DAY 4 – 24th May – 113195064 – Bondili Bhavya Charitha

**Navigating to Database: -**

ReferenceError: db1 is not defined

use db1

switched to db db1

use db

switched to db db

use db1

switched to db db1

**Creating Collection: -**

db.createCollection("students")

{ ok: 1 }

show collections

students

**Inserting Documents: -**

db.students.insertMany([

{

name:"roger",

subject:"maths"

},

{

name:"anita",

subject:"chemistry"

},

{

name:"thomas",

subject:"physics"

},

{

name:"tina",

subject:"maths"

},

{

name:"jonty",

subject:"english"

}

]

)

{

acknowledged: true,

insertedIds: {

'0': ObjectId('683172a764be9171e9a97d54'),

'1': ObjectId('683172a764be9171e9a97d55'),

'2': ObjectId('683172a764be9171e9a97d56'),

'3': ObjectId('683172a764be9171e9a97d57'),

'4': ObjectId('683172a764be9171e9a97d58')

}

}

**Querying the collection: -**

db.students.find()

{

\_id: ObjectId('683172a764be9171e9a97d54'),

name: 'roger',

subject: 'maths'

}

{

\_id: ObjectId('683172a764be9171e9a97d55'),

name: 'anita',

subject: 'chemistry'

}

{

\_id: ObjectId('683172a764be9171e9a97d56'),

name: 'thomas',

subject: 'physics'

}

{

\_id: ObjectId('683172a764be9171e9a97d57'),

name: 'tina',

subject: 'maths'

}

{

\_id: ObjectId('683172a764be9171e9a97d58'),

name: 'jonty',

subject: 'english'

}

**Querying Collection with Filter: -**

db.students.find({name:"anita"})

{

\_id: ObjectId('683172a764be9171e9a97d55'),

name: 'anita',

subject: 'chemistry'

}

show dbs

admin 40.00 KiB

config 48.00 KiB

db1 40.00 KiB

local 40.00 KiB

db.students.find()

{

\_id: ObjectId('683172a764be9171e9a97d54'),

name: 'roger',

subject: 'maths'

}

{

\_id: ObjectId('683172a764be9171e9a97d55'),

name: 'anita',

subject: 'chemistry'

}

{

\_id: ObjectId('683172a764be9171e9a97d56'),

name: 'thomas',

subject: 'physics'

}

{

\_id: ObjectId('683172a764be9171e9a97d57'),

name: 'tina',

subject: 'maths'

}

{

\_id: ObjectId('683172a764be9171e9a97d58'),

name: 'jonty',

subject: 'english'

}

**Deleting a specific Document from Collection: -**

db1.students.deleteOne({'name':'jonty'})

ReferenceError: db1 is not defined

db.students.deleteOne({'name':'jonty'})

.............................

db.students.find()

{

\_id: ObjectId('683172a764be9171e9a97d54'),

name: 'roger',

subject: 'maths'

}

{

\_id: ObjectId('683172a764be9171e9a97d55'),

name: 'anita',

subject: 'chemistry'

}

{

\_id: ObjectId('683172a764be9171e9a97d56'),

name: 'thomas',

subject: 'physics'

}

{

\_id: ObjectId('683172a764be9171e9a97d57'),

name: 'tina',

subject: 'maths'

}

db.students.drop

[Function: drop] AsyncFunction {

apiVersions: [ 1, Infinity ],

returnsPromise: true,

serverVersions: [ '0.0.0', '999.999.999' ],

topologies: [ 'ReplSet', 'Sharded', 'LoadBalanced', 'Standalone' ],

returnType: { type: 'unknown', attributes: {} },

deprecated: false,

platforms: [ 'Compass', 'Browser', 'CLI' ],

isDirectShellCommand: false,

acceptsRawInput: false,

shellCommandCompleter: undefined,

help: [Function (anonymous)] Help

}

show collections

students

db.students.drop()

true

show collections

db.createCollection("students")

{ ok: 1 }

db.students.insertMany([

{

name:"roger",

subject:"maths"

},

{

name:"anita",

subject:"chemistry"

},

{

name:"thomas",

subject:"physics"

