Baza danych Neo4J

Celem ćwiczenia jest zapoznanie studentek i studentów z możliwościami modelowania rzeczywistości przy użyciu grafowej bazy danych Neo4J oraz strukturą i składnią języka zapytań Cypher do grafowej bazy danych

praca z bazą danych

1. Wejdź na stronę https://beta.graphenedb.com/ i utwórz konto a następnie utwórz bazę danych i zaloguj się do serwisu lub wejdź na stronę http://www.neo4j.org/download i pobierz oraz zainstaluj lokalnie bazę danych.

2. Uruchom narzędzie `Neo4j Web Admin interface`

3. Rozpocznij pracę od utworzenia pojedynczego węzła reprezentującego Toma Hanksa

```
CREATE (n:Actor { name:"Tom Hanks" });
```

4. Wydaj zapytanie i wyświetl utworzony wierzchołek

```
MATCH (actor:Actor { name: "Tom Hanks" })
RETURN actor;
```

5. Odnotuj fakt występu T.Hanksa w filmie „Bezsenność w Seattle”

```
MATCH (actor:Actor)
WHERE actor.name = "Tom Hanks"
CREATE (movie:Movie { title:'Sleepless IN Seattle' })
CREATE (actor)-[:ACTED_IN]->(movie);
```

6. Znajdź Toma Hanksa i dodaj do własności wierzchołka informacje o dacie urodzenia aktora

```
MATCH (actor:Actor { name: "Tom Hanks" })
SET actor.DoB = 1944
RETURN actor.name, actor.DoB;
```

7. Rozszerz bazę danych o kilkanaście filmów

```
CREATE (matrix1:Movie { title : 'The Matrix', year : '1999-03-31' })
CREATE (matrix2:Movie { title : 'The Matrix Reloaded', year : '2003-05-07'
})
CREATE (matrix3:Movie { title : 'The Matrix Revolutions', year :
'2003-10-27' })
CREATE (keanu:Actor { name:'Keanu Reeves' })
CREATE (laurence:Actor { name:'Laurence Fishburne' })
CREATE (carrieanne:Actor { name:'Carrie-Anne Moss' })
CREATE (keanu)-[:ACTS_IN { role : 'Neo' }]->(matrix1)
CREATE (keanu)-[:ACTS_IN { role : 'Neo' }]->(matrix2)
```

```
CREATE (keanu)-[:ACTS_IN { role : 'Neo' }]->(matrix3)
CREATE (laurence)-[:ACTS_IN { role : 'Morpheus' }]->(matrix1)
CREATE (laurence)-[:ACTS_IN { role : 'Morpheus' }]->(matrix2)
CREATE (laurence)-[:ACTS_IN { role : 'Morpheus' }]->(matrix3)
CREATE (carrieanne)-[:ACTS_IN { role : 'Trinity' }]->(matrix1)
CREATE (carrieanne)-[:ACTS_IN { role : 'Trinity' }]->(matrix2)
CREATE (carrieanne)-[:ACTS_IN { role : 'Trinity' }]->(matrix3)
```

8. Policz liczbę wierzchołków w grafie

```
MATCH (n)
RETURN "Hello Graph with " + count(*)+ " Nodes!" AS welcome;
```

9. Wyświetl tytuł i rok produkcji filmu „Matrix”

```
MATCH (movie:Movie { title: 'The Matrix' })
RETURN movie.title, movie.year;
```

10. Wyświetl nazwiska wszystkich aktorów uporządkowane alfabetycznie

```
MATCH (actor:Actor)
RETURN actor.name
ORDER BY actor.name;
```

11. Znajdź aktorów których nazwiska kończą sie na literę „s”

```
MATCH (actor:Actor)
WHERE actor.name =~ ".*s$"
RETURN actor.name;
```

