



МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО”

Факультет прикладної математики
Кафедра програмного забезпечення комп’ютерних систем

Лабораторна робота № 3
з дисципліни “Бази даних 2”
тема “Практика використання графової бази даних Neo4J”

Виконав
студент III курсу
групи КП-83

Мричко Богдан Тарасович
(*прізвище, ім'я, по батькові*)

варіант № 15

Зарахована
“ ____ ” “ ____ ” 2021 р.
викладачем

Петрашенко Андрій Васильович
(*прізвище, ім'я, по батькові*)

Київ 2021

Завдання

Реалізувати можливості формування графової бази даних в онлайн-режимі на основі модифікованої програми лабораторної роботи №2. На основі побудованої графової бази даних виконати аналіз сформованих даних.

Окремі програмні компоненти

1. Інфраструктура лабораторної роботи №2:
 - 1.1. Redis server.
 - 1.2. Програма емуляції активності користувачі (вхід/вихід, відправка/отримання повідомлення).
 - 1.3. Виконувач задач (Worker).
2. Сервер Neo4J.
3. Інтерфейс користувача Neo4J.

Порядок виконання роботи

1. В ЛР№2 залишити єдиний режим роботи - емуляція активності.
2. Внести доповнення у програму ЛР№2 шляхом додавання у повідомлення тегу або тегів з переліку, заданого у вигляді констант, обраних студентом.
3. Встановити сервер [Neo4J Community Edition](#).
4. Розробити схему бази даних Neo4J для збереження інформації про активності користувачів (вхід/вихід, відправлення/отримання повідомлень) та Worker (перевірка на спам). Визначити вузли та зв'язки між ними на графі.
5. Розширити функціональність ЛР№2 шляхом збереження будь-якої активності (див. п. 4) у базу даних Neo4J у момент збереження даних у Redis.
6. У програмі “Інтерфейс користувача Neo4J” виконати і вивести результат наступних запитів до сервера Neo4J:
 - 6.1. Задано список тегів (*tags*). Знайти всіх користувачів, що відправили або отримали повідомлення з набором тегів *tags*.
 - 6.2. Задано довжину зв'язку N - кількість спільних повідомлень між користувачами. Знайти усі пари користувачів, що мають зв'язок

довжиною N через відправлені або отримані повідомлення. Наприклад, якщо користувач A відправив повідомлення користувачу B , а B відправив повідомлення C , то довжина зв'язку між A і C є $N=2$.

6.3. Задано два користувача. Знайти на графі найкоротший шлях між ними через відправлені або отримані повідомлення.

6.4. Знайти авторів повідомлень, які пов'язані між собою лише повідомленнями, позначеними як "спам".

6.5. Задано список тегів (*tags*). Знайти всіх користувачів, що відправили або отримали повідомлення з набором тегів *tags*, але ці користувачі не пов'язані між собою.

Вимоги до засобів емуляції даних

Забезпечити генерацію даних відносно невеликого обсягу, що підтверджують коректність виконання завдання пунктів 6.1 - 6.5.

Вимоги до інтерфейсу користувача

Використовувати консольний (текстовий) інтерфейс користувача.