},

{

name:"tina",

subject:"maths"

},

{

name:"jonty",

subject:"english"

}

]

)

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6831921364be9171e9a97d59'),

'1': ObjectId('6831921364be9171e9a97d5a'),

'2': ObjectId('6831921364be9171e9a97d5b'),

'3': ObjectId('6831921364be9171e9a97d5c'),

'4': ObjectId('6831921364be9171e9a97d5d')

}

}

db.students.find()

{

\_id: ObjectId('6831921364be9171e9a97d59'),

name: 'roger',

subject: 'maths'

}

{

\_id: ObjectId('6831921364be9171e9a97d5a'),

name: 'anita',

subject: 'chemistry'

}

{

\_id: ObjectId('6831921364be9171e9a97d5b'),

name: 'thomas',

subject: 'physics'

}

{

\_id: ObjectId('6831921364be9171e9a97d5c'),

name: 'tina',

subject: 'maths'

}

{

\_id: ObjectId('6831921364be9171e9a97d5d'),

name: 'jonty',

subject: 'english'

}

**Updating Collection with Data: -**

db.students.updateOne({name : 'jonty'}, {$set: {'subject':'zoology'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

db.students.find()

{

\_id: ObjectId('6831921364be9171e9a97d59'),

name: 'roger',

subject: 'maths'

}

{

\_id: ObjectId('6831921364be9171e9a97d5a'),

name: 'anita',

subject: 'chemistry'

}

{

\_id: ObjectId('6831921364be9171e9a97d5b'),

name: 'thomas',

subject: 'physics'

}

{

\_id: ObjectId('6831921364be9171e9a97d5c'),

name: 'tina',

subject: 'maths'

}

{

\_id: ObjectId('6831921364be9171e9a97d5d'),

name: 'jonty',

subject: 'zoology'

}

**Creating another Collection with some Data: -**

db.createCollection("aggregateProduct")

{ ok: 1 }

db.aggregateProduct.insertMany( [

{ product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },

{ product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" },

{ product: "sheet", quantity: 100, size: { height: 8.5, weight: 11, unit: "in" }, status: "D" },

{ product: "timetable", quantity: 75, size: { height: 22.85, weight: 30, unit: "cm" }, status: "D" },

{ product: "envelop", quantity: 45, size: { height: 10, weight: 15.25, unit: "cm" }, status: "A" },

{ product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },

{ product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" },

{ product: "sheet", quantity: 100, size: { height: 8.5, weight: 11, unit: "in" }, status: "D" },

{ product: "timetable", quantity: 75, size: { height: 22.85, weight: 30, unit: "cm" }, status: "D" },

{ product: "envelop", quantity: 45, size: { height: 10, weight: 15.25, unit: "cm" }, status: "A" },

{ product: "register", quantity: 25, size: { height: 14, weight: 21, unit: "cm" }, status: "A" },

{ product: "book", quantity: 50, size: { height: 8.5, weight: 11, unit: "in" }, status: "A" }

]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6831941664be9171e9a97d5e'),

'1': ObjectId('6831941664be9171e9a97d5f'),

'2': ObjectId('6831941664be9171e9a97d60'),

'3': ObjectId('6831941664be9171e9a97d61'),

'4': ObjectId('6831941664be9171e9a97d62'),

'5': ObjectId('6831941664be9171e9a97d63'),

'6': ObjectId('6831941664be9171e9a97d64'),

'7': ObjectId('6831941664be9171e9a97d65'),

'8': ObjectId('6831941664be9171e9a97d66'),

'9': ObjectId('6831941664be9171e9a97d67'),

'10': ObjectId('6831941664be9171e9a97d68'),

'11': ObjectId('6831941664be9171e9a97d69')

}

}

db.aggregateProduct.find()

{

\_id: ObjectId('6831941664be9171e9a97d5e'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d5f'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d60'),

product: 'sheet',

quantity: 100,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'D'

}

{

\_id: ObjectId('6831941664be9171e9a97d61'),

product: 'timetable',

quantity: 75,

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

status: 'D'

}

{

\_id: ObjectId('6831941664be9171e9a97d62'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d63'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d64'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d65'),

product: 'sheet',

quantity: 100,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'D'