12. Wyświetl wszystkie wierzchołki wraz z łączącymi je związkami

```
MATCH (n)-[r]->(m)
RETURN n AS from, r AS `->`, m AS to;
```

modyfikacja danych

13. Dodaj do grafu wierzchołek reprezentujący Ciebie

```
CREATE (me:User { name: "Mikolaj" })
RETURN me;
```

14. Dodaj do grafu swoją ocenę filmu „Matrix”

```
MATCH (me:User { name: "Mikolaj" }),(movie:Movie { title: "The Matrix" })
CREATE (me)-[:RATED { stars : 5, comment : "Uwielbiam ten film" }]->(movie);
```

15. Wyświetl swoje oceny filmów

```
MATCH (me:User { name: "Mikolaj" }),(me)-[rating:RATED]->(movie)
RETURN movie.title, rating.stars, rating.comment;
```

16. Dodaj do bazy swoją koleżankę/swojego kolegę

```
CREATE (friend:User { name: "Marek" })
RETURN friend;
```

17. Dodaj związek łączący Ciebie z koleżanką/kolegą. Poniższe polecenie wykonaj parę razy i sprawdź, czy tworzonych jest wiele związków

```
MATCH (me:User { name: "Mikolaj" }),(friend:User { name: "Marek" })
CREATE UNIQUE (me)-[friendship:FRIEND]->(friend)
RETURN friendship;
```

18. Dodaj sobie dziesięciu fikcyjnych przyjaciół

```
MATCH (me:User { name: "Mikolaj" })
FOREACH (i IN range(1,10)| CREATE (friend:User { name: "Friend " + i
}),(me)-[:FRIEND]->(friend));
```

19. Wyświetl wszystkich swoich przyjaciół

```
MATCH (me:User { name: "Mikolaj" })-[r:FRIEND]->(friend)
RETURN type(r) AS friendship, friend.name;
```

wyszukiwanie w grafowej bazie danych

20. Znajdź wszystkie filmy w których grali aktorzy i aktorki występujący w filmie „Matrix”

```
MATCH (:Movie { title: "The Matrix" })<-[:ACTS_IN]-(actor)-
[:ACTS_IN]->(movie)
RETURN movie.title, count(*)
ORDER BY count(*) DESC ;

MATCH (:Movie { title: "The Matrix" })<-[:ACTS_IN]-(actor)-
[:ACTS_IN]->(movie)
RETURN movie.title, collect(actor.name), count(*) AS count
ORDER BY count DESC ;
```

21. Znajdź pary aktorów którzy wspólnie występowali

```
MATCH (:Movie { title: "The Matrix"})<-[:ACTS_IN]-(actor)-
[:ACTS_IN]->(movie)<-[:ACTS_IN]-(colleague)
RETURN actor.name, collect(DISTINCT colleague.name);
```

22. Znajdź wszystkie ścieżki między Keanu Reevesem i Carrie-Anne Moss

```
MATCH p =(:Actor { name: "Keanu Reeves" })-[:ACTS_IN*0..5]-(:Actor { name:
"Carrie-Anne Moss" })
RETURN p, length(p)
LIMIT 10;
```

23. Ogranicz poprzednie zapytanie jedynie do nazwisk i tytułów filmów

```
MATCH p =(:Actor { name: "Keanu Reeves" })-[:ACTS_IN*0..5]-(:Actor { name:
"Carrie-Anne Moss" })
RETURN extract(n IN nodes(p)| coalesce(n.title,n.name)) AS `names AND
titles`, length(p)
ORDER BY length(p)
LIMIT 10;
```

Etykiety i ograniczenia

24. Zdefiniuj ograniczenie unikalnościowe na tytule filmu

```
CREATE CONSTRAINT ON (movie:Movie) ASSERT movie.title IS UNIQUE
```

25. Załóż indeks na nazwisku aktora

```
CREATE INDEX ON :Actor(name)
```

26. Dodaj ponownie informację o tym, że Tom Hanks grał w „Bezsenności w Seattle”

```
CREATE (actor:Actor { name:"Tom Hanks" }),(movie:Movie { title:'Sleepless IN
Seattle' }),(actor)-[:ACTED_IN]->(movie);
```

