Microsoft Windows [Version 10.0.19044.2006]

(c) Microsoft Corporation. All rights reserved.

**=============Run mongosh command to start mongo shell==========**

C:\Users\Welcome>**mongosh**

Current Mongosh Log ID: 658243121526d7e6b6de6d46

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6

Using MongoDB: 7.0.4

Using Mongosh: 1.10.6

mongosh 2.1.1 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------

The server generated these startup warnings when booting

2023-12-19T11:28:06.676+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

**==========Show dbs or shows databases for showing all dbs=========**

test> **show dbs**

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

test> **show databases**

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

**===================Creating database=====================**

test> **use abc;**

switched to db abc

**===================Creating collection x=====================**

abc> **db.createCollection('x');**

{ ok: 1 }

**===================Showing collections=====================**

abc**> show collections**

x

**===================Creating collection y=====================**

abc> **db.createCollection('y');**

{ ok: 1 }

**===================Showing collections=====================**

abc> **show collections**

x

y

**===================Deleting collection x=====================**

abc> **db.x.drop();**

true

**===================Showing collections=====================**

abc> **show collections**

y

**===================Deleting collection y=====================**

abc> **db.y.drop();**

true

**===================Showing collections=====================**

abc> **show collections**

**===================Showing databases=====================**

abc> **show databases**

admin 40.00 KiB

config 96.00 KiB

local 40.00 KiB

**===================Deleting database=====================**

abc> **db.dropDatabase();**

{ ok: 1, dropped: 'abc' }

**=================== Showing databases=====================**

abc> **show databases**

admin 40.00 KiB

config 96.00 KiB

local 40.00 KiB

abc> show dbs

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

**=================== Insert Operation=====================**

**================ Creating database students=====================**

abc> **use students;**

switched to db students

**=================== insertOne=====================**

students> **db.data.insertOne({'name':'vinod',age:29});**

{

acknowledged: true,

insertedId: ObjectId("6583fdb31526d7e6b6de6d47")

}

**=================== insertMany=====================**

students> **db.data.insertMany([{'name':'vinamra',age:29},{'name':'sudha',age:20}]);**

{

acknowledged: true,

insertedIds: {

'0': ObjectId("6583fe241526d7e6b6de6d48"),

'1': ObjectId("6583fe241526d7e6b6de6d49")

}

}

**=======================find()===========================**

students> **db.data.find();**

[

{ \_id: ObjectId("6583fdb31526d7e6b6de6d47"), name: 'vinod', age: 29 },

{

\_id: ObjectId("6583fe241526d7e6b6de6d48"),

name: 'vinamra',

age: 29

},

{ \_id: ObjectId("6583fe241526d7e6b6de6d49"), name: 'sudha', age: 20 }

]

**===================Ordered – unordered======================**

students> **show dbs**

admin 40.00 KiB

config 60.00 KiB

local 40.00 KiB

students 72.00 KiB

students> **show collections**

data

students> **db.data.find();**

[

{ \_id: ObjectId("6583fdb31526d7e6b6de6d47"), name: 'vinod', age: 29 },

{

\_id: ObjectId("6583fe241526d7e6b6de6d48"),

name: 'vinamra',

age: 29

},

{ \_id: ObjectId("6583fe241526d7e6b6de6d49"), name: 'sudha', age: 20 }

]

**======================Ordered insert=======================**

students> **db.data.insertMany([{'name':'a',age:30},{'name':'b',age:29, \_id: ObjectId("6583fdb31526d7e6b6de6d47")},{'name':'c',age:28}]);**

Uncaught:

MongoBulkWriteError: E11000 duplicate key error collection: students.data index: \_id\_ dup key: { \_id: ObjectId('6583fdb31526d7e6b6de6d47') }

Result: BulkWriteResult {

insertedCount: 1,

matchedCount: 0,

modifiedCount: 0,

deletedCount: 0,

upsertedCount: 0,

upsertedIds: {},

insertedIds: {

'0': ObjectId("6585624b1526d7e6b6de6d4a"),

'1': ObjectId("6583fdb31526d7e6b6de6d47"),

'2': ObjectId("6585624b1526d7e6b6de6d4b")

}

}

Write Errors: [

WriteError {

err: {

index: 1,

code: 11000,

errmsg: "E11000 duplicate key error collection: students.data index: \_id\_ dup key: { \_id: ObjectId('6583fdb31526d7e6b6de6d47') }",

errInfo: undefined,

op: { name: 'b', age: 29, \_id: ObjectId("6583fdb31526d7e6b6de6d47") }

}

}

]

students> **db.data.find();**

[

{ \_id: ObjectId("6583fdb31526d7e6b6de6d47"), name: 'vinod', age: 29 },

{

\_id: ObjectId("6583fe241526d7e6b6de6d48"),

name: 'vinamra',

age: 29

},

{ \_id: ObjectId("6583fe241526d7e6b6de6d49"), name: 'sudha', age: 20 },

{ \_id: ObjectId("6585624b1526d7e6b6de6d4a"), name: 'a', age: 30 }

]

**======================UnOrdered insert=======================**

students> **db.data.insertMany([{'name':'a1',age:30},{'name':'b1',age:29, \_id: ObjectId("6583fdb31526d7e6b6de6d47")},{'name':'c1',age:28}],{ordered:false});**

Uncaught:

MongoBulkWriteError: E11000 duplicate key error collection: students.data index: \_id\_ dup key: { \_id: ObjectId('6583fdb31526d7e6b6de6d47') }

Result: BulkWriteResult {

insertedCount: 2,

matchedCount: 0,

modifiedCount: 0,

deletedCount: 0,

upsertedCount: 0,

upsertedIds: {},

insertedIds: {

'0': ObjectId("6585627e1526d7e6b6de6d4c"),

'1': ObjectId("6583fdb31526d7e6b6de6d47"),

'2': ObjectId("6585627e1526d7e6b6de6d4d")

}

}

Write Errors: [

WriteError {

err: {

index: 1,

code: 11000,

errmsg: "E11000 duplicate key error collection: students.data index: \_id\_ dup key: { \_id: ObjectId('6583fdb31526d7e6b6de6d47') }",

errInfo: undefined,

op: {

name: 'b1',

age: 29,

\_id: ObjectId("6583fdb31526d7e6b6de6d47")

}

}

}

]

students> db.data.find();

[

{ \_id: ObjectId("6583fdb31526d7e6b6de6d47"), name: 'vinod', age: 29 },

{

\_id: ObjectId("6583fe241526d7e6b6de6d48"),

name: 'vinamra',

age: 29

},

{ \_id: ObjectId("6583fe241526d7e6b6de6d49"), name: 'sudha', age: 20 },

{ \_id: ObjectId("6585624b1526d7e6b6de6d4a"), name: 'a', age: 30 },

{ \_id: ObjectId("6585627e1526d7e6b6de6d4c"), name: 'a1', age: 30 },

{ \_id: ObjectId("6585627e1526d7e6b6de6d4d"), name: 'c1', age: 28 }

]

**=================== Read Operation=====================**

**========================find()===========================**

students**> db.data.find({'name':'a'});**

[ { \_id: ObjectId("6585624b1526d7e6b6de6d4a"), name: 'a', age: 30 } ]

**========================findOne()=======================**

students> **db.data.findOne({'name':'a'});**

{ \_id: ObjectId("6585624b1526d7e6b6de6d4a"), name: 'a', age: 30 }

**=================== Importing Json file=====================**

**//products.json & categories.json are not json array but sales.json is json array**

C:\Users\Welcome>**mongoimport C:\Users\Welcome\Desktop\mongo\_json\products.json -d shop -c products**

2023-12-23T06:32:10.070+0530 connected to: mongodb://localhost/

2023-12-23T06:32:10.698+0530 10355 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\Welcome>**mongoimport C:\Users\Welcome\Desktop\mongo\_json\sales.json -d shop -c sales**

2023-12-23T18:06:56.125+0530 connected to: mongodb://localhost/

2023-12-23T18:06:56.131+0530 Failed: cannot decode array into a primitive.D

2023-12-23T18:06:56.132+0530 0 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\Welcome>**mongoimport C:\Users\Welcome\Desktop\mongo\_json\sales.json -d shop -c sales --jsonArray**

2023-12-23T18:07:12.937+0530 connected to: mongodb://localhost/

2023-12-23T18:07:12.983+0530 5 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\Welcome>**mongoimport C:\Users\Welcome\Desktop\mongo\_json\categories.json -d shop -c categories**

2023-12-23T18:10:00.417+0530 connected to: mongodb://localhost/

2023-12-23T18:10:00.441+0530 20 document(s) imported successfully. 0 document(s) failed to import.

**=================== Exporting Json file=====================**

C:\Users\Welcome>**mongoexport -d shop -c sales -o C:\Users\Welcome\Desktop\mongo\_json\salestest.json**

2023-12-23T18:18:12.753+0530 connected to: mongodb://localhost/

2023-12-23T18:18:12.796+0530 exported 5 records

**=========== Checking imported data of Json file============**

C:\Users\Welcome>mongosh

Current Mongosh Log ID: 6586d84ea4368ee40d5a8a72

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6

Using MongoDB: 7.0.4

Using Mongosh: 1.10.6

mongosh 2.1.1 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------

The server generated these startup warnings when booting

2023-12-19T11:28:06.676+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

test> show dbs

admin 40.00 KiB

config 72.00 KiB

local 40.00 KiB

shop 716.00 KiB

students 72.00 KiB

test> use shop;

switched to db shop

shop> show collections

categories

products

sales

shop> db.products.find();