}

{

\_id: ObjectId('6831941664be9171e9a97d66'),

product: 'timetable',

quantity: 75,

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

status: 'D'

}

{

\_id: ObjectId('6831941664be9171e9a97d67'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d68'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A'

}

{

\_id: ObjectId('6831941664be9171e9a97d69'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A'

}

**Using Aggregate Function with Match & Group Stage: -**

db.aggregateProduct.aggregate([

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

{ $group: { \_id: "$product", total: { $sum: "$quantity" } } }

])

{

\_id: 'register',

total: 75

}

{

\_id: 'book',

total: 150

}

{

\_id: 'envelop',

total: 90

}

**Using Aggregate Function with Match, Group & Sort Stages: -**

db.aggregateProduct.aggregate([

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

{ $group: { \_id: "$product", total: { $sum: "$quantity" } } },

{$sort : { total : 1 }

}

])

{

\_id: 'register',

total: 75

}

{

\_id: 'envelop',

total: 90

}

{

\_id: 'book',

total: 150

}

**Using Aggregate Function and Adding Fields using Max & Min: -**

db.aggregateProduct.aggregate([

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

{ $addFields: {

maxWeight: { $max: "$size.weight" },

minSizeHeight: { $min: "$size.height" }

} }

])

{

\_id: ObjectId('6831941664be9171e9a97d5e'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxWeight: 21,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d5f'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxWeight: 11,

minSizeHeight: 8.5

}

{

\_id: ObjectId('6831941664be9171e9a97d62'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A',

maxWeight: 15.25,

minSizeHeight: 10

}

{

\_id: ObjectId('6831941664be9171e9a97d63'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxWeight: 21,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d64'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxWeight: 11,

minSizeHeight: 8.5

}

{

\_id: ObjectId('6831941664be9171e9a97d67'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A',

maxWeight: 15.25,

minSizeHeight: 10

}

{

\_id: ObjectId('6831941664be9171e9a97d68'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxWeight: 21,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d69'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxWeight: 11,

minSizeHeight: 8.5

}

db.aggregateProduct.aggregate([

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

{ $addFields: {

maxQ: { $max: "$quantity" },

minSizeHeight: { $min: "$size.height" }

} }

])

{

\_id: ObjectId('6831941664be9171e9a97d5e'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxQ: 25,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d5f'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxQ: 50,

minSizeHeight: 8.5

}

{

\_id: ObjectId('6831941664be9171e9a97d62'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A',

maxQ: 45,

minSizeHeight: 10

}

{

\_id: ObjectId('6831941664be9171e9a97d63'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxQ: 25,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d64'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxQ: 50,

minSizeHeight: 8.5

}

{

\_id: ObjectId('6831941664be9171e9a97d67'),

product: 'envelop',

quantity: 45,

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

status: 'A',

maxQ: 45,

minSizeHeight: 10

}

{

\_id: ObjectId('6831941664be9171e9a97d68'),

product: 'register',

quantity: 25,

size: {

height: 14,

weight: 21,

unit: 'cm'

},

status: 'A',

maxQ: 25,

minSizeHeight: 14

}

{

\_id: ObjectId('6831941664be9171e9a97d69'),

product: 'book',

quantity: 50,

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

status: 'A',

maxQ: 50,

minSizeHeight: 8.5

}

**Using Aggregate Function with Projection, Group and Sort Stages: -**

db.aggregateProduct.aggregate([

{ $project : { product : 1,quantity : 1 } } , //optional line

{ $group: { \_id: "$product", total: { $sum: "$quantity" } } },

{$sort : { total : 1 }

}

])

{

\_id: 'register',

total: 75

}

{

\_id: 'envelop',

total: 90

}

{

\_id: 'book',

total: 150

}

{

\_id: 'timetable',

total: 150

}

{

\_id: 'sheet',

total: 200

}

**Another Exercise for Aggregation Concept: -**

db.createCollection("new")

{ ok: 1 }

db.new.insertMany({food:'biriyani', quantity: 2,}, {food:'samosa', quantity:4})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

db.new.insertMany({food:'biriyani', quantity: 2}, {food:'samosa', quantity:4})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

db.new.insertMany({food:'biriyani', quantity: 2},{food:'samosa', quantity:4})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

db.new.insertOne({food:'biriyani', quantity:2})

