Assignment 3 Report

Team: Ankita Kamalkishor Tiwari and Vasuki Manoharan

1. Twitter API is used to get the data from twitter,developer account is created on twitter and API tokens are generated the tokens are used along with python code to extract the data and create collection in MongoDB: .**ipynb attached**
   Keywords used for extraction: ['#hoops', '#lakers', '#basketballneverstops', '#lebronjames', '#dunk', '#kobebryant','#NBA', '#lebron']

2. Queries (in terminal)

Popularity : Finding the followers of every username

```
db.tweets.aggregate(
  {$group: {_id: '$username', followers: {$max: "$followers"}}},
  {$sort: {followers: -1}}
);
```

```
Last login: Tue Apr  7 12:29:47 on ttys001
Vasukis-MacBook-Air:~ vasukimanoharan$ mongo
MongoDB shell version v4.2.5
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("d6b0157d-7952-48d4-b820-6655baa1f72e") }
MongoDB server version: 4.2.5
Server has startup warnings:
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten]
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] ** WARNING: Access control is not enabled for the database.
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          Read and write access to data and configuration is unrestricted.
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten]
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] ** WARNING: This server is bound to localhost.
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          Remote systems will be unable to connect to this server.
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          Start the server with --bind_ip <address> to specify which IP
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          addresses it should serve responses from, or with --bind_ip_all to
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          bind to all interfaces. If this behavior is desired, start the
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] **          server with --bind_ip 127.0.0.1 to disable this warning.
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten]
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten]
2020-04-01T17:41:26.061-0400 I  CONTROL  [initandlisten] ** WARNING: soft rlimits too low. Number of files is 256, should be at least 1000
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

> db
test
> show dbs
TwitterStream  0.000GB
Twitternew     0.000GB
admin          0.000GB
config         0.000GB
local          0.000GB
twitterdb      0.001GB
> use Twitternew
switched to db Twitternew
> show collections
tweets
> db.tweets.aggregate(
...   {$group: {_id: '$username', followers: {$max: "$followers"}}},
...   {$sort: {followers: -1}}
... );
{ "_id" : "Gazzetta_it", "followers" : 1731636 }
{ "_id" : "News4SA", "followers" : 195792 }
{ "_id" : "etalkCTV", "followers" : 149546 }
{ "_id" : "TheUndefeated", "followers" : 130743 }
{ "_id" : "BigOShow", "followers" : 101170 }
{ "_id" : "RonaldOllie", "followers" : 87156 }
{ "_id" : "KABBFOX29", "followers" : 74814 }
{ "_id" : "MurphyKeith", "followers" : 71133 }
{ "_id" : "TooAthletic", "followers" : 66787 }
{ "_id" : "vozdelfanatico", "followers" : 62540 }
{ "_id" : "mercenarygeo", "followers" : 51263 }
{ "_id" : "2KCrewFinder", "followers" : 48076 }
{ "_id" : "Jeff_Mans", "followers" : 47773 }
{ "_id" : "RonBohning", "followers" : 38066 }
```

Trending: Finding the highest retweet of every tweet in descending

db.tweets.aggregate(
{$group: {_id: '$text', favourites: {$max: "$favourites"}}},
{$sort: {favourites: -1}}
);

