**Using Demo DB, animals Collections**

Create demo db with data, copy-paste from the following.   
use demo  
//Enter these documents into the collection 'animals'

db.animals.save({ \_id: 1, name: 'cat', tags:['land','cute'] ,info:{type:'mammal', color:'red' }})

db.animals.save({ \_id: 2, name: 'rabbit', tags:['land','cute'],info:{type:'mammal', color:'white' } })  
db.animals.save({ \_id: 3, name: 'shark', tags:['ocean'] ,info:{type:'fish', color:'grey' }})

db.animals.save({ \_id: 4, name: 'dolphin', tags:['ocean','cute'] ,info:{type:'mammal', color:'grey', canFly: null }})

db.animals.save({ \_id: 5, name: 'penguin', tags:['land','ocean','cute'] ,info:{type:'bird', canFly:false }})

db.animals.save({ \_id: 6, name: 'duck', tags:['land','cute'], info: {type:'bird', canFly: true} })

1. Create new document with \_id = 1

{ "\_id" : 1 }

1. Save again document with \_id = 1 (using save/insert Command)

WriteResult({

"nInserted" : 0,

"writeError" : {

"code" : 11000,

"errmsg" : "E11000 duplicate key error collection: demo.animals index: \_id\_ dup key: { \_id: 1.0 }"

1. Add a field: name to the document you just created, and assign its

{ "\_id" : 1, "name" : "lion" }

1. value of lion. fetch the document. (findAndReplace/update)  
   { "\_id" : 1, "name" : "lion" }
2. Add a field: Age to the document you just created, and assign its value of 30. fetch the document. (findAndReplace)  
   { "\_id" : 1, "name" : "lion", "Age" : 30 }
3. Increment the Age by 2, fetch the result  
   { "\_id" : 1, "name" : "lion", "Age" : 32 }
4. Decrement the age by 4, fetch the result  
   { "\_id" : 1, "name" : "lion", "Age" : 28 }
5. Multiple the Age value by 10.  
   { "\_id" : 1, "name" : "lion", "Age" : 280 }
6. Add 2 to the Age field, only for documents who's Age value is lessthan 38. Run this update multiple times till the nModified will be 0. Show the document.

???

1. For document : \_id :1 , Add “cute” to tags **array**. Show the document  
   { "\_id" : 1, "Age" : 20, "name" : "lion", "tags" : [ "cute" ] }
2. Add “great” to the tags array.  
   { "\_id" : 1, "Age" : 20, "name" : "lion", "tags" : [ "cute", "Great" ] }
3. Remove the Age field. Using the unset Command.

{ "\_id" : 1, "name" : "lion", "tags" : [ "cute", "Great" ] }

1. Add multiple values into tags Array: King, Predator (addToSet)

{ "\_id" : 1, "name" : "lion", "tags" : [ "cute", "Great", "King", "Predator" ] }

1. Same as above, using push command to insert in position no.2

{ "\_id" : 1, "name" : "lion", "tags" : [ "cute", "Great", "King", "Predator" ] }

???

1. Run the above again. Does the value insert again?   
   Check the matchedCount and ModifiedCount

????

1. Replace the addToSet with Push command.
2. Remove from the tags array some values – using pull and in Command.

{ "\_id" : 1, "name" : "lion", "tags" : [ "cute", "Predator" ] }

1. Replace the Cute item in tags Array with “Not Cute”

{ "\_id" : 1, "name" : "lion", "tags" : [ " not cute", "Predator" ] }

Pop all the tags items – one by one   
{ "\_id" : 1, "name" : "lion", "tags" : [ "Predator" ] }

{ "\_id" : 1, "name" : "lion", "tags" : [ ] }

1. Write loop to add 10 new documents with id ranging from 10 to 20.   
   name field = “No name”. You can use a javascript for-loop for this.