{

acknowledged: true,

insertedId: ObjectId('6831983964be9171e9a97d6a')

}

db.new.find()

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

show collections

aggregateProduct

new

students

db.new.insertMany([{food:'pizza', quantity: 6},{food:'samosa', quantity:4}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId('68319a1164be9171e9a97d6b'),

'1': ObjectId('68319a1164be9171e9a97d6c')

}

}

db.new.find()

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

food: 'pizza',

quantity: 6

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 4

}

db.new.aggregate([

{$sort : { total : 1 }

}

])

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

food: 'pizza',

quantity: 6

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 4

}

db.new.aggregate([

{$sort : { quantity : 1 }

}

])

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 4

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

food: 'pizza',

quantity: 6

}

db.new.upsert([{food:'biriyani', quantity:3}])

TypeError: db.new.upsert is not a function

db.new.upsert({food:'biriyani', quantity:3})

TypeError: db.new.upsert is not a function

db.new.updateOne()

MongoshInvalidInputError: [COMMON-10001] Missing required argument at position 0 (Collection.updateOne)

db.new.updateOne({food: 'biriyani', {$set: {quantity:3} }, {upsert:true})

SyntaxError: Unexpected token (1:36)

[0m[31m[1m>[22m[39m[90m 1 |[39m db[33m.[39m[36mnew[39m[33m.[39mupdateOne({food[33m:[39m [32m'biriyani'[39m[33m,[39m {$set[33m:[39m {quantity[33m:[39m[35m3[39m} }[33m,[39m {upsert[33m:[39m[36mtrue[39m})

[90m |[39m [31m[1m^[22m[39m[0m

db.new.aggregate([

{ $project : { quantity : 1 } } , //optional line

{$sort : { quantity : 1 }

}

])

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

quantity: 4

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

quantity: 6

}

db.new.insertOne({food:"pizza", quantity: 9})

{

acknowledged: true,

insertedId: ObjectId('68319e7d64be9171e9a97d6d')

}

db.new.aggregate([{$project: {quantity:1}}, {$group:{ \_id: "$pizza"}}, {$sort: {quantity:1}}])

{

\_id: null

}

db.new.aggregate([{$project: {quantity:1}}, {$group:{ food: "$pizza"}}, {$sort: {quantity:1}}])

MongoServerError[Location40234]: The field 'food' must be an accumulator object

db.new.aggregate([

{ $project : { quantity : 1 } } , //optional line

{$sort : { quantity : 1 }

}

])

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

quantity: 4

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

quantity: 6

}

{

\_id: ObjectId('68319e7d64be9171e9a97d6d'),

quantity: 9

}

db.new.aggregate([

{ $project : { quantity : 1 } } , //optional line

{$sort : { quantity : 1 }

}

])

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

quantity: 4

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

quantity: 6

}

{

\_id: ObjectId('68319e7d64be9171e9a97d6d'),

quantity: 9

}

db.new.aggregate([

{$group: {-id : '$food'}},

{$sort : { quantity : 1 }

}

])

SyntaxError: Unexpected token (2:13)

[0m [90m 1 |[39m db[33m.[39m[36mnew[39m[33m.[39maggregate([

[31m[1m>[22m[39m[90m 2 |[39m {$group[33m:[39m {[33m-[39mid [33m:[39m [32m'$food'[39m}}[33m,[39m

[90m |[39m [31m[1m^[22m[39m

[90m 3 |[39m {$sort [33m:[39m { quantity [33m:[39m [35m1[39m }

[90m 4 |[39m }

[90m 5 |[39m ])[0m

db.new.aggregate([

{$group: {\_id : '$food'}},

{$sort : { quantity : 1 }

}

])

{

\_id: 'biriyani'

}

{

\_id: 'pizza'

}

{

\_id: 'samosa'

}

db.new.aggregate([

{ $project : { quantity : 1 } } ,

{$group: {-id : '$food'}},

{$sort : { quantity : 1 }

}

])

