**DBMS Lab**

**09/10/2020** **Assignment No. 13 (Inventory Collection) Sanjay Rawat(7341)**

**Execute following queries in Mongo DB:**

1. Consider a collection “Inventory” with the following documents:

> show dbs

admin 0.000GB

bank 0.000GB

config 0.000GB

local 0.000GB

orders 0.000GB

> use Inventory

switched to db Inventory

> db.Inventory.insert({"\_id":1,item:"f1",type:"food",quantity:500})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":2,item:"f2",type:"food",quantity:100})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":3,item:"p1",type:"paper",quantity:200})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":4,item:"p2",type:"paper",quantity:150})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":5,item:"f3",type:"snacks",quantity:300})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":6,item:"t1",type:"toys",quantity:500})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":7,item:"a1",type:"apparel",quantity:250})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":8,item:"a2",type:"apparel",quantity:400})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":9,item:"t2",type:"toys",quantity:50})

WriteResult({ "nInserted" : 1 })

> db.Inventory.insert({"\_id":10,item:"f4",type:"snacks",quantity:75})

WriteResult({ "nInserted" : 1 })

> db.Inventory.find().pretty()

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 9, "item" : "t2", "type" : "toys", "quantity" : 50 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

> show dbs

Inventory 0.000GB

admin 0.000GB

bank 0.000GB

config 0.000GB

local 0.000GB

orders 0.000GB

1. Insert a new document into a collection named “Inventory”

> db.Inventory.insert({"\_id":11,item:"f5",type:"food",quantity:120})

WriteResult({ "nInserted" : 1 })

> db.Inventory.find().pretty()

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 9, "item" : "t2", "type" : "toys", "quantity" : 50 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : 11, "item" : "f5", "type" : "food", "quantity" : 120 }

1. Find all documents where the type field has the value snacks

> db.Inventory.find({type:"snacks"}).pretty()

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

1. Insert an Array of Documents

> var mydocuments =

... [

... {

... item: "ABC2",

... details: { model: "14Q3", manufacturer: "M1 Corporation" },

... stock: [ { size: "M", qty: 50 } ],

... category: "clothing"

... },

... {

... item: "MNO2",

... details: { model: "14Q3", manufacturer: "ABC Company" },

... stock: [ { size: "S", qty: 5 }, { size: "M", qty: 5 }, { size: "L", qty: 1 } ],

... category: "clothing"

... }

... ]

> db.Inventory.insert(mydocuments)

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 2,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

> db.Inventory.find().forEach(printjson)

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 9, "item" : "t2", "type" : "toys", "quantity" : 50 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : 11, "item" : "f5", "type" : "food", "quantity" : 120 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "14Q3",

"manufacturer" : "ABC Company"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "clothing"

}

1. Update a document

> db.Inventory.update({"\_id":9},{$set:{"quantity":200,"item":"t3"}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.Inventory.find().forEach(printjson)

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 9, "item" : "t3", "type" : "toys", "quantity" : 200 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : 11, "item" : "f5", "type" : "food", "quantity" : 120 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "14Q3",

"manufacturer" : "ABC Company"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "clothing"

}

1. Update Specific Fields

**For the document with item equal to "MNO2", update the category field and the details field**.

> db.Inventory.update({"item":"MNO2"},{$set:{"category":"mobile accessories","details":{"model":"15Q1","manufacturer":"Nokia"}}})

WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })

> db.Inventory.find().forEach(printjson)

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 9, "item" : "t3", "type" : "toys", "quantity" : 200 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : 11, "item" : "f5", "type" : "food", "quantity" : 120 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Remove a single document where type is toys and qty is 200

> db.Inventory.remove({"type":"toys","quantity":200})

WriteResult({ "nRemoved" : 1 })

> db.Inventory.find().forEach(printjson)

{ "\_id" : 1, "item" : "f1", "type" : "food", "quantity" : 500 }

{ "\_id" : 2, "item" : "f2", "type" : "food", "quantity" : 100 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : 11, "item" : "f5", "type" : "food", "quantity" : 120 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Remove all documents where type is “food”

> db.Inventory.remove({"type":"food"})

WriteResult({ "nRemoved" : 3 })

> db.Inventory.find().forEach(printjson)

