Revision date: 11/22/2022

COMS 363 Fall 2022
Assignment 5

Percentage in your final grade: 9%
Maximum score for the assignment: 100 points

Objectives

1. Practice importing data into a Neo4j graph database from csv files.
2. Practice writing Cypher queries for the Neo4j graph database management system. Cypher is a query language of Neo4j. Neo4j is by far the most supported and used graph DBMS. Some portion of its query language could be included in an ongoing development of the Graph Query Language standard.

Instructions:

The assignment is individual work, not group work. Put all your answers in <netid>HW5.zip where <netid> is replaced by your Iowa State's netid.

<netid>Q1.jpeg for the answer of Question 1
<netid>Q2.cypher for the answers of Question 2
<netid>Q3.cypher for the answer of Question 3.

The file with the cypher extension is a text file with Cypher statements in it. See an example in LoadTweets.cypher. Hints are suggestions, not a requirement.

1. (10 points) Use Neo4j Desktop to create a new graph database called "tweetsdb" and load the data from the given csv files using the provided LoadTweets.cypher to import the data. The script works for Neo4j DBMS version 5.1.0. It may not work for older versions of Neo4j since Neo4j does not support backward compatibility well in the past. Make sure to put the csv files in the import folder of the database. Watch the recording on Canvas on how to install Neo4j. Watch the recording of class participation on Friday Nov. 18.

   The design of this database follows the basic graph database design guidelines. Nodes represent entities, and an edge represents a relationship between two nodes. Nodes can have properties. Edges can have properties.

   It is crucial to get the data loaded as soon as possible since your system may have a configuration that prevents successful importing of the data. See the teaching staff ASAP if you have a problem.

   Take a screenshot of the pseudo schema of the database. Be sure that node labels and edge labels are shown in the screenshot. Name the file <netid>Q1.jpeg. To see the pseudo schemas of this database, do the following statement at the Neo4j console.

   ```
   call db.schema.visualization()
   ```

2. (70 points) Write Neo4j Cypher queries to answer all the questions. Put all the answers in <netid>Q2.cypher. Indicate which Cypher query is for which question. Add a comment that includes your name to indicate that you are the person who writes the queries.

   Hint: Use the pseudo schema in Question 1 to help you when writing Cypher queries.

   a. (12 points) Find five tweets posted during February 2016. Show the posting user's screen name of the tweet, the party (sub_category property) of the posting user, and the retweet count of the tweet in descending order of the retweet count.

   Output when displayed in "Text":

"screenname"	"party"	"retweetCount"
"jessiehellmann"	"na"	75555
"BernieSanders"	"democrat"	44461
"HillaryClinton"	"democrat"	44461
"RepMarkCardenas"	"democrat"	44456
"jasonmdstein"	"na"	27460

   Hint: Use the match statement with node variables bound to the Tweet and the User node labels via the POSTED edge label.

   b. (10 points) List five unique hashtags posted by the users who lived in one of these states, Ohio and Alaska. Show the names of the hashtags in ascending order.

   Output when displayed in "Text":

"hashtagName"
"12MONTHBIRTHCONTROL"
"131STGA"
"1U"
"6MOREWEEKS"
"ABLEACT"

Hint: The location attribute of the User node label has information about the state in which the user lived. Use the match statement with node variables bound to the Hashtag, Tweet, and the User node labels and the where clause to specify the condition on the state the user lived. Since multiple users may use the same hashtag, list the hashtag name only once.

c. (10 points) Find five users in the "Democrat" party with the most number of followers. Show the user's screen name, party, and the number of followers of these users in descending order of the number of followers.

Output when displayed in "Text":

"screenname"	"party"	"numFollowers"
"HillaryClinton"	"democrat"	6187008
"BernieSanders"	"democrat"	2137079
"MartinOMalley"	"democrat"	130920
"FlaDems"	"democrat"	46559
"CASenateDems"	"democrat"	30868

Hint: The sub_category attribute has information about which party the user belongs to. We only need a match statement with one node variable in this query.

d. (20 points) Find five users mentioned in the most number of tweets of users in the GOP party. Show the mentioned user's screen name and the list of the mentioning users' screen name in descending order of the number of tweets mentioning this user.

