**MONGODB LABBOOK**

**LAB 1 – Introduction to MongoDB**

* 1. Create Your Own Database With Name “CgProductDB”.

Sol 🡪 **use CgProductDB**.

switched to db CgProductDB..

1.2- Create Collection with name “ProductCollection” in the above database.

Sol 🡪 **db.createCollection("ProductCollection")**

{ "ok" : 1 }

1.3- Insert unique product id , item , product category , product price , product quantity ,order Info, coords, email , mobile in above “ProductCollection”

• “Order info” has unique order id, order date and address

• “Address” has street, city, state and zip code.

• “cords” is the array having x, y coordinates of GPS

Refer “MDB AssignmnetData.txt “file for insert operation in “ProductCollection” collection.

Sol 🡪

1. **db.ProductCollection.insert({ \_id:1,item:"Laptop",prodCat :"Electronics",price:30000.0,quantity: 2,orderInfo:{\_id:"O001",orderdate: ISODate("2014-01-01"),address:{street:"66,Airoli",city:"Mumbai",state:"MS",zipcode:700987,coords: [ -73.856077, 40.848447 ]},email:"capgemini@capgemini.com",mobiles:[8888108810]}})**
2. **db.ProductCollection.insert({ \_id:2,item:"Mobile",prodCat :"Electronics",price:15000.0,quantity: 10,orderInfo:{"\_id":"O002","orderdate" : ISODate("2010-03-12"),address:{street:"FC Road",city:"Pune",state:"MS",zipcode:40081,coords: [ -23.80007, 30.1123456 ]},email:"icres@ibm.com",mobiles:[9856342189]}})**
3. **db.ProductCollection.insert({ \_id:3,item:"Google PixelMobile",prodCat :"Electronics",price:15000.0,quantity: 10,orderInfo:{"\_id":"O003","orderdate" : ISODate("2010-03-12"),address:{street:"FC Road",city:"Pune",state:"MS",zipcode:40081,coords: [ -23.80007, 30.1123456 ]},email:"icres@ibm.com",mobiles:[9856342189]}})**
4. **db.ProductCollection.insert({ \_id:4,item:"IPhoneX",prodCat :"Electronics",price:105000.0,quantity: 10,orderInfo:{"\_id":"O004","orderdate" : ISODate("2010-03-12"),address:{street:"FC Road",city:"Pune",state:"MS",zipcode:40081,coords: [ -23.80007, 30.1123456 ]},email:"icres@ibm.com",mobiles:[9856342189]}})**
5. **db.ProductCollection.insert({ \_id:5,item:"TV",prodCat :"Electronics",price:24000.0,quantity: 10,orderInfo:{"\_id":"O003","orderdate" : ISODate("2012-06-17"),address:{street:"GT Road",city:"Sahibabad",state:"UP",zipcode:567777,coords: [ -11.80007, 13.1123456 ]},email:"techm@techm.com",mobiles:[7865452222]}})**
6. **db.ProductCollection.insert({ \_id:6,item:"Bangles",prodCat :"Jewellery",price:4000.0,quantity: 1,orderInfo:{"\_id":"O004","orderdate" : ISODate("2010-05-16"),address:{street:"Salt Lake",city:"Kolkata",state:"West Bengol",zipcode:222224,coords: [ -67.850007, 9.456666 ]},email:"vaishali@gmail.com",mobiles:[8888108850,9402312123]}})**
7. **db.ProductCollection.insert({ \_id:7,item:"Redmi A1",prodCat :"Electronics",price:15000.0,quantity: 10,orderInfo:{"\_id":"O007","orderdate" : ISODate("2010-03-12"),address:{street:"FC Road",city:"Gwalior",state:"MP",zipcode:474011,coords: [ -23.80007, 30.1123456 ]},email:"icres@ibm.com",mobiles:[9856342189]}})**
8. **db.ProductCollection.insert([{ \_id:8,item:"Redmi Note 5 Pro",prodCat :"Electronics",price:15000.0,quantity: 10,orderInfo:{"\_id":"O008","orderdate" : ISODate("2010-03-12"),address:{street:"FC Road",city:"Gwalior",state:"MP",zipcode:474011,coords: [ -23.80007, 30.1123456 ]},email:"icres@ibm.com",mobiles:[9856342189]}}])**

1.4- Fetch the product based on unique product id.