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9265"),

name: 'Kitchen Blender',

company: '64c23350e32f4a51b19b9233',

price: 89,

colors: [ '#ffffff', '#ff9900', '#990000' ],

image: '/images/product-blender.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

name: 'Smartphone Model X',

company: '64c23350e32f4a51b19b9230',

price: 699,

colors: [ '#000000', '#ffffff', '#ff0000' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9264"),

name: 'Smart Watch',

company: '64c23350e32f4a51b19b9232',

price: 249,

colors: [ '#000000', '#ff0000', '#00ff00' ],

image: '/images/product-smartwatch.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9267"),

name: 'Skin Care Set',

company: '64c23350e32f4a51b19b9235',

price: 59,

colors: [ '#ffffff', '#ff99cc', '#00cc99' ],

image: '/images/product-skincare.png',

category: '64c2342de32f4a51b19b9251',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9266"),

name: 'Fashionable Sunglasses',

company: '64c23350e32f4a51b19b9234',

price: 129,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-sunglasses.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

name: 'Outdoor Camping Tent',

company: '64c23350e32f4a51b19b9236',

price: 299,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-tent.png',

category: '64c2342de32f4a51b19b9252',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9269"),

name: 'Building Blocks Set',

company: '64c23350e32f4a51b19b9237',

price: 49,

colors: [ '#ff0000', '#00ff00', '#0000ff' ],

image: '/images/product-building-blocks.png',

category: '64c2342de32f4a51b19b9253',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926a"),

name: 'Car Jump Starter',

company: '64c23350e32f4a51b19b9238',

price: 89,

colors: [ '#000000', '#999999', '#00ff00' ],

image: '/images/product-jump-starter.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926d"),

name: 'Gym Equipment Set',

company: '64c23350e32f4a51b19b923b',

price: 799,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-gym-equipment.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926e"),

name: 'Car Cleaning Kit',

company: '64c23350e32f4a51b19b923c',

price: 29,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-car-cleaning.png',

category: '64c2342de32f4a51b19b9254',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926f"),

name: 'Cookbook Collection',

company: '64c23350e32f4a51b19b923d',

price: 69,

colors: [ '#ffffff', '#ff9900', '#0099ff' ],

image: '/images/product-cookbook.png',

category: '64c2342de32f4a51b19b9255',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9270"),

name: 'Healthy Snack Pack',

company: '64c23350e32f4a51b19b923e',

price: 19,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-snack-pack.png',

category: '64c2342de32f4a51b19b9256',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926b"),

name: 'Cookware Set',

company: '64c23350e32f4a51b19b9239',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-cookware.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

name: 'Designer Handbag',

company: '64c23350e32f4a51b19b923a',

price: 349,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-handbag.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9273"),

name: 'Silver Necklace',

company: '64c23350e32f4a51b19b9240',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-necklace.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9274"),

name: 'Outdoor Dining Set',

company: '64c23350e32f4a51b19b9241',

price: 599,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-dining-set.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9275"),

name: 'Air Fryer',

company: '64c23350e32f4a51b19b9242',

price: 129,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-air-fryer.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9272"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9276"),

name: 'Kids Educational Toys',

company: '64c23350e32f4a51b19b9243',

price: 39,

colors: [ '#ffffff', '#ff9900', '#0099ff' ],

image: '/images/product-educational-toys.png',

category: '64c2342de32f4a51b19b9253',

isFeatured: true

}

]

Type "it" for more

shop**> it**

[

{

\_id: ObjectId("64c2363be32f4a51b19b9277"),

name: 'Wireless Earbuds',

company: '64c23350e32f4a51b19b9230',

price: 79,

colors: [ '#000000', '#ff0000', '#00ff00' ],

image: '/images/product-earbuds.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9278"),

name: 'Robot Vacuum Cleaner',

company: '64c23350e32f4a51b19b9231',

price: 299,

colors: [ '#ffffff', '#ffcc00', '#00cc99' ],

image: '/images/product-vacuum.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9271"),

name: 'Pet Grooming Kit',

company: '64c23350e32f4a51b19b923e',

price: 39,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-pet-grooming.png',

category: '64c2342de32f4a51b19b9257',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b927b"),

name: 'Designer Watch',

company: '64c23350e32f4a51b19b9234',

price: 299,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-designer-watch.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927a"),

name: 'Slow Cooker',

company: '64c23350e32f4a51b19b9233',

price: 79,

colors: [ '#333333', '#ff9900', '#990000' ],

image: '/images/product-slow-cooker.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b927c"),

name: 'Fitness Tracker',

company: '64c23350e32f4a51b19b9235',

price: 129,

colors: [ '#ffffff', '#ff99cc', '#00cc99' ],

image: '/images/product-fitness-tracker.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9279"),

name: 'Wireless Gaming Mouse',

company: '64c23350e32f4a51b19b9232',

price: 49,

colors: [ '#000000', '#999999', '#00ff00' ],

image: '/images/product-gaming-mouse.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927d"),

name: 'Car Seat Covers',

company: '64c23350e32f4a51b19b9236',

price: 59,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-car-seat-covers.png',

category: '64c2342de32f4a51b19b9254',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927e"),

name: 'Notebook Collection',

company: '64c23350e32f4a51b19b9237',

price: 19,

colors: [ '#ff0000', '#00ff00', '#0000ff' ],

image: '/images/product-notebook.png',

category: '64c2342de32f4a51b19b9255',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927f"),

name: 'Pet Food Bowl Set',

company: '64c23350e32f4a51b19b9238',

price: 29,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-pet-bowl-set.png',

category: '64c2342de32f4a51b19b9257',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9280"),

name: 'Elegant Dining Table',

company: '64c23350e32f4a51b19b9239',

price: 699,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-dining-table.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9283"),

name: 'Blender & Juicer Combo',

company: '64c23350e32f4a51b19b923c',

price: 149,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-blender-juicer.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9282"),

name: 'Outdoor Patio Set',

company: '64c23350e32f4a51b19b923b',

price: 1199,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-outdoor-patio.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9285"),

name: 'Healthy Snack Pack',

company: '64c23350e32f4a51b19b923e',

price: 19,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-snack-pack.png',

category: '64c2342de32f4a51b19b9256',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9284"),

name: 'Watercolor Paint Set',

company: '64c23350e32f4a51b19b923d',

price: 29,

colors: [ '#ffffff', '#ff9900', '#0099ff' ],

image: '/images/product-watercolor-paint.png',

category: '64c2342de32f4a51b19b925f',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9287"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9286"),

name: 'Pet Grooming Kit',

company: '64c23350e32f4a51b19b923e',

price: 39,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-pet-grooming.png',

category: '64c2342de32f4a51b19b9257',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9289"),

name: 'Outdoor Dining Set',

company: '64c23350e32f4a51b19b9241',

price: 599,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-dining-set.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b928a"),

name: 'Air Fryer',

company: '64c23350e32f4a51b19b9242',

price: 129,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-air-fryer.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: false

}

]

Type "it" for more

**=================== Comparison Operators =================**

shop> **db.products.find({'price':{$eq:699}})**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

name: 'Smartphone Model X',

company: '64c23350e32f4a51b19b9230',

price: 699,

colors: [ '#000000', '#ffffff', '#ff0000' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9280"),

name: 'Elegant Dining Table',

company: '64c23350e32f4a51b19b9239',

price: 699,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-dining-table.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c23707e32f4a51b19b9295"),

name: 'Elegant Dining Table',

company: '64c23350e32f4a51b19b9239',

price: 699,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-dining-table.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c23825e32f4a51b19b92ba"),

name: 'Smartphone',

company: '64c23350e32f4a51b19b9236',

price: 699,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b9251',

isFeatured: true

},

{

\_id: ObjectId("64c23eb1f1b9c8985b80fd33"),

name: 'Tasty Concrete Mouse',

company: ObjectId("64c23350e32f4a51b19b9242"),

price: 699,

colors: [ '#2d0a6a' ],

image: '/images/product-Modern.png',

category: ObjectId("64c2342de32f4a51b19b9251"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f8560206e"),

name: 'Electronic Bronze Mouse',

company: ObjectId("64c23350e32f4a51b19b9238"),

price: 699,

colors: [ '#2b3311' ],

image: '/images/product-Incredible.png',

category: ObjectId("64c2342de32f4a51b19b925e"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f856026ba"),

name: 'Fantastic Bronze Bacon',

company: ObjectId("64c23350e32f4a51b19b9237"),

price: 699,

colors: [ '#335c62' ],

image: '/images/product-Modern.png',

category: ObjectId("64c2342de32f4a51b19b9261"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f856026e7"),

name: 'Sleek Soft Shirt',

company: ObjectId("64c23350e32f4a51b19b9236"),

price: 699,

colors: [ '#396b56' ],

image: '/images/product-Ergonomic.png',

category: ObjectId("64c2342de32f4a51b19b925c"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f8560293e"),

name: 'Oriental Rubber Gloves',

company: ObjectId("64c23350e32f4a51b19b924c"),

price: 699,

colors: [ '#2f4575' ],

image: '/images/product-Ergonomic.png',

category: ObjectId("64c2342de32f4a51b19b925e"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4f3e0"),

name: 'Fantastic Soft Ball',

company: ObjectId("64c23350e32f4a51b19b9230"),

price: 699,

colors: [ '#2f787d' ],

image: '/images/product-Luxurious.png',

category: ObjectId("64c2342de32f4a51b19b925c"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4f6c5"),

name: 'Licensed Granite Table',

company: ObjectId("64c23350e32f4a51b19b924d"),

price: 699,

colors: [ '#394827' ],

image: '/images/product-Luxurious.png',

category: ObjectId("64c2342de32f4a51b19b925a"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4fa7f"),

name: 'Licensed Soft Table',

company: ObjectId("64c23350e32f4a51b19b9240"),

price: 699,

colors: [ '#233376' ],

image: '/images/product-Handmade.png',

category: ObjectId("64c2342de32f4a51b19b924e"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4fbbb"),

name: 'Small Concrete Table',

company: ObjectId("64c23350e32f4a51b19b923c"),

price: 699,

colors: [ '#0d621e' ],

image: '/images/product-Electronic.png',

category: ObjectId("64c2342de32f4a51b19b925c"),

isFeatured: false,

\_\_v: 0

}

]

shop> **db.products.find({'price':{$eq:699}}).count()**

13

shop> **db.products.find({'price':{$gt:699}}).count()**

3053

shop> **db.products.find({'price':{$in:[699,129,69]}}).count()**

51

shop> **db.products.find({'price':{$nin:[699,129,69]}}).count()**

10304

**================= Cursor methods ===================**

-count(),limit(),skip(),sort()

shop> **db.products.find({'price':{$gt:250}}).count()**

7666

shop> **db.products.find({'price':{$gt:250}}).limit(3)**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

name: 'Smartphone Model X',

company: '64c23350e32f4a51b19b9230',

price: 699,

colors: [ '#000000', '#ffffff', '#ff0000' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

name: 'Outdoor Camping Tent',

company: '64c23350e32f4a51b19b9236',

price: 299,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-tent.png',

category: '64c2342de32f4a51b19b9252',

isFeatured: false

}

]

shop> **db.products.find({'price':{$gt:250}}).limit(5).skip(2)**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

name: 'Outdoor Camping Tent',

company: '64c23350e32f4a51b19b9236',

price: 299,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-tent.png',

category: '64c2342de32f4a51b19b9252',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926d"),

name: 'Gym Equipment Set',

company: '64c23350e32f4a51b19b923b',

price: 799,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-gym-equipment.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

name: 'Designer Handbag',

company: '64c23350e32f4a51b19b923a',

price: 349,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-handbag.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9274"),

name: 'Outdoor Dining Set',

company: '64c23350e32f4a51b19b9241',

price: 599,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-dining-set.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9272"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