Tags associated: number of hashtags used

db.tweets.aggregate([{ $unwind: "$hashtags" }, { $sortByCount: "$hashtags" }])

```
> db.tweets.aggregate( [ { $unwind: "$hashtags" },  { $sortByCount: "$hashtags" } ] )
{ "_id" : { "text" : "itsreal85", "indices" : [ 59, 69 ] }, "count" : 35 }
{ "_id" : { "text" : "nba", "indices" : [ 54, 58 ] }, "count" : 35 }
{ "_id" : { "text" : "tookmeahourlol", "indices" : [ 70, 85 ] }, "count" : 35 }
{ "_id" : { "text" : "WoodenAward", "indices" : [ 66, 78 ] }, "count" : 34 }
{ "_id" : { "text" : "Spurs", "indices" : [ 110, 116 ] }, "count" : 9 }
{ "_id" : { "text" : "Lakers", "indices" : [ 117, 124 ] }, "count" : 9 }
{ "_id" : { "text" : "NBA", "indices" : [ 64, 68 ] }, "count" : 8 }
{ "_id" : { "text" : "NBA", "indices" : [ 76, 80 ] }, "count" : 7 }
{ "_id" : { "text" : "lakeshow", "indices" : [ 46, 55 ] }, "count" : 7 }
{ "_id" : { "text" : "lakers", "indices" : [ 56, 63 ] }, "count" : 7 }
{ "_id" : { "text" : "NBA", "indices" : [ 83, 87 ] }, "count" : 6 }
{ "_id" : { "text" : "NBA", "indices" : [ 43, 47 ] }, "count" : 5 }
{ "_id" : { "text" : "vanessabryant", "indices" : [ 59, 73 ] }, "count" : 5 }
{ "_id" : { "text" : "sacramentoproud", "indices" : [ 117, 133 ] }, "count" : 5 }
{ "_id" : { "text" : "kobe", "indices" : [ 104, 109 ] }, "count" : 5 }
{ "_id" : { "text" : "nataliabryant", "indices" : [ 75, 89 ] }, "count" : 5 }
{ "_id" : { "text" : "nba", "indices" : [ 112, 116 ] }, "count" : 5 }
{ "_id" : { "text" : "kobebryant", "indices" : [ 91, 102 ] }, "count" : 5 }
{ "_id" : { "text" : "gospursgo", "indices" : [ 101, 111 ] }, "count" : 5 }
{ "_id" : { "text" : "NBA", "indices" : [ 89, 93 ] }, "count" : 4 }
Type "it" for more
> it
{ "_id" : { "text" : "soccer", "indices" : [ 127, 134 ] }, "count" : 4 }
{ "_id" : { "text" : "BetOnAceD", "indices" : [ 119, 129 ] }, "count" : 4 }
{ "_id" : { "text" : "WWE", "indices" : [ 135, 139 ] }, "count" : 4 }
{ "_id" : { "text" : "HockeyTwitter", "indices" : [ 74, 88 ] }, "count" : 4 }
{ "_id" : { "text" : "NBATwitter", "indices" : [ 94, 105 ] }, "count" : 4 }
{ "_id" : { "text" : "NBA", "indices" : [ 128, 132 ] }, "count" : 4 }
{ "_id" : { "text" : "NBATWITTER", "indices" : [ 15, 26 ] }, "count" : 4 }
{ "_id" : { "text" : "NBA", "indices" : [ 78, 82 ] }, "count" : 4 }
{ "_id" : { "text" : "NFL", "indices" : [ 122, 126 ] }, "count" : 4 }
{ "_id" : { "text" : "MLB", "indices" : [ 106, 110 ] }, "count" : 4 }
{ "_id" : { "text" : "NHL", "indices" : [ 69, 73 ] }, "count" : 4 }
{ "_id" : { "text" : "MLS", "indices" : [ 111, 115 ] }, "count" : 4 }
{ "_id" : { "text" : "AceD", "indices" : [ 131, 136 ] }, "count" : 4 }
{ "_id" : { "text" : "NLL", "indices" : [ 117, 121 ] }, "count" : 4 }
{ "_id" : { "text" : "1970s", "indices" : [ 132, 138 ] }, "count" : 3 }
{ "_id" : { "text" : "NBA", "indices" : [ 35, 39 ] }, "count" : 3 }
{ "_id" : { "text" : "NBATwitter", "indices" : [ 0, 11 ] }, "count" : 3 }
{ "_id" : { "text" : "nba", "indices" : [ 126, 130 ] }, "count" : 3 }
{ "_id" : { "text" : "NewOrleans", "indices" : [ 107, 118 ] }, "count" : 3 }
{ "_id" : { "text" : "WeALLEN", "indices" : [ 95, 103 ] }, "count" : 3 }
Type "it" for more
>
```

Similar social media users:

Let us assume the user data collected belong to set A.From the top hashtags use, we pick the ones that are not related to nba and find the tweets associated with it and the user, now these users are the ones related that belong to set B and the some of them belong to Set A too. Hence the intersection of set A and B will have the users from both domain. And B is the related social domain users.