27. Dodaj, a następnie usuń, dodatkową etykietę opisującą Toma Hanksa

```
MATCH (actor:Actor { name: "Tom Hanks" }) SET actor :American;

MATCH (actor:Actor { name: "Tom Hanks" }) REMOVE actor:American;
```

ćwiczenie samodzielne

28. Pobierz poniższy kod aby utworzyć graf zawierający postacie z „Gry o Tron”

```
CREATE (westeros { name: "Westeros" })
CREATE (targaryen { house:"Targaryen" }),(stark { house:"Stark"
}),(lannister { house:"Lannister" }),(baratheon { house:"Baratheon"
}),(tully { house:"Tully" })
FOREACH (house IN [stark,lannister,baratheon,targaryen,tully] |
  CREATE house-[:HOUSE]->westeros)
  CREATE (danaerys { name:"Danaerys" }), danaerys-[:OF_HOUSE]->targaryen,
         (drogo { name:"Khal Drogo" }), danaerys-[:MARRIED_TO]->drogo,
```

```
        (tywin { name:"Tywin" }), tywin-[:OF_HOUSE]->lannister,
        (steffon { name:"Steffon" }), steffon-[:OF_HOUSE]->baratheon,
        (rickard { name:"Rickard" }), rickard-[:OF_HOUSE]->stark,
        (ned { name:"Eddard" }), ned-[:CHILD_OF]->rickard,
        (catelyn { name:"Catelyn" }), catelyn-[:MARRIED_TO]->ned, catelyn-
[:OF_HOUSE]->tully,
        (jon { name:"Jon" }), jon-[:CHILD_OF]->ned
FOREACH (child IN ["Robb", "Bran", "Arya", "Sansa", "Rickon"]| CREATE UNIQUE
ned<-[:CHILD_OF]-({ name:child })-[:CHILD_OF]->catelyn)
FOREACH (child IN ["Cersei", "Jamie", "Tyrion"]| CREATE UNIQUE tywin<-
[:CHILD_OF]-({ name:child }))
FOREACH (brother IN ["Robert", "Renly", "Stannis"]| CREATE UNIQUE steffon<-
[:CHILD_OF]-({ name:brother }))
FOREACH (child IN ["Joffrey", "Myrcella", "Tommen"]| CREATE UNIQUE tywin<-
[:CHILD_OF]-(jamie { name:"Jamie" })<-[:CHILD_OF]-({ name:child })-
[:CHILD_OF]->(cersei { name:"Cersei" })-[:CHILD_OF]->tywin)
                                                    CREATE UNIQUE steffon<-
[:CHILD_OF]-(robert { name:"Robert" })<-[:MARRIED_TO]-(cersei {
name:"Cersei" })-[:CHILD_OF]->tywin
                                                    CREATE UNIQUE ned<-
[:CHILD_OF]-(sansa { name:"Sansa" })-[:PROMISED_TO]->(joffrey {
name:"Joffrey" })-[:CHILD_OF]->cersei
                                                    CREATE UNIQUE ned<-
[:CHILD_OF]-(sansa)-[:MARRIED_TO]->(tyrion { name:"Tyrion" })-
[:CHILD_OF]->steffon
```

Napisz poniższe zapytania:

- dla każdego rodu wyświetl kolekcję imion przedstawicielek i przdstawicieli rodu
- wyświetl wszystkich potomków Rickarda Starka
- wyświetl wszystkie osoby związane rodzinnie lub małżeńsko z rodem Lannisterów

30. Utwórz swoją własną grafową bazę danych opisującą postacie występujące w dowolnej książce polskiej autorki lub autora, oraz zdefiniuj relacje między tymi postaciami oraz dowolne inne cechy tych postaci (np. własności odnoszące się do miejsca zamieszkania, zawodu, czynności, itp.)