

Assignment 5.

Nilufar Hatamova

Exercise 1.

A.

To load data into Neo4j I have done several steps:

- Generating Cypher for one row of data using the Arrows tool (as sample)
- Writing python program that uses this sample Cypher model to generate Cypher which creates a graph. The graph contains all data set which is the result of Question 1 in Assignment 4.
- Saving Cypher as a txt file.
- Using Cypher in neo4j database to create a graph.

I did not create a University node separately. Because the university is a subclass of the organization class, all university nodes inserted as an organization node but with yearFounded property key.

B.

Query to find the names of people who work in the same place (e.g. Gothenburg) where they graduated:

```
MATCH
(person:Person)-[:employeeOf]->(org:Organisation),(org:Organisation)-[:locatedIn]->
(placeorg:Place) ,
(person:Person)-[:alumnusOf]->(uniorg:Organisation),(uniorg:Organisation)-[:located
In]->(placeuni:Place)
WHERE placeuni.placeName=placeorg.placeName // work in the same place where they
graduated
//WHERE placeuni.placeName="Gothenburg" and placeorg.placeName="Gothenburg" will
return work and graduated in Gothenburg
RETURN person.personName as Person ,org.organisationName as organisationName,
uniorg.organisationName as universityName, placeorg.placeName as organisationPlace,
placeuni.placeName as universityPlace
```

Some results of the query:

"Person"	"organisationName"	"universityName"	"organisationPlace"	"universityPlace"
"Gunnar Carlquist"	"Lund University Library"	"Lund University"	"Lund"	"Lund"
"Gerard De Geer"	"Geological Survey of Sweden"	"Uppsala University"	"Uppsala"	"Uppsala"
"Per Alström"	"Swedish University of Agricultural Sciences"	"Uppsala University"	"Uppsala"	"Uppsala"
"Gunilla Wünsche"	"Swedish University of Agricultural Sciences"	"Uppsala University"	"Uppsala"	"Uppsala"
"Maria Elmquist"	"Chalmers University of Technology"	"University of Gothenburg"	"Gothenburg"	"Gothenburg"
"Ulf Grenander"	"Royal Institute of Technology"	"Stockholm University"	"Stockholm"	"Stockholm"

Query to find which Chalmers alumni are employed by each organization.

```
MATCH (person:Person)-[:employeeOf]->(org:Organisation) ,
      (person:Person)-[:alumnusOf]->(uni:Organisation{organisationName:"Chalmers University of Technology"})
RETURN person.personName as Person ,org.organisationName as organisationName,
       uni.organisationName as universityName
```

Some results of query:

"Person"	"organisationName"	"universityName"
"Harald Hammarström"	"Uppsala University"	"Chalmers University of Technology"
"Birgit Krantz"	"Lund University"	"Chalmers University of Technology"
"Erik G. Hallén"	"Royal Institute of Technology"	"Chalmers University of Technology"
"Yngve Hamnerius"	"Chalmers University of Technology"	"Chalmers University of Technology"
"Åke Öberg"	"Linköping University"	"Chalmers University of Technology"

Query to count the number of Chalmers alumni are employed by each organization.

```
MATCH (person:Person)-[:employeeOf]->(org:Organisation) ,
(person:Person)-[:alumnusOf]->(uni:Organisation{organisationName:"Chalmers
University of Technology"})
RETURN org.organisationName as organisation_name, count(person) as
number_of_Chalmers_alumni
ORDER BY (number_of_Chalmers_alumni) //ordering
```

Result of the query:

"organisation_name"	"number_of_Chalmers_alumni"
"Karolinska Institute"	1
" Jönköping University"	1
"Massachusetts Institute of Technology"	1
" Geological Survey of Sweden"	1
" Hamburg-Bergedorf Observatory"	1
"Umeå University"	1
"Lund University"	2
"Linköping University"	2
"University of Gothenburg"	2
"Uppsala University"	3
"Royal Institute of Technology"	7
"Chalmers University of Technology"	21

Query to calculate the age of Swedish universities and define status as “oldest”, “old”, “new” and “newest”:

```
MATCH (uni:Organisation)-[:locatedIn]-(place:Place{countryName:"Sweden"})
WHERE uni.yearFounded is not null
RETURN uni.organisationName as University, 2020-toInteger(uni.yearFounded) as AGE,
CASE
WHEN (2020-toInteger(uni.yearFounded))>=300
THEN "OLDEST"
WHEN 300>(2020-toInteger(uni.yearFounded))>=150
THEN "OLD"
WHEN 150>(2020-toInteger(uni.yearFounded))>=50
THEN "NEW"
ELSE "NEWEST" END AS STATUS
```

Some results of the query:

"University"	"AGE"	"STATUS"
"Lund University"	354	"OLDEST"
"Uppsala University"	543	"OLDEST"
"Karolinska Institute"	210	"OLD"
"University of Gothenburg"	66	"NEW"
"Stockholm University"	142	"NEW"
"Umeå University"	55	"NEW"
"Linköping University"	45	"NEWEST"
"Royal Institute of Technology"	193	"OLD"
"Chalmers University of Technology"	191	"OLD"
"Örebro University"	43	"NEWEST"
"Karlstad University"	43	"NEWEST"
"Luleå University of Technology"	49	"NEWEST"
"Mid Sweden University"	27	"NEWEST"
"Swedish University of Agricultural Sciences"	43	"NEWEST"
"Linnaeus University"	10	"NEWEST"

Exercise 2.

Identify stakeholders

A stakeholder is a person who is impacted by ethical decisions. Because Covid-19 is an infectious disease, stakeholders are the drivers involved, all passengers who traveled at the same time, and all the people who interacted with them (family, friends, colleagues, etc). Moreover, the stakeholders are not limited to only the list specified. It keeps increasing as people interact with others.

Discussing possible approaches, their potential benefits, and possible harms.

Taking no action will be unethical since it will lead to the possibility of lost lives and spread the disease further. However, the goal must be taking actions within ethical considerations and users' privacy in mind.

The ticketing software must be designed in such a way that, persons who are not supposed to have access to personal data should not be able to see it. The first mistake has been done since I, as an engineer already have access to the personal data of customers such as phone numbers.

The exposure of personal data can be limited if I can limit any other human factor to access it. Therefore, I will design a system that can send an automated message to all passengers who have traveled the same route and at the same time as myself. The message will warn the first wave of stakeholders about the danger that they are potentially facing and the preventative actions will be left to the individuals to take. There is no harm in this approach unless the created software securely handles the matter. If there are harms, the benefits would outweigh the harms.

Another approach would be sharing the data (phone numbers) with authorities, and hospitals and letting them take action. The following could be possible harms in this case.

1. The means of sharing the data might not be secure, therefore the data might end up in the wrong hands such as hackers, scammers, etc.
2. The authorities/hospitals might use the data for other purposes such as sending ads.

This would not be very ethical as the other stakeholders will not want their phone numbers to be shared with third parties and the company does not have consent to do so.

Weighing the benefits against possible harm

The selected approach is to send an automated message to all passengers who have traveled the same time as me. This approach does not have such harm as making data accessible for third parties, hackers, etc. But this approach has a lot of benefits such as saving a lot of lives, slowing the spreading speed of the virus. The benefits of the decision would exceed the harms since the lives of other individuals are at stake.