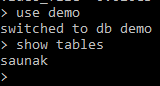
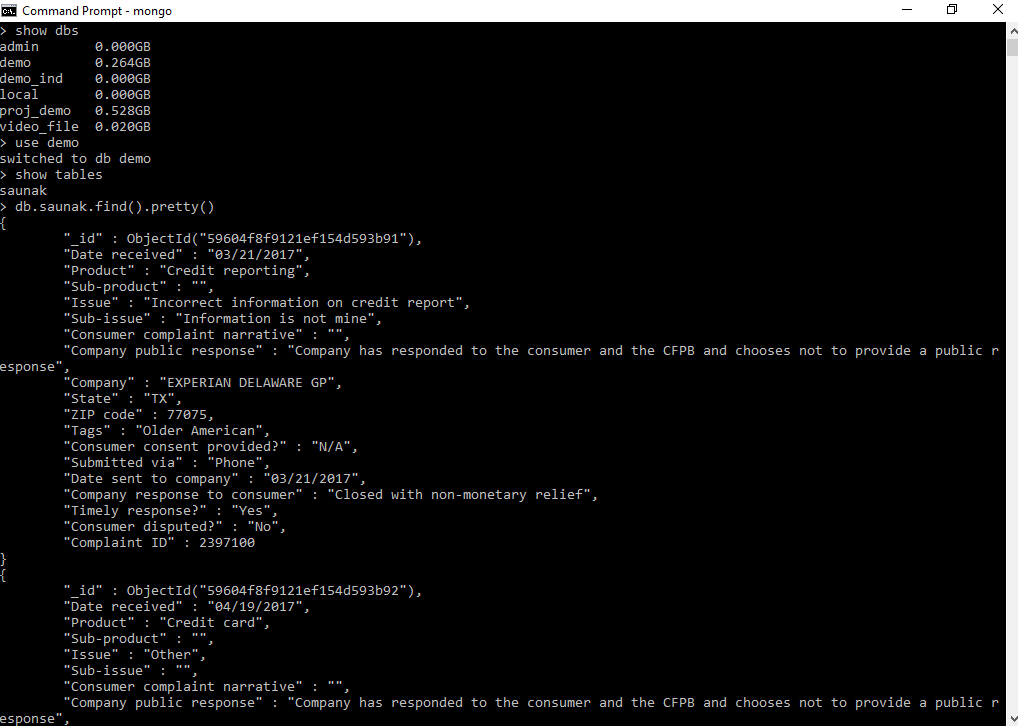


Checking what are the database available.



Now we are selecting a particular database and checking what are the tables present inside it,here find a table named “saunak”

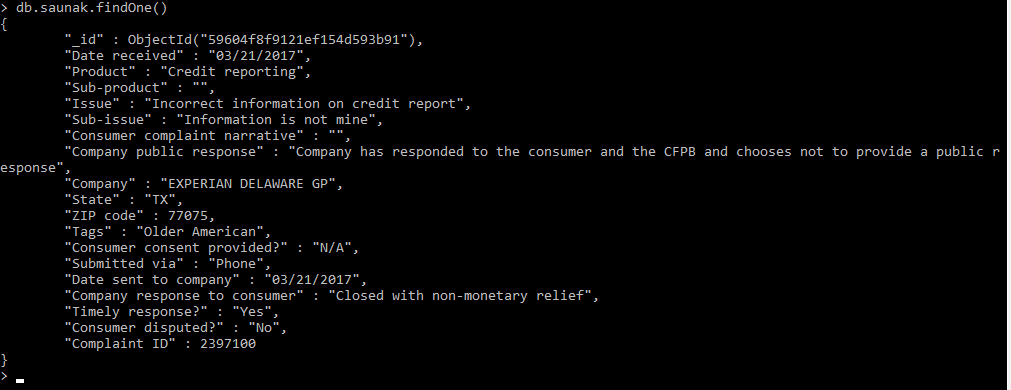
>>db.saunak.find().pretty()



We can see that these are the data that are available.

Now we will do some operations on this data and get various results.