Output when displayed in "Text":

"mentionedUser"	"listMentioningUsers"
"tedcruz"	["FloridaGOP","IdahoGOP","JebBush","JohnKasich","NDGOP","NewYorkGOP","OhioSenateGOP","RhodeIslandGOP","TNGOP","gophawaii","marcorubio","realDonaldTrump","tedcruz"]
"JohnKasich"	["IdahoGOP","JebBush","JohnKasich","MSGOP","NDGOP","NewYorkGOP","OhioSenateGOP","TNGOP","gophawaii","realDonaldTrump","tedcruz"]
"realDonaldTrump"	["FloridaGOP","IdahoGOP","JebBush","JohnKasich","NewYorkGOP","OhioSenateGOP","RealBenCarson","TNGOP","UtahGOP","WVGOP","gophawaii","gov_gilmore","indgop","lagop","marcorubio","realDonaldTrump","tedcruz"]
"HillaryClinton"	["FloridaGOP","IdahoGOP","JebBush","JohnKasich","MSGOP","MissouriGOP","NCGOP","NHGOP","NewYorkGOP","RealBenCarson","TNGOP","WVGOP","gov_gilmore","mainegop","marcorubio","massgop","realDonaldTrump","tedcruz"]
"GOP"	["ColoSenGOP","FloridaGOP","IdahoGOP","JohnKasich","KYSenateGOP","MSGOP","MissouriGOP","NCGOP","NEGOP","NHGOP","NewMexicoGOP","NewYorkGOP","RealBenCarson","RhodeIslandGOP","TNGOP","WVGOP","gophawaii","gov_gilmore","lagop","mainegop","massgop","realDonaldTrump","tedcruz"]

Hint: Use different node variables for the posting users and the mentioned users. Use the "with" clause that acts like a group-by clause in SQL. Use collect() to create the list like group_concat() did in MySQL in Homework 3.

e. (18 points) List three hashtag names in descending order of the number of states the hashtag was used in tweets posted by the users who lived in those states. List the hashtag name, the total number of states, and the list of the distinct state names. Do not include the state whose name is na or empty.

The first row shows that the hashtag "GOPDEBATE" was used in tweets posted by the users of 29 states with the names of the states shown in the last column.

Output when displayed in "Text":

"hashtag"	"numstates"	"statelist"
"GOPDEBATE"	29	["North Carolina","Michigan","Delaware","Florida","Hawaii","Oregon","Idaho","Indiana","Pennsylvania","Wisconsin","South Carolina","Illinois","Ohio","Missouri","Rhode Island","Louisiana","Maine","Mississippi","Nebraska","New Mexico","New York","New Hampshire","Nevada","Texas","South Dakota","New Jersey","Tennessee","Virginia","West Virginia"]
"DEMDEBATE"	27	["North Carolina","Michigan","Delaware","Florida","Mississippi","Idaho","Pennsylvania","Wisconsin","South Carolina","Illinois","Massachusetts","Rhode Island","Louisiana","Maine","California","Missouri","Nebraska","New York","New Hampshire","Nevada","Arizona","Texas","South Dakota","New Jersey","Tennessee","Virginia","West Virginia"]
"SOTU"	26	["Connecticut","California","Colorado","Delaware","Florida","Hawaii","Oregon","Idaho","Indiana","Pennsylvania","Iowa","Illinois","Rhode Island","Louisiana","Massachusetts","Michigan","Missouri","Mississippi","North Carolina","Nebraska","Washington","South Dakota","Tennessee","Virginia","Wisconsin","West Virginia"]

Hint: The state where a user has lived is specified in the location attribute. Use the "with" clause to group the tweets by the hashtag occurring in these tweets. Use the collect() function to create the list.

3. (20 points) Create a Neo4j graph database from the ER diagram below. Write Cypher statements in a text file <netid>Q3.cypher to create the graph database. Use a node label to group nodes with similar properties. Use an edge label to group edges with similar properties. Attributes of entities and relationships specified in the ER diagram need to be captured in the Neo4j graph database. The constraints on these attributes must also be captured. Create three nodes for each node label. Create three edges for each edge label.