**ASSIGNMENT NO:09**

Name:Mrunal R Somankar

Roll No:TCOA46

1. **Collection “orderinfo“ which contains the documents given as below(Perform on Mongo Terminal) { cust\_id:123 cust\_name:”abc”, status:”A”, price:250 }**

department> use ASS9

switched to db ASS9

ASS9> db.createCollection("orderinfo")

{ ok: 1 }

ASS9> db.orderinfo.insertMany([

... {

... cust\_id:123 ,

... cust\_name:"abc",

... status:"A",

... price:250

... },

... {

... cust\_id:124,

... cust\_name:"xyz",

... status:"A",

... price:950

... },

... {

... cust\_id:125,

... cust\_name:"pqr",

... status:"B",

... price:1000

... },

... {

... cust\_id:126,

... cust\_name:"ijk",

... status:"A",price:1100

... },

... {

... cust\_id:127,

... cust\_name:"xyu",

... status:"A",

... price:870

... }

... ]

... );

{

acknowledged: true,

insertedIds: {

'0': ObjectId("65368490638c33fba3c0d71e"),

'1': ObjectId("65368490638c33fba3c0d71f"),

'2': ObjectId("65368490638c33fba3c0d720"),

'3': ObjectId("65368490638c33fba3c0d721"),

'4': ObjectId("65368490638c33fba3c0d722")

}

}

1. **find the average price for each customers having status 'A**

ASS9> db.orderinfo.aggregate([

... { $match: { status: "A" } },

... { $group: { \_id: "$cust\_name", avg\_price: { $avg: "$price" } } }

... ])

[

{ \_id: 'xyu', avg\_price: 870 },

{ \_id: 'xyz', avg\_price: 950 },

{ \_id: 'ijk', avg\_price: 1100 },

{ \_id: 'abc', avg\_price: 250 }

]

1. **Display the status of the customers whose amount/price lie between 100 and 1000**

ASS9> db.orderinfo.find({ price: { $gte: 100, $lte: 1000 } }, { status: 1 })

[

{ \_id: ObjectId("65368490638c33fba3c0d71e"), status: 'A' },

{ \_id: ObjectId("65368490638c33fba3c0d71f"), status: 'A' },

{ \_id: ObjectId("65368490638c33fba3c0d720"), status: 'B' },

{ \_id: ObjectId("65368490638c33fba3c0d722"), status: 'A' }

]

1. **Display the customers information without “\_id” .**

ASS9> db.orderinfo.find({}, { \_id: 0 })

[

{ cust\_id: 123, cust\_name: 'abc', status: 'A', price: 250 },

{ cust\_id: 124, cust\_name: 'xyz', status: 'A', price: 950 },

{ cust\_id: 125, cust\_name: 'pqr', status: 'B', price: 1000 },

{ cust\_id: 126, cust\_name: 'ijk', status: 'A', price: 1100 },

{ cust\_id: 127, cust\_name: 'xyu', status: 'A', price: 870 }

]

**iv. create a simple index on onderinfo collection and fire the queries.**

ASS9> db.orderinfo.createIndex({ cust\_name: 1 })

cust\_name\_1

ASS9> db.orderinfo.find({ status: "A" }, { avg\_price: { $avg: "$price" } })

[

{ \_id: ObjectId("65368490638c33fba3c0d71e"), avg\_price: 250 },

{ \_id: ObjectId("65368490638c33fba3c0d71f"), avg\_price: 950 },

{ \_id: ObjectId("65368490638c33fba3c0d721"), avg\_price: 1100 },

{ \_id: ObjectId("65368490638c33fba3c0d722"), avg\_price: 870 }

]

**2.Collection “movies“ which contains the documents given as below(Perform on Mongo Terminal)**

**{ name: “Movie1”, type: “action”, budget:1000000 producer:**

**{ name: “producer1”, address:”PUNE” }**

**}**

test> db.createCollection("movies")

{ ok: 1 }

test> db.movies.insertOne(

... {

... name: "Movie1",

... type: "action",

... budget: 1000000,

... producer: {

... name: "producer1",

... address: "PUNE"

... }

... })

{

acknowledged: true,

insertedId: ObjectId("65368d5ce5f6201783b2804c")

}

test> db.movies.insertOne(

... {

... name: "Movie2",

... type: "comedy",

... budget: 800000,

... producer: {

... name: "producer2",

... address: "Los Angeles"

... }

... })

{

acknowledged: true,

insertedId: ObjectId("65368d69e5f6201783b2804d")

}

test> db.movies.insertOne(

... {

... name: "Documentary1",

... type: "documentary",

... budget: 50000,

... producer: {

... name: "producer3",

... address: "New York"

... }

... })

{

acknowledged: true,

insertedId: ObjectId("65368d7ce5f6201783b2804e")

}

test> db.movies.insertOne({

... name: "Movie4",

... type: "foreign",

... budget: 1500000,

... producer: {

... name: "producer4",

... address: "Paris"

... }

... })

{

acknowledged: true,

insertedId: ObjectId("65368d8ce5f6201783b2804f")

}

test> db.movies.insertOne({

... name: "SciFiMovie1",

... type: "science fiction",

... budget: 2000000,

... producer: {

... name: "producer5",

... address: "London"

... }

... })

{

acknowledged: true,

insertedId: ObjectId("65368d9ae5f6201783b28050")

}

1. **Find the name of the movie having budget greater than 1,00,000.**

test> db.movies.find({ budget: { $gt: 100000 } }, { name: 1, \_id: 0 })

[

{ name: 'Movie1' },

{ name: 'Movie2' },

{ name: 'Movie4' },

{ name: 'SciFiMovie1' }

]

1. **Find the name of producer who lives in Pune**

test> db.movies.find({ "producer.address": "PUNE" }, { "producer.name": 1, \_id: 0 })

[ { producer: { name: 'producer1' } } ]

1. **Update the type of movie “action” to “horror”**

test> db.movies.updateMany(

... { type: "action" },

... { $set: { type: "horror" } }

... )

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

1. **Find all the documents produced by name “producer1” with their address**

test> db.movies.find(

... { "producer.name": "producer1" },

... { "producer.name": 1, "producer.address": 1, \_id: 0 }

... )

[ { producer: { name: 'producer1', address: 'PUNE' } } ]