23. Write a MongoDB query to find the restaurant Id, name and grades for those restaurants

where the 2nd element of grades array contains a grade of "A" and score 9 on an ISODate

"2014-08-11T00:00:00Z".

db.restaurent.find({‘grades.2.grade’:’A’,’grades.2.score’:9, ’grades.2.date’: ISODate ("2014-08-11T00:00:00Z")

})

update and index

update ----- you can update many documents or you can update 1 document

updateOne ------ you can update only first matching document

updateMany-------You can update all matching document

-------update function

1. if you want add new key value pair
2. delete existing key value pair
3. rename key name
4. overwrite the value of existing key
5. increment or decrement the value key.
6. multiplication the existing value of a key
7. min and max replace the value
8. replace entire document
9. adding or deleting element from the array

db.collection.update(<query>,<update>,<options>)

db.collection.updateOne(<query>,<update>)

db.collection.updateMany(<query>,<update>)

$set,$inc,$mul,$min,$max,$currentDate,$unset

------increase the price by 100 for all movies with name starts with kahani

db.movie.update({ name:/^kahani/},

{$inc:{price:100}},

{multi:true})

or

db.movie.updateMany({ name:/^kahani/},

{$inc:{price:100}})

------ increase the price by 100 for first matching movie with name starts with kahani

db.movie.update({ name:/^kahani/},

{$inc:{price:100}})

or

db.movie.updateOne({ name:/^kahani/},

{$inc:{price:100}})

-------increase ticketnumber by 20 for all movies with rating > 4

db.movie.update({rating:{$gt:4}},

{$inc:{tickect\_no:20}},

{multi:true})

-------decrease ticketnumber by 20 and increase price by 50 for all movies with rating > 4

db.movie.update({rating:{$gt:4}},

{$inc:{tickect\_no:-20,price:50}},

{multi:true})

--------overwrite the rating by 4 and increase price by 30 for all movies with price>300

db.movie.update({price:{$gt:300}},

{$set:{rating:4},$inc:{price:30}},

{multi:true})

--------increase the price by 20% for all the movies

db.movie.update({},{$mul:{price:1.20}},{multi:true})

--------to set the rating to 3 if current rating is > 3 for all movies with price<400

rating 2 3 ------- will not get replaced

retiring 4 3 ------ will get replaced

db.movie.update({price:{$lt:400}},{$min:{rating :3}},{multi:true})

-------to set the rating to 3 if current rating is < 3 for all movies with price<400

rating 2 3--------replaced by 3

rating 4 3 -------will not get replaced

db.movie.update({price:{$lt:400}},{$max:{rating :3,price:200}},{multi:true})

---------delete a key value pair ----- $unset

--------to remove key tickect\_no from all document

db.movie.update({},{$unset:{tickect\_no:’’ }},{multi:true})

------- to increase the price of padmavat movie by 20 and also want to store data on which modification has happened

db.movie.update({name:’padmavat’},{$inc:{price:20},$currentDate:{‘modified’:true}},{multi:true})

lastchanged:{

reason:’public demand’,

modified: ISODate(‘2021-07-17T005:10:05z’)

}

db.movie.update({name:'padmavat'},

{$inc:{price:20},$currentDate:{'lastchange.modified':true},$set:{'lastchange.reason':'public demand'}},{multi:true})

--------to decrease ticket\_no by 10 and increase price by 10% for all movies with rating > 4

ticketno+ (-10)

price\*1.10

db.movie.update({rating:{$gt:4}},{$inc:{ticket\_no:-10},$mul:{price:1.10}},{multi:true})

------to overwrite rating by 4 also assign today’s date to lastmodified

for all movies with price is = either 350 or 260 or 500

or movie name has n anywhere.

price:{$in:[350,260,500]}

name:/n/

$or:[{ price:{$in:[350,260,500]}},{ name:/n/}]

db.movie.update({$or:[{ price:{$in:[350,260,500]}},{ name:/n/}]},

{$set:{rating:4},$currentDate:{lastmodified:true}},{multi:true})

------------ arrays

$,$[],$[item],$push,$pull,$pop, $each, $position

--------- to add mohit raina in uri movie

db.movie.update({name:'uri'},{$push:{actor:'mohit raina'}},{multi:true})

--------to add 'kirti kilhari','rajit kapur','akash deep' in movie uri

db.movie.update({name:'uri'},{$push:{actor:{$each:['kirti kilhari','rajit kapur','akash deep']}}},{multi:true})

------ add r madhavan and sharman joshi in 3 idiot

db.movie.update({name:’3 idiots’},

{$push:{actor:{$each:[‘r madhavan’,’sharman joshi’],$position:2}}},

{multi:true})

-------delete data from array

$pop ------ deletes the values from the end of the array

-1 ----- deletes from the beginning

1 -----deletes from the end

db.movie.update({name:'3 idiots'},{$pop:{actor:-1}},{multi:true})

------- to delete the known values

-------to delete zzz from chichore

db.movie.update({name:'chichore'},{$pull:{actor:'zzz'}},{multi:true})

------- to delete multiple values use $in with $pull

---------to delete actors ‘aaaa’ and ‘bbbb’ from actor

> db.movie.update({name:'kahani'},{$pull:{actor:{$in:['aaaa','bbbb']}}},{multi:true})

$,$[],$[elem]

-------to update all means of matched documents

db.student.update({\_id:6},{$inc:{'grades.$[].mean':3}},{multi:true})

---- to update matched documents, first matched value in the array are matched

db.student.update({\_id:6,'grades.grade':{$gt:88}},{$inc:{'grades.$.mean':3}},{multi:true})

---- to update documents which are matched

db.student.update({\_id:6 },{$inc:{'grades.$[elem].mean':3}},

{multi:true,

arrayFilters:[{‘elem.grade’:{$gt:88}}]

})

------to increment all prices of all products with name eatable by 10

{

"\_id" : ObjectId("6068412624ef6361fb602089"),

"id" : 13,

"name" : "eatable",

"items" : [

{

"category" : "chips",

"price" : 50

},

{

"category" : "chocolate",

"price" : 50

},

{

"category" : "biscuits",

"price" : 50

}

]

}

db.product.update({name:’eatable’},{$inc:{items.$[].price:10}},{multi:true})

-------to update price of all items with category chocolate

db.product.update({'items.category':'chocolate'},{$inc:{'items.$.price':10}},{multi:true})

$rename to rename the key

db.movie.update({},{$rename:{oldkey: new key,oldkey:new key}},{multi:true})

------overwrite entore document of movie padmavat by new document

db.movie.replaceOne({name:’padmavat’},{name:’xxx’,price:260,ticket\_no:456})

----index in mongodb

------to create a new index

ensureIndex

createIndex

db.movie.createIndex({name:1})

db.movie.createIndex({name:1,price:-1})

---to see list of all indexes on a collection

db.movie.getIndexes()

-------to delete indexes by name

db.movie.dropIndex(‘name\_1\_price\_-1’)