METROPOLITAN STATE UNIVERSITY OF DENVER
Department of Mathematical and Computer Sciences

CS 3810 - Principles of Database Systems (Fall 2019)

Database Assignment 04

Deadline: December 6th (11:59pm)

Introduction

Many human genetic disorders and diseases are known to be related to each other through frequently observed co-occurrences[1]. By studying disease-gene associations it is possible to build Weighted Human Disease Networks (WHDNs). In WHDNs, two diseases are connected if they have at least one association gene in common. Weights are then assigned to connection edges based on the number of shared association genes, as well as the strengths of those associations. In this assignment you will model a WHDN on a Neo4J graph database. Then you will perform a cluster analysis to shed some light on how groups of diseases are related to each other.

Data Load

The dataset for this assignment is publicly available here. However, we recommend that you use the diseases_network.json file that only represents `411` diseases: the ones that showed stronger connections to other diseases. The shared JSON file has a list of `<disease_pair>`, defined as:

```
<disease_pair> ::= {
    'a': { 'code': <String>, 'name': <String> },
    'b': { 'code': <String>, 'name': <String> },
    'weight': <Float>
}
```

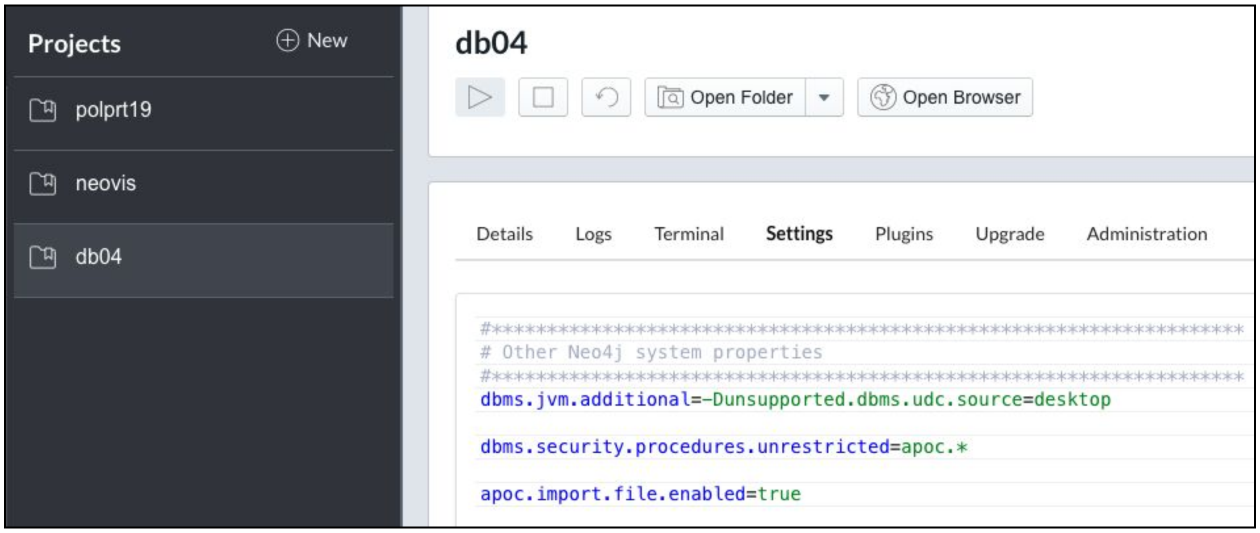
Begin your project by creating a database named `db04` using Neo4J Desktop. In order to populate your database using the shared JSON file you will write a data load script named `network.cypher` (the first deliverable of this assignment). We suggest using APOC (Awesome Procedures on Cypher), a Neo4J data manipulation library. This link here explains how to get APOC installed.