}

]

shop> **db.products.find({'price':{$gt:1250}}).limit(3).sort({price:1});**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c23707e32f4a51b19b9296"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

}

]

**=================== Logical Operators ===============**

* **$and , $nor , $or , $not**

**=================== $and===========================**

Q:find products with price greater than 100 and name equals to Diamond Ring

shop> **db.products.find({$and:[{'price':{$gt:100}},{'name':'Diamond Ring'}]})**

[

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c23707e32f4a51b19b9296"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

}

]

**======================== Implicit and =========================**

shop> **db.products.find({'price':{$gt:100},'name':'Diamond Ring'})**

[

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c23707e32f4a51b19b9296"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

}

]

**======================= $or===========================**

shop> **db.products.find({$or:[{'price':{$gt:100}},{'name':'Diamond Ring'}]})**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

name: 'Smartphone Model X',

company: '64c23350e32f4a51b19b9230',

price: 699,

colors: [ '#000000', '#ffffff', '#ff0000' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9264"),

name: 'Smart Watch',

company: '64c23350e32f4a51b19b9232',

price: 249,

colors: [ '#000000', '#ff0000', '#00ff00' ],

image: '/images/product-smartwatch.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9266"),

name: 'Fashionable Sunglasses',

company: '64c23350e32f4a51b19b9234',

price: 129,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-sunglasses.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

name: 'Outdoor Camping Tent',

company: '64c23350e32f4a51b19b9236',

price: 299,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-tent.png',

category: '64c2342de32f4a51b19b9252',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926d"),

name: 'Gym Equipment Set',

company: '64c23350e32f4a51b19b923b',

price: 799,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-gym-equipment.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926b"),

name: 'Cookware Set',

company: '64c23350e32f4a51b19b9239',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-cookware.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

name: 'Designer Handbag',

company: '64c23350e32f4a51b19b923a',

price: 349,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-handbag.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9273"),

name: 'Silver Necklace',

company: '64c23350e32f4a51b19b9240',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-necklace.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9274"),

name: 'Outdoor Dining Set',

company: '64c23350e32f4a51b19b9241',

price: 599,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-dining-set.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9275"),

name: 'Air Fryer',

company: '64c23350e32f4a51b19b9242',

price: 129,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-air-fryer.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9272"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9278"),

name: 'Robot Vacuum Cleaner',

company: '64c23350e32f4a51b19b9231',

price: 299,

colors: [ '#ffffff', '#ffcc00', '#00cc99' ],

image: '/images/product-vacuum.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927b"),

name: 'Designer Watch',

company: '64c23350e32f4a51b19b9234',

price: 299,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-designer-watch.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927c"),

name: 'Fitness Tracker',

company: '64c23350e32f4a51b19b9235',

price: 129,

colors: [ '#ffffff', '#ff99cc', '#00cc99' ],

image: '/images/product-fitness-tracker.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9280"),

name: 'Elegant Dining Table',

company: '64c23350e32f4a51b19b9239',

price: 699,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-dining-table.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9283"),

name: 'Blender & Juicer Combo',

company: '64c23350e32f4a51b19b923c',

price: 149,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-blender-juicer.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9282"),

name: 'Outdoor Patio Set',

company: '64c23350e32f4a51b19b923b',

price: 1199,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-outdoor-patio.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9287"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

}

]

**======================= $nor===========================**

db.products.find({$nor:[{'price':{$gt:100}},{'name':'Diamond Ring'}]}).count()

1139

**======================= $not===========================**

db.products.find({'price':{$not:{$eq:100}}}).count()

10348

**==================== Complex expression ===================**

**Q.1:find products that have price greater than 1340 by both simple and complex expression**

//simple expression

shop> **db.products.find({'price':{$gt:1340}}).count()**

2

//complex expression

shop> **db.products.find({$expr:{$gt:['$price',1340]}}).count()**

2

**Q.2:find sales where (quantity\*price) is greater than the target price**

shop> **db.sales.find({$expr:{$gt:[{$multiply:['$quantity','$price']},'$targetPrice']}})**

[

{ \_id: 2, quantity: 5, price: 25, targetPrice: 100 },

{ \_id: 1, quantity: 10, price: 15, targetPrice: 120 },

{ \_id: 4, quantity: 5, price: 55, targetPrice: 150 },

{ \_id: 5, quantity: 5, price: 55, targetPrice: 150 },

{ \_id: 3, quantity: 6, price: 35, targetPrice: 100 }

]

**Q.2:find sales where (quantity+price) is lesser than the target price**

shop> **db.sales.find({$expr:{$lt:[{$add:['$quantity','$price']},'$targetPrice']}})**

[

{ \_id: 2, quantity: 5, price: 25, targetPrice: 100 },

{ \_id: 1, quantity: 10, price: 15, targetPrice: 120 },

{ \_id: 4, quantity: 5, price: 55, targetPrice: 150 },

{ \_id: 5, quantity: 5, price: 55, targetPrice: 150 },

{ \_id: 3, quantity: 6, price: 35, targetPrice: 100 }

]

**==================== Elements Operators ===================**

1: **$exists**: Matches documents that have a specific field, regardless of its value.

Ex.

db.products.find({ price: { $exists: true } }).count();

shop> **db.products.find({ price: { $exists: true } }).count();**

**10355**

*Q. Find documents with the "price" field present, and if it's present then check the value greater then 1200*

*Ans. db.products.find({ price: { $exists: true },cls price: { $gt: 1250 } });*

shop**> db.products.find({ price: { $exists: true },price: { $gt: 1250 } });**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c23707e32f4a51b19b9296"),

name: 'Diamond Ring',

company: '64c23350e32f4a51b19b923a',

price: 1999,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-diamond-ring.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

}

]

**===============================$type======================**

test> db.products.find({price:{$type:'string'}}).count()

0

test> db.products.find({price:{$type:2}}).count()

0

test> db.products.find({price:{$type:1}}).count()

0

test> db.products.find({price:{$type:6}}).count()

0

test> db.products.find({price:{$type:16}}).count()

0

test> db.products.find({price:{$type:18}}).count()

0

test> db.products.find({price:{$type:19}}).count()

0

**//importing commnets.json**

C:\Users\Welcome>mongoimport C:\Users\Welcome\Desktop\mongo\_json\comments.json -d shop -c comments

2023-12-28T07:50:32.360+0530 connected to: mongodb://localhost/

2023-12-28T07:50:32.457+0530 7 document(s) imported successfully. 0 document(s) failed to import.

C:\Users\Welcome>mongosh

Current Mongosh Log ID: 658cdb9da7a9c380bdf065aa

Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+1.10.6

Using MongoDB: 7.0.4

Using Mongosh: 1.10.6

mongosh 2.1.1 is available for download: https://www.mongodb.com/try/download/shell

For mongosh info see: https://docs.mongodb.com/mongodb-shell/

------

The server generated these startup warnings when booting

2023-12-19T11:28:06.676+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted

------

test> use shop

switched to db shop

shop> show collections

categories

comments

products

sales

shop> db.comments.find()

[

{

\_id: 3,

title: 'Getting Started with NoSQL Databases',

content: 'NoSQL databases provide flexible data models...',

author: 'Sarah Williams',

comments: [

{ user: 'Eva', text: 'Very helpful!' },

{ user: 'Frank', text: 'Clear explanations.' }

],

metadata: { views: 1200, likes: 40 }

},

{

\_id: 4,

title: 'Advanced Queries in MongoDB',

content: 'Learn how to write complex queries...',

author: 'Michael Johnson',

comments: [

{ user: 'Grace', text: 'Mind-blowing content!' },

{ user: 'Henry', text: 'Impressive examples.' }

],

metadata: { views: 1500, likes: 60 }

},

{

\_id: 1,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'John Doe',

comments: [

{ user: 'Alice', text: 'Great article!' },

{ user: 'Bob', text: 'Thanks for sharing.' },

{ user: 'Eva', text: 'Its beatifull!' },

{ user: 'jessy' }

],

metadata: { views: 1000, likes: 50 }

},

{

\_id: 5,

title: 'Optimizing MongoDB Performance',

content: 'Tips and tricks to improve MongoDB performance...',

author: 'Alex Turner',

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' }

],

metadata: { views: 900, likes: 55 }

},

{

\_id: 6,

title: 'Schema Design Best Practices',

content: 'Designing schemas for optimal performance...',

author: 'Emily Brown',

comments: [

{ user: 'Kevin', text: 'Invaluable insights!' },

{ user: 'Lily', text: 'Well-structured explanations.' }

],

metadata: { views: 700, likes: 65 }

},

{

\_id: 7,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'Vinod Thapa',

comments: [

{ user: 'Alice', text: 'Awesome article!' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

],

metadata: { views: 1000, likes: 70 }

},

{

\_id: 2,

title: 'Deep Dive into Aggregation Framework',

content: 'The aggregation framework in MongoDB...',

author: 'Jane Smith',

comments: [

{ user: 'Charlie', text: 'Very informative!' },

{ user: 'David', text: 'Well explained.' }

],

metadata: { views: 800, likes: 70 }

}

]

**======================== $size ============================**

shop> db.comments.find({comments:{$size:1}}).count()

0

shop> db.comments.find({comments:{$size:4}}).count()

1

shop> db.comments.find({comments:{$size:2}}).count()

6

**--------------------------------PROJECTIONS--------------------------------------------**

shop> **db.comments.find({'comments':{$size:2}},{comments:1,\_id:0})**

[

{

comments: [

{ user: 'Eva', text: 'Very helpful!' },

{ user: 'Frank', text: 'Clear explanations.' }

]

},

{

comments: [

{ user: 'Grace', text: 'Mind-blowing content!' },

{ user: 'Henry', text: 'Impressive examples.' }

]

},

{

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' }

]

},

{

comments: [

{ user: 'Kevin', text: 'Invaluable insights!' },

{ user: 'Lily', text: 'Well-structured explanations.' }

]

