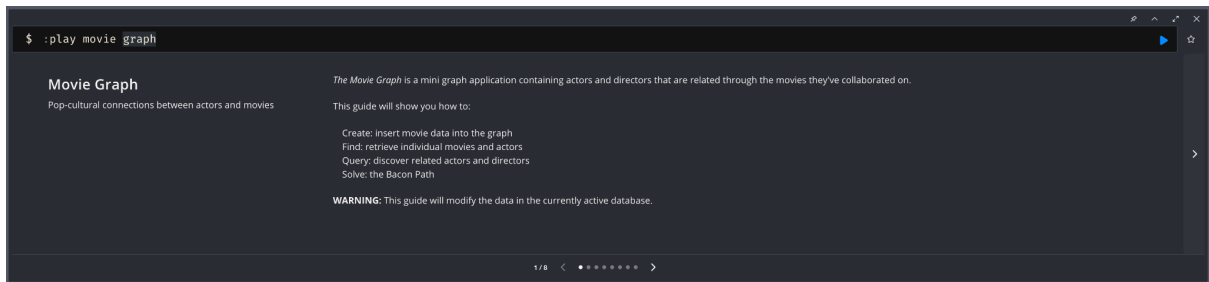
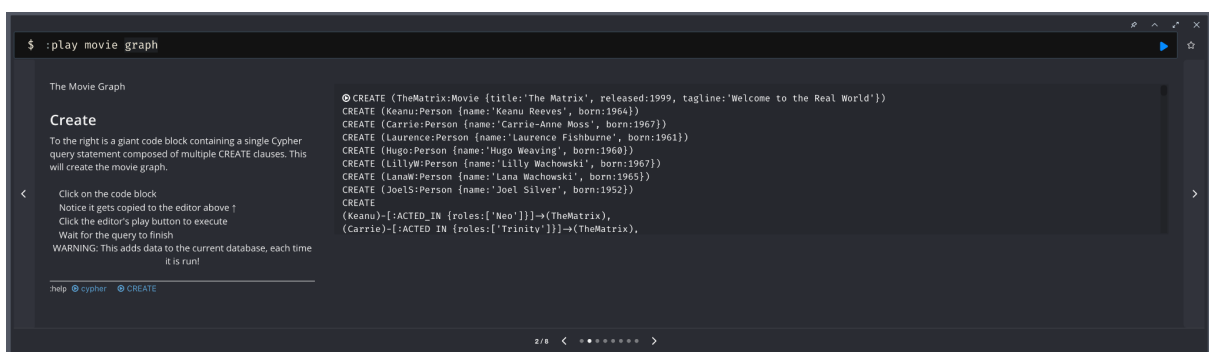


Neo4j – Podstawowe zapytania

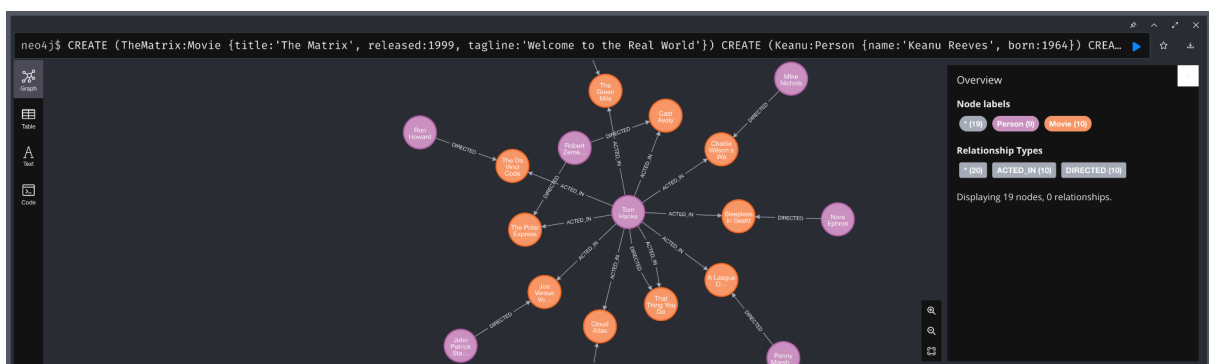
Gdy masz już uruchomiony pusty sandbox Neo4j, wypełnij bazę danych informacjami o filmach. Wykonasz to wpisując do wiersza poleceń Neo4j: `:play movie graph` i klikając przycisk odtwarzania lub `ctrl+enter`.



