**Working With Mongo**

**Create a Database**

Use DatabaseName

If the database already existing , will switches to that database

If not existing , creates a new Database and Switches to that Database

Ex: use Books

**Creating a Collection**

db.createCollection("CollectionName")

or

Db.CollectionName.Insert({json Object})

Will creates a collection and inserts the Json Object if Collection Doesn’t Exists

Will Just Inserts the Json Object if the Collection exist already

Ex:

db.Books.insert({

Id:1001,

BookName:"Web Api",

Description:"Web Apiet Reference Book",

AuthorName:["Sonu","Swapnil"],

Year:2014,

Price:320,

Comments:[

{

Email : "w@r.r",

Comment : "Nice Book",

Date: 2/2/2014

},

{

Email : "s@r.r",

Comment : "Nice Book",

Date: 2/2/2014

}

]

})

**To Show the Database**

Show dbs

**To Show the Collections**

show collections –

Show the collections of the current db - Shows the collections which have atleast one document

**To Drop a Collection**

db.CollectionName.drop()

db.Student.drop()

**To Drop a Database**

db.dropDatabase()

**Invertory Collection**

db.inventory.insertMany([

{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },

{ item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },

{ item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },

{ item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },

{ item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }

]);

**Select operations**

Db.CollectionName.find({Condition})

Db.inventory.find() - Will shows all the Documents in the Collection

Db.inventory.find({qty:25}) - Will Display all the documents whose qty is equal to 25

**Operators in MongoDB**

$eq Matches values that are equal to a specified value.

$gt Matches values that are greater than a specified value.

$gte Matches values that are greater than or equal to a specified value.

$lt Matches values that are Lesstha than to a specified value.

$lte Matches values that are Less than or equal to a specified value.

$ne Matches all values that are not equal to a specified value.

$nin Matches none of the values specified in an array.

db.inventory.find({qty:{$lte:25}})

db.inventory.find({qty:{$gt:25}})

db.inventory.find({qty:{$ne:25}})

**using in Operator**

db.inventory.find({status:{$in:["A","W"]}})

**using AND** Logical Checking : conditions can be separated by , can be treated as AND

db.inventory.find({status:"A",qty:{$lt:30}})

**Using $Or**

db.inventory.find({$or:[{status:"A",qty:{$lt:30}}]})

**Using AND , Or Operators**

db.inventory.find({status:"A",$or:[{qty:20},{item:/^p/}]

Hint : in mongoDb we can use Regular for LIKE Keyword item LIKE ‘p%’

**Query An Array**

db.inventory.insertMany([

{ item: "journal", qty: 25, tags: ["blank", "red"], dim\_cm: [ 14, 21 ] },

{ item: "notebook", qty: 50, tags: ["red", "blank"], dim\_cm: [ 14, 21 ] },

{ item: "paper", qty: 100, tags: ["red", "blank", "plain"], dim\_cm: [ 14, 21 ] },

{ item: "planner", qty: 75, tags: ["blank", "red"], dim\_cm: [ 22.85, 30 ] },

{ item: "postcard", qty: 45, tags: ["blue"], dim\_cm: [ 10, 15.25 ] }

]);

db.inventory1.find({tags:"red"})

Displays the Documenst which is having red in the tags array

db.inventory1.find({tags:["red","blank"]})

Displays the Documenst which is having red , blank in the tags array in the same order

db.inventory1.find({tags:{$all:["red","blank"]}})

Displays the Documenst which having red , blank in the tags array irrespective of order

db.inventory1.find({dim\_cm:{$gt:25}})

Displays the Documenst which having dim\_cm array atleast one element >25

db.inventory1.find({dim\_cm:{$gt:15,$lt:20}})

one element can satisfy first , one can satisfy second , others and all will be displayed

db.inventory1.find({dim\_cm:{elementMatch:{$gt:22,$lt:30}}})

Display the Documents where at least one element matches all the criteria

db.inventory1.find({tags:{$size:3}})

displays the document only when the size of the array is 3

**Querying an Array of Embed Docs**

db.inventory.insertMany( [

{ item: "journal", instock: [ { warehouse: "A", qty: 5 }, { warehouse: "C", qty: 15 } ] },

{ item: "notebook", instock: [ { warehouse: "C", qty: 5 } ] },

{ item: "paper", instock: [ { warehouse: "A", qty: 60 }, { warehouse: "B", qty: 15 } ] },

{ item: "planner", instock: [ { warehouse: "A", qty: 40 }, { warehouse: "B", qty: 5 } ] },

{ item: "postcard", instock: [ { warehouse: "B", qty: 15 }, { warehouse: "C", qty: 35 } ] }

]);

db.inventoryed.find({instock:{warehouse:"A", qty:5}})

Display the docs having instock subdoc with the specified conditions

db.inventoryed.find({'instock.qty':{$gt:25}})

apply condition on field of a subdoc

db.inventoryed.find({'instock.0.qty':{$gt:25}})

checking the Array 0th postion quantity gt 25

db.inventoryed.find( { "instock.qty": { $gt: 10, $lte: 20 } } )

Multiple conditions on array of subdocs

db.inventory.find( { item: **null** } )

checking the item is null

**Update Documents**

db.inventory.insertMany( [

{ item: "canvas", qty: 100, size: { h: 28, w: 35.5, uom: "cm" }, status: "A" },

{ item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },

{ item: "mat", qty: 85, size: { h: 27.9, w: 35.5, uom: "cm" }, status: "A" },

{ item: "mousepad", qty: 25, size: { h: 19, w: 22.85, uom: "cm" }, status: "P" },

{ item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "P" },

{ item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },

{ item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },

{ item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" },

{ item: "sketchbook", qty: 80, size: { h: 14, w: 21, uom: "cm" }, status: "A" },

{ item: "sketch pad", qty: 95, size: { h: 22.85, w: 30.5, uom: "cm" }, status: "A" }

] );

updates the first doc

db.inventory.update({item:"canvas"},{$set:{status:"p"}})

Updates all the docs which match the criteria

db.inventory.updateMany(

{ "qty": { $lt: 50 } },

{

$set: { "size.uom": "in", status: "P" },

$currentDate: { lastModified: true }

}

)

**Array Updations**

https://docs.mongodb.com/manual/reference/operator/update/

$push - Adds an item to an array

$addToSet - Adds elements to an array only if they do not already exist in the set

$pop - Removes the first or last item of an array

$pull - Removes all array elements that match a specified query.

$pullAll

**Deleting Documents**

db.inventory.deleteMany({})

db.inventory.deleteMany({ status : "A" })

db.inventory.deleteOne( { status: "D" } )