Big Data Algorithms: Assignment 4

Asrorbek Orzikulov

**Understanding the data**

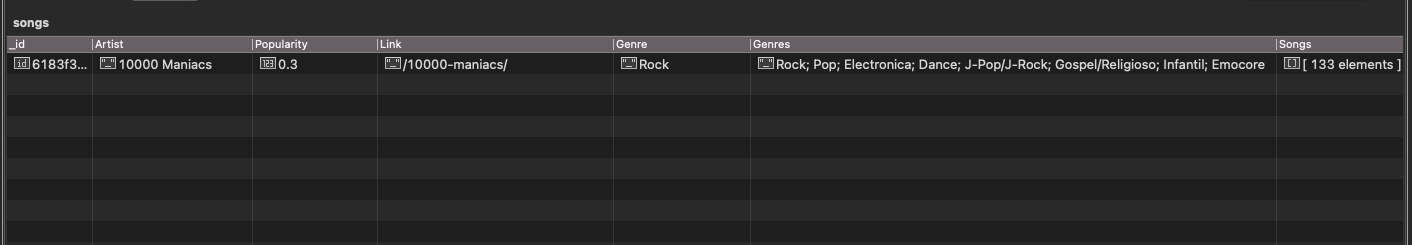
The dataset consists of seven fields or attributes that are present in all provided documents. Those fields are Artist (string -- name of an artist), Genre (string -- main genre of an artist), Genres (string -- genres of an artist separated by semicolons), Link (string -- the link at which an artist’s songs are available), Popularity (numeric -- score of an artist), Songs (array -- information about the songs of an artist), and \_id.

Songs is an array of nested documents where we can find information about a specific song. It consists of SName (string -- name of the song), SLink (string -- the link to the song), lyrics and language of a song.

**MongoDB queries**

1. (1 point) - Retrieve one element of the collection.

db.songs.find({}).limit(1);



2. (1 point) - Retrieve the artist name (Artist attribute) and the main genre of any artist for which the link to the webpage is provided.

db.songs.find(

{

"Link" : {

"$ne" : null

}

},

{

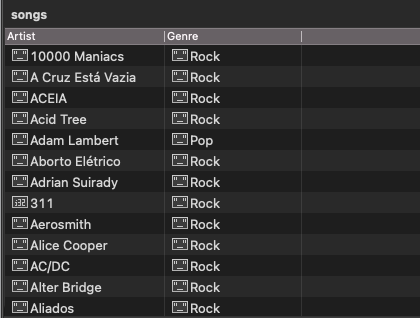
"Artist" : "$Artist",

"Genre" : "$Genre",

"\_id" : 0

}

);



3. (1 point) - Show the list of artists (just their name) that contain Stones in the name.

db.songs.find(

{

"Artist" : /^.\*Stones.\*$/i

},

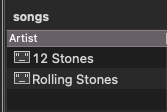
{

"Artist" : "$Artist",

"\_id" : 0

}

);



4. (1 point) - Show the number of artists that are returned by the previous query.

db.songs.aggregate(

[

{

"$match" : {

"Artist" : /^.\*Stones.\*$/i

}

},

{

"$group" : {

"\_id" : {},

"artist\_count" : {

"$sum" : 1

}

}

},

{

"$project" : {

"artist\_count " : "$artist\_count",

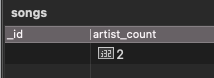
"\_id" : 0

}

}

]

);



5. (1 point) - Provide the name of the artists that have popularity greater than 4.

db.songs.find(

{

"Popularity" : {

"$gt" : 4

}

},

{

"Artist" : "$Artist",

"\_id" : 0

}

);



6. (1 point) - Show the artists with the lowest popularity.

db.songs.aggregate(

[

{

"$group" : {

"\_id" : {},

"minimum\_popularity" : {

"$min" : "$Popularity"

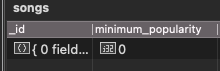
}

}

},

],

);



So, the minimum popularity score is 0. Now, I will retrieve all artists whose popularity is 0.