Na drugim slajdzie znajdziesz gotowe metody potrzebne do dodania danych.



Wykonaj je.

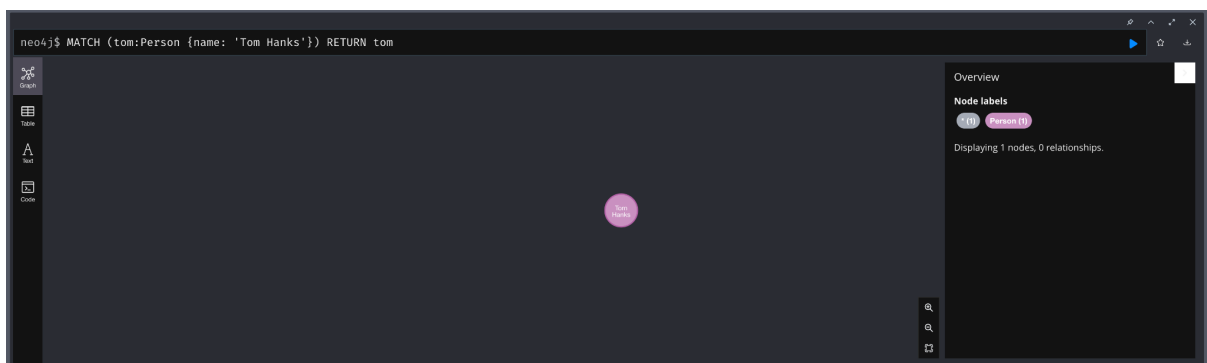


Dokonaj wizualnej eksploracji bazy danych klikając na węzły i połączenia między nimi.

Podstawowe zapytania

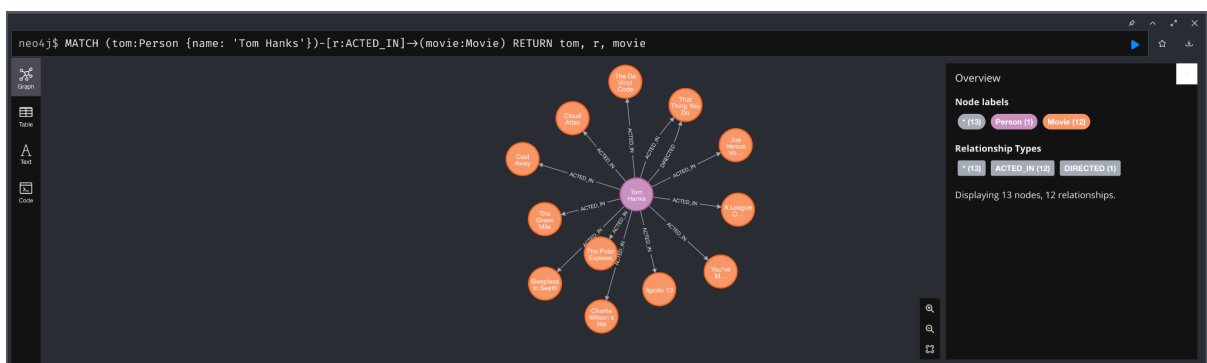
Po wizualnej eksploracji danych powinieneś poznać atrybuty węzłów oraz połączeń. Na początek wykonajmy zapytanie które zwróci nam jednego interesującego nas aktora.

`MATCH (tom:Person {name: 'Tom Hanks'}) RETURN tom`



Po znalezieniu interesującego nas aktora możemy pobrać wszystkie jego filmy. Wykorzystajmy w zapytaniu relacje `ACTED_IN` która pomoże osiągnąć nam cel. Twoje wyniki powinny wyglądać jak sieć powyżej.

`MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie:Movie) RETURN tom, r, movie`