SyntaxError: Unexpected token (3:13)

[0m [90m 1 |[39m db[33m.[39m[36mnew[39m[33m.[39maggregate([

[90m 2 |[39m { $project [33m:[39m { quantity [33m:[39m [35m1[39m } } [33m,[39m

[31m[1m>[22m[39m[90m 3 |[39m {$group[33m:[39m {[33m-[39mid [33m:[39m [32m'$food'[39m}}[33m,[39m

[90m |[39m [31m[1m^[22m[39m

[90m 4 |[39m {$sort [33m:[39m { quantity [33m:[39m [35m1[39m }

[90m 5 |[39m }

[90m 6 |[39m ])[0m

db.new.aggregate([

{ $project : { quantity : 1 } } ,

{$group: {\_id : '$food'}},

{$sort : { quantity : 1 }

}

])

{

\_id: null

}

db.new.aggregate([

{$group: {\_id : '$food'}},

{$sort : { quantity : 1 }

}

])

{

\_id: 'samosa'

}

{

\_id: 'pizza'

}

{

\_id: 'biriyani'

}

db.new.insertMany({})

MongoInvalidArgumentError: Argument "docs" must be an array of documents

db.createCollection("Sales")

{ ok: 1 }

show collections

aggregateProduct

new

Sales

students

db.Sales.insertMany([{

"\_id":1,

"item":"Mobile",

"quantity":2,

"price":200,

"category":"electronics",

"status":"delivered"

},

{

"\_id":2,

"item":"Shampoo",

"quantity":2,

"price":50,

"category":"hair care",

"status":"cancelled"

},

{

"\_id":3,

"item":"Laptop",

"quantity":1,

"price":700,

"category":"electronics",

"status":"dispatched"

},

])

{

acknowledged: true,

insertedIds: {

'0': 1,

'1': 2,

'2': 3

}

}

db.Sales.find()

{

\_id: 1,

item: 'Mobile',

quantity: 2,

price: 200,

category: 'electronics',

status: 'delivered'

}

{

\_id: 2,

item: 'Shampoo',

quantity: 2,

price: 50,

category: 'hair care',

status: 'cancelled'

}

{

\_id: 3,

item: 'Laptop',

quantity: 1,

price: 700,

category: 'electronics',

status: 'dispatched'

}

db.Sales.insertMany([{

"\_id":4,

"item":"Ear Phones",

"quantity":1,

"price":100,

"category":"electronics",

"status":"delivered"

},

{

"\_id":5,

"item":"Conditioner",

"quantity":2,

"price":200,

"category":"hair care",

"status":"cancelled"

},

{

"\_id":6,

"item":"iPad",

"quantity":1,

"price":600,

"category":"electronics",

"status":"dispatched"

},

])

{

acknowledged: true,

insertedIds: {

'0': 4,

'1': 5,

'2': 6

}

}

db.Sales.find()

{

\_id: 1,

item: 'Mobile',

quantity: 2,

price: 200,

category: 'electronics',

status: 'delivered'

}

{

\_id: 2,

item: 'Shampoo',

quantity: 2,

price: 50,

category: 'hair care',

status: 'cancelled'

}

{

\_id: 3,

item: 'Laptop',

quantity: 1,

price: 700,

category: 'electronics',

status: 'dispatched'

}

{

\_id: 4,

item: 'Ear Phones',

quantity: 1,

price: 100,

category: 'electronics',

status: 'delivered'

}

{

\_id: 5,

item: 'Conditioner',

quantity: 2,

price: 200,

category: 'hair care',

status: 'cancelled'

}

{

\_id: 6,

item: 'iPad',

quantity: 1,

price: 600,

category: 'electronics',

status: 'dispatched'

}

db.Sales.aggregate([

{$match:{status:"delivered"}},

{$project:{item:1,totalSale:{$multiply:["$quantity","$price"]},category:1}},

{$group:{\_id:"$category",totalRevenue:{$sum:"$totalSale"},totalSales:{$sum:1}}},

{$sort:{totalRevenue:-1}}

])

