**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

db. animals.save({\_id:1})

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

db.animals.save({\_id : 1})

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

value of lion. fetch the document. (findAndReplace/update)

db.animals.updateOne({\_id : 1 } , {$set : { name : "lion"}})

db. animals.findAndModify({query:{\_id:1}, update:{$set:{Name:”lion”}}}

1. Add a field: Age to the document you just created, and assign its value of 30. fetch the document. (findAndReplace)

db. animals.findAndModify({query:{\_id:1}, update:{$set:{Age:30}}})

db. animals.updateOne( {\_id:1}, {$set:{age:30} } )

1. Increment the Age by 2, fetch the result

db.animals.updateOne ( {\_id:1} , { $inc : {Age :2} })

1. Decrement the age by 4, fetch the result

db.animals.updateOne ( {\_id:1} , { $inc : {Age :-4} }

1. Multiple the Age value by 10.

db.animals.updateOne( {Age:{$lt : 38} }, { $inc : {Age :2} })

1. Add 2 to the Age field, only for documents who's Age value is less than 38. Run this update multiple times till the nModified will be 0. Show the document.

db.animals.update({\_id : 1 } , {$mul :{"age" : 10}} )

1. For document : \_id :1 , Add “cute” to tags **array**. Show the document

db. animals.updateOne( {\_id:1}, {$set:{tags: ["cute"] }} )

1. Add “great” to the tags array.

db.animals.updateOne( {\_id:1}, {$addToSet : {tags :"Great"} })

1. Remove the Age field. Using the unset Command.

db.animals.updateOne({\_id : 1} , {$unset :{"age" : ""}} )

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

db.animals.updateOne( {\_id:1}, {$addToSet : { $each : [“King” , “Predator” ] }

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

db.animals.updateOne({\_id:1},{$addToSet : {"tags":{$each : ["a", "b”] , $position : 2} }})

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

db.animals.updateOne( {\_id:1}, {$addToSet : { $each : [“King” , “Predator” ] } })

1. Replace the addToSet with Push command.

db.animals.updateOne( {\_id:1}, {$push : { $each : [“King” , “Predator” ] } }) db.animals.updateOne( {\_id:1}, {$pull: { tags :{$in : ["Great" , "King" ]} }} )

1. Remove from the tags array some values – using pull and in Command.

db.animals.updateOne( {\_id:1}, {$pull: { tags :{$in : ["Great" , "King" ]} }} )

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

db.animals.updateMany( {\_id:1 , tags : "cute"}, { $set: { "tags.$" : "not Cute"}})

1. Pop all the tags items – one by one

db.animals.updateMany( {\_id:1}, { $pop: {tags : -1} })

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

db.animals.updateMany({\_id:{$gte:10 , $lte:20}} , {$set: {"info.color" :"grey"}} )

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

db.animals.updateOne ({\_id:{$eq :4 } } , {$pop : {"tags" : -1}})

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

db.animals.updateOne ({\_id:{$eq :4 } } , {$unset : {"tags" : 1}})

1. Delete docs with \_id between 10 and 20

db.animals.deleteMany({\_id: {$gte:10 , $lte:20}})

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

db.animals.updateMany({} , {$push : {tags: "animal" }} , )

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.

db.animals.replaceOne({\_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”]

db.animals.updateOne( {\_id : } , {$set : {"name" : }} , {"upsert":true } )

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} )