

1. Zainstalować serwer neo4j lokalnie (<https://neo4j.com/download-center/#releases>)

2. Wgrać bazę:

```
LOAD CSV WITH HEADERS FROM 'https://neo4j.com/docs/cypher-manual/3.5/csv/query-tuning/movies.csv' AS
line
MERGE (m:Movie { title: line.title })
ON CREATE SET m.released = toInteger(line.released), m.tagline = line.tagline
LOAD CSV WITH HEADERS FROM 'https://neo4j.com/docs/cypher-manual/3.5/csv/query-tuning/actors.csv' AS
line
MATCH (m:Movie { title: line.title })
MERGE (p:Person { name: line.name })
ON CREATE SET p.born = toInteger(line.born)
MERGE (p)-[:ACTED_IN { roles:split(line.roles, ';')}]>(m)
LOAD CSV WITH HEADERS FROM 'https://neo4j.com/docs/cypher-manual/3.5/csv/query-tuning/directors.csv'
AS line
MATCH (m:Movie { title: line.title })
MERGE (p:Person { name: line.name })
ON CREATE SET p.born = toInteger(line.born)
MERGE (p)-[:DIRECTED]>(m)
```

3. Zaimplementować funkcję (wystarczy wykonać jedno zapytanie typu MATCH WHERE i wyświetlić wynik). Np. można korzystać z JDBC:

<https://neo4j.com/blog/neo4j-jdbc-driver-3-0-3-and-3-1-0-released>,
https://neo4j.com/developer/java/#_neo4j_for_java_developers

4. Stworzyć kilka nowych węzły reprezentujących film oraz aktorów w nim występujących, następnie stworzyć relacje ich łączące (np. ACTED_IN)

5. Dodać zapytaniem nowe właściwości nowo dodanych węzłów reprezentujących aktor (np. birthdate oraz birthplace).

6. Ułożyć zapytanie, które zmieni wartość atrybutu węzłów danego typu, jeżeli innych atrybut węzła spełnia zadane kryterium

7. Zapytanie o aktorów którzy grali w conajmniej 2 filmach (użyć *collect* i *length*) i policzyć średnią występów w filmach dla grupy aktorów, którzy wystąpili w conajmniej 3 filmach.

9. Zmienić wartość wybranego atrybutu w węzłach na ścieżce pomiędzy dwoma podanymi węzłami

10. Wyświetlić węzły, które znajdują się na 2 miejscu na ścieżkach o długości 4 pomiędzy dwoma wybranymi węzłami.

11. Porównać czas wykonania zapytania o wybranego aktora bez oraz z indeksem w bazie nałożonym na atrybut name (DROP INDEX i CREATE INDEX oraz użyć komendy PROFILE/EXPLAIN).

12. Spróbować dokonać optymalizacji wybranych dwóch zapytań z poprzednich zadań (załączyć przykłady w sprawozdaniu).

13. Napisać kod, które wygeneruje drzewo rozpinające z bazy (z poziomu javy lub pythona, nie musi być minimalne) - (można wygenerować własny mały graf do realizacji zadania, zadanie na liczbę punktów powyżej 5.0)

Dowolność w wyborze języka wysokiego poziomu (Java, Python itd.).

Termin oddania dla wszystkich grup 10.01.2020