{

\_id: 'electronics',

totalRevenue: 500,

totalSales: 2

}

db.Sales.aggregate([

{$match:{status:"delivered"}},

{$project:{item:1,totalSale:{$multiply:["$quantity","$price"]},category:1}},

{$group:{\_id:"$category",totalRevenue:{$sum:"$totalSale"},totalSales:{$sum:1}}},

])

{

\_id: 'electronics',

totalRevenue: 500,

totalSales: 2

}

db.new.find()

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

food: 'pizza',

quantity: 6

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 4

}

{

\_id: ObjectId('68319e7d64be9171e9a97d6d'),

food: 'pizza',

quantity: 9

}

db.new.findAndModify({

{food:"pizza"},

{$set: {quantity: 8}},

upsert:true

})

SyntaxError: Unexpected token (2:4)

[0m [90m 1 |[39m db[33m.[39m[36mnew[39m[33m.[39mfindAndModify({

[31m[1m>[22m[39m[90m 2 |[39m {food[33m:[39m[32m"pizza"[39m}[33m,[39m

[90m |[39m [31m[1m^[22m[39m

[90m 3 |[39m {$set[33m:[39m {quantity[33m:[39m [35m8[39m}}[33m,[39m

[90m 4 |[39m upsert[33m:[39m[36mtrue[39m

[90m 5 |[39m })[0m

db.new.findAndModify({

{food:"samosa"},

{$set: {quantity: 8}},

upsert:true

})

SyntaxError: Unexpected token (2:4)

[0m [90m 1 |[39m db[33m.[39m[36mnew[39m[33m.[39mfindAndModify({

[31m[1m>[22m[39m[90m 2 |[39m {food[33m:[39m[32m"samosa"[39m}[33m,[39m

[90m |[39m [31m[1m^[22m[39m

[90m 3 |[39m {$set[33m:[39m {quantity[33m:[39m [35m8[39m}}[33m,[39m

[90m 4 |[39m upsert[33m:[39m[36mtrue[39m

[90m 5 |[39m })[0m

db.new.findAndModify({

query: {food:"samosa"},

update: {$set: {quantity: 8}},

upsert:true

})

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 4

}

db.new.findAndModify({

query: {food:"chai"},

update: {$set: {quantity: 8}},

upsert:true

})

null

db.new.findAndModify({

query: {food:"samosa"},

update: {$set: {quantity: 8}},

upsert:true

})

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 8

}

**Count ():**

db.new.count({food:'pizza'})

DeprecationWarning: Collection.count() is deprecated. Use countDocuments or estimatedDocumentCount.

2

show collections

aggregateProduct

new

Sales

Students

**Limit ():**

db.new.find().limit(3)

{

\_id: ObjectId('6831983964be9171e9a97d6a'),

food: 'biriyani',

quantity: 2

}

{

\_id: ObjectId('68319a1164be9171e9a97d6b'),

food: 'pizza',

quantity: 6

}

{

\_id: ObjectId('68319a1164be9171e9a97d6c'),

food: 'samosa',

quantity: 8

}

db.sales.find().limit(2)

db.Sales.find().limit(2)

{

\_id: 1,

item: 'Mobile',

quantity: 2,

price: 200,

category: 'electronics',

status: 'delivered'

}

{

\_id: 2,

item: 'Shampoo',

quantity: 2,

price: 50,

category: 'hair care',

status: 'cancelled'

}

**Indexing:**

db.new.createIndex({field:1})

field\_1

db.new.createIndex({quantity:1})

quantity\_1

db.new.createIndex({quantity:-1})

quantity\_-1

db.new.createIndex({food:1}, {unique:true})

MongoServerError[DuplicateKey]: Index build failed: bd61bb65-b959-4a58-b7dc-b6986bd4b07b: Collection db1.new ( 557895fe-eb0a-4b60-94b6-758a9e5c2270 ) :: caused by :: E11000 duplicate key error collection: db1.new index: food\_1 dup key: { food: "pizza" }

db.new.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { field: 1 }, name: 'field\_1' },

{ v: 2, key: { quantity: 1 }, name: 'quantity\_1' },

{ v: 2, key: { quantity: -1 }, name: 'quantity\_-1' }

]

db.createCollection("indexcollection")