20

1. Add a field named “info.color” to all documents with id between 10 and 20 . Set the field value to grey.

{ "\_id" : 1, "name" : "lion", "tags" : [ ] }

{ "\_id" : 2, "name" : "rabbit", "tags" : [ "land", "cute" ], "info" : { "type" : "mammal", "color" : "white" } }

{ "\_id" : 3, "name" : "shark", "tags" : [ "ocean" ], "info" : { "type" : "fish", "color" : "grey" } }

{ "\_id" : 4, "name" : "dolphin", "tags" : [ "ocean", "cute" ], "info" : { "type" : "mammal", "color" : "grey", "canFly" : null } }

{ "\_id" : 5, "name" : "penguin", "tags" : [ "land", "ocean", "cute" ], "info" : { "type" : "bird", "canFly" : false } }

{ "\_id" : 6, "name" : "duck", "tags" : [ "land", "cute" ], "info" : { "type" : "bird", "canFly" : true } }

{ "\_id" : 10, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 11, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 12, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 13, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 14, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 15, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 16, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 17, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 18, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 19, "name" : "no name", "Info" : { "color" : "grey" } }

{ "\_id" : 20, "name" : "no name", "Info" : { "color" : "grey" } }

1. For Doc No 4, remove the first item from tags Array($pop)

{ "\_id" : 4, "name" : "dolphin", "tags" : [ "cute" ], "info" : { "type" : "mammal", "color" : "grey", "canFly" : null } }

1. Remove the tags array from document with id 4 .

{ "\_id" : 4, "name" : "dolphin", "info" : { "type" : "mammal", "color" : "grey", "canFly" : null } }

1. Delete docs with \_id between 10 and 20

{ "\_id" : 1, "name" : "lion", "tags" : [ ] }

{ "\_id" : 2, "name" : "rabbit", "tags" : [ "land", "cute" ], "info" : { "type" : "mammal", "color" : "white" } }

{ "\_id" : 3, "name" : "shark", "tags" : [ "ocean" ], "info" : { "type" : "fish", "color" : "grey" } }

{ "\_id" : 4, "name" : "dolphin", "info" : { "type" : "mammal", "color" : "grey", "canFly" : null } }

{ "\_id" : 5, "name" : "penguin", "tags" : [ "land", "ocean", "cute" ], "info" : { "type" : "bird", "canFly" : false } }

{ "\_id" : 6, "name" : "duck", "tags" : [ "land", "cute" ], "info" : { "type" : "bird", "canFly" : true } }

1. For each of the docs add item: animal int the tag Array. Use UpdateMany with push commands

{ "\_id" : 1, "name" : "lion", "tags" : [ "animals" ] }

{ "\_id" : 2, "name" : "rabbit", "tags" : [ "land", "cute", "animals" ], "info" : { "type" : "mammal", "color" : "white" } }

{ "\_id" : 3, "name" : "shark", "tags" : [ "ocean", "animals" ], "info" : { "type" : "fish", "color" : "grey" } }

{ "\_id" : 4, "name" : "dolphin", "info" : { "type" : "mammal", "color" : "grey", "canFly" : null }, "tags" : [ "animals" ] }

{ "\_id" : 5, "name" : "penguin", "tags" : [ "land", "ocean", "cute", "animals" ], "info" : { "type" : "bird", "canFly" : false } }

{ "\_id" : 6, "name" : "duck", "tags" : [ "land", "cute", "animals" ], "info" : { "type" : "bird", "canFly" : true } }

>

1. Use ReplaceOne to replace the lion doc with {name: lioness, tags array with one item: “animal”, info as a sub-document with color=grey, and canFly=false, keep the same id.

{ "\_id" : 1, "name" : "lioness", "tags" : [ "animal" ], "info" : { "color" : "grey", "canFly" : true } }

1. Use UpdateOne to update doc id =15 with name : elephant , tags array : “big” info as a sub document with color = grey.In case the doc is not exists insert new with one with the items:   
   name : “elep” , tags :[“upsert value”]

{ "\_id" : 15, "name" : "elephant" }

1. Using one command bulkWrite - insert new doc, delete one doc, and update one doc

db.animals.bulkWrite( [ {"updateOne" : {"filter" : {"\_id" : 2} ,"update" :{"name" : {$set: "dfgdfgdfgdfg" } } } } , {"deleteOne": {"filter" : {"\_id" : 1}}} ] , {"ordered" : true} )