Sol🡪 **db.ProductCollection.find({\_id:1}).pretty()**

{

"\_id" : 1,

"item" : "Laptop",

"prodCat" : "Electronics",

"price" : 30000,

"quantity" : 2,

"orderInfo" : {

"\_id" : "O001",

"orderdate" : ISODate("2014-01-01T00:00:00Z"),

"address" : {

"street" : "66,Airoli",

"city" : "Mumbai",

"state" : "MS",

"zipcode" : 700987,

"coords" : [

-73.856077,

40.848447

]

},

"email" : "capgemini@capgemini.com",

"mobiles" : [

8888108810

]

}

}

1.5- Fetch all product details except coord , email .

Sol 🡪 **db.ProductCollection.find({},{"orderInfo.address.coords":0,"orderInfo.email":0}).pretty().**

{

"\_id" : 1,

"item" : "Laptop",

"prodCat" : "Electronics",

"price" : 30000,

"quantity" : 2,

"orderInfo" : {

"\_id" : "O001",

"orderdate" : ISODate("2014-01-01T00:00:00Z"),

"address" : {

"street" : "66,Airoli",

"city" : "Mumbai",

"state" : "MS",

"zipcode" : 700987

},

"mobiles" : [

8888108810

]

}

}

{

"\_id" : 2,

"item" : "Mobile",

"prodCat" : "Electronics",

"price" : 15000,

"quantity" : 10,

"orderInfo" : {

"\_id" : "O002",

"orderdate" : ISODate("2010-03-12T00:00:00Z"),

"address" : {

"street" : "FC Road",

"city" : "Pune",

"state" : "MS",

"zipcode" : 40081

},

"mobiles" : [

9856342189

]

}

}

{

"\_id" : 5,

"item" : "TV",

"prodCat" : "Electronics",

"price" : 24000,

"quantity" : 10,

"orderInfo" : {

"\_id" : "O003",

"orderdate" : ISODate("2012-06-17T00:00:00Z"),

"address" : {

"street" : "GT Road",

"city" : "Sahibabad",

"state" : "UP",

"zipcode" : 567777

},

"mobiles" : [

7865452222

]

}

}

{

"\_id" : 6,

"item" : "Bangles",

"prodCat" : "Jewellery",

"price" : 4000,

"quantity" : 1,

"orderInfo" : {

"\_id" : "O004",

"orderdate" : ISODate("2010-05-16T00:00:00Z"),

"address" : {

"street" : "Salt Lake",

"city" : "Kolkata",

"state" : "West Bengol",

"zipcode" : 222224

},

"mobiles" : [

8888108850,

9402312123

]

}

}

{

"\_id" : 3,

"item" : "Pen",

"prodCat" : "Stationary",

"price" : 25,

"quantity" : 5,

"orderInfo" : {

"\_id" : "O005",

"orderdate" : ISODate("2015-01-01T00:00:00Z"),

"address" : {

"street" : "66,Airoli",

"city" : "Mumbai",

"state" : "MS",

"zipcode" : 700987

},

"mobiles" : [

8888108810

]

}

}

{

"\_id" : 4,

"item" : "Jeans",

"prodCat" : "Clothing",

"price" : 1500,

"quantity" : 3,

"orderInfo" : {

"\_id" : "O006",

"orderdate" : ISODate("2010-03-12T00:00:00Z"),

"address" : {

"street" : "FC Road",

"city" : "Pune",

"state" : "MS",

"zipcode" : 40081

},

"mobiles" : [

9856342189

]

}

}

{

"\_id" : 7,

"item" : "TV",

"prodCat" : "Jewellery",

"price" : 24000,

"quantity" : 1,

"orderInfo" : {

"\_id" : "O015",

"orderdate" : ISODate("2012-06-17T00:00:00Z"),

"address" : {

"street" : "GT Road",

"city" : "Sahibabad",

"state" : "UP",

"zipcode" : 567777

},

"mobiles" : [

7865452222

]

}

}

{

"\_id" : 8,

"item" : "Shirt",

"prodCat" : "Clothing",

"price" : 400,

"quantity" : 6,

"orderInfo" : {

"\_id" : "O004",

"orderdate" : ISODate("2010-05-16T00:00:00Z"),

"address" : {

"street" : "Salt Lake",

"city" : "Kolkata",

"state" : "West Bengol",

"zipcode" : 222224

},

"mobiles" : [

8888108850,

9402312123

]

}

}

1.6- Add the orderstatus = “pending” field for product id “3” .