},

{

comments: [

{ user: 'Alice', text: 'Awesome article!' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

]

},

{

comments: [

{ user: 'Charlie', text: 'Very informative!' },

{ user: 'David', text: 'Well explained.' }

]

}

]

**//except id , you cant include and exclude at same time**

shop> **db.comments.find({'comments':{$size:2}},{comments:1,\_id:0,name:0})**

MongoServerError: Cannot do exclusion on field name in inclusion projection

**-----------------------EMBEDED DOCUMENTS---------------------**

shop> **db.comments.find({'comments.user':Lily})**

ReferenceError: Lily is not defined

shop> **db.comments.find({'comments.user':'Lily'})**

[

{

\_id: 6,

title: 'Schema Design Best Practices',

content: 'Designing schemas for optimal performance...',

author: 'Emily Brown',

comments: [

{ user: 'Kevin', text: 'Invaluable insights!' },

{ user: 'Lily', text: 'Well-structured explanations.' }

],

metadata: { views: 700, likes: 65 }

}

]

**//Q.find the documents where the views in metadata field > 1200**

shop> **db.comments.find({'metadata.views':{$gt:1200}})**

[

{

\_id: 4,

title: 'Advanced Queries in MongoDB',

content: 'Learn how to write complex queries...',

author: 'Michael Johnson',

comments: [

{ user: 'Grace', text: 'Mind-blowing content!' },

{ user: 'Henry', text: 'Impressive examples.' }

],

metadata: { views: 1500, likes: 60 }

}

]

**//Q.find docs. where user in comments =henry and also in the metadata likes views >50**

**shop> db.comments.find({'comments.user':'Henry','metadata.likes':{$gt:50}})**

[

{

\_id: 4,

title: 'Advanced Queries in MongoDB',,

content: 'Learn how to write complex queries...',ce...',

author: 'Michael Johnson',

comments: [

{ user: 'Grace', text: 'Mind-blowing content!' },

{ user: 'Henry', text: 'Impressive examples.' }ions.' }

],

metadata: { views: 1500, likes: 60 }

}

]

**Q.we need to return comments array which must have this 2 users = alice and vinod elements on it, for this we need to use $all**

**------------------------------$all-------------------------------------------------------**

**shop> db.comments.find({'comments.user':{$all:['Alice','Vinod']}})**

[ ],

{ metadata: { views: 1000, likes: 70 }

\_id: 7,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'Vinod Thapa', Aggregation Framework',

comments: [he aggregation framework in MongoDB...',

{ user: 'Alice', text: 'Awesome article!' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

],{ user: 'Charlie', text: 'Very informative!' },

metadata: { views: 1000, likes: 70 }ined.' }

} ],

] metadata: { views: 800, likes: 70 }

**---------------------------------$elemMatch------------------------------------------------**

**shop>** db.comments.find({'comments':{$elemMatch:{'user':'Alice','text':'Awesome article!'}}})

[

{

\_id: 7,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'Vinod Thapa',

comments: [

{ user: 'Alice', text: 'Awesome article!' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

],

metadata: { views: 1000, likes: 70 }

}

]

Or we can do that as below too

shop> db.comments.find({'comments.user':'Alice','comments.text':'Awesome article!'})

[

{

\_id: 7,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'Vinod Thapa',

comments: [

{ user: 'Alice', text: 'Awesome article!' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

],

metadata: { views: 1000, likes: 70 }

}

]

**-------------------------Update Operation-------------------------**

**-------------updateOne() and updateMany()------------------**

Q.1.Update the price value = 45 in a products collections, where the \_id = ObjectId("64c2363be32f4a51b19b9271")

shop> **db.products.find({ \_id : ObjectId("64c2363be32f4a51b19b9271")})**

[

{

\_id: ObjectId("64c2363be32f4a51b19b9271"),

name: 'Pet Grooming Kit',

company: '64c23350e32f4a51b19b923e',

price: 39,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-pet-grooming.png',

category: '64c2342de32f4a51b19b9257',

isFeatured: false

}

]

shop> **db.products.updateOne({ \_id : ObjectId("64c2363be32f4a51b19b9271")},{$set:{'price':45}})**

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop**> db.products.find({ \_id : ObjectId("64c2363be32f4a51b19b9271")})**

[

{

\_id: ObjectId("64c2363be32f4a51b19b9271"),

name: 'Pet Grooming Kit',

company: '64c23350e32f4a51b19b923e',

price: 45,

colors: [ '#ffffff', '#ffcc00', '#99cc00' ],

image: '/images/product-pet-grooming.png',

category: '64c2342de32f4a51b19b9257',

isFeatured: false

}

]

**Q.2. Update the isFeatures value = true in a products collections, where the name = Designer Handbag**

shop> db.products.updateOne({name:'Designer Handbag'},{$set:{'isFeatured':false}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.products.find({'name':'Designer Handbag'})

[

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

name: 'Designer Handbag',

company: '64c23350e32f4a51b19b923a',

price: 349,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-handbag.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: false

}

]

**Q.3. Update all the isFeatures value = true in a products collections, where the  price = 120**

shop> db.products.updateMany({'price':120},{$set:{'isFeatured':true}})

{

acknowledged: true,

insertedId: null,

matchedCount: 5,

modifiedCount: 5,

upsertedCount: 0

}

shop> db.products.find({'price':120})

[

{

\_id: ObjectId("64c23edf9dc1fb2f85602168"),

name: 'Ergonomic Granite Sausages',

company: ObjectId("64c23350e32f4a51b19b9240"),

price: 120,

colors: [ '#641e3b' ],

image: '/images/product-Luxurious.png',

category: ObjectId("64c2342de32f4a51b19b925f"),

isFeatured: true,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f85602c38"),

name: 'Unbranded Frozen Chicken',

company: ObjectId("64c23350e32f4a51b19b9234"),

price: 120,

colors: [ '#146d77' ],

image: '/images/product-Tasty.png',

category: ObjectId("64c2342de32f4a51b19b9261"),

isFeatured: true,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4f16e"),

name: 'Electronic Concrete Bacon',

company: ObjectId("64c23350e32f4a51b19b923f"),

price: 120,

colors: [ '#383d3d' ],

image: '/images/product-Fantastic.png',

category: ObjectId("64c2342de32f4a51b19b9258"),

isFeatured: true,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4fe9e"),

name: 'Practical Concrete Ball',

company: ObjectId("64c23350e32f4a51b19b9230"),

price: 120,

colors: [ '#34221e' ],

image: '/images/product-Sleek.png',

category: ObjectId("64c2342de32f4a51b19b9250"),

isFeatured: true,

\_\_v: 0

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4ff92"),

name: 'Incredible Plastic Sausages',

company: ObjectId("64c23350e32f4a51b19b9242"),

price: 120,

colors: [ '#332648' ],

image: '/images/product-Ergonomic.png',

category: ObjectId("64c2342de32f4a51b19b9256"),

isFeatured: true,

\_\_v: 0

}

]

**-------------renaming and removing fields------------------**

**Q.1.Rename the products collection isFeatured field to isFeature, where the price = 123**

shop> **db.products.updateMany({'price':123},{$rename:{'isFeatured':'isFeature'}})**

{

acknowledged: true,

insertedId: null,

matchedCount: 7,

modifiedCount: 7,

upsertedCount: 0

}

shop> **db.products.find({'price':123})**

[

{

\_id: ObjectId("64c23e543e180eb8a2682ff4"),

name: 'Rustic Frozen Towels',

company: ObjectId("64c23350e32f4a51b19b9235"),

price: 123,

colors: [ '#50215b' ],

image: '/images/product-Rustic.png',

category: ObjectId("64c2342de32f4a51b19b9255"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23edf9dc1fb2f85601add"),

name: 'Modern Frozen Soap',

company: ObjectId("64c23350e32f4a51b19b9243"),

price: 123,

colors: [ '#2c042e' ],

image: '/images/product-Gorgeous.png',

category: ObjectId("64c2342de32f4a51b19b9258"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23edf9dc1fb2f85602697"),

name: 'Elegant Granite Gloves',

company: ObjectId("64c23350e32f4a51b19b924d"),

price: 123,

colors: [ '#136a01' ],

image: '/images/product-Tasty.png',

category: ObjectId("64c2342de32f4a51b19b924e"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23edf9dc1fb2f85602a48"),

name: 'Electronic Soft Keyboard',

company: ObjectId("64c23350e32f4a51b19b9240"),

price: 123,

colors: [ '#63615a' ],

image: '/images/product-Electronic.png',

category: ObjectId("64c2342de32f4a51b19b9254"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23edf9dc1fb2f85602c60"),

name: 'Incredible Steel Keyboard',

company: ObjectId("64c23350e32f4a51b19b9235"),

price: 123,

colors: [ '#7e2d2f' ],

image: '/images/product-Oriental.png',

category: ObjectId("64c2342de32f4a51b19b925a"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23f13f8ffc4957dc4f4ae"),

name: 'Licensed Fresh Table',

company: ObjectId("64c23350e32f4a51b19b923a"),

price: 123,

colors: [ '#09323b' ],

image: '/images/product-Refined.png',

category: ObjectId("64c2342de32f4a51b19b9258"),

\_\_v: 0,

isFeature: false

},

{

\_id: ObjectId("64c23f13f8ffc4957dc50181"),

name: 'Awesome Steel Ball',

company: ObjectId("64c23350e32f4a51b19b923b"),

price: 123,

colors: [ '#6b7b0b' ],

image: '/images/product-Unbranded.png',

category: ObjectId("64c2342de32f4a51b19b9260"),

\_\_v: 0,

isFeature: false

}

]

**---------------------adding and deleting field-----------------------------**

**=============adding======================**

shop> db.comments.updateOne({\_id:5},{$set:{'user':'jessy'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.comments.find({\_id:5})

[

{

\_id: 5,

title: 'Optimizing MongoDB Performance',

content: 'Tips and tricks to improve MongoDB performance...',

author: 'Alex Turner',

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' }

],

metadata: { views: 900, likes: 55 },

user: 'jessy'

}

]

**=============deleting======================**

shop> db.comments.updateOne({\_id:5},{$unset:{'user':'jessy'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.comments.find({\_id:5})

[

{

\_id: 5,

title: 'Optimizing MongoDB Performance',

content: 'Tips and tricks to improve MongoDB performance...',

author: 'Alex Turner',

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' }

],

metadata: { views: 900, likes: 55 }

}

]