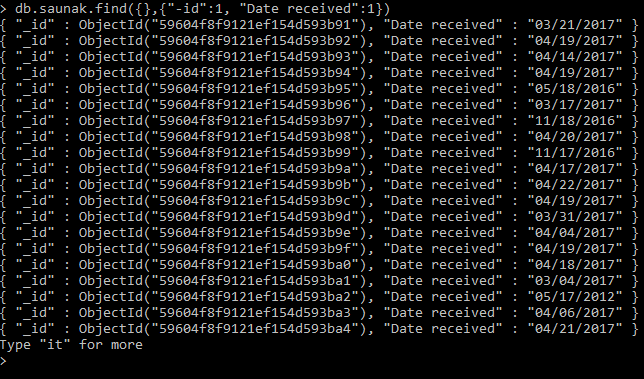
>>db.saunak.findOne()



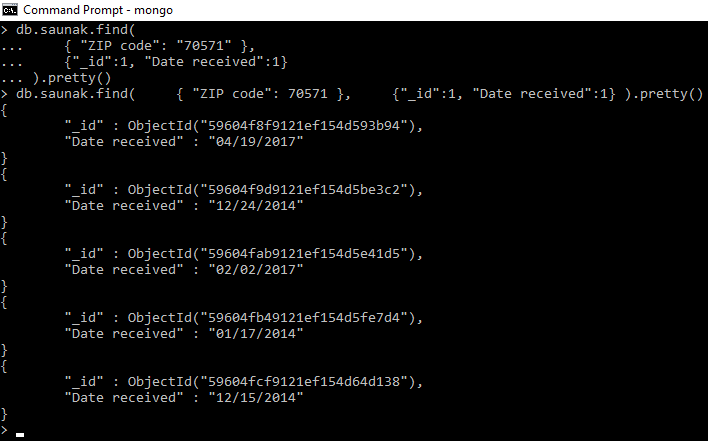
“findOne()” is used only to show only one row of the collection.

>>Projection and Condition

>>db.saunak.find({},{"\_id":1, "Date received":1})

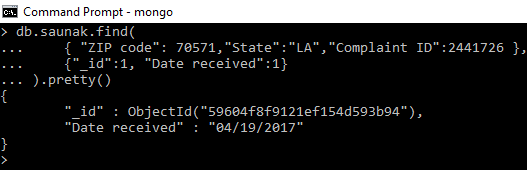


db.saunak.find( { "ZIP code": 70571 }, {"\_id":1, "Date received":1} ).pretty()



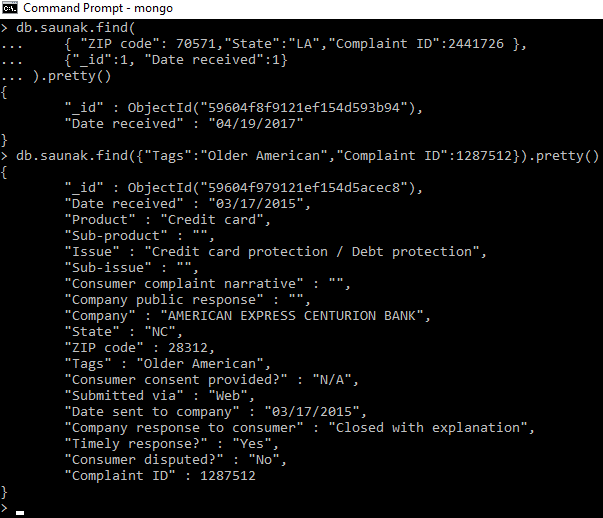
db.saunak.find({ "ZIP code": 70571,"State":"LA","Complaint ID":2441726 },

{"\_id":1, "Date received":1} ).pretty()



>>Select with where & and condition

>db.saunak.find({"Tags":"Older American","Complaint ID":1287512}).pretty()



