

## Лабораторна робота № 3.

### Практика використання графової бази даних Neo4J

*Метою роботи* є здобуття практичних навичок створення програм, орієнтованих на використання графової бази даних Neo4J за допомогою мови Python.

*Завдання* на роботу полягає у наступному:

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 (див. [завдання](#)):
  - 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.

#### *Вимоги до інструментарію*

1. Мова програмування – Python 3.x.
2. Засоби відлагодження запитів мовою Cypher - Neo4j Browser, доступний після встановлення Neo4J за адресою <http://localhost:7474/browser/>
3. Бібліотека доступу до Neo4J – [Neo4 Python Driver](#).
4. Середовище розробки програмного забезпечення – PyCharm Community Edition (*опціонально*)

#### *Вибір варіанту*

Робота виконується індивідуально. Хоча індивідуальних варіантів не передбачено, усі вимоги, наведені вище, є обов'язковими до виконання.

*Вимоги до оформлення лабораторної роботи у електронному вигляді*

Опис лабораторної роботи у репозиторії включає: назву лабораторної роботи, **приклад схеми бази даних Neo4J**, копії екранних форм (screenshots) результатів виконання запитів.

*Контрольні запитання*

1. Визначити сфери застосування графової бази даних Neo4J.
2. Пояснити призначення ключових слів MATCH, WHERE, RETURN мови запитів Cypher.

*Джерела*

1. Книга [Graph Algorithms Neo4J](#)
2. Книга [Graph Databases](#)
3. Документація [Cypher Query Language](#)
4. Стаття [Graph Databases for Beginners: Graph Theory & Predictive Modeling](#)
5. [Neo4j Cypher Refcard 4.0](#)