Make sure you set parameter `apoc.import.file.enabled` to `true` in order to enable JSON APOC data loads[2]. Using Neo4J Desktop tool, click on "Manage" and then "Settings"; go

[1] See https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6430629

[2] Failing to do so will have the data load fail with the exception: `Import from files not enabled, please set apoc.import.file.enabled=true in your neo4j.conf`.

to the end of your database configuration file and paste `apoc.import.file.enabled = true`. You will be asked to restart the database. Refer to the picture below for clarification.



```
#********************************************************************
# Other Neo4j system properties
#********************************************************************
dbms.jvm.additional=-Dunsupported.dbms.udc.source=desktop

dbms.security.procedures.unrestricted=apoc.*

apoc.import.file.enabled=true
```

Using JSON APOC to perform a data load is not difficult. Use Activity 20 as a starting point. Before moving forward, make sure you have `411` nodes in your graph and that the nodes have the correct properties and relationships.

For example, the query below should return `411` nodes.

```
MATCH (n) RETURN count(n)
```

The query below should display two nodes and an edge between them. Click on each node and verify their names. For example, node C0023892 should be 'hanot's cirrhosis' and node C0859942 is 'biliary cirrhosis'.

```
MATCH( a { code: 'C0023892' } )
MATCH( b { code: 'C0859942' } )
RETURN a, b
```

Community Detection

After importing the JSON file and generating the graph, you are asked to apply the Louvain's community detection algorithm to identify groups of diseases in the graph that are highly interconnected. Again, use Activity 20 notes to help you running the community detection procedure. Save the results for further processing exporting them using JSON format. This file should be named `nodes.json` and it is the second deliverable of this assignment.

Your next step is to set a label to each of the disease nodes based on their community classification. Unfortunately Neo4J does not currently support using variables when applying labels to nodes through a cypher script. Therefore you will have to write a script on your own using an external PL, like Java or Python.

Your script should read `nodes.json` and produce as output a cypher script named `labeling.cypher` (the third deliverable of this assignment). This script consists of a sequence of cypher commands to apply labels to each node based on their community. Because the label of a node must begin with a letter, we suggest appending the letter "c" on each of the community codes before applying the label to a node. Below is an excerpt of `labeling.cypher` so you understand its format.