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Select all documents in a Collection

> db.Inventory.find().pretty()

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Specify Equality Condition

**Find all documents where the type field has the value snacks**

> db.Inventory.find({type:"snacks"})

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

1. Specify AND Conditions

**All documents where the type field has the value 'apparel' and the value of the quantity field is less than 250**

> db.Inventory.find({$and:[{type:"apparel"},{quantity:{$lte:300}}]});

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

1. Specify OR Conditions

**All documents in the collection where the field qty has a value greater than 100 or the value of the type field is 'paper'**

> db.Inventory.find({$or:[{type:"paper"},{quantity:{$gte:100}}]});

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

1. Specify AND as well as OR Conditions

**Find all documents in the collection where the value of the type field is 'paper' and either the qty has a value greater than 50 or the value of the item field “p1”**:

> db.Inventory.find({$and: [{"type": "paper"},{$or:[{"quantity": {$gte:100}},{"item":"p1"}] } ]});

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

1. Exact Match on the Embedded Document

**Find all documents where the value of the field stock is an embedded document that contains only the field size with the value 'M’ and the field qty with the value '50’, in the exact order**

> db.Inventory.find({"stock":{"size":"M","qty":50}}).pretty();

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

1. Exact Match on an Array

**Find all documents where the field stock is an array that holds exactly three elements, S, M, and L**

> db.Inventory.find({stock:{$size:3}}).pretty()

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Match an Array Element

**Find all documents where stock is an array that contains S as one of its elements**

> db.Inventory.find({"stock":{$elemMatch:{size:"S"}}}).pretty()

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Match a Specific Element of an Array

**Find all documents where the stock array contains S as the first element**

> db.Inventory.find({"stock.0":{"size":"S","qty":5}}).pretty()

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}

1. Return Specified Fields Only

**Return toys only with item and qty fields from the documents.**

> db.Inventory.find({type:"toys"},{"\_id":0,type:0});

{ "item" : "t1", "quantity" : 500 }

1. Sort documents using one field

**Sort documents on qty in descending order**

> db.Inventory.find().sort({quantity:-1});

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{ "\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"), "item" : "ABC2", "details" : { "model" : "14Q3", "manufacturer" : "M1 Corporation" }, "stock" : [ { "size" : "M", "qty" : 50 } ], "category" : "clothing" }

{ "\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"), "item" : "MNO2", "details" : { "model" : "15Q1", "manufacturer" : "Nokia" }, "stock" : [ { "size" : "S", "qty" : 5 }, { "size" : "M", "qty" : 5 }, { "size" : "L", "qty" : 1 } ], "category" : "mobile accessories" }

1. Create Index on some field

**Create Index on item field**

> db.Inventory.createIndex({"\_id":5});

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_"

},

{

"v" : 2,

"key" : {

"\_id" : 5

},

"name" : "\_id\_5"

}

]

> db.Inventory.find().pretty();

{ "\_id" : 3, "item" : "p1", "type" : "paper", "quantity" : 200 }

{ "\_id" : 4, "item" : "p2", "type" : "paper", "quantity" : 150 }

{ "\_id" : 5, "item" : "f3", "type" : "snacks", "quantity" : 300 }

{ "\_id" : 6, "item" : "t1", "type" : "toys", "quantity" : 500 }

{ "\_id" : 7, "item" : "a1", "type" : "apparel", "quantity" : 250 }

{ "\_id" : 8, "item" : "a2", "type" : "apparel", "quantity" : 400 }

{ "\_id" : 10, "item" : "f4", "type" : "snacks", "quantity" : 75 }

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f0"),

"item" : "ABC2",

"details" : {

"model" : "14Q3",

"manufacturer" : "M1 Corporation"

},

"stock" : [

{

"size" : "M",

"qty" : 50

}

],

"category" : "clothing"

}

{

"\_id" : ObjectId("5f7fe0a06ebf7cb125d055f1"),

"item" : "MNO2",

"details" : {

"model" : "15Q1",

"manufacturer" : "Nokia"

},

"stock" : [

{

"size" : "S",

"qty" : 5

},

{

"size" : "M",

"qty" : 5

},

{

"size" : "L",

"qty" : 1

}

],

"category" : "mobile accessories"

}