{ ok: 1 }

db.indexcollection.insertMany( [

{ product: "register", status: "A", size: { height: 14, weight: 21, unit: "cm" }, instock: [ { warehouse: "A", quantity: 5 } ] , creationts: ISODate("2014-11-01T12:30:15Z")},

{ product: "book", status: "A", size: { height: 8.5, weight: 11, unit: "in" }, instock: [ { warehouse: "C", quantity: 5 } ], creationts: ISODate("2016-11-01T12:30:15Z") },

{ product: "sheet", status: "D", size: { height: 8.5, weight: 11, unit: "in" }, instock: [ { warehouse: "A", quantity: 60 } ] , creationts: ISODate("2017-11-01T12:30:15Z")},

{ product: "timetable", status: "D", size: { height: 22.85, weight: 30, unit: "cm" }, instock: [ { warehouse: "A", quantity: 40 } ] , creationts: ISODate("2018-11-01T12:30:15Z")},

{ product: "envelop", status: "A", size: { height: 10, weight: 15.25, unit: "cm" }, instock: [ { warehouse: "B", quantity: 15 }, { warehouse: "C", quantity: 35 } ], creationts: ISODate("2019-11-01T12:30:15Z") },

{ product: "register", status: "A", size: { height: 14, weight: 21, unit: "cm" }, instock: [ { warehouse: "A", quantity: 5 } ] , creationts: ISODate("2014-11-01T12:30:15Z")},

{ product: "register", status: "A", size: { height: 8.5, weight: 11, unit: "in" }, instock: [ { warehouse: "C", quantity: 5 } ], creationts: ISODate("2016-11-01T12:30:15Z") },

{ product: "register", status: "D", size: { height: 8.5, weight: 11, unit: "in" }, instock: [ { warehouse: "A", quantity: 60 } ] , creationts: ISODate("2017-11-01T12:30:15Z")},

{ product: "book", status: "D", size: { height: 22.85, weight: 30, unit: "cm" }, instock: [ { warehouse: "A", quantity: 40 } ] , creationts: ISODate("2018-11-01T12:30:15Z")},

{ product: "book", status: "A", size: { height: 10, weight: 15.25, unit: "cm" }, instock: [ { warehouse: "B", quantity: 15 }, { warehouse: "C", quantity: 35 } ], creationts: ISODate("2019-11-01T12:30:15Z") }

]);

{

acknowledged: true,

insertedIds: {

'0': ObjectId('6831b7f464be9171e9a97d6e'),

'1': ObjectId('6831b7f464be9171e9a97d6f'),

'2': ObjectId('6831b7f464be9171e9a97d70'),

'3': ObjectId('6831b7f464be9171e9a97d71'),

'4': ObjectId('6831b7f464be9171e9a97d72'),

'5': ObjectId('6831b7f464be9171e9a97d73'),

'6': ObjectId('6831b7f464be9171e9a97d74'),

'7': ObjectId('6831b7f464be9171e9a97d75'),

'8': ObjectId('6831b7f464be9171e9a97d76'),

'9': ObjectId('6831b7f464be9171e9a97d77')

}

}

**Retrieving Indexes: -**

db.indexcollection.getIndexes()

[ { v: 2, key: { \_id: 1 }, name: '\_id\_' } ]

db.indexcollection.createIndex({product:1})

product\_1

db.indexcollection.createIndex({status:1})

status\_1

db.indexcollection.find({product:'register'})

{

\_id: ObjectId('6831b7f464be9171e9a97d6e'),

product: 'register',

status: 'A',

size: {

height: 14,

weight: 21,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 5

}

],

creationts: 2014-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d73'),

product: 'register',

status: 'A',

size: {

height: 14,

weight: 21,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 5

}

],

creationts: 2014-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d74'),

product: 'register',

status: 'A',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'C',

quantity: 5

}

],

creationts: 2016-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d75'),

product: 'register',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}

db.indexcollection.find({status:'D'})

