc_Type: 'Z'},
...    {Cust_id: 4, Acc_Bal: 2000, Acc_Type: 'Z'},
...    {Cust_id: 5, Acc_Bal: 1300, Acc_Type: 'Z'}
... ])
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("660a7dd47d2bf3fa60a4559a"),
    '1': ObjectId("660a7dd47d2bf3fa60a4559b"),
    '2': ObjectId("660a7dd47d2bf3fa60a4559c"),
    '3': ObjectId("660a7dd47d2bf3fa60a4559d"),
    '4': ObjectId("660a7dd47d2bf3fa60a4559e")
  }
}
```

3. Write a query to display those records whose total account balance is greater than 1200 of account type ‘Z’ for each customer_id.

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.Customers.aggregate([
...    {$match: {Acc_Type: 'Z'}},
...    {$group: {_id: "$Cust_id", totalBalance: {$sum: "$Acc_Bal"}}},
...    {$match: {totalBalance: {$gt: 1200}}}
... ])
[
  { _id: 4, totalBalance: 2000 },
  { _id: 2, totalBalance: 1500 },
  { _id: 5, totalBalance: 1300 }
]
```

4. Determine Minimum and Maximum account balance for each customer_i

Ans.

```
Atlas atlas-u03yqp-shard-0 [primary] Student> db.Customers.aggregate([
...    {$group: {_id: "$Cust_id", minBalance: {$min: "$Acc_Bal"}, maxBalance: {$max: "$Acc_Bal"}}}
... ])
[
  { _id: 1, minBalance: 1000, maxBalance: 1000 },
  { _id: 4, minBalance: 2000, maxBalance: 2000 },
  { _id: 5, minBalance: 1300, maxBalance: 1300 },
  { _id: 3, minBalance: 800, maxBalance: 800 },
  { _id: 2, minBalance: 1500, maxBalance: 1500 }
]
```