**-------------------------------update embedded documents---------------------**

**------------- pushing(adding element)--------------------------------------------------------------**

shop> db.comments.updateOne({\_id:5},{$push:{comments:{'user':'eva','text':'subscribe to my channel'}}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.comments.find({\_id:5})

[

{

\_id: 5,

title: 'Optimizing MongoDB Performance',

content: 'Tips and tricks to improve MongoDB performance...',

author: 'Alex Turner',

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' },

{ user: 'eva', text: 'subscribe to my channel' }

],

metadata: { views: 900, likes: 55 }

}

]

**------------- pushing(deleting element)--------------------------------------------------------------**

shop> db.comments.updateOne({\_id:5},{$pop:{comments:1}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.comments.find({\_id:5})

[

{

\_id: 5,

title: 'Optimizing MongoDB Performance',

content: 'Tips and tricks to improve MongoDB performance...',

author: 'Alex Turner',

comments: [

{ user: 'Ivy', text: 'Very practical advice!' },

{ user: 'Jack', text: 'Helped me a lot.' }

],

metadata: { views: 900, likes: 55 }

}

]

**------------- pushing(deleting element)--------------------------------------------------------------**

**Q.1.Update the text value within an comments array = "Awesome article!", where the id=7 & username=alice.**

shop> db.comments.updateOne({\_id:7,'comments.user':'Alice'},{$set:{'comments.$.text':'awesome Thapa'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

shop> db.comments.find({\_id:7,'comments.user':'Alice'})

[

{

\_id: 7,

title: 'Introduction to MongoDB',

content: 'MongoDB is a popular NoSQL database...',

author: 'Vinod Thapa',

comments: [

{ user: 'Alice', text: 'awesome Thapa' },

{ user: 'Vinod', text: 'Thanks for sharing.' }

],

metadata: { views: 1000, likes: 70 }

}

]

**----------------- Delete operation-----------------------------------------**

In MongoDB, the DELETE operations are used to remove documents from a collection. There are two main methods to perform deletion: deleteOne() and deleteMany().

Delete a Single Document:

syntax : db.collectionName.deleteOne({ \_id: ObjectId("12345") });

db.sales.deleteOne({ \_id: 1 });

Delete Multiple Documents:

Syntax: db.collectionName.deleteMany({ field: "value" });

db.sales.deleteMany({'price':55})

**----------------- explain()-------------------------------------------**

shop> db.products.find({price:{$gt:100}}).count()

9216

shop> db.products.find({price:{$gt:100}}).explain()

{

explainVersion: '2',

queryPlanner: {

namespace: 'shop.products',

indexFilterSet: false,

parsedQuery: { price: { '$gt': 100 } },

queryHash: 'A8E7254E',

planCacheKey: 'D9BF4228',

maxIndexedOrSolutionsReached: false,

maxIndexedAndSolutionsReached: false,

maxScansToExplodeReached: false,

winningPlan: {

queryPlan: {

stage: 'COLLSCAN',

planNodeId: 1,

filter: { price: { '$gt': 100 } },

direction: 'forward'

},

slotBasedPlan: {

slots: '$$RESULT=s5 env: { s7 = 100, s1 = TimeZoneDatabase(Pacific/Efate...Europe/Kaliningrad) (timeZoneDB), s2 = Nothing (SEARCH\_META), s3 = 1704171231092 (NOW) }',

stages: '[1] filter {traverseF(s4, lambda(l1.0) { ((l1.0 > s7) ?: false) }, false)} \n' +

'[1] scan s5 s6 none none none none lowPriority [s4 = price] @"7812c093-6d5b-4c95-a342-9bf149bfb9ca" true false '

}

},

rejectedPlans: []

},

command: {

find: 'products',

filter: { price: { '$gt': 100 } },

'$db': 'shop'

},

serverInfo: {

host: 'DESKTOP-NSHR7J8',

port: 27017,

version: '7.0.4',

gitVersion: '38f3e37057a43d2e9f41a39142681a76062d582e'

},

serverParameters: {

internalQueryFacetBufferSizeBytes: 104857600,

internalQueryFacetMaxOutputDocSizeBytes: 104857600,

internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,

internalDocumentSourceGroupMaxMemoryBytes: 104857600,

internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,

internalQueryProhibitBlockingMergeOnMongoS: 0,

internalQueryMaxAddToSetBytes: 104857600,

internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,

internalQueryFrameworkControl: 'trySbeEngine'

},

ok: 1

}

**----------------------explain(‘executionStats’)----------------------------**

shop> db.products.find({price:{$gt:100}}).explain('executionStats')

{

explainVersion: '2',

queryPlanner: {

namespace: 'shop.products',

indexFilterSet: false,

parsedQuery: { price: { '$gt': 100 } },

queryHash: 'A8E7254E',

planCacheKey: 'D9BF4228',

maxIndexedOrSolutionsReached: false,

maxIndexedAndSolutionsReached: false,

maxScansToExplodeReached: false,

winningPlan: {

queryPlan: {

stage: 'COLLSCAN',

planNodeId: 1,

filter: { price: { '$gt': 100 } },

direction: 'forward'

},

slotBasedPlan: {

slots: '$$RESULT=s5 env: { s7 = 100, s1 = TimeZoneDatabase(Pacific/Efate...Europe/Kaliningrad) (timeZoneDB), s2 = Nothing (SEARCH\_META), s3 = 1704189364204 (NOW) }',

stages: '[1] filter {traverseF(s4, lambda(l1.0) { ((l1.0 > s7) ?: false) }, false)} \n' +

'[1] scan s5 s6 none none none none lowPriority [s4 = price] @"7812c093-6d5b-4c95-a342-9bf149bfb9ca" true false '

}

},

rejectedPlans: []

},

executionStats: {

executionSuccess: true,

nReturned: 9216,

executionTimeMillis: 15,

totalKeysExamined: 0,

totalDocsExamined: 10355,

executionStages: {

stage: 'filter',

planNodeId: 1,

nReturned: 9216,

executionTimeMillisEstimate: 13,

opens: 1,

closes: 1,

saveState: 10,

restoreState: 10,

isEOF: 1,

numTested: 10355,

filter: 'traverseF(s4, lambda(l1.0) { ((l1.0 > s7) ?: false) }, false) ',

inputStage: {

stage: 'scan',

planNodeId: 1,

nReturned: 10355,

executionTimeMillisEstimate: 13,

opens: 1,

closes: 1,

saveState: 10,

restoreState: 10,

isEOF: 1,

numReads: 10355,

recordSlot: 5,

recordIdSlot: 6,

fields: [ 'price' ],

outputSlots: [ Long("4") ]

}

}

},

command: {

find: 'products',

filter: { price: { '$gt': 100 } },

'$db': 'shop'

},

serverInfo: {

host: 'DESKTOP-NSHR7J8',

port: 27017,

version: '7.0.4',

gitVersion: '38f3e37057a43d2e9f41a39142681a76062d582e'

},

serverParameters: {

internalQueryFacetBufferSizeBytes: 104857600,

internalQueryFacetMaxOutputDocSizeBytes: 104857600,

internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,

internalDocumentSourceGroupMaxMemoryBytes: 104857600,

internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,

internalQueryProhibitBlockingMergeOnMongoS: 0,

internalQueryMaxAddToSetBytes: 104857600,

internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,

internalQueryFrameworkControl: 'trySbeEngine'

},

ok: 1

}

**Q. Find name= air fryer from the products collections**

shop> db.products.explain('executionStats').find({'name':'Air Fryer'})

{

explainVersion: '2',

queryPlanner: {

namespace: 'shop.products',

indexFilterSet: false,

parsedQuery: { name: { '$eq': 'Air Fryer' } },

queryHash: '1AD872C6',

planCacheKey: '96BB7350',

maxIndexedOrSolutionsReached: false,

maxIndexedAndSolutionsReached: false,

maxScansToExplodeReached: false,

winningPlan: {

queryPlan: {

stage: 'COLLSCAN',

planNodeId: 1,

filter: { name: { '$eq': 'Air Fryer' } },

direction: 'forward'

},

slotBasedPlan: {

slots: '$$RESULT=s5 env: { s2 = Nothing (SEARCH\_META), s3 = 1704190591547 (NOW), s1 = TimeZoneDatabase(Pacific/Efate...Europe/Kaliningrad) (timeZoneDB), s7 = "Air Fryer" }',

stages: '[1] filter {traverseF(s4, lambda(l1.0) { ((l1.0 == s7) ?: false) }, false)} \n' +

'[1] scan s5 s6 none none none none lowPriority [s4 = name] @"7812c093-6d5b-4c95-a342-9bf149bfb9ca" true false '

}

},

rejectedPlans: []

},

executionStats: {

executionSuccess: true,

nReturned: 3,

executionTimeMillis: 12,

totalKeysExamined: 0,

totalDocsExamined: 10355,

executionStages: {

stage: 'filter',

planNodeId: 1,

nReturned: 3,

executionTimeMillisEstimate: 11,

opens: 1,

closes: 1,

saveState: 10,

restoreState: 10,

isEOF: 1,

numTested: 10355,

filter: 'traverseF(s4, lambda(l1.0) { ((l1.0 == s7) ?: false) }, false) ',

inputStage: {

stage: 'scan',

planNodeId: 1,

nReturned: 10355,

executionTimeMillisEstimate: 10,

opens: 1,

closes: 1,

saveState: 10,

restoreState: 10,

isEOF: 1,

numReads: 10355,

recordSlot: 5,

recordIdSlot: 6,

fields: [ 'name' ],

outputSlots: [ Long("4") ]

}

}

},

command: { find: 'products', filter: { name: 'Air Fryer' }, '$db': 'shop' },

serverInfo: {

host: 'DESKTOP-NSHR7J8',

port: 27017,

version: '7.0.4',

gitVersion: '38f3e37057a43d2e9f41a39142681a76062d582e'

},

serverParameters: {

internalQueryFacetBufferSizeBytes: 104857600,

internalQueryFacetMaxOutputDocSizeBytes: 104857600,

internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,

internalDocumentSourceGroupMaxMemoryBytes: 104857600,

internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,

internalQueryProhibitBlockingMergeOnMongoS: 0,

internalQueryMaxAddToSetBytes: 104857600,

internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,

internalQueryFrameworkControl: 'trySbeEngine'

},

ok: 1

}

**--------------------------------Indexes----------------------------------**

shop> **db.products.getIndexes()**

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

shop> **db.products.createIndex({'name':1})**

name\_1

shop> db.products.getIndexes()