{

\_id: ObjectId('6831b7f464be9171e9a97d70'),

product: 'sheet',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d71'),

product: 'timetable',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d75'),

product: 'register',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d76'),

product: 'book',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

**Creating Indexes: -**

db.indexcollection.createIndex({creations:1})

creations\_1

db.indexcollection.find({creations: {$gt:ISODate('2017-01-01T00:00:00Z')}})

db.indexcollection.createIndex({creationts:1})

creationts\_1

db.indexcollection.find({creationts: {$gt:ISODate('2017-01-01T00:00:00Z')}})

{

\_id: ObjectId('6831b7f464be9171e9a97d70'),

product: 'sheet',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d75'),

product: 'register',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d71'),

product: 'timetable',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d76'),

product: 'book',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d72'),

product: 'envelop',

status: 'A',

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

instock: [

{

warehouse: 'B',

quantity: 15

},

{

warehouse: 'C',

quantity: 35

}

],

creationts: 2019-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d77'),

product: 'book',

status: 'A',

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

instock: [

{

warehouse: 'B',

quantity: 15

},

{

warehouse: 'C',

quantity: 35

}

],

creationts: 2019-11-01T12:30:15.000Z

}

db.indexcollection.createIndex({'size.height':1, 'size.weight':1})

size.height\_1\_size.weight\_1

db.indexcollection.find({'size.height': {$gt: 14}})

{

\_id: ObjectId('6831b7f464be9171e9a97d71'),

product: 'timetable',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d76'),

product: 'book',

status: 'D',

size: {

height: 22.85,

weight: 30,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 40

}

],

creationts: 2018-11-01T12:30:15.000Z

}

db.indexcollection.createIndex({'instock.warehouse':1})

instock.warehouse\_1

db.indexcollection.find({'instock.warehouse':'C'})

{

\_id: ObjectId('6831b7f464be9171e9a97d6f'),

product: 'book',

status: 'A',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'C',

quantity: 5

}

],

creationts: 2016-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d72'),

product: 'envelop',

status: 'A',

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

instock: [

{

warehouse: 'B',

quantity: 15

},

{

warehouse: 'C',

quantity: 35

}

],

creationts: 2019-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d74'),

product: 'register',

status: 'A',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'C',

quantity: 5

}

],

creationts: 2016-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d77'),

product: 'book',

status: 'A',

size: {

height: 10,

weight: 15.25,

unit: 'cm'

},

instock: [

{

warehouse: 'B',

quantity: 15

},

{

warehouse: 'C',

quantity: 35

}

],

creationts: 2019-11-01T12:30:15.000Z

}

show collections

aggregateProduct

indexcollection

new

Sales

students

db.new.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { field: 1 }, name: 'field\_1' },

{ v: 2, key: { quantity: 1 }, name: 'quantity\_1' },

{ v: 2, key: { quantity: -1 }, name: 'quantity\_-1' }

]

db.indexcollection.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { product: 1 }, name: 'product\_1' },

{ v: 2, key: { status: 1 }, name: 'status\_1' },

{ v: 2, key: { creations: 1 }, name: 'creations\_1' },

{ v: 2, key: { creationts: 1 }, name: 'creationts\_1' },

{

v: 2,

key: { 'size.height': 1, 'size.weight': 1 },

name: 'size.height\_1\_size.weight\_1'

},

{

v: 2,

key: { 'instock.warehouse': 1 },

name: 'instock.warehouse\_1'

}

]

**Regex:**

db.indexcollection.find({product:{$regex: /^re/,$options:'i'}})

{

\_id: ObjectId('6831b7f464be9171e9a97d6e'),

product: 'register',

status: 'A',

size: {

height: 14,

weight: 21,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 5

}

],

creationts: 2014-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d73'),

product: 'register',

status: 'A',

size: {

height: 14,

weight: 21,

unit: 'cm'

},

instock: [

{

warehouse: 'A',

quantity: 5

}

],

creationts: 2014-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d74'),

product: 'register',

status: 'A',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'C',

quantity: 5

}

],

creationts: 2016-11-01T12:30:15.000Z

}

{

\_id: ObjectId('6831b7f464be9171e9a97d75'),

product: 'register',

status: 'D',

size: {

height: 8.5,

weight: 11,

unit: 'in'

},

instock: [

{

warehouse: 'A',

quantity: 60

}

],

creationts: 2017-11-01T12:30:15.000Z

}