**Mongodb Readme**

Step0: pre-pocessing in the Posts csv file, copy the column “Tags” to NotePad++, Search—>Replace all Double quotes to nothing, and use the data to replace the original Tags column.

Step1: create database 5338group, create four collections: posts, users, tags, and votes in Robo3T, then using

./mongoimport -d 5338group -c collection name --type csv --file /file store path/ --headerline to import data from corresponding csv file which is downloaded from elearning system and is pre-processed in Step 0.

Step2: modify the CreationDate format to DATE type in posts and votes collections

db.posts.find().forEach(function(doc){ doc.CreationDate = new Date(doc.CreationDate); db.CreationDate.save(doc) });

db.votes.find().forEach(function(doc){ doc.CreationDate = new Date(doc.CreationDate); db.CreationDate.save(doc) });

Step3: By using the aggregate pipeline, get questions collection and answers collection from the posts collection

db.getCollection('posts').aggregate([

{$match:{PostTypeId:1}},

{$project:{Id:1,AcceptedAnswerId:1,CreationDate:1,ViewCount:1, OwnerUserId:1,Title:1,Tags:1}},

{$out:"questions"}

])

db.getCollection('posts').aggregate([

{$match:{PostTypeId:2}},

{$project:{Id:1,CreationDate:1, OwnerUserId:1}},

{$out:"answers"}

])

Step4: get the acceptedanswer field in the questions collection by

db.getCollection('questions').aggregate([

{$lookup:{

from:"answers",

localField:"AcceptedAnswerId",

foreignField:"Id",

as:"acceptedanswer"

}},

{$out:"v1"}

])

Step5: get the answers field in the questions collection by

db.getCollection('questions').aggregate([

{$lookup:{

from:"answers",

localField:"Id",

foreignField:"ParentId",

as:"answers"

}},

{$out:"v2"}

])

Step 6: get unacceptedanswer field in questions collection and split tags by

db.getCollection('v2').aggregate([

{$project:{id:"$Id",AcceptedAnswerId:1,CreationDate:1,ViewCount:1,OwnerUserId:1,Title:1,Tags:{$split:[“$Tags”,”,”]},acceptedanswer:1,unacceptedanswer:{$setDifference:["$answers","$acceptedanswer"]}}},

{$out:"v3"}

Step 7: get acceptedvote and unacceptedvote fields in questions collection by

db.getCollection('v3').aggregate([

{$lookup:{

from:"votes",

localField:"acceptedanswer.Id",

foreignField:"PostId",

as:"acceptedvote"

} },

{$lookup:{

from:"votes",

localField:"unacceptedanswer.Id",

foreignField:"PostId",

as:"unacceptedvote"

}},

{$out:"v4"}

])

Step 8: get questionuser, accepteduser and unaccepteduser in questions collection by

db.getCollection('v4').aggregate([

{$lookup:{

from:"users",

localField:"OwnerUserId",

foreignField:"Id",

as:"questionuser"

}},

{$lookup:{

from:"users",

localField:"acceptedanswer.OwnerUserId",

foreignField:"Id",

as:"accepteduser"

}},

{$lookup:{

from:"users",

localField:"unacceptedanswer.OwnerUserId",

foreignField:"Id",

as:"unaccepteduser"

}},

{$out:"v5"}

])

Step 9: delete non-relevant fields by

db.getCollection('v5').aggregate([

{$project:{\_id:"$id",CreationDate:1,ViewCount:1,Title:1,"acceptedanswer.Id":1,"acceptedanswer.CreationDate":1,tags:1,"unacceptedanswer.Id":1,

"unacceptedanswer.CreationDate":1, "unacceptedanswer.OwnerUserId":1,

"acceptedvote.VoteTypeId":1,"acceptedvote.CreationDate":1,"unacceptedvote.VoteTypeId":1,"unacceptedvote.CreationDate":1,

"questionuser.Id":1,"questionuser.CreationDate":1,"questionuser.DisplayName":1,"questionuser.UpVotes":1,"questionuser.DownVotes":1,

"accepteduser.Id":1,"accepteduser.CreationDate":1,"accepteduser.DisplayName":1,"accepteduser.UpVotes":1,"accepteduser.DownVotes":1,

"unaccepteduser.Id":1,"unaccepteduser.CreationDate":1,"unaccepteduser.DisplayName":1,"unaccepteduser.UpVotes":1,"unaccepteduser.DownVotes":1}},

{$out:"v6"}

])

Step 10: adjust array type field to object field because the field should have one object only in logic

db.getCollection('v6').aggregate([

{$project:{\_id:1,CreationDate:1,ViewCount:1,Title:1,acceptedanswer:{$arrayElemAt:["$acceptedanswer",0]},tags:1,unacceptedanswer:1,acceptedvote:1,

unacceptedvote:1,questionuser:{$arrayElemAt:["$questionuser",0]},accepteduser:{$arrayElemAt:["$accepteduser",0]},unaccepteduser:1

}},

{$out:"v7"}

])

Step 11: get decisionday field in questions collection by

db.getCollection('v7').aggregate([

{$unwind:"$acceptedvote"},

{$match:{"acceptedvote.VoteTypeId":1}},

{$project:{decisionday:"$acceptedvote.CreationDate"}},

{$out:"v8"}

])

db.getCollection('v7').aggregate([

{$lookup:{

from:"v8",

localField:"\_id",

foreignField:"\_id",

as: "decisionDay"

}},

{$project:{CreationDate:1,ViewCount:1,Title:1,tags:1,unacceptedanswer:1,acceptedvote:1,unacceptedvote:1,unaccepteduser:1,acceptedanswer:1,questionuser:1,

accepteduser:1, decisionday:{$arrayElemAt:["$decisionDay.decisionday",0]}}},

{$out:"v9"}

])

Step 12: Create indexes

db.getCollection(‘v9’).createIndex({ViewCount:1})

db.getCollection(‘v9’).createIndex({CreationDate:1})

db.getCollection(‘v9’).createIndex({tags:”text”})

db.getCollection(‘v9’).createIndex({“acceptedanswer.CreationDate”:1})

db.getCollection(‘v9’).createIndex({“unacceptedanswer.CreationDate”:1})

Step 13: Keep ‘v9’, and delete all other collections in the database. And use the query script in “Target Query scripts” file, to run the queries and watch the results in Robo3T.