Реалізація

Схема бази даних

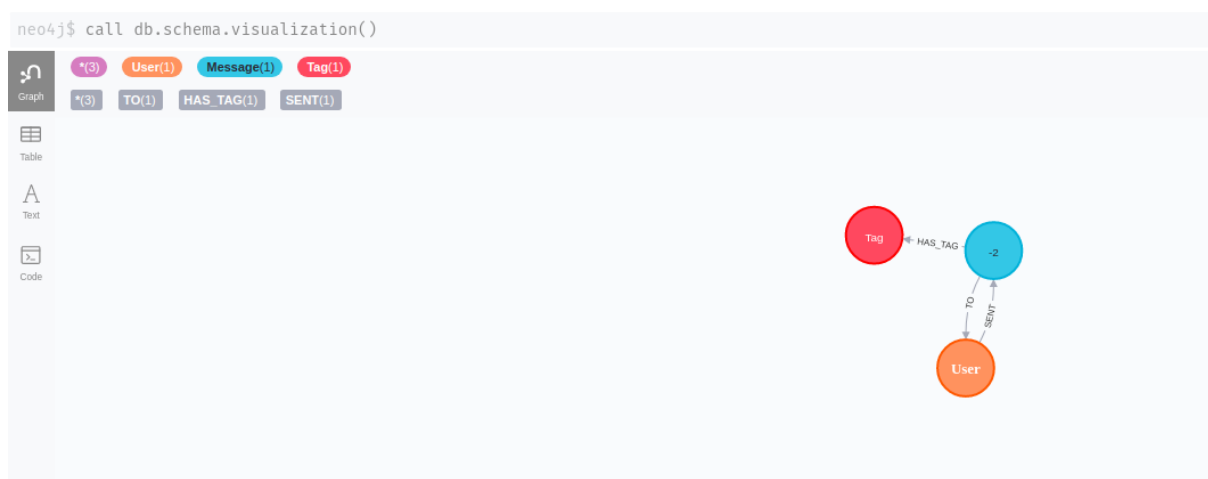
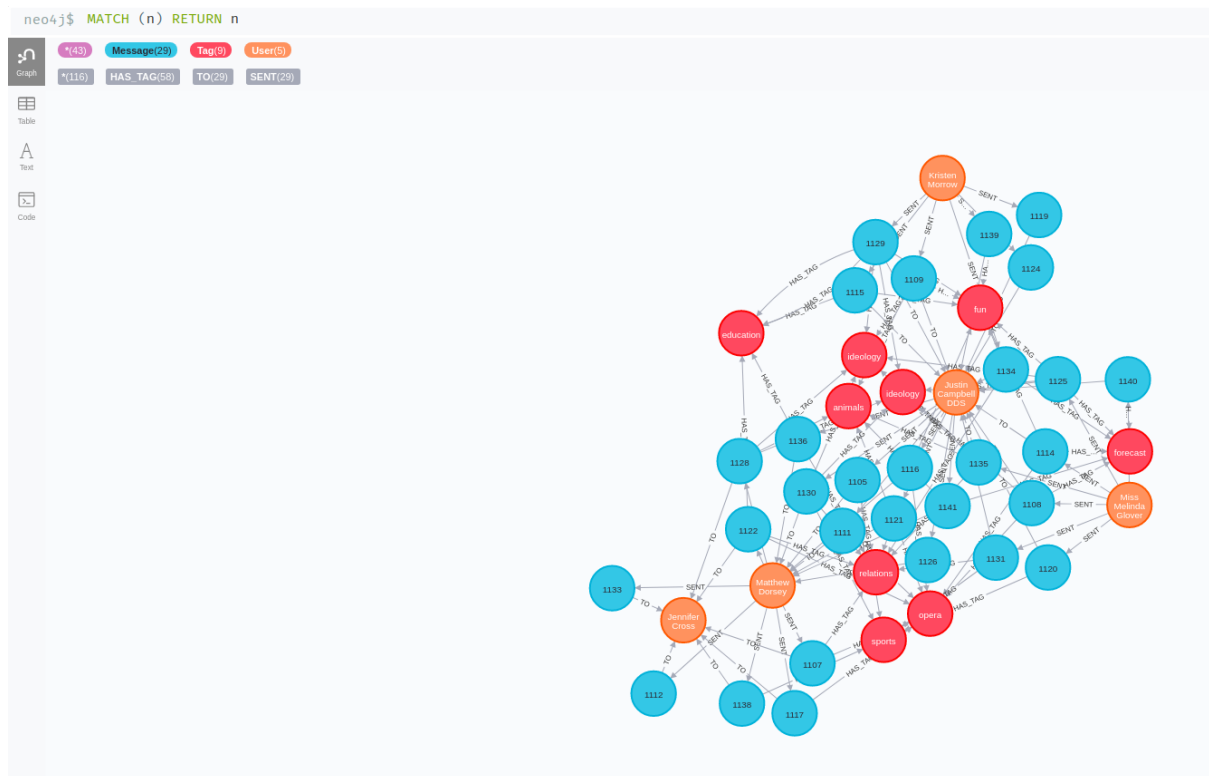


Схема графу усієї бази даних



Задано список тегів (*tags*). Знайти всіх користувачів, що відправили або отримали повідомлення з набором тегів *tags*.

```
match (tag:Tag) where tag.name in ['fun']
match (msg:Message)→(tag)
match (sender:User)-[:SENT]→(msg)-[:TO]→
(receiver:User)
return sender.username as username
union
match (tag:Tag) where tag.name in ['fun']
match (msg:Message)→(tag)
match (sender:User)-[:SENT]→(msg)-[:TO]→
(receiver:User)
return receiver.username as username
```

username

1	"Kristen Morrow"
2	"Miss Melinda Glover"
3	"Justin Campbell DDS"
4	"Matthew Dorsey"

Задано довжину зв'язку N - кількість спільних повідомлень між користувачами. Знайти усі пари користувачів, що мають зв'язок довжиною N через відправлені або отримані повідомлення. Наприклад, якщо користувач А відправив повідомлення користувачу В, а В відправив повідомлення С, то довжина зв'язку між А і С є $N=2$.

```
match (user1:User)
match (user2:User)
match (user1)-[:SENT|TO*2]-(user2) where user1.username
<> user2.username
return user1.username as
User1, user2.username as
User2
```

	User1	User2
1	"Justin Campbell DDS"	"Matthew Dorsey"
2	"Justin Campbell DDS"	"Miss Melinda Glover"
3	"Justin Campbell DDS"	"Kristen Morrow"
4	"Justin Campbell DDS"	"Matthew Dorsey"
5	"Justin Campbell DDS"	"Miss Melinda Glover"
6	"Justin Campbell DDS"	"Kristen Morrow"
7	"Justin Campbell DDS"	"Matthew Dorsey"

ted streaming 58 records in less than 1 ms and completed after 1 ms.

```

match (user1:User)
match (user2:User)
match (user1)-[:SENT|TO*4]-(user2) where user1.username
<> user2.username
return user1.username as
User1, user2.username as
User2