.ipynb for extraction attached

Keywords used for extraction: ['#WoodenAward', '#soccer', '#NFL', '#HockeyTwitter']

```
> use socialmedia
switched to db socialmedia
> show collections
tweets
> db.social.aggregate( [ { $unwind: "$hashtags" },  { $sortByCount: "$hashtags" } ] )
> db.tweets.aggregate( [ { $unwind: "$hashtags" },  { $sortByCount: "$hashtags" } ] )
{ "_id" : { "text" : "NFL", "indices" : [ 0, 4 ] }, "count" : 2 }
{ "_id" : { "text" : "HardKnocks", "indices" : [ 51, 62 ] }, "count" : 1 }
{ "_id" : { "text" : "better", "indices" : [ 33, 40 ] }, "count" : 1 }
{ "_id" : { "text" : "football", "indices" : [ 54, 63 ] }, "count" : 1 }
{ "_id" : { "text" : "Chicago", "indices" : [ 77, 85 ] }, "count" : 1 }
{ "_id" : { "text" : "sports", "indices" : [ 80, 87 ] }, "count" : 1 }
{ "_id" : { "text" : "argument", "indices" : [ 41, 50 ] }, "count" : 1 }
{ "_id" : { "text" : "sport", "indices" : [ 88, 94 ] }, "count" : 1 }
{ "_id" : { "text" : "NEVERFORGETLOYALTY", "indices" : [ 116, 135 ] }, "count" : 1 }
{ "_id" : { "text" : "futbol", "indices" : [ 72, 79 ] }, "count" : 1 }
{ "_id" : { "text" : "NFL", "indices" : [ 60, 64 ] }, "count" : 1 }
{ "_id" : { "text" : "NFL", "indices" : [ 59, 63 ] }, "count" : 1 }
{ "_id" : { "text" : "NYGiants", "indices" : [ 53, 62 ] }, "count" : 1 }
{ "_id" : { "text" : "fifa", "indices" : [ 95, 100 ] }, "count" : 1 }
{ "_id" : { "text" : "soccer", "indices" : [ 64, 71 ] }, "count" : 1 }
{ "_id" : { "text" : "FlyEaglesFly", "indices" : [ 24, 37 ] }, "count" : 1 }
{ "_id" : { "text" : "Philadelphia", "indices" : [ 63, 76 ] }, "count" : 1 }
{ "_id" : { "text" : "futebol", "indices" : [ 123, 131 ] }, "count" : 1 }
{ "_id" : { "text" : "Shopping", "indices" : [ 6, 15 ] }, "count" : 1 }
{ "_id" : { "text" : "NFL", "indices" : [ 86, 90 ] }, "count" : 1 }
Type "it" for more
>
```

3. Extraction of data from Assignment 2 where the Normalized tables were created.
   There are multiple ways to create collection and add data in MongoDB like, import JSON data,import CSV files directly.We used python scripting to convert the csv files to json format and insert the data to mongodb using python code .ipynb file for the same is attached.

Audit Validity/Acurracy :

In this fast moving digital world, having accurate data is one of the most important aspects of data collection. Incorrect data may result from migration of data from one database to another, presence of incorrect values, or even time-bound data changes.Reviewing is an efficient way to check the correctness of the data.To get the correct accurate data we have used trending hashtags from twitter to get the tweet and data related to the hashtags. The documents in collections are created with specific filesd like tweet_id,hashtags,username,followers,etc to remove unwanted metadata and other data associated and which is not required.

Audit Completeness :

Data completeness refers to whether there are any gaps in the data from what was expected to be collected, and what was actually collected.The problem of incomplete data can be resolved by ensuring that the data cannot be submitted, unless all expected data is present. Having a mandatory feild of PlayerID, Game ID and Team ID,Tweet_Id etc has made sure there is completeness and has resulted in less time consumption for audting completeness.

Audit Consistency/Uniformity:

Data consistency becomes a challenge to check while using No-SQL,but since the data that we have used is converted from SQL,and the SQL data is checked for consistency using primary key,foreign keys which were created while normalization the data is consistent as the keys are still present and can be used to check for the consistency.
Similarly the data from Twitter is consistent as it is created using specific fields and removing all that is not required(meta data,timestamps,etc).

Contirbution:
We contributed By Own: 30%
Provided by the professor : 30%
By External source: 40%

Github :
Ankita Tiwari : https://github.com/fx2044/tiwari_ank_dmdd
Vasuki Manoharan : https://github.com/Vasuki-Manoharan/NBA-Stats-Webscraping-API

Citation:

https://docs.mongodb.com/manual/tutorial/query-documents/

https://developer.twitter.com/en/apps/17674256

https://docs.mongodb.com/manual/reference/operator/query/

https://www.pythonforbeginners.com

https://towardsdatascience.com/streaming-twitter-data-into-a-mysql-database-d62a02b050d6

https://www.complex.com/sports/2019/05/all-30-nba-twitter-accounts-ranked-for-2019/

LICENSE:

Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.