db.songs.find(

{

"Popularity" : NumberLong(0)

},

{

"Artist" : "$Artist",

"Popularity" : "$Popularity",

"\_id" : 0

}

);



7. (1 point) - Give the list of artists that have at least 10 songs listed.

db.songs.find(

{"$expr": {$gte:

[{$size: "$Songs"}, 10]}

},

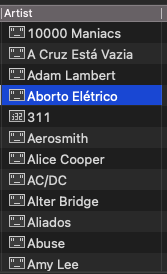
{

"Artist" : 1,

"\_id" : 0

}

);



8. (1 point) - Give the artists (if any) that do not have songs listed.

db.songs.find(

{"Songs" : {"$size": 0}},

{

"Artist" : 1,

"\_id" : 0

}

);



9. (3 point) - Give the list of artists that have at list 10 distinct songs listed and the criteria

that you used for this distinct operation.

db.songs.aggregate([

{ "$project": {

"\_id": 0,

"Artist": 1,

"distinct\_songs": {

"$size": { "$setDifference": [ "$Songs", [] ] }

}

}

},

{

"$match": {

"distinct\_songs": {"$gte": 10}

}

},

{ "$project": {

"\_id": 0,

"Artist": 1}

}

]);



For the distinct operation, I used the "$setDifference" function between Songs and an empty set. It will just return Songs as a set, which by definition contains only unique elements. Then, I obtained the number of distinct songs using “$size” function.

10. (1 point) - Show the complete set of genres for 40 artists.

db.songs.find({},

{

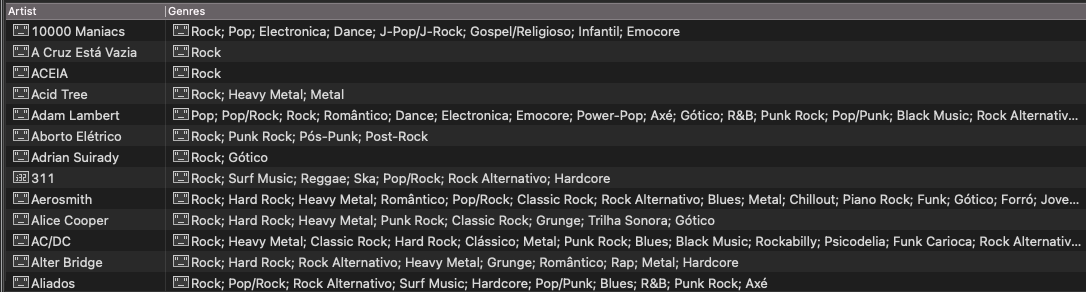
"Artist" : "$Artist",

"Genres" : "$Genres",

"\_id" : 0

}

).limit(40);



11. (1 point) - Comment on how the list is specified and on how it is possible to query this

attribute.

Different genres are separated by semicolon. We can use $match command with regular expressions to query this attribute. For example, to find the list of artists whose genre included Classic Rock, we can use the following command.

db.songs.find(

{

"Genres" : /^.\*Classic Rock.\*$/i

},

{

"Artist" : "$Artist",

"\_id" : 0

}

);

12. (3 points) - Give the number of artists having Folk among the genres.

db.songs.aggregate(

[

{

"$match" : {

"Genres" : /^.\*Folk.\*$/i

}

},

{

"$group" : {

"\_id" : {},

"folk\_artist\_count" : {

"$sum" : 1

}

}

},

{

"$project" : {

"folk\_artist\_count" : "$folk\_artist\_count",

"\_id" : 0

}

}

]

);



13. (3 points) Give the number of songs by main genre of the artist and language.

db.songs.aggregate([

{

"$unwind" : "$Songs"

},

{

"$group":

{

"\_id": {

"Genre": "$Genre",

"Language": "$Songs.Idiom"

},

"Song\_count": {

"$sum": NumberInt(1)

}

}

},

{

"$project" :

{

"\_id" : 0,

"Genre" : "$\_id.Genre",

"Language" : "$\_id.Language",

"Song\_count" : 1,

}

},

]);