>>Select with where & or condition

db.saunak.find(

{

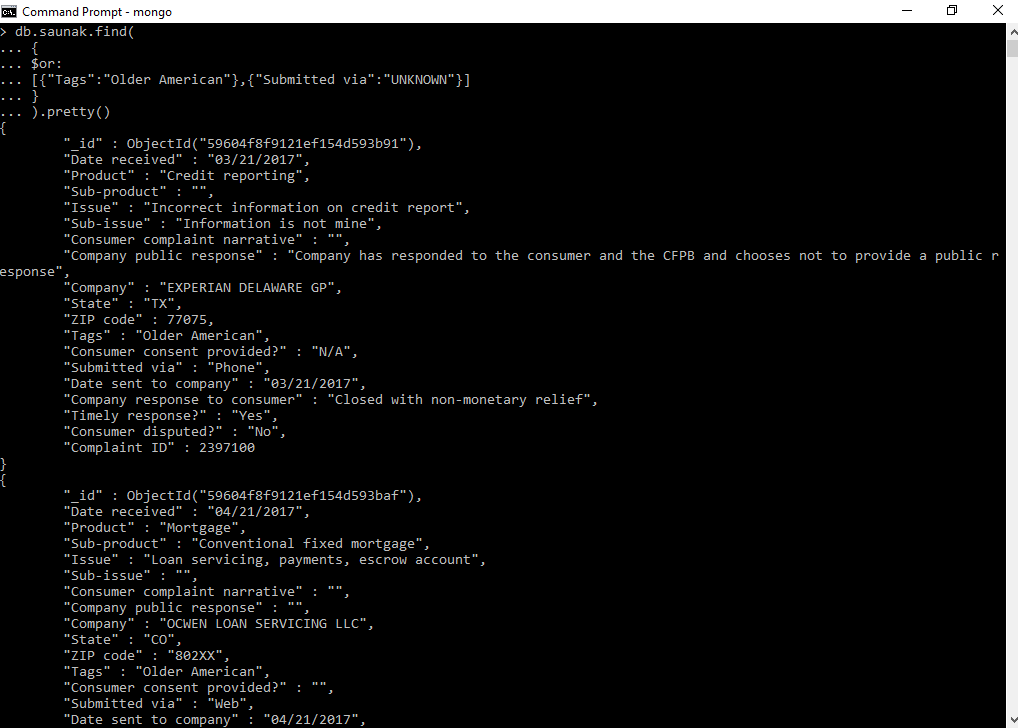
$or:

[{"Tags":"Older American"},{"Submitted via":"UNKNOWN"}]

}

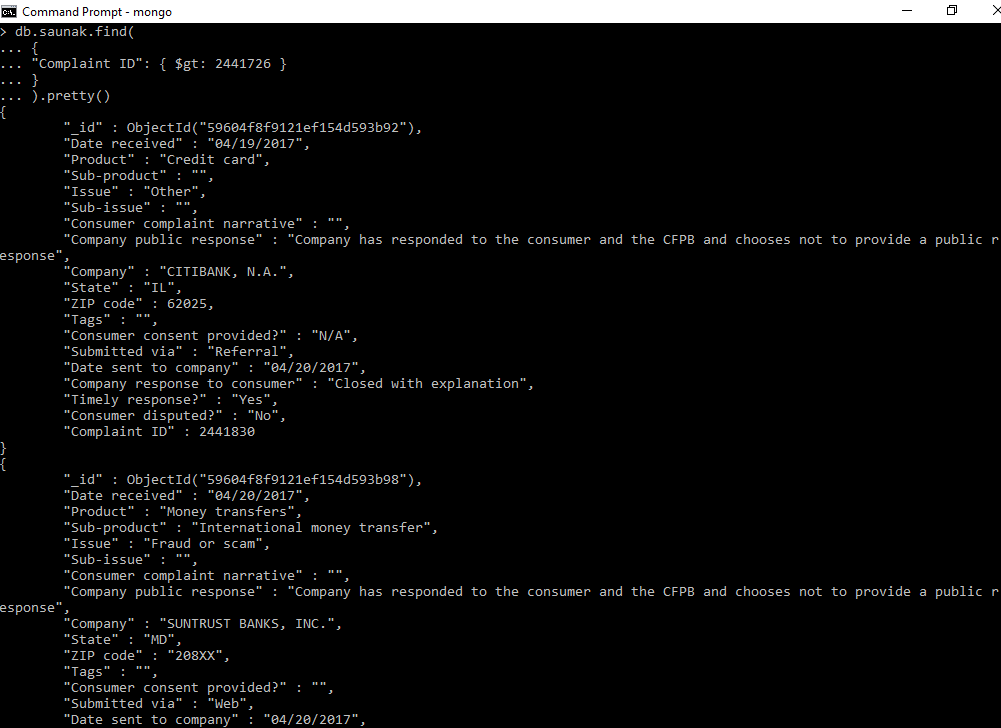
).pretty()

