

LAB - 02

Perform following DB operation using MongoDB

* Cust-id, Acc-Bal, Acc-Type

db.Customer.insertMany ({

{ Cust-id: 1, Acc-Bal: 1500, Acc-Type: '2' },

{ Cust-id: 2, Acc-Bal: 900, Acc-Type: '2' },

{ Cust-id: 3, Acc-Bal: 9000, Acc-Type: '1' },

{ Cust-id: 4, Acc-Bal: 1800, Acc-Type: '2' },

{ Cust-id: 5, Acc-Bal: 1200, Acc-Type: '2' }

});

{

acknowledged: true

3) db.Customer.find ({ ACC-Bal: { \$gt: 1000, Acc-Type: '2' }});

{

-id: ObjectId('67cfffb45f50ec56b108'),

Cust-id: 1,

ACC-Bal: 1500,

Acc-Type: '2'

},

{

-id: ObjectId('67cfffb45f50ec56b108'),

Cust-id: 4,

ACC-Bal: 1800,

Acc-Type: '2'

},

4) db.Customer.aggregate ({

\$group: {

-id: "\$Cust-id",

min_balance: { \$min: "\$ACC-Bal" },

```
max_balance: { $max: "Acc-Bal" }  
}  
}  
};  
[ { _id: 4, min_balance: 1800, max_balance: 1800 },  
{ _id: 3, min_balance: 2000, max_balance: 2000 },  
{ _id: 1, min_balance: 1800, max_balance: 1800 },  
{ _id: 2, min_balance: 900, max_balance: 900 },  
{ _id: 5, min_balance: 1200, max_balance: 1200 } ]
```

Product

- use e-commerce
- switched to db ecommerce

```
db.createCollection("Products")
```

```
db.products.insertMany([
```

```
  {  
    product_id: "P001",  
    name: "Laptop",  
    category: "Electronics",  
    price: 999.99,  
    quantity: 50,  
    description: "High-end gaming laptop",  
  },  
  {  
    product_id: "P002",  
    name: "Headphones",  
    category: "Electronics",  
    price: 199.99,  
    quantity: 100,  
    description: "Noise-cancelling Headphones"  
  }])
```

```
])
```

{ acknowledged : true

(User
db.createCollection("user"))

db.user.insertOne({

username : "john_doe",

password : "hashedpassword123",

email : "john.doe@example.com",

phone_no : "103-656-7890",

Shipping_address : {

street : "123 Main St",

city : "Somewhere",

state : "(A",

postal_code : "90001",

country : "USA"

},

created_at : new Date(),

updated_at : new Date()

})

{

acknowledged : true

Carts

db.createCollection("cart")

db.cart.insertOne({

user_id : ObjectID("user_id_new"),

products : [

product_id : ObjectID("product_id_new"),
quantity : 5,

price_at_time : 999.99

},

Total price: 1299.98,

Created at: new Date(),

update_at: new Date()

})

Orders

a. create Collection("orders")

ab. orders. newOne ({

user_id: ObjectId("user_id here"),

order_status: "Pending",

shipping_address: {

street: "123 Main St",

City: "Somewhere",

state: "CA",

Postal code: "90001",

Country: "USA",

products: ["P1", {

product_id: ObjectId("Product_id here"),

quantity: 1,

price_at_time: 999.99

},],

total price: 999.99,

created_at: new Date(),

update_at: new Date()

})

a) ab.products.find({}):

{}
id: ObjectId("67 ---"),

product_id: "P001",

Name: "Laptop",

Category: "Electronics",

Price: 999.99,

quantity: 50,
description: 'High-end gaming laptop',
mag: [], mag1.jpg, .mag2.jpg

{
 id: ObjectID('67 - - -'),
 Product_id: 'P003'

 name: "Headphones",
 category: 'Electronics',

 price: 199.99,

 quantity: 100,

}] description: 'Noise-cancelling headphones'

b) db.product.find({ category: "Electronics" })

{
 id: ObjectID('67 - - -'),
 Product_id: 'P001'

 name: 'Laptop',

 category: 'Electronics',

 price: 999.99,

 quantity: 50,

 description: 'High-end gaming laptop'

{
 id: ObjectID('67 - - -'),
 Product_id: 'P003'

 name: "Headphones",

 category: 'Electronics',

 price: 199.99,

 quantity: 100,

 description: 'Noise-cancelling headphones'

db.products.find({ "quarterly": { \$gt: 0 } })

{
 - id: object_id('67--'),
 product_id: 'P001',
 name: 'Laptop',
 category: 'Electronics',
 price: 999.99,
 quantity: 50,
 description: 'High-end gaming laptop'
}

{
 - id: object_id("67-"),
 product_id: "P002",
 name: "Headphones",
 category: 'Electronics',
 price: 199.99,
 quantity: 100,
 description: 'Noise-cancelling headphones'
}

}

db.products.find({}).sort({ \$price": 1 })

{
 - id: object_id('67--'),
 product_id: "P002",
 name: "Headphones",
 category: 'Electronics',
 price: 199.99,
 quantity: 100
}

~~{~~ ~~- id: object_id('67-')~~

~~product_id: "P001",~~

~~name: 'Laptop',~~

~~category: 'Electronics',~~

~~price: 999.99,~~

~~quantity: 100 }~~

e) db.products.find({ "Price": { \$gt: 189.99 } })

db.products.aggregate([
{\$match: { "id": ObjectId("5e83a5f123456789") }},
{\$unwind: "\$cart"},
{\$lookup: {
from: "products",
localField: "cart.product_id",
foreignField: "id",
as: "product_details" }},
{\$project: { "product_details": 1, "cart.quantity": 1 } }])

g) db.orders.aggregate([
{\$match: { "id": ObjectId("5e83a5f123456789") }},
{\$unwind: "\$order"},
{\$lookup: {
from: "orders",
localField: "order.order_id",
foreignField: "id",
as: "order_details" }},
{\$project: { "order_details": 1 } }])

h) db.orders.aggregate([
{\$match: { "id": ObjectId("5e83a5f123456789") }},
{\$unwind: "\$order"},
{\$group: {
id: "\$id",
total_order_price: { \$sum: "\$order.Price" } } }])

Aggregation queries

1) db.products.aggregate([

{\$group: { _id: "\$category", total_product:

{ \$sum: 1 } }}

])

2) db.products.aggregate([

{\$group: { _id: "\$category", total_price:

{ \$sum: { \$multiply: ["\$price", "\$quantity"] } }}

])

3) db.products.aggregate([

{\$group: { _id: null, avg_price: { \$avg:

"\$price" } } }])

4) db.products.find({ "quantity": { \$lt: 10 } })

5) db.products.find({ }).sort({ "price": -1 })

6) db.orders.aggregate([

{\$unwind: "\$product"},

{\$group: {

_id: "\$super_id",

total_order_price: { \$sum: { \$multiply:

{ \$product.price }, "\$product.quantity" } }

}}

])

7) db.orders.aggregate([

{\$unwind: "\$product"},

{\$group: {

_id: "\$super_id",

total_order_price: ("sum", "Total order price",
"I product price", "I product quantity")
})

{ \$ord: { "total_order_price": -1 } }
(\$sum: 1)
})

8) db.orders.aggregate([
{ \$group: { _id: null, avg_order_price:
{ \$avg: "I total_price" } } }
])

80
21/3/25