[

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

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

]

//now checking execution time etc

shop> db.products.explain('executionStats').find({'name':'Air Fryer'})

{

explainVersion: '2',

queryPlanner: {

namespace: 'shop.products',

indexFilterSet: false,

parsedQuery: { name: { '$eq': 'Air Fryer' } },

queryHash: '1AD872C6',

planCacheKey: 'F4D525D2',

maxIndexedOrSolutionsReached: false,

maxIndexedAndSolutionsReached: false,

maxScansToExplodeReached: false,

winningPlan: {

queryPlan: {

stage: 'FETCH',

planNodeId: 2,

inputStage: {

stage: 'IXSCAN',

planNodeId: 1,

keyPattern: { name: 1 },

indexName: 'name\_1',

isMultiKey: false,

multiKeyPaths: { name: [] },

isUnique: false,

isSparse: false,

isPartial: false,

indexVersion: 2,

direction: 'forward',

indexBounds: { name: [ '["Air Fryer", "Air Fryer"]' ] }

}

},

slotBasedPlan: {

slots: '$$RESULT=s11 env: { s2 = Nothing (SEARCH\_META), s3 = 1704191224908 (NOW), s1 = TimeZoneDatabase(Pacific/Efate...Europe/Kaliningrad) (timeZoneDB), s5 = KS(3C416972204672796572000104), s6 = KS(3C41697220467279657200FE04), s10 = {"name" : 1} }',

stages: '[2] nlj inner [] [s4, s7, s8, s9, s10] \n' +

' left \n' +

' [1] cfilter {(exists(s5) && exists(s6))} \n' +

' [1] ixseek s5 s6 s9 s4 s7 s8 [] @"7812c093-6d5b-4c95-a342-9bf149bfb9ca" @"name\_1" true \n' +

' right \n' +

' [2] limit 1 \n' +

' [2] seek s4 s11 s12 s7 s8 s9 s10 [] @"7812c093-6d5b-4c95-a342-9bf149bfb9ca" true false \n'

}

},

rejectedPlans: []

},

executionStats: {

executionSuccess: true,

nReturned: 3,

executionTimeMillis: 1,

totalKeysExamined: 3,

totalDocsExamined: 3,

executionStages: {

stage: 'nlj',

planNodeId: 2,

nReturned: 3,

executionTimeMillisEstimate: 0,

opens: 1,

closes: 1,

saveState: 0,

restoreState: 0,

isEOF: 1,

totalDocsExamined: 3,

totalKeysExamined: 3,

collectionScans: 0,

collectionSeeks: 3,

indexScans: 0,

indexSeeks: 1,

indexesUsed: [ 'name\_1' ],

innerOpens: 3,

innerCloses: 1,

outerProjects: [],

outerCorrelated: [ Long("4"), Long("7"), Long("8"), Long("9"), Long("10") ],

outerStage: {

stage: 'cfilter',

planNodeId: 1,

nReturned: 3,

executionTimeMillisEstimate: 0,

opens: 1,

closes: 1,

saveState: 0,

restoreState: 0,

isEOF: 1,

numTested: 1,

filter: '(exists(s5) && exists(s6)) ',

inputStage: {

stage: 'ixseek',

planNodeId: 1,

nReturned: 3,

executionTimeMillisEstimate: 0,

opens: 1,

closes: 1,

saveState: 0,

restoreState: 0,

isEOF: 1,

indexName: 'name\_1',

keysExamined: 3,

seeks: 1,

numReads: 4,

indexKeySlot: 9,

recordIdSlot: 4,

snapshotIdSlot: 7,

indexIdentSlot: 8,

outputSlots: [],

indexKeysToInclude: '00000000000000000000000000000000',

seekKeyLow: 's5 ',

seekKeyHigh: 's6 '

}

},

innerStage: {

stage: 'limit',

planNodeId: 2,

nReturned: 3,

**executionTimeMillisEstimate: 0,**

opens: 3,

closes: 1,

saveState: 0,

restoreState: 0,

isEOF: 1,

limit: 1,

inputStage: {

stage: 'seek',

planNodeId: 2,

nReturned: 3,

**executionTimeMillisEstimate: 0,**

opens: 3,

closes: 1,

saveState: 0,

restoreState: 0,

isEOF: 0,

numReads: 3,

recordSlot: 11,

recordIdSlot: 12,

seekKeySlot: 4,

snapshotIdSlot: 7,

indexIdentSlot: 8,

indexKeySlot: 9,

indexKeyPatternSlot: 10,

fields: [],

outputSlots: []

}

}

}

},

command: { find: 'products', filter: { name: 'Air Fryer' }, '$db': 'shop' },

serverInfo: {

host: 'DESKTOP-NSHR7J8',

port: 27017,

version: '7.0.4',

gitVersion: '38f3e37057a43d2e9f41a39142681a76062d582e'

},

serverParameters: {

internalQueryFacetBufferSizeBytes: 104857600,

internalQueryFacetMaxOutputDocSizeBytes: 104857600,

internalLookupStageIntermediateDocumentMaxSizeBytes: 104857600,

internalDocumentSourceGroupMaxMemoryBytes: 104857600,

internalQueryMaxBlockingSortMemoryUsageBytes: 104857600,

internalQueryProhibitBlockingMergeOnMongoS: 0,

internalQueryMaxAddToSetBytes: 104857600,

internalDocumentSourceSetWindowFieldsMaxMemoryBytes: 104857600,

internalQueryFrameworkControl: 'trySbeEngine'

},

ok: 1

}

Shop> **db.products.dropIndex({'name':1})**

{ nIndexesWas: 2, ok: 1 }

Shop> **db.products.getIndexes()**

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

**--------------------------------------Aggregation---------------------------**

**-----------------------------------------------$match------------------------------**

**Q.1. Retrieve all products with a name = Sleek Wooden Tuna***.*

shop> **db.products.aggregate([{$match:{'name':'Sleek Wooden Tuna'}}])**

[

{

\_id: ObjectId("64c23edf9dc1fb2f85601ae5"),

name: 'Sleek Wooden Tuna',

company: ObjectId("64c23350e32f4a51b19b9235"),

price: 101,

colors: [ '#29505b' ],

image: '/images/product-Recycled.png',

category: ObjectId("64c2342de32f4a51b19b9255"),

isFeatured: false,

\_\_v: 0

},

{

\_id: ObjectId("64c23edf9dc1fb2f856027db"),

name: 'Sleek Wooden Tuna',

company: ObjectId("64c23350e32f4a51b19b9241"),

price: 363,

colors: [ '#464438' ],

image: '/images/product-Sleek.png',

category: ObjectId("64c2342de32f4a51b19b9256"),

isFeatured: false,

\_\_v: 0

}

]

**Q.2. Retrieve all products with a price greater than 50.**

shop> db.products.aggregate([{ $match: { price: { $gt: 50 } } }]);

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

name: 'Laptop Pro',

company: '64c23350e32f4a51b19b9231',

price: 1299,

colors: [ '#333333', '#cccccc', '#00ff00' ],

image: '/images/product-laptop.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9265"),

name: 'Kitchen Blender',

company: '64c23350e32f4a51b19b9233',

price: 89,

colors: [ '#ffffff', '#ff9900', '#990000' ],

image: '/images/product-blender.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

name: 'Smartphone Model X',

company: '64c23350e32f4a51b19b9230',

price: 699,

colors: [ '#000000', '#ffffff', '#ff0000' ],

image: '/images/product-smartphone.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9264"),

name: 'Smart Watch',

company: '64c23350e32f4a51b19b9232',

price: 249,

colors: [ '#000000', '#ff0000', '#00ff00' ],

image: '/images/product-smartwatch.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9267"),

name: 'Skin Care Set',

company: '64c23350e32f4a51b19b9235',

price: 59,

colors: [ '#ffffff', '#ff99cc', '#00cc99' ],

image: '/images/product-skincare.png',

category: '64c2342de32f4a51b19b9251',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b9266"),

name: 'Fashionable Sunglasses',

company: '64c23350e32f4a51b19b9234',

price: 129,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-sunglasses.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

name: 'Outdoor Camping Tent',

company: '64c23350e32f4a51b19b9236',

price: 299,

colors: [ '#333333', '#99cc00', '#ffcc00' ],

image: '/images/product-tent.png',

category: '64c2342de32f4a51b19b9252',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926a"),

name: 'Car Jump Starter',

company: '64c23350e32f4a51b19b9238',

price: 89,

colors: [ '#000000', '#999999', '#00ff00' ],

image: '/images/product-jump-starter.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926d"),

name: 'Gym Equipment Set',

company: '64c23350e32f4a51b19b923b',

price: 799,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-gym-equipment.png',

category: '64c2342de32f4a51b19b925e',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926f"),

name: 'Cookbook Collection',

company: '64c23350e32f4a51b19b923d',

price: 69,

colors: [ '#ffffff', '#ff9900', '#0099ff' ],

image: '/images/product-cookbook.png',

category: '64c2342de32f4a51b19b9255',

isFeatured: true

},

{

\_id: ObjectId("64c23601e32f4a51b19b926b"),

name: 'Cookware Set',

company: '64c23350e32f4a51b19b9239',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-cookware.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