(Note:- submitted via – doesn’t have anyone named as “UNKONWN”)

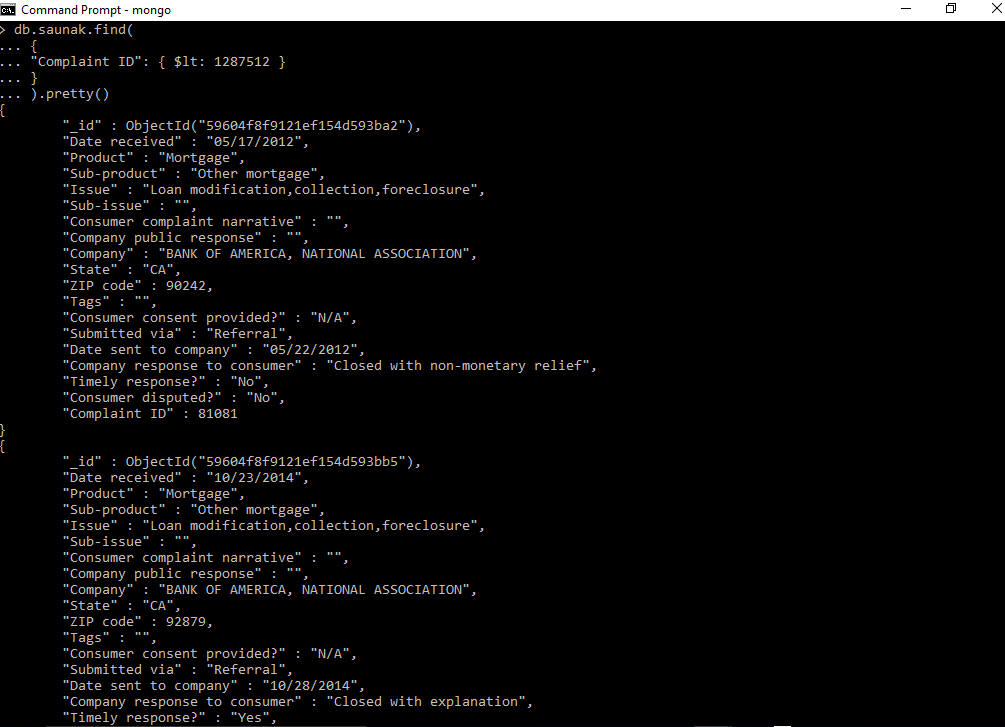


>>Select with Greater Than or Less Than condition

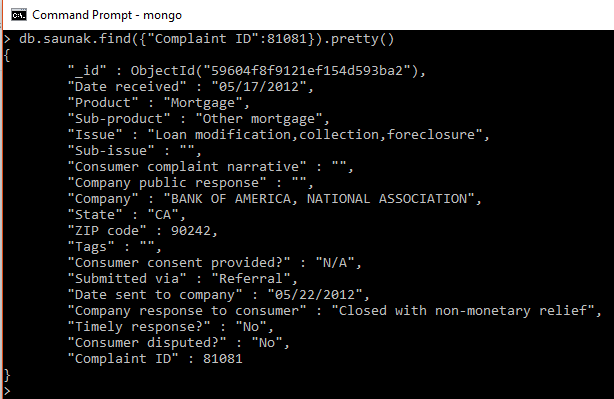
db.saunak.find( {"Complaint ID": { $gt: 2441726 }}).pretty()



db.saunak.find({"Complaint ID": { $lt: 1287512}}).pretty()