Sol 🡪

**db.ProductCollection.update({\_id:3},{$set:{"orderInfo.orderstatus":"pending"}})**

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

> **db.ProductCollection.find({\_id:3}).pretty()**

1.7- Update the product price by 5000 for product name “Laptop”

Sol 🡪 **db.ProductCollection.update({item:"Laptop"},{$set:{price:5000}})**

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

> **db.ProductCollection.find({item:"Laptop"}).pretty()**

1.8- Update the product price by 5000 for all product having product name “Laptop”

Sol 🡪 **db.ProductCollection.update({item:"Laptop"},{$set:{price:5000}},{multi:true}).**

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

> **db.ProductCollection.find({item:"Laptop"}).pretty()**

1.9- Add one more mobile number in mobiles field of order information for order id “2”

Sol 🡪 **db.ProductCollection.update({\_id:2},{$push:{"orderInfo.mobiles":7509406124}}).**

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

> **db.ProductCollection.find({item:"Laptop"}).pretty()**

1.10- Update the product price to “40.00 “and product category to “electronics” for the product having item name “CD”. Insert the product if it is not existing.

Sol 🡪 **db.ProductCollection.update({item:"CD"},{$set:{prodCat:"Electronics",price:40}}).**

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

> **db.ProductCollection.find({item:"TV"}).pretty()**

1.11- Rename “item” field to “productName” in the above collection

Sol 🡪 **db.ProductCollection.update({},{$rename: { item: "productName" } })**

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

> **db.ProductCollection.find({\_id:1}).pretty()**

1.12- Remove the product with \_id=4.

Sol 🡪 **db.ProductCollection.remove({\_id:4});**

WriteResult({ "nRemoved" : 1 })

1.13- Remove all products which product name starts with "C"

Sol 🡪 **db.ProductCollection.remove({ productName: { $in: [ /^T/] } });**

**(Or)**

**db.ProductCollection.remove({productName:{$regex: /^C/}})**

WriteResult({ "nRemoved" : 1 })

1.14- Find the product having productType as either “Electronics” OR “TV”

Sol 🡪

**db.ProductCollection.find({$or:[{prodCat:"Electronics"},{productName:"TV"}]}).pretty()**

1.15- Show the list of those product having price greater than 40000.

Sol 🡪

**db.ProductCollection.find({price:{$gt:40000}}).pretty()**

1.16- Display only product name, product type and product price for those product having price greater than 40000.

Sol 🡪

**db.ProductCollection.find({price:{$gt:40000}},{productName:1,prodCat:1,price:1}).pretty()**

1.17- Display only product name, product type and product price for those product having price greater than 40000. But show only first 3 documents.

Sol 🡪

**db.ProductCollection.find({price:{$gt:40000}},{productName:1,prodCat:1,price:1}).limit(3).pretty()**

1.18- Find all documents which are delivered in state “MS”.

Sol 🡪

**db.ProductCollection.find({"orderInfo.address.state":"MS"}).pretty()**

1.19- Sort products as per price in descending order.

Sol 🡪

**db.ProductCollection.find({}).sort({price:-1}).pretty()**

1.20- Sort product as per product category.

Sol 🡪

**db.ProductCollection.find({}).sort({prodCat:1}).pretty()**

1.21- Display only first 3 products

Sol 🡪

**db.ProductCollection.find({}).limit(3).pretty()**

1.22- Create product order report as follows.

Display-

Product Name: \_\_\_\_ Product Category: \_\_\_\_\_\_\_\_ Product Price:\_\_\_\_\_ Order Date:\_\_\_\_\_\_ Order State:\_\_\_\_\_\_\_

Sol🡪

**db.ProductCollection.find({}).forEach(function(doc){print("\nProduct Name: "+doc.productName+"\nProduct Category:"+doc.prodCat+"\nOrder Date: "+doc.orderInfo.orderdate+"\nOrder State: "+doc.orderInfo.address.state)})**

1.23 Find the products document maching state in “MS” OR “UP”.

Sol 🡪

**db.ProductCollection.find({$or:[{"orderInfo.address.state":"MS"},{"orderInfo.address.state":"UP"}]}).pretty()**

1.24 Find those products having price >40000 and delivered in state “MS”.

Sol 🡪

**db.ProductCollection.find({$and:[{price :{$gt:40000}},{"orderInfo.address.state":"MS"}]}).pretty()**

1.25 Find List of all “electronics” delivered in city “Pune”.

Sol 🡪

**db.ProductCollection.find({prodCat:"Electronics", "orderInfo.address.city":"Pune"}).pretty()**

1.26. Sort the product by product category in ascending order and print it one by one using cursor.

Sol 🡪

**var myCursor = db.ProductCollection.find().sort({prodCat:1})**

**while(myCursor.hasNext()) { print(tojson(myCursor.next()))}**

**Lab 2- Aggregation**

2.1- Display the product with “electronic” category.