},

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

name: 'Designer Handbag',

company: '64c23350e32f4a51b19b923a',

price: 349,

colors: [ '#000000', '#cc6600', '#663300' ],

image: '/images/product-handbag.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9273"),

name: 'Silver Necklace',

company: '64c23350e32f4a51b19b9240',

price: 149,

colors: [ '#ffffff', '#ffcc00', '#cc9900' ],

image: '/images/product-necklace.png',

category: '64c2342de32f4a51b19b9259',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9274"),

name: 'Outdoor Dining Set',

company: '64c23350e32f4a51b19b9241',

price: 599,

colors: [ '#333333', '#ff9900', '#660000' ],

image: '/images/product-dining-set.png',

category: '64c2342de32f4a51b19b925a',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9275"),

name: 'Air Fryer',

company: '64c23350e32f4a51b19b9242',

price: 129,

colors: [ '#000000', '#666666', '#cc9900' ],

image: '/images/product-air-fryer.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9272"),

name: 'Modern Sofa Set',

company: '64c23350e32f4a51b19b923f',

price: 899,

colors: [ '#000000', '#999999', '#cc6600' ],

image: '/images/product-sofa-set.png',

category: '64c2342de32f4a51b19b9258',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b9277"),

name: 'Wireless Earbuds',

company: '64c23350e32f4a51b19b9230',

price: 79,

colors: [ '#000000', '#ff0000', '#00ff00' ],

image: '/images/product-earbuds.png',

category: '64c2342de32f4a51b19b924e',

isFeatured: false

},

{

\_id: ObjectId("64c2363be32f4a51b19b9278"),

name: 'Robot Vacuum Cleaner',

company: '64c23350e32f4a51b19b9231',

price: 299,

colors: [ '#ffffff', '#ffcc00', '#00cc99' ],

image: '/images/product-vacuum.png',

category: '64c2342de32f4a51b19b925b',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927b"),

name: 'Designer Watch',

company: '64c23350e32f4a51b19b9234',

price: 299,

colors: [ '#000000', '#666666', '#ffcc00' ],

image: '/images/product-designer-watch.png',

category: '64c2342de32f4a51b19b9250',

isFeatured: true

},

{

\_id: ObjectId("64c2363be32f4a51b19b927a"),

name: 'Slow Cooker',

company: '64c23350e32f4a51b19b9233',

price: 79,

colors: [ '#333333', '#ff9900', '#990000' ],

image: '/images/product-slow-cooker.png',

category: '64c2342de32f4a51b19b924f',

isFeatured: false

}

]

**-----------------------------------------------$group------------------------------**

shop> db.products.aggregate([{$group:{\_id:"$company",totalProducts:{$sum:1}}}])

[

{ \_id: '64c23350e32f4a51b19b9239', totalProducts: 6 },

{ \_id: ObjectId("64c23350e32f4a51b19b9239"), totalProducts: 366 },

{ \_id: ObjectId("64c23350e32f4a51b19b9231"), totalProducts: 333 },

{ \_id: '64c23350e32f4a51b19b9244', totalProducts: 2 },

{ \_id: '64c23350e32f4a51b19b9251', totalProducts: 2 },

{ \_id: ObjectId("64c23350e32f4a51b19b9235"), totalProducts: 356 },

{ \_id: '64c23350e32f4a51b19b9256', totalProducts: 1 },

{ \_id: '64c23350e32f4a51b19b924c', totalProducts: 2 },

{ \_id: ObjectId("64c23350e32f4a51b19b9230"), totalProducts: 319 },

{ \_id: ObjectId("64c23350e32f4a51b19b9237"), totalProducts: 353 },

{ \_id: ObjectId("64c23350e32f4a51b19b9242"), totalProducts: 309 },

{ \_id: ObjectId("64c23350e32f4a51b19b9243"), totalProducts: 351 },

{ \_id: '64c23350e32f4a51b19b9250', totalProducts: 2 },

{ \_id: ObjectId("64c23350e32f4a51b19b923f"), totalProducts: 324 },

{ \_id: '64c23350e32f4a51b19b9233', totalProducts: 6 },

{ \_id: '64c23350e32f4a51b19b9232', totalProducts: 6 },

{ \_id: '64c23350e32f4a51b19b923e', totalProducts: 7 },

{ \_id: '64c23350e32f4a51b19b923b', totalProducts: 6 },

{ \_id: '64c23350e32f4a51b19b9249', totalProducts: 2 },

{ \_id: '64c23350e32f4a51b19b9254', totalProducts: 1 }

]

Type "it" for more

In above, we grouped some fields of collection named ‘products’ and new collection is created whose id field is filled wih ‘company’ field data.

And we alsp created second field ‘totalProducts’ in which we performed sum means it will give the no of how many times the same \_id is there or product collection’s ‘’company’’ field is repeated

Ex. { \_id: '64c23350e32f4a51b19b9232', totalProducts: 6 },

Means company field having same 64c23350e32f4a51b19b9232 no is repeated 6 times in whole product collection

Here $sum:1 means do the sum of first field data repeatation means it works on \_id and counts its repeatation if we give $sum:’$price’ then it would sum the price of same \_id as below:

shop> db.products.aggregate([{$group:{\_id:"$company",totalProducts:{$sum:"$price"}}}])

[

{ \_id: ObjectId("64c23350e32f4a51b19b9240"), totalProducts: 164622 },

{ \_id: '64c23350e32f4a51b19b9245', totalProducts: 598 },

{ \_id: ObjectId("64c23350e32f4a51b19b923c"), totalProducts: 177608 },

{ \_id: ObjectId("64c23350e32f4a51b19b924c"), totalProducts: 169507 },

{ \_id: '64c23350e32f4a51b19b9246', totalProducts: 978 },

{ \_id: '64c23350e32f4a51b19b9247', totalProducts: 398 },

{ \_id: '64c23350e32f4a51b19b924a', totalProducts: 478 },

{ \_id: '64c23350e32f4a51b19b9253', totalProducts: 69 },

{ \_id: ObjectId("64c23350e32f4a51b19b9233"), totalProducts: 157935 },

{ \_id: ObjectId("64c23350e32f4a51b19b9234"), totalProducts: 180935 },

{ \_id: '64c23350e32f4a51b19b9257', totalProducts: 79 },

{ \_id: '64c23350e32f4a51b19b924b', totalProducts: 328 },

{ \_id: ObjectId("64c23350e32f4a51b19b924b"), totalProducts: 171802 },

{ \_id: '64c23350e32f4a51b19b9237', totalProducts: 784 },

{ \_id: ObjectId("64c23350e32f4a51b19b9245"), totalProducts: 153241 },

{ \_id: '64c23350e32f4a51b19b9236', totalProducts: 1344 },

{ \_id: '64c23350e32f4a51b19b923a', totalProducts: 4884 },

{ \_id: ObjectId("64c23350e32f4a51b19b9232"), totalProducts: 170827 },

{ \_id: '64c23350e32f4a51b19b9238', totalProducts: 393 },

{ \_id: '64c23350e32f4a51b19b9248', totalProducts: 228 }

]

Type "it" for more

**Q.1 find the products which have price>900 ,group them and do sum of their price**

shop> db.products.aggregate([{$match:{price:{$gt:900}}},{$group:{\_id:"$company",totalProducts:{$sum:"$price"}}}])

[

{ \_id: ObjectId("64c23350e32f4a51b19b9243"), totalProducts: 29271 },

{ \_id: ObjectId("64c23350e32f4a51b19b9247"), totalProducts: 36148 },

{ \_id: ObjectId("64c23350e32f4a51b19b9241"), totalProducts: 27397 },

{ \_id: ObjectId("64c23350e32f4a51b19b924a"), totalProducts: 31476 },

{ \_id: '64c23350e32f4a51b19b9231', totalProducts: 1299 },

{ \_id: ObjectId("64c23350e32f4a51b19b923b"), totalProducts: 41725 },

{ \_id: '64c23350e32f4a51b19b923b', totalProducts: 2398 },

{ \_id: ObjectId("64c23350e32f4a51b19b923e"), totalProducts: 38053 },

{ \_id: ObjectId("64c23350e32f4a51b19b9248"), totalProducts: 31617 },

{ \_id: ObjectId("64c23350e32f4a51b19b9231"), totalProducts: 24780 },

{ \_id: ObjectId("64c23350e32f4a51b19b9244"), totalProducts: 35119 },

{ \_id: ObjectId("64c23350e32f4a51b19b9249"), totalProducts: 38105 },

{ \_id: ObjectId("64c23350e32f4a51b19b924d"), totalProducts: 23985 },

{ \_id: ObjectId("64c23350e32f4a51b19b9238"), totalProducts: 27539 },

{ \_id: ObjectId("64c23350e32f4a51b19b9239"), totalProducts: 27534 },

{ \_id: ObjectId("64c23350e32f4a51b19b9246"), totalProducts: 28691 },

{ \_id: ObjectId("64c23350e32f4a51b19b9236"), totalProducts: 30492 },

{ \_id: ObjectId("64c23350e32f4a51b19b923a"), totalProducts: 33379 },

{ \_id: ObjectId("64c23350e32f4a51b19b923d"), totalProducts: 22005 },

{ \_id: ObjectId("64c23350e32f4a51b19b9232"), totalProducts: 35876 }

]

Type "it" for more

**Q.2.find the quantity = 5, group them with same quantity and find the average price**

shop> db.sales.aggregate([

... { $match: { quantity: 5 } },

... {

... $group: {

... \_id: { quan: "$quantity" },

... avgPrice: { $avg: "$price" },

... },

... },

... ]);

[ { \_id: { quan: 5 }, avgPrice: 45 } ]

**-----------------------------------------------$sort------------------------------**

shop> **db.products.aggregate([**

**... { $match: { price: { $gt: 1200 } } },**

**... {**

**... $group: {**

**... \_id: "$category",**

**... totalPrice: { $sum: "$price" },**

**... },**

**... },**

**... { $sort: { totalPrice: 1 } },**

**... ]);**

[

{ \_id: '64c2342de32f4a51b19b924e', totalPrice: 1299 },

{ \_id: '64c2342de32f4a51b19b9259', totalPrice: 3998 }

]

shop> **db.products.aggregate([ { $match: { price: { $gt: 1200 } } }, { $group: { \_id: "$category", totalPrice: { $sum: "$price" } } }, { $sort: { totalPrice: -1 } }] );**

[

{ \_id: '64c2342de32f4a51b19b9259', totalPrice: 3998 },

{ \_id: '64c2342de32f4a51b19b924e', totalPrice: 1299 }

]

**-----------------------------------------------$Project------------------------------**

**Q.1. project the docs. Including price only(id is default field if id is not excluded it would be in result defaultly)**

shop> **db.products.aggregate([{$project:{price:1}}])**

[

{ \_id: ObjectId("64c23601e32f4a51b19b9263"), price: 1299 },

{ \_id: ObjectId("64c23601e32f4a51b19b9265"), price: 89 },

{ \_id: ObjectId("64c23601e32f4a51b19b9262"), price: 699 },

{ \_id: ObjectId("64c23601e32f4a51b19b9264"), price: 249 },

{ \_id: ObjectId("64c23601e32f4a51b19b9267"), price: 59 },

{ \_id: ObjectId("64c23601e32f4a51b19b9266"), price: 129 },

{ \_id: ObjectId("64c23601e32f4a51b19b9268"), price: 299 },

{ \_id: ObjectId("64c23601e32f4a51b19b9269"), price: 49 },

{ \_id: ObjectId("64c23601e32f4a51b19b926a"), price: 89 },

{ \_id: ObjectId("64c23601e32f4a51b19b926d"), price: 799 },

{ \_id: ObjectId("64c23601e32f4a51b19b926e"), price: 29 },

{ \_id: ObjectId("64c23601e32f4a51b19b926f"), price: 69 },

{ \_id: ObjectId("64c23601e32f4a51b19b9270"), price: 19 },

{ \_id: ObjectId("64c23601e32f4a51b19b926b"), price: 149 },

{ \_id: ObjectId("64c23601e32f4a51b19b926c"), price: 349 },

{ \_id: ObjectId("64c2363be32f4a51b19b9273"), price: 149 },

{ \_id: ObjectId("64c2363be32f4a51b19b9274"), price: 599 },

{ \_id: ObjectId("64c2363be32f4a51b19b9275"), price: 129 },

{ \_id: ObjectId("64c2363be32f4a51b19b9272"), price: 899 },

{ \_id: ObjectId("64c2363be32f4a51b19b9276"), price: 39 }

]