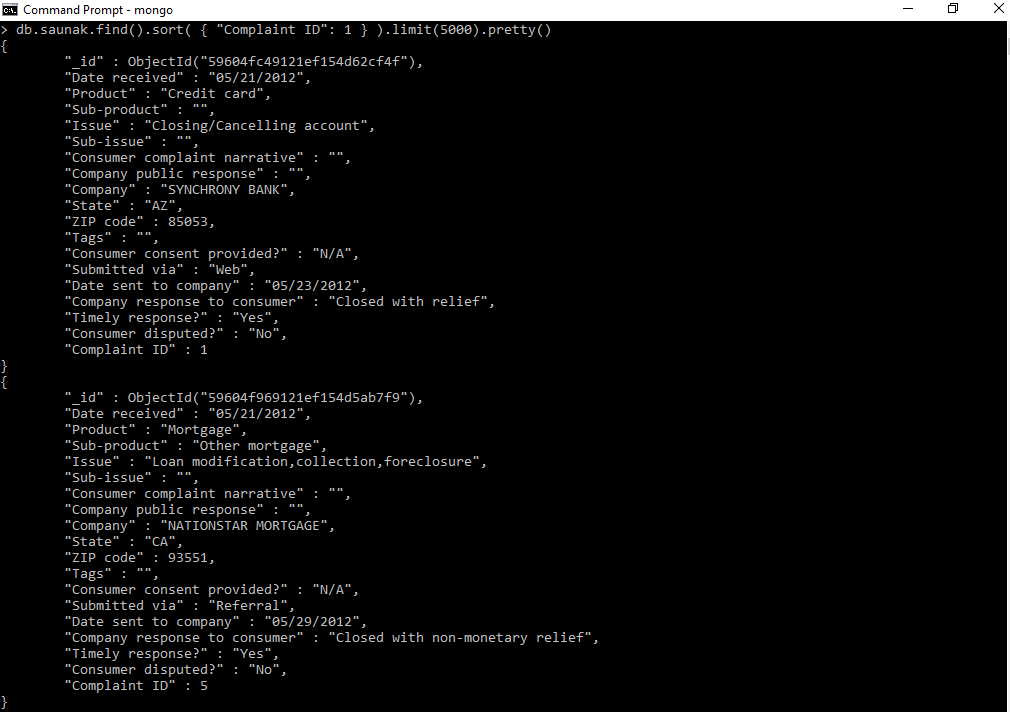


db.saunak.find({"Complaint ID":81081}).pretty()

