ASSIGNMENT No:5C

Q. Implement the aggregation and indexing for following in MongoDB.

Product(id, item\_nm, quantity, price)

> db.product.insert([

{id:1,item\_nm:'Shirt',quantity:2,price:1000},

{id:2,item\_nm:'Shorts',quantity:5,price:1500}, {id:3,item\_nm:'Shoes',quantity:1,price:4500},

{id:4,item\_nm:'Jacket',quantity:1,price:2000},

{id:5,item\_nm:'BAT',quantity:2,price:750},

{id:6,item\_nm:'Socks',quantity:2,price:69},

{id:7,item\_nm:'Tie',quantity:4,price:400}

])

BulkWriteResult({

"writeErrors" : [ ],

"writeConcernErrors" : [ ],

"nInserted" : 7,

"nUpserted" : 0,

"nMatched" : 0,

"nModified" : 0,

"nRemoved" : 0,

"upserted" : [ ]

})

> db.product.find({},{\_id:0}).pretty()

{

"id" : 1,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 2,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 3,

"item\_nm" : "Shoes",

"quantity" : 1,

"price" : 4500

}

{

"id" : 4,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 2000

}

{

"id" : 5,

"item\_nm" : "BAT",

"quantity" : 2,

"price" : 750

}

{

"id" : 6,

"item\_nm" : "Socks",

"quantity" : 2,

"price" : 69

}

{

"id" : 7,

"item\_nm" : "Tie",

"quantity" : 4,

"price" : 400

}

-----------------------------------------------------------------

a. Demonstrate the following aggregation operation (Sum, Average, min,max ) on suitable key.

db.product.insert([

{id:8,item\_nm:'Jacket',quantity:1,price:1500}

])

> db.product.aggregate([

{$group:{\_id:"$item\_nm",totalAmount:{$sum:{$multiply:["$price","$quantity"]}} , count:{$sum:1}}}])

{ "\_id " : "Shirt", "totalAmount" : 2000 , "count" : 1}

{ "\_id " : "Shorts", "totalAmount" : 7500 , "count" : 1}

{ "\_id " : "Shoes", "totalAmount" : 4500 , "count" : 1}

{ "\_id " : "Jacket", "totalAmount" : 3500 , "count" : 2}

{ "\_id " : "BAT", "totalAmount" : 1500 , "count" : 1}

{ "\_id " : "Socks", "totalAmount" : 138 , "count" : 1}

{ "\_id " : "Tie", "totalAmount" : 1600 , "count" : 1}

> db.product.aggregate([{

$group:{\_id:"$item\_nm",avgAmount:{$avg:{$multiply:["$price","$quantity"]}},avgQuantity:{$avg:"$quantity"}}}])

{ "\_id" : "Shirt", "avgAmount" : 2000, "avgQuantity" : 2 }

{ "\_id" : "Shorts", "avgAmount" : 7500, "avgQuantity" : 5 }

{ "\_id" : "Shoes", "avgAmount" : 4500, "avgQuantity" : 1 }

{ "\_id" : "Jacket", "avgAmount" : 1750, "avgQuantity" : 1 }

{ "\_id" : "BAT", "avgAmount" : 1500, "avgQuantity" : 2 }

{ "\_id" : "Socks", "avgAmount" : 138, "avgQuantity" : 2 }

{ "\_id" : "Tie", "avgAmount" : 1600, "avgQuantity" : 4 }

> db.product.aggregate([

{$group:{\_id:"$item\_nm",minAmount:{$min:"$price"}}}])

{ "\_id" : "Shirt", "minAmount" : 1000 }

{ "\_id" : "Shorts", "minAmount" : 1500 }

{ "\_id" : "Shoes", "minAmount" : 4500 }

{ "\_id" : "Jacket", "minAmount" : 1500 }

{ "\_id" : "BAT", "minAmount" : 750 }

{ "\_id" : "Socks", "minAmount" : 69 }

{ "\_id" : "Tie", "minAmount" : 400 }

> db.product.aggregate([

{$group:{\_id:"$item\_nm",maxAmount:{$max:"$price"}}}])

{ "\_id" : "Shirt", "maxAmount" : 1000 }

{ "\_id" : "Shorts", "maxAmount" : 1500 }

{ "\_id" : "Shoes", "maxAmount" : 4500 }

{ "\_id" : "Jacket", "maxAmount" : 2000 }

{ "\_id" : "BAT", "maxAmount" : 750 }

{ "\_id" : "Socks", "maxAmount" : 69 }

{ "\_id" : "Tie", "maxAmount" : 400 }

-----------------------------------------------------------------

b. Create and drop different types of indexes and explain to show the advantage of the indexes.