```



	User1	User2
	"Matthew Dorsey"	"Miss Melinda Glover"
82	"Matthew Dorsey"	"Kristen Morrow"
83	"Matthew Dorsey"	"Kristen Morrow"
84	"Matthew Dorsey"	"Miss Melinda Glover"
85	"Matthew Dorsey"	"Miss Melinda Glover"
86	"Matthew Dorsey"	"Kristen Morrow"
87	"Matthew Dorsey"	"Miss Melinda Glover"

orted streaming 434 records in less than 1 ms and completed after 4 ms.

```

match (user1:User)
match (user2:User)
match (user1)-[:SENT|TO*8]-(user2) where user1.username
<> user2.username
return user1.username as
User1, user2.username as
User2

```



	User1	User2
1	"Justin Campbell DDS"	"Jennifer Cross"
2	"Justin Campbell DDS"	"Jennifer Cross"
3	"Justin Campbell DDS"	"Jennifer Cross"
4	"Justin Campbell DDS"	"Jennifer Cross"
5	"Justin Campbell DDS"	"Jennifer Cross"
6	"Justin Campbell DDS"	"Jennifer Cross"
7	"Justin Campbell DDS"	"Jennifer Cross"

rted streaming 63784 records in less than 1 ms and completed after 2 ms, displaying first 1000 rows.

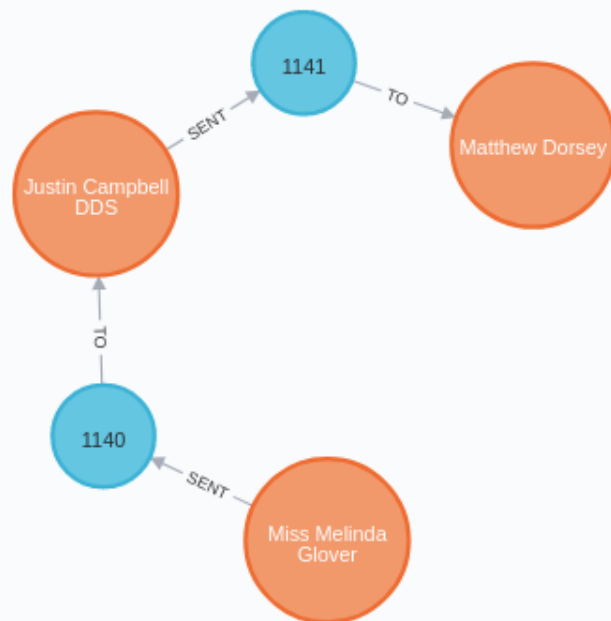
Задано два користувача. Знайти на графі найкоротший шлях між ними через відправлені або отримані повідомлення.

```

match (user1:User {username: 'Matthew Dorsey'})
match (user2:User {username: 'Miss Melinda Glover'})
with shortestPath((user1)-[:SENT|TO*]-(user2)) as
path
return path

```

*(5) User(3) Message(2)
 *(4) TO(2) SENT(2)



Displaying 5 nodes, 4 relationships.

Знайти авторів повідомлень, які пов'язані між собою лише повідомленнями, позначеними як “fun”


```

match (tag:Tag {name: 'fun'})
match (msg:Message)→(tag)
match (sender:User)-[:SENT]→(msg)-[:TO]→
(receiver:User)
return sender.username as Sender,
receiver.username as Receiver

```



	Sender	Receiver
3	"Kristen Morrow"	"Justin Campbell DDS"
4	"Miss Melinda Glover"	"Justin Campbell DDS"
5	"Justin Campbell DDS"	"Matthew Dorsey"
6	"Miss Melinda Glover"	"Justin Campbell DDS"
7	"Kristen Morrow"	"Justin Campbell DDS"
8	"Kristen Morrow"	"Justin Campbell DDS"

ted streaming 8 records after 1 ms and completed after 1 ms.

Задано список тегів (*tags*). Знайти всіх користувачів, що відправили або отримали повідомлення з набором тегів *tags*, але ці користувачі не пов'язані між собою.

```

1 WITH ['fun', 'forecast'] as tags
2 MATCH (tag:Tag)
3 WHERE tag.name in tags
4 WITH collect(tag) as tags
5 MATCH (msg:Message)
6 WHERE ALL(t in tags WHERE (msg)-[:HAS_TAG]→(t))
7 MATCH (user:User)-[:SENT|TO]→(msg)
8 RETURN user.username as Users

```

Users	
1	"Miss Melinda Glover"
2	"Miss Melinda Glover"
3	"Kristen Morrow"

Started streaming 3 records in less than 1 ms and completed after 1 ms.

Програмний код

Програмний код розміщений в GitHub репозиторії за покликанням - <https://github.com/Bodichelly/DataBaseTermSix>

Висновок

В результаті виконання лабораторної роботи здобуто практичні навички створення програм, орієнтованих на використання графової бази даних Neo4J за допомогою мови Python.