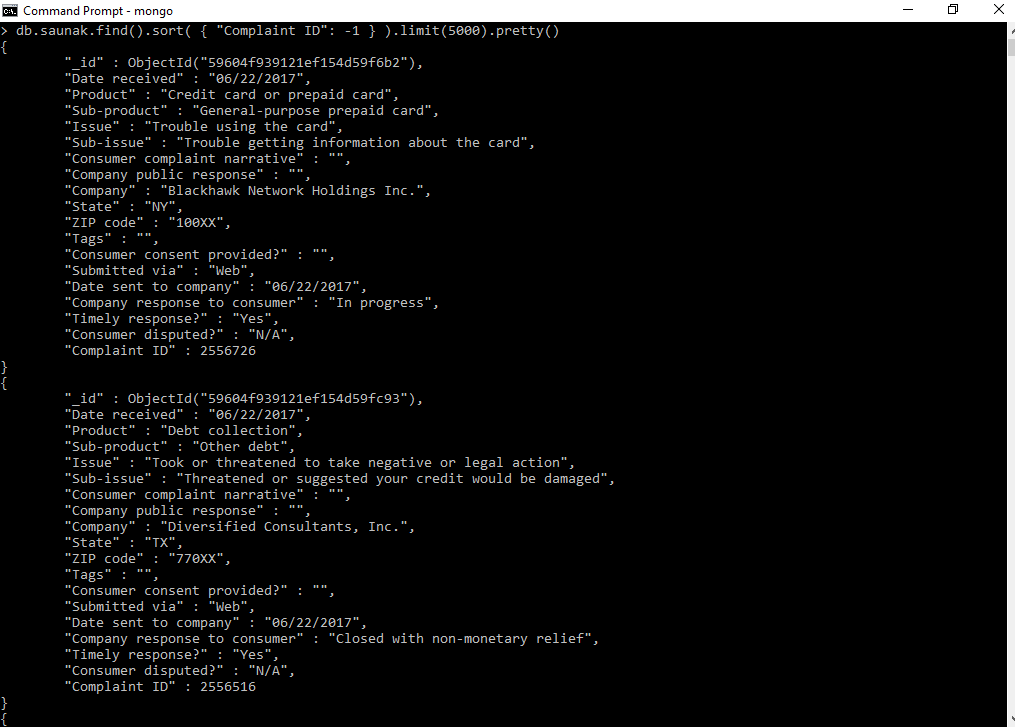


>>Sorting Documents

db.saunak.find().sort( { "Complaint ID": 1 } ).limit(5000).pretty()

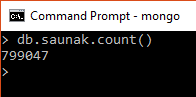


db.saunak.find().sort( { "Complaint ID": -1 } ).limit(5000).pretty()

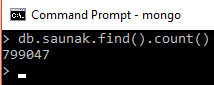


>>Counting Documents

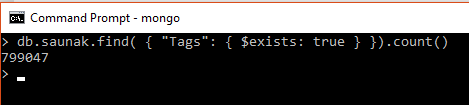
>db.saunak.count()



>db.saunak.find().count()

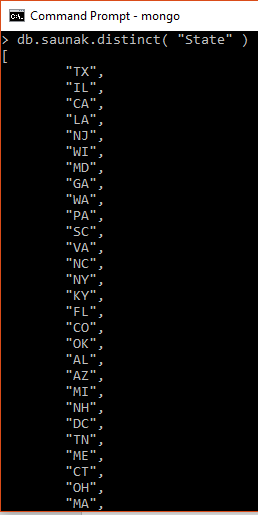


db.saunak.find( { "Tags": { $exists: true } }).count()



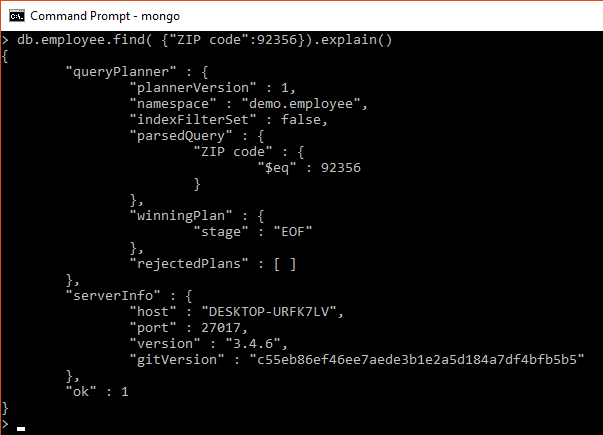
>>Selecting Distinct Documents

>db.saunak.distinct( "State" )



>>Explain about query

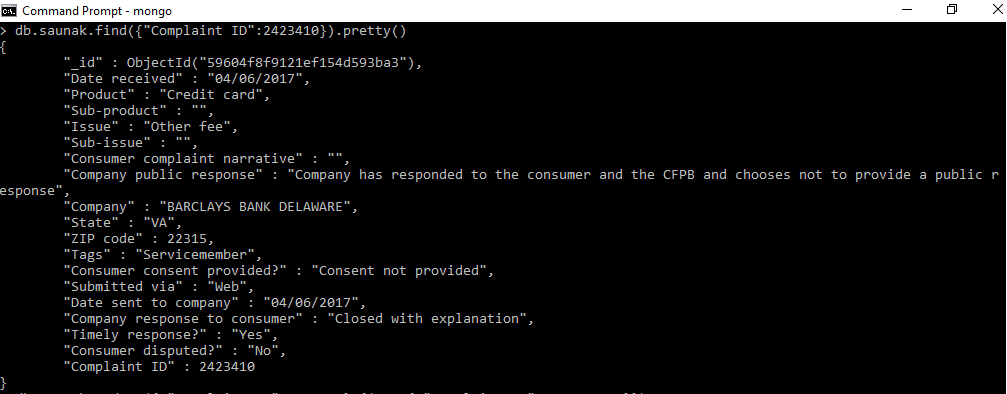
>db.employee.find( {"ZIP code":92356}).explain()



