CS 348 - Homework 6: Document databases (MongoDB).
(75 Points)

Fall 2020
Due on: **11/24/2020 at 11:59 pm**
This assignment is to be completed by individuals. You should only talk to the instructor, and the TAs about this assignment. You may also post questions (and not answers) to Campuswire.

There will be a 10% penalty if the homework is submitted 24 hours after the due date, a 15% penalty if the homework is submitted 48 hours after the due date, or a 20% penalty if the homework is submitted 72 hours after the due date. The homework will not be accepted after 72 hours, as a solution will be posted by then.

Submission Instruction: write your answers (queries) in the HW6_Mongo.py file. Upload the file to Brightspace. The HW6_Mongo.py file and further instructions will be posted later this week.

For all questions, consider the countries collection included in the countries.json file. You can import the file to your local MongoDB database using the mongoimport tool. Check the following url for instructions to install mongoimport.
https://docs.mongodb.com/database-tools/installation/installation/

Question 1) (5 points)
Find countries whose area is less than 10 or greater than 10,000,000. Use the area field.

Expected Result:

```
{ "area" : 14000000, "name" : "Antarctica" }
{ "area" : 6, "name" : "Gibraltar" }
{ "area" : 2.02, "name" : "Monaco" }
{ "area" : 17098242, "name" : "Russia" }
{ "area" : -1, "name" : "Svalbard and Jan Mayen" }
{ "area" : 0.44, "name" : "Vatican City" }
```

Question 2) (5 points)

List the capital of the country whose official name in Italian is “Repubblica di Malta”. Use the translations.ita.official field.

Expected Result:

```
{ "capital" : "Valletta" }
```

Question 3) (5 points)

List the capital and the country name for capital cities located between latitudes 40 and 45 inclusive and between longitudes 10 and 20 inclusive. The latlng field contains the latitude and longitude for each country’s capital.

Expected Result:

```
{ "capital" : "Sarajevo", "name" : "Bosnia and Herzegovina" }
{ "capital" : "Tirana", "name" : "Albania" }
{ "capital" : "Rome", "name" : "Italy" }
{ "capital" : "Podgorica", "name" : "Montenegro" }
{ "capital" : "City of San Marino", "name" : "San Marino" }
{ "capital" : "Vatican City", "name" : "Vatican City" }
```

Question 4) (10 points)

List the country name and the number of currencies the country has. Sort you result by (number of currencies desc, country-name asc). Keep only the top five rows.

Notes:

- You may need to use $sort and $limit operators in MongoDB aggregates.
- You may need to use $size to get the size of the currencies array.

Check the MongoDB documentation for more information and examples.

Expected Result:

```
{ "name" : "Switzerland", "numberOfCurrencies" : 3 }
{ "name" : "United States", "numberOfCurrencies" : 3 }
{ "name" : "Western Sahara", "numberOfCurrencies" : 3 }
{ "name" : "Bhutan", "numberOfCurrencies" : 2 }
{ "name" : "Bolivia", "numberOfCurrencies" : 2 }
```

Question 5) (10 points)

Using MongoDB aggregates, list the total area for each subregion in the world.

Expected Result:

```
{ "_id" : "Northern America", "totalArea" : 21523696.2 }
{ "_id" : "Eastern Europe", "totalArea" : 18846226 }
{ "_id" : "South America", "totalArea" : 17837285 }
{ "_id" : "", "totalArea" : 14008208 }
{ "_id" : "Eastern Asia", "totalArea" : 11907076 }

……
```

Output is trimmed (check full output in the fullOutput.json file)

Question 6) (10 points)

Write the query in question 5 using the map-reduce framework in MongoDB.

Expected Result:

```
{
    "results" : [
        {
            "_id" : "Eastern Africa",
            "value" : 6361716
        },
        {
            "_id" : "Southern Asia",
            "value" : 6868982
        },

……
```

Output is trimmed (check full output in the fullOutput.json file)

Question 7) (15 points)

Using the map-reduce framework in MongoDB, perform a word count for the name.official field for countries in the southern Asia subregion.

Expected result:

```
{
    "results" : [ {  "_id" : "Kingdom",
                "value" : 1},
                {  "_id" : "the",
                  "value" : 1 },
                {  "_id" : "of",
                  "value" : 9  },
              {    "_id" : "Republic",
                  "value" : 8 },
                 …..
```

Output is trimmed (check full output in the fullOutput.json file)

Hints:

- Use the split function in JavaScript to get individual words from the name.official string. Here is an example:

```
db.countries.find({"name.common": "United States"}).forEach( function(myDoc) {
   var words = myDoc.name.official.split(" ");
   for (var idx=0; idx<words.length; idx++) {
        print( "Word: " +  words[idx] );
    }
} );
```

- To restrict the map-reduce processing to only countries in southern Asia, use the query option in MongoDB map-reduce. Here is an example:

```
db.countries.mapReduce(mapFunction1, reduceFunction1,
 {out:  {inline:1},
 query: { "subregion":"Southern Asia" } })
```

Question 8) (zero points, solution is provided)

List country name for countries that share the same demonym as the United States.

Expected result:

```
{      "demonym" : "American",
       "name" : "United States",
       "countryWithSameDemonym" : "Northern Mariana Islands"
}
{       "demonym" : "American",
      "name" : "United States",
       "countryWithSameDemonym" : "United States Minor Outlying Islands"
}
{       "demonym" : "American",
       "name" : "United States",
       "countryWithSameDemonym" : "United States"
}
```

Solution:

```
db.countries.aggregate([
   { $match: { "name.common" : "United States"}},
   { $project: { _id: 0, name: "$name.common", demonym: 1 } },
   { $lookup:
      {
        from: "countries",
        localField: "demonym",
        foreignField: "demonym",
        as: "countriesWithSameDemonym"
      }
   },
   { $unwind: "$countriesWithSameDemonym" },
   { $project: { _id: 0, name: 1, demonym: 1 , countryWithSameDemonym:
"$countriesWithSameDemonym.name.common"} },
]).pretty()
```

Question 9) (15 points)

Use MongoDB aggregates to list the neighbor countries for the United States. Each country has a cca3 field that contains a 3-character code for the country. The borders filed (array) contains the cca3 codes for neighbor countries.

Expected result:

```
{ "name" : "United States", "neighborCountryname" : "Canada" }
{ "name" : "United States", "neighborCountryname" : "Mexico" }
```