**PreProcessing Instructions**

1. **Changes in CSV data files- Before importing into MongoDB**
2. First we ensure that the date values in all the CSVs are in a common format “yyyy-mm-ddThh:mm:ss.000”.
3. The tags field in the posts csv was in the form “tag1,tag2,tag3”. We removed the quotes so that it would be easy for us to store the tags in an array form.
4. **Upload all the 4 CSV files using below commands**.
5. mongoimport -d dbm -c Posts --type csv --file C:\comp5338\mongodb\odqa\Posts.csv --headerline
6. mongoimport -d mydb -c Tags --type csv --file C:\comp5338\mongodb\odqa\Tags.csv –headerline
7. mongoimport -d mydb -c Users --type csv --file C:\comp5338\mongodb\odqa\Tags.csv –headerline
8. mongoimport -d mydb -c Votes --type csv --file C:\comp5338\mongodb\odqa\Votes.csv –headerline
9. **Changed the date format of creation date, Last Editor Date from post collection to ISO format using below query.**

db.posts.find().forEach(function(element){

element.CreationDate = new Date(element.CreationDate);

db.posts.save(element);

})

db.posts.find().forEach(function(element){

element.LastEditDate = new Date(element.LastEditDate);

db.posts.save(element);

})

1. **Adding Votes as an embeded Array to Posts collection:**

db.Posts.aggregate(

[{ $lookup:

{

from:"votes",

localField:"Id",

foreignField:"PostId",

as:"votes"

}},

{

$out:"PostsVotes"

}])

1. **Converting tags to embeded array type in Posts Collection:**

db.PostsVotes.aggregate(

[{"$addFields":

{ "tags":

{ "$split":

[ "$Tags",","

] } } },

{"$out":"PostsVotesTags" }])

6**)Adding Owner User Details as an embeded Array to ‘PostsVotesTags’ collection:**

db.PostsVotesTags.aggregate(

[{

$lookup:

{

from:"users",

localField: "OwnerUserId",

foreignField: "Id",

as:"owneruserprofile"

}},

{

$out:"PostsVotesTagsUsers"

}])

**7)Adding Last Editor User Details as an Array to PostsVotesTagsUsers collection:**

db.PostsVotesTagsUsers.aggregate(

[{

$lookup:

{

from:"users",

localField: "LastEditorUserId",

foreignField:"Id",

as:"Lasteditoruserprofile"

}},

{

$out:"Combined "

}])

**8) Create a new Collection for adding accepted answer details to questions**

db.Combined.aggregate([

{$lookup: {

from: "Combined",

localField: "AcceptedAnswerId" ,

foreignField: "Id",

as: "Answers"}

},

{$unwind : "$tags"},

{$match:{AcceptedAnswerId:{$gte:1}}},

{$unwind : "$Answers"},

{

$group:

{"\_id": {"Topic":"$tags"},

titleInfo:{$push:{time:{$subtract:["$Answers.CreationDate","$CreationDate"]},Title:"$Title",OwnerUserID:"$Answers.OwnerUserId",PostID:"$Id"}}

}

},

{$out:"TagTime"}

])

9) **We had to create a collection were the questions and its’ respective answers shared a common query:**

db.Combined.aggregate(

  [

 {"$match":{$or:[{PostTypeId:1},{PostTypeId:2}]}},

     {$project:{ OwnerUserId:"$OwnerUserId", Id:"$Id",\_id:3, Date:"$CreationDate",

           NewId:{

                $cond: { if: { $eq: [ "$PostTypeId", 1 ] }, then: "$Id", else: "$ParentId" }

              },

              tags:

              {$cond: { if: { $eq: [ "$PostTypeId", 1 ] }, then: "$tags", else: "" }

              }

         }

     },

     {

"$out":"PostAnalytic"

} ])

**10) To introduce the Decision date to each question we had to add a new collection for analytical question 5**

db.Combined.aggregate(

[

{$lookup: {

from: "temp",

localField: "Id" ,

foreignField: "id",

as: "Decision1"}

},

{

"$out":"Combined1"

}])

**11) Create a collection ‘**AuthorCollection**“ to help calculate the co-authors for analytical question 6**

db.Combined.aggregate([

{

$lookup:

{

from:"Combined",

localField:"Id",

foreignField:"ParentId",

as:"answers"}},

     {"$out":"AuthorCollection"}])