> db.product.getIndexes();

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "sase.product"

}

]

> db.product.find().sort({price:1}).explain();

{

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "sase.product",

"indexFilterSet" : false,

"parsedQuery" : {

},

"winningPlan" : {

"stage" : "SORT",

"sortPattern" : {

"price" : 1

},

"inputStage" : {

"stage" : "SORT\_KEY\_GENERATOR",

"inputStage" : {

"stage" : "COLLSCAN",

"direction" : "forward"

}

}

},

"rejectedPlans" : [ ]

},

"serverInfo" : {

"host" : "HP-Pavilion-7757",

"port" : 27017,

"version" : "3.4.17",

"gitVersion" : "7c14a47868643bb691a507a92fe25541f998eca4"

},

"ok" : 1

}

> db.product.createIndex({price:1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 1,

"numIndexesAfter" : 2,

"ok" : 1

}

> db.product.find().sort({price:1}).explain()

{

"queryPlanner" : {

"plannerVersion" : 1,

"namespace" : "sase.product",

"indexFilterSet" : false,

"parsedQuery" : {

},

"winningPlan" : {

"stage" : "FETCH",

"inputStage" : {

"stage" : "IXSCAN",

"keyPattern" : {

"price" : 1

},

"indexName" : "price\_1",

"isMultiKey" : false,

"multiKeyPaths" : {

"price" : [ ]

},

"isUnique" : false,

"isSparse" : false,

"isPartial" : false,

"indexVersion" : 2,

"direction" : "forward",

"indexBounds" : {

"price" : [

"[MinKey, MaxKey]"

]

}

}

},

"rejectedPlans" : [ ]

},

"serverInfo" : {

"host" : " HP-Pavilion-7757",

"port" : 27017,

"version" : "3.4.17",

"gitVersion" : "7c14a47868643bb691a507a92fe25541f998eca4"

},

"ok" : 1

}

> db.product.find({},{\_id:0}).sort({price:1}).pretty()

{

"id" : 6,

"item\_nm" : "Socks",

"quantity" : 2,

"price" : 69

}

{

"id" : 7,

"item\_nm" : "Tie",

"quantity" : 4,

"price" : 400

}

{

"id" : 5,

"item\_nm" : "BAT",

"quantity" : 2,

"price" : 750

}

{

"id" : 1,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 2,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 8,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 1500

}

{

"id" : 4,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 2000

}

{

"id" : 3,

"item\_nm" : "Shoes",

"quantity" : 1,

"price" : 4500

}

> db.product.find({},{\_id:0}).sort({price:-1}).pretty()

{

"id" : 3,

"item\_nm" : "Shoes",

"quantity" : 1,

"price" : 4500

}

{

"id" : 4,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 2000

}

{

"id" : 8,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 1500

}

{

"id" : 1,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 2,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 5,

"item\_nm" : "BAT",

"quantity" : 2,

"price" : 750

}

{

"id" : 7,

"item\_nm" : "Tie",

"quantity" : 4,

"price" : 400

}

{

"id" : 6,

"item\_nm" : "Socks",

"quantity" : 2,

"price" : 69

}

> db.product.createIndex({price:1,id:-1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 2,

"numIndexesAfter" : 3,

"ok" : 1

}

> db.product.find({},{\_id:0}).sort({price:-1,id:-1}).pretty()

{

"id" : 3,

"item\_nm" : "Shoes",

"quantity" : 1,

"price" : 4500

}

{

"id" : 4,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 2000

}

{

"id" : 8,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 1500

}

{

"id" : 2,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 1,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 5,

"item\_nm" : "BAT",

"quantity" : 2,

"price" : 750

}

{

"id" : 7,

"item\_nm" : "Tie",

"quantity" : 4,

"price" : 400

}

{

"id" : 6,

"item\_nm" : "Socks",

"quantity" : 2,

"price" : 69

}

> db.product.createIndex({id:-1,price:-1})

{

"createdCollectionAutomatically" : false,

"numIndexesBefore" : 3,

"numIndexesAfter" : 4,

"ok" : 1

}

> db.product.find({},{\_id:0}).sort({id:-1,price:-1}).pretty()

{

"id" : 7,

"item\_nm" : "Tie",

"quantity" : 4,

"price" : 400

}

{

"id" : 6,

"item\_nm" : "Socks",

"quantity" : 2,

"price" : 69

}

{

"id" : 5,

"item\_nm" : "BAT",

"quantity" : 2,

"price" : 750

}

{

"id" : 4,

"item\_nm" : "Jacket",

"quantity" : 1,

"price" : 2000

}

{

"id" : 3,

"item\_nm" : "Shoes",

"quantity" : 1,

"price" : 4500

}

{

"id" : 2,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

{

"id" : 1,

"item\_nm" : "Shirt",

"quantity" : 2,

"price" : 1000

}

> db.product.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "sase.product"

},

{

"v" : 2,

"key" : {

"price" : 1

},

"name" : "price\_1",

"ns" : "sase.product"

},

{

"v" : 2,

"key" : {

"price" : 1,

"id" : -1

},

"name" : "price\_1\_id\_-1",

"ns" : "sase.product"

},

{

"v" : 2,

"key" : {

"id" : -1,

"price" : -1

},

"name" : "id\_-1\_price\_-1",

"ns" : "sase.product"

}

]

> db.product.dropIndex("id\_-1\_price\_-1")

{ "nIndexesWas" : 4, "ok" : 1 }

> db.product.getIndexes()

[

{

"v" : 2,

"key" : {

"\_id" : 1

},

"name" : "\_id\_",

"ns" : "sase.product"

},

{

"v" : 2,

"key" : {

"price" : 1

},

"name" : "price\_1",

"ns" : "sase.product"

},

{

"v" : 2,

"key" : {

"price" : 1,

"id" : -1

},

"name" : "price\_1\_id\_-1",

"ns" : "sase.product"

}

]

> db.product.dropIndex({'price':1})

{ "nIndexesWas" : 3, "ok" : 1 }

-----------------------------------------------------------------