Type "it" for more

**Q.2. project the docs. Including price and excluding id**

shop> **db.products.aggregate([{$project:{price:1,\_id:0}}])**

[

{ price: 1299 }, { price: 89 },

{ price: 699 }, { price: 249 },

{ price: 59 }, { price: 129 },

{ price: 299 }, { price: 49 },

{ price: 89 }, { price: 799 },

{ price: 29 }, { price: 69 },

{ price: 19 }, { price: 149 },

{ price: 349 }, { price: 149 },

{ price: 599 }, { price: 129 },

{ price: 899 }, { price: 39 }

]

Type "it" for more

**Q.3. project the docs. Including price and find discounted price which is 0.8\*price**

shop> **db.products.aggregate([{$project:{price:1,discountPrice:{$multiply:['$price',0.8]}}}])**

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

price: 1299,

discountPrice: 1039.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9265"),

price: 89,

discountPrice: 71.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9262"),

price: 699,

discountPrice: 559.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9264"),

price: 249,

discountPrice: 199.20000000000002

},

{

\_id: ObjectId("64c23601e32f4a51b19b9267"),

price: 59,

discountPrice: 47.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9266"),

price: 129,

discountPrice: 103.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9268"),

price: 299,

discountPrice: 239.20000000000002

},

{

\_id: ObjectId("64c23601e32f4a51b19b9269"),

price: 49,

discountPrice: 39.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b926a"),

price: 89,

discountPrice: 71.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b926d"),

price: 799,

discountPrice: 639.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b926e"),

price: 29,

discountPrice: 23.200000000000003

},

{

\_id: ObjectId("64c23601e32f4a51b19b926f"),

price: 69,

discountPrice: 55.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b9270"),

price: 19,

discountPrice: 15.200000000000001

},

{

\_id: ObjectId("64c23601e32f4a51b19b926b"),

price: 149,

discountPrice: 119.2

},

{

\_id: ObjectId("64c23601e32f4a51b19b926c"),

price: 349,

discountPrice: 279.2

},

{

\_id: ObjectId("64c2363be32f4a51b19b9273"),

price: 149,

discountPrice: 119.2

},

{

\_id: ObjectId("64c2363be32f4a51b19b9274"),

price: 599,

discountPrice: 479.20000000000005

},

{

\_id: ObjectId("64c2363be32f4a51b19b9275"),

price: 129,

discountPrice: 103.2

},

{

\_id: ObjectId("64c2363be32f4a51b19b9272"),

price: 899,

discountPrice: 719.2

},

{

\_id: ObjectId("64c2363be32f4a51b19b9276"),

price: 39,

discountPrice: 31.200000000000003

}

]

Type "it" for more

**Q.3. project the docs. Including price and find discounted price which is 0.8\*price for only those docs. Who have price>1200**

shop> db.products.aggregate([{$match:{price:{$gt:1200}}},{$project:{price:1,discountPrice:{$multiply:['$price',0.8]}}}])

[

{

\_id: ObjectId("64c23601e32f4a51b19b9263"),

price: 1299,

discountPrice: 1039.2

},

{

\_id: ObjectId("64c236a2e32f4a51b19b9281"),

price: 1999,

discountPrice: 1599.2

},

{

\_id: ObjectId("64c23707e32f4a51b19b9296"),

price: 1999,

discountPrice: 1599.2

}

]

**-----------------------------------$push-----------------------------------------**

**Q.1.Find documents with a price greater than 1200, then group them by price and create an array of colors for each group.**

shop> **db.products.aggregate([**

**... { $match: { price: { $gt: 1200 } } },**

**... {**

**... $group: {**

**... \_id: { priceGroup: "$price" },**

**... colors: { $push: "$colors" },**

**... },**

**... },**

**... ]);**

[

{

\_id: { priceGroup: 1999 },

colors: [

[ '#000000', '#cc6600', '#663300' ],

[ '#000000', '#cc6600', '#663300' ]

]

},

{

\_id: { priceGroup: 1299 },

colors: [ [ '#333333', '#cccccc', '#00ff00' ] ]

}

]

Here,after pushing we find problem of 1)duplication of colors in array and 2)instead of creating array it is creating array of array

1st problem can be solved by $addtoset and 2nd by $unwind as follows:

**-----------------------------------$unwind-----------------------------------------**

shop> **db.products.aggregate([{$unwind:'$colors'},{ $match: { price: { $gt: 1200 } } }, { $group: { \_id: { priceGroup: "$price" }, colors: { $push: "$colors" } } }] );**

[

{

\_id: { priceGroup: 1999 },

colors: [

'#000000',

'#cc6600',

'#663300',

'#000000',

'#cc6600',

'#663300'

]

},

{

\_id: { priceGroup: 1299 },

colors: [ '#333333', '#cccccc', '#00ff00' ]

}

]

**-----------------------------------$addToSet-----------------------------------------**

shop> **db.products.aggregate([ { $match: { price: { $gt: 1200 } } }, { $unwind: "$colors" }, { $group: { \_id: { priceGroup: "$price" }, colors: { $addToSet: "$colors" } } }] );**

[

{

\_id: { priceGroup: 1999 },

colors: [ '#663300', '#000000', '#cc6600' ]

},

{

\_id: { priceGroup: 1299 },

colors: [ '#00ff00', '#cccccc', '#333333' ]

}

]

**-----------------------------------$size-----------------------------------------**

shop> db.products.aggregate([

... { $match: { price: { $gt: 1200 } } },

... { $unwind: "$colors" },

... {

... $group: {

... \_id: { priceGroup: "$price" },

... colors: { $addToSet: "$colors" },

... colorLength: { $size: "$colors" },

... },

... },

... ]);

**MongoServerError: unknown group operator '$size'**

**Note:$size is not allowed in $group**

shop> db.products.aggregate([ { $match: { price: { $gt: 1200 } } }, { $unwind: "$colors" }, { $group: { \_id: { priceGroup: "$price" }, allColors: { $addToSet: "$colors" } } }, { $project: { \_id: 1, allColors: 1, colorLength: { $size: "$allColors" } } }] );

[

{

\_id: { priceGroup: 1999 },

allColors: [ '#663300', '#000000', '#cc6600' ],

colorLength: 3

},

{

\_id: { priceGroup: 1299 },

allColors: [ '#00ff00', '#cccccc', '#333333' ],

colorLength: 3

}

]

**------------------------------$limit and $skip-----------------------------------**

**shop> db.products.aggregate([**

**... { $match: { price: { $gt: 1200 } } },**

**... { $unwind: "$colors" },**

**... {**

**... $group: {**

**... \_id: { priceGroup: "$price" },**

**... allColors: { $addToSet: "$colors" },**

**... },**

**... },**

**... {**

**... $project: {**

**... \_id: 1,**

**... allColors: 1,**

**... colorLength: { $size: "$allColors" },**

**... },**

**... },**

**... { $limit: 1 },**

**... ]);**

[

{

\_id: { priceGroup: 1299 },

allColors: [ '#333333', '#cccccc', '#00ff00' ],

colorLength: 3

}

]

**//Skip**

shop> db.products.aggregate([ { $match: { price: { $gt: 1200 } } }, { $unwind: "$colors" }, { $group: { \_id: { priceGroup: "$price" }, allColors: { $addToSet: "$colors" } } }, { $project: { \_id: 1, allColors: 1, colorLength: { $size: "$allColors" } } }, { $skip: 1 }] );

[

{

\_id: { priceGroup: 1299 },

allColors: [ '#00ff00', '#333333', '#cccccc' ],

colorLength: 3

}

]

**//////////////////----Inserting col.json----/////////////////////////**

shop> db.col.insertMany([

... {\_id:"64c23350e32f4a51b19b9201",name:"Document 1",values:[10,20,30,40,50]},

... {\_id:"64c23350e32f4a51b19b9202",name:"Document 2",values:[15,25,35,45,55]},

... {\_id:"64c23350e32f4a51b19b9203",name:"Document 3",values:[5,15,25,35,45]},

... {\_id:"64c23350e32f4a51b19b9204",name:"Document 4",values:[30,40,50,60,70]},

... {\_id:"64c23350e32f4a51b19b9205",name:"Document 5",values:[25,35,45,55,65]}]);

{

acknowledged: true,

insertedIds: {

'0': '64c23350e32f4a51b19b9201',

'1': '64c23350e32f4a51b19b9202',

'2': '64c23350e32f4a51b19b9203',

'3': '64c23350e32f4a51b19b9204',

'4': '64c23350e32f4a51b19b9205'

}

}

**-----------------------------------$filter----------------------------------------**

shop> db.col.aggregate([

... {

... $project: {

... name: 1,

... values: {

... $filter: {

... input: "$values",

... as: "value",

... cond: { $gt: ["$$value", 30] },

... },

... },

... },

... },

... ]);

[

{

\_id: '64c23350e32f4a51b19b9201',

name: 'Document 1',

values: [ 40, 50 ]

},

{

\_id: '64c23350e32f4a51b19b9202',

name: 'Document 2',

values: [ 35, 45, 55 ]

},

{

\_id: '64c23350e32f4a51b19b9203',

name: 'Document 3',

values: [ 35, 45 ]

},

{

\_id: '64c23350e32f4a51b19b9204',

name: 'Document 4',

values: [ 40, 50, 60, 70 ]

},

{

\_id: '64c23350e32f4a51b19b9205',

name: 'Document 5',

values: [ 35, 45, 55, 65 ]

}

]