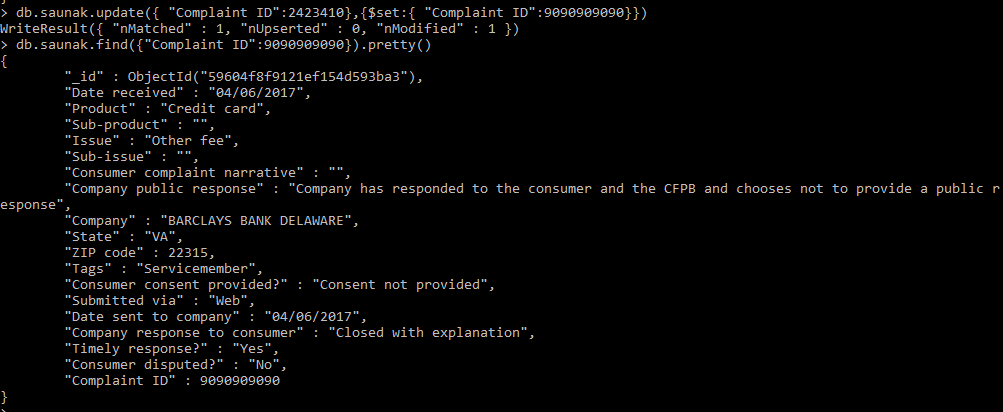
>>Modify Documents

db.saunak.update({ "Complaint ID":2423410},{$set:{ "Complaint ID":9090909090}})

