**SOFTWARE LABORATORY 1**

**GROUP A – EXPERIMENT 4**

**TITLE:**

MongoDB – Aggregation and Indexing: Design and Develop MongoDB Queries using aggregation and indexing with suitable example using MongoDB.

MongoDB – Map-reduce operations: Implement Map-reduce operation with suitable example using MongoDB.

**CODE:**

# Aggregation

(base) student@student-HP-Pro-Tower-280-G9-PCI-Desktop-PC:~$ mongosh

test> use unzo

switched to db unzo

unzo> db.fruits.insertMany([{id:1, name:"Mango", quantity:20, price:50, status:"shipped" }, {id:2, name:"Apple", quantity:25, price:35, status:"pending"}, {id:3, name:"Lychee", quantity:55, price:45, status:"shipped"}, {id:4, name:"Blueberry", quantity:100, price:45, status:"pending"}, {id:5, name:"Pomogranate", quantity:50, price:50,status:"shipped"}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6891d21d207a23613bc59f35'),

'1': ObjectId('6891d21d207a23613bc59f36'),

'2': ObjectId('6891d21d207a23613bc59f37'),

'3': ObjectId('6891d21d207a23613bc59f38'),

'4': ObjectId('6891d21d207a23613bc59f39')

}

}

unzo> db.fruits.aggregate([{$match:{status:"shipped"}}])

[

{

\_id: ObjectId('6891d21d207a23613bc59f35'),

id: 1,

name: 'Mango',

quantity: 20,

price: 50,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f37'),

id: 3,

name: 'Lychee',

quantity: 55,

price: 45,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f39'),

id: 5,

name: 'Pomogranate',

quantity: 50,

price: 50,

status: 'shipped'

}

]

unzo> db.fruits.aggregate([{$group:{\_id:"$name", totalQuantity:{$sum:"$quantity"}}}])

[

{ \_id: 'Apple', totalQuantity: 25 },

{ \_id: 'Blueberry', totalQuantity: 100 },

{ \_id: 'Mango', totalQuantity: 20 },

{ \_id: 'Pomogranate', totalQuantity: 50 },

{ \_id: 'Lychee', totalQuantity: 55 }

]

unzo> db.fruits.aggregate([{$sort:{quantity:-1}}])

[

{

\_id: ObjectId('6891d21d207a23613bc59f38'),

id: 4,

name: 'Blueberry',

quantity: 100,

price: 45,

status: 'pending'

},

{

\_id: ObjectId('6891d21d207a23613bc59f37'),

id: 3,

name: 'Lychee',

quantity: 55,

price: 45,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f39'),

id: 5,

name: 'Pomogranate',

quantity: 50,

price: 50,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f36'),

id: 2,

name: 'Apple',

quantity: 25,

price: 35,

status: 'pending'

},

{

\_id: ObjectId('6891d21d207a23613bc59f35'),

id: 1,

name: 'Mango',

quantity: 20,

price: 50,

status: 'shipped'

}

]

unzo> db.fruits.aggregate([{$sort:{quantity:1}}])

[

{

\_id: ObjectId('6891d21d207a23613bc59f35'),

id: 1,

name: 'Mango',

quantity: 20,

price: 50,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f36'),

id: 2,

name: 'Apple',

quantity: 25,

price: 35,

status: 'pending'

},

{

\_id: ObjectId('6891d21d207a23613bc59f39'),

id: 5,

name: 'Pomogranate',

quantity: 50,

price: 50,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f37'),

id: 3,

name: 'Lychee',

quantity: 55,

price: 45,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f38'),

id: 4,

name: 'Blueberry',

quantity: 100,

price: 45,

status: 'pending'

}

]

unzo> db.fruits.aggregate([{$limit:2}])

[

{

\_id: ObjectId('6891d21d207a23613bc59f35'),

id: 1,

name: 'Mango',

quantity: 20,

price: 50,

status: 'shipped'

},

{

\_id: ObjectId('6891d21d207a23613bc59f36'),

id: 2,

name: 'Apple',

quantity: 25,

price: 35,

status: 'pending'

}

]

unzo> db.fruits.createIndex({ price: 1 })

price\_1

# Map-reduce operations

// Select or create the database

use mydb;

// Insert sample data into the 'orders' collection

db.orders.insertMany([

{ "\_id": 1, "product": "apple", "quantity": 5 },

{ "\_id": 2, "product": "banana", "quantity": 3 },

{ "\_id": 3, "product": "apple", "quantity": 8 },

{ "\_id": 4, "product": "banana", "quantity": 2 },

{ "\_id": 5, "product": "orange", "quantity": 7 }

]);

{

"acknowledged": true,

"insertedIds": {

"0": 1,

"1": 2,

"2": 3,

"3": 4,

"4": 5

}

}

// Define the map function

var mapFunction = function() {

emit(this.product, this.quantity);

};

// Define the reduce function

var reduceFunction = function(keyProduct, valuesQuantities) {

return Array.sum(valuesQuantities);

};

// Run the Map-Reduce operation

db.orders.mapReduce(

mapFunction,

reduceFunction,

{

out: "product\_totals"

}

);

// Display the results

print("Map-Reduce Results:");

db.product\_totals.find().forEach(printjson);

{ "\_id": "apple", "value": 13 }

{ "\_id": "orange", "value": 7 }

{ "\_id": "banana", "value": 5 }

// Display the count of unique products

var count = db.product\_totals.count();

print("Total unique products:", count);

Total unique products: 3