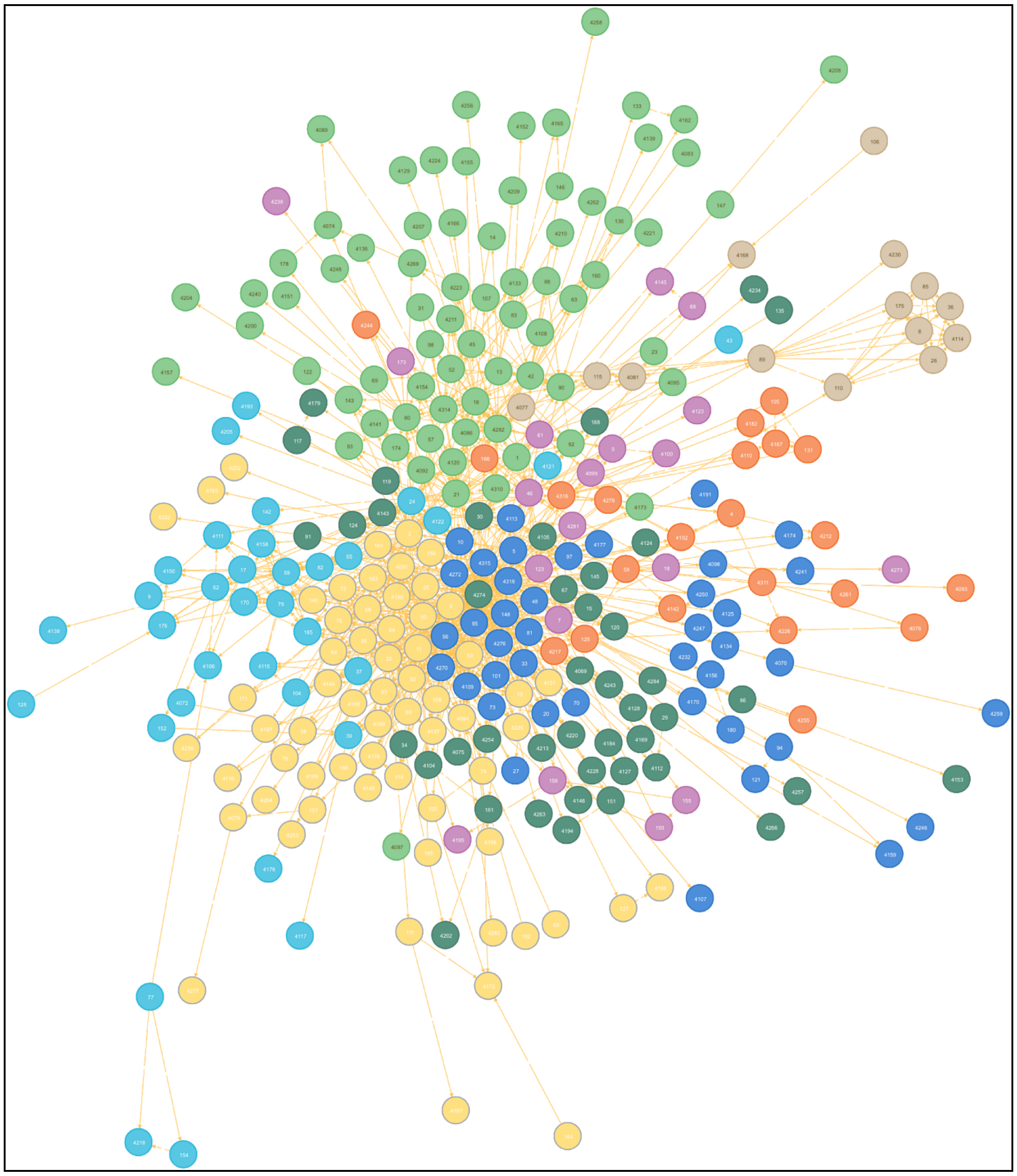
```
...
MATCH( node { code: 'C0410550'} ) SET node:c7
WITH 1 as dummy
MATCH( node { code: 'C0028960'} ) SET node:c7
WITH 1 as dummy
MATCH( node { code: 'C3544092'} ) DETACH DELETE node
WITH 1 as dummy
MATCH( node { code: 'C4020898'} ) DETACH DELETE node
WITH 1 as dummy
MATCH( node { code: 'C0085606'} ) DETACH DELETE node
WITH 1 as dummy
MATCH( node { code: 'C0014550'} ) SET node:c9
WITH 1 as dummy
...
MATCH (n)
RETURN n
```

In this assignment we will focus on communities of diseases with at least 10 nodes. Your cypher script should delete the nodes that are part of smaller communities of less than 10 nodes. In the example above, nodes `C3544092`, `C4020898`, and `C0085606` are marked for deletion (together they are part of a small community of three nodes).

Between each node labeling (or deletion) you should add a "`WITH 1 as dummy`" statement. This is a workaround so you can run multiple statements in a single script file. Finally, note that your cypher script should end in "`MATCH(n) RETURN n`" so the resulting graph is displayed when the script completes.

Data Analysis

Use Neo4J Browser to display nodes with different colors based on their community labels. Your graph should look like the one below (colors don't have to match). Pick one community and write a hypothesis describing why do you think the diseases in the chosen community are related to each other. Give some examples of diseases in that community to support your hypothesis. Write your answer in a file named `hyphothesis.txt` (your fourth and last deliverable on this assignment).



Deliverables

- `network.cypher` (data load program written in cypher);
- `nodes.json` (communities exported using JSON format);
- `labeling.cypher` (cypher script to delete nodes in small communities and to attach community labels to the other nodes);
- `hypothesis.txt` (data analysis).

Rubric

TBD