BEFORE>>

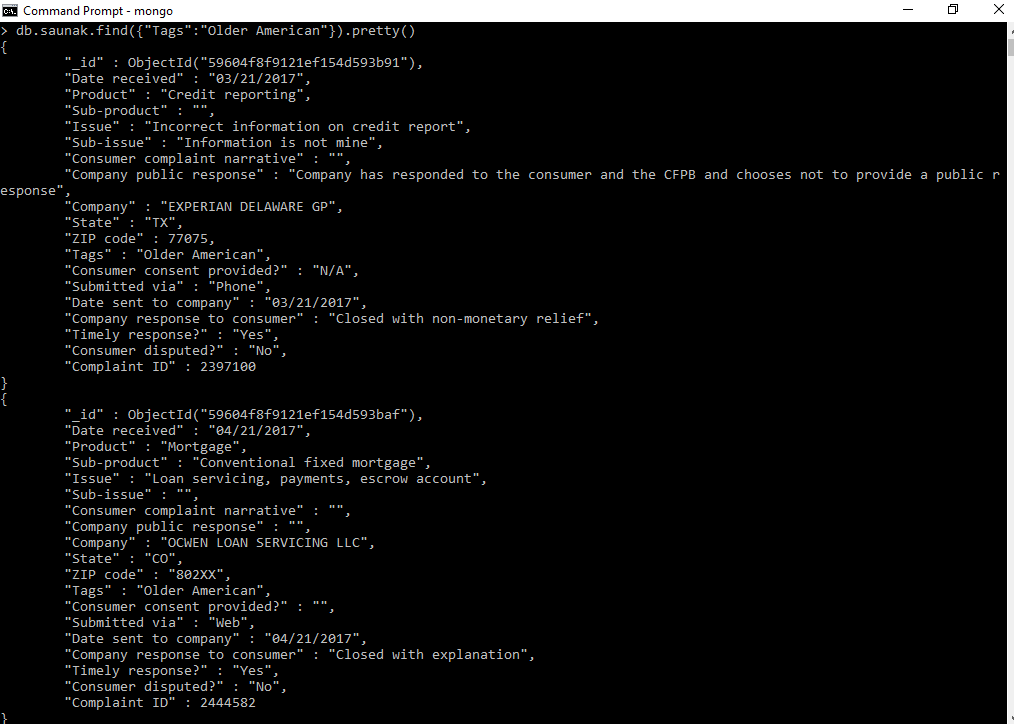


AFTER>>

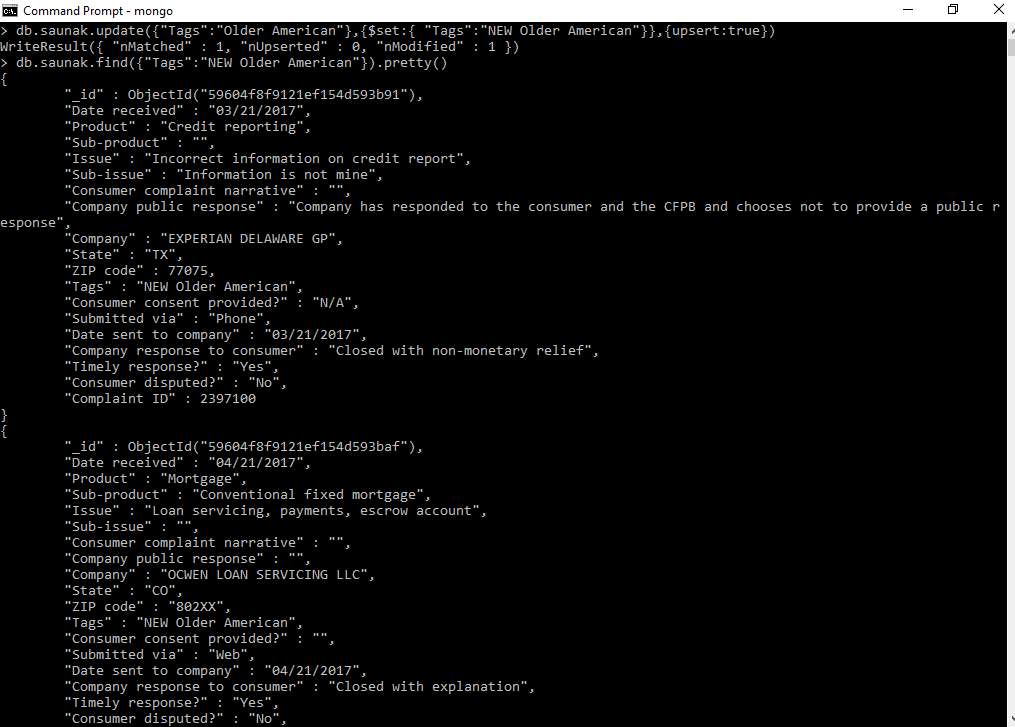


>db.saunak.update({"Tags":"Older American"},{$set:{ "Tags":"NEW Older American"}},{upsert:true})

BEFORE>>



AFTER>>



>>Counter Collection Implementation

db.counters.insert({\_id: "userid",seq: 0})

function getNextSequence(name) {var ret = db.counters.findAndModify(

{query: { \_id: name },update: { $inc: { seq: 1 } },new: true});return ret.seq;}

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "Amit"

}

);

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "Akshay"

}

);

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "Anant"

}

);

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "Das"

}

);

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "harendra"

}

);

db.users.insert(

{

\_id: getNextSequence("userid"),

name: "Kiran"

}

);

db.users.insert(

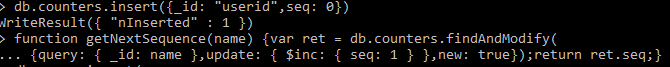
{

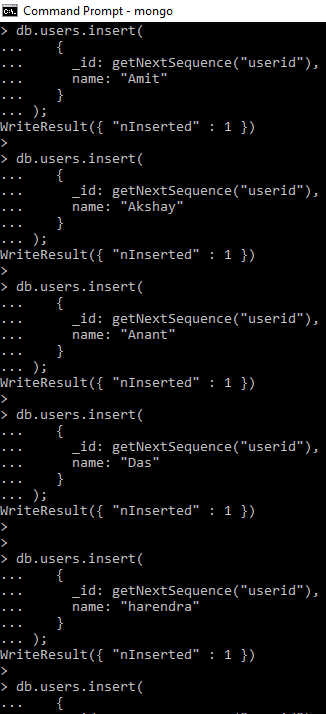
\_id: getNextSequence("userid"),

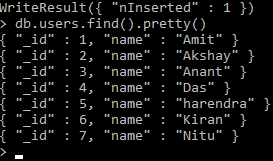
name: "Nitu"

}

);







>>Map Reduce