Sol🡪

db.ProductCollection.aggregate({$match:{prodCat:"Electronics"}}).pretty();

2.2-Group the product category wise and display the count.

Sol🡪

db.ProductCollection.aggregate([{$group:{\_id:"$prodCat",count:{$sum:1}}}]);

{ "\_id" : "Jewellery", "count" : 1 }

{ "\_id" : "Electronics", "count" : 5 }

{ "\_id" : "Clothing", "count" : 1 }

2.3- Display how many number of products available for “Jewellery” category.

Sol🡪 db.ProductCollection.aggregate([{$match:{prodCat:"Jewellery"}},{$group:{\_id:"$

prodCat",count:{$sum:1}}}]);

{ "\_id" : "Jewellery", "count" : 1 }

2.4- Calculate the average amount and average quantity for each category of product.

Sol🡪

db.ProductCollection.aggregate([{$group:{\_id:"$prodCat",AverageAmount:{$avg:{$multiply:["$price",$quantity"]}},AverageQuantity:{$avg:"$quantity"}}}]);

{ "\_id" : "Jewellery", "AverageAmount" : 4000, "AverageQuantity" : 1 }

{ "\_id" : "Electronics", "AverageAmount" : 789166.6666666666, "AverageQuantity" : 18.33333333333333

}

{ "\_id" : "Clothing", "AverageAmount" : 102500, "AverageQuantity" : 20.5 }

2.5- Find product name, product category, price for those product having quantity greater than 30.

Show only first 3 record.

Sol🡪 db.ProductCollection.find({quantity:{$gt:30}},{productName:1,prodCat:1,price:1

}).pretty().limit(3);

{

"\_id" : 9,

"productName" : "Keyboard",m

"prodCat" : "Electronics",

"price" : 50000

}

{

"\_id" : 10,

"productName" : "Mouse",

"prodCat" : "Electronics",

"price" : 50000

}

{

"\_id" : 15,

"productName" : "Shirts",

"prodCat" : "Clothing",

"price" : 5000

}

2.6-Sort the product quantity wise in ascending order

Sol🡪

db.ProductCollection.aggregate([{$sort:{quantity:1}}]).pretty();

2.7- Display the sum of total sale price category wise.

Sol🡪 db.ProductCollection.aggregate([{$group:{\_id:"$prodCat",TotalSales:{$sum:{$multiply:["$price","$quantity"]}}}}]).pretty();

{ "\_id" : "Jewellery", "TotalSales" : 4000 }

{ "\_id" : "Electronics", "TotalSales" : 4735000 }

{ "\_id" : "Clothing", "TotalSales" : 205000 }

2.8-Find out which product is most selling product.

Sol🡪 db.ProductCollection.aggregate([{$group:{\_id:"$productName",MaxSales:{$max:{$multiply:["$price","$quantity"]}}}},{$sort:{MaxSales:-1}},{$limit:1}]).pretty();

{ "\_id" : "Keyboard", "MaxSales" : 2500000 }

2.9- Find the product list which was ordered in year “2010”

Sol🡪

db.ProductCollection.aggregate([{$project:{productName:1, year:{$year: "$orderInfo.orderdate"}}}{$match:{year:2010}}])

(or)

db.ProductCollection.aggregate([{$project:{year:{$year:"$orderInfo.orderdate"}}},{$match:{year:2010}}])

2.10 Display category wise product name and product price.

Sol🡪 db.ProductCollection.aggregate([{$group:{\_id:"$prodCat",ProductDetails:{$push:

{ProductName:"$productName",Price:"$price"}}}}]).pretty();

{

"\_id" : "Jewellery",

"ProductDetails" : [

{

"ProductName" : "Bangles",

"Price" : 4000

}

]

}

{

"\_id" : "Electronics",

"ProductDetails" : [

{

"ProductName" : "Mobile",

"Price" : 15000

},

{

"ProductName" : "TV",

"Price" : 24000

},

{

"ProductName" : "Pen drive",

"Price" : 5000

},

{

"ProductName" : "Laptop",

"Price" : 40000

},

{

"Price" : 40

},

{

"ProductName" : "Keyboard",

"Price" : 50000

},

{

"ProductName" : "Mouse",

"Price" : 50000

}

]

}

{

"\_id" : "Clothing",

"ProductDetails" : [

{

"ProductName" : "Trousers",

"Price" : 5000

},

{

"ProductName" : "Shirts",

"Price" : 5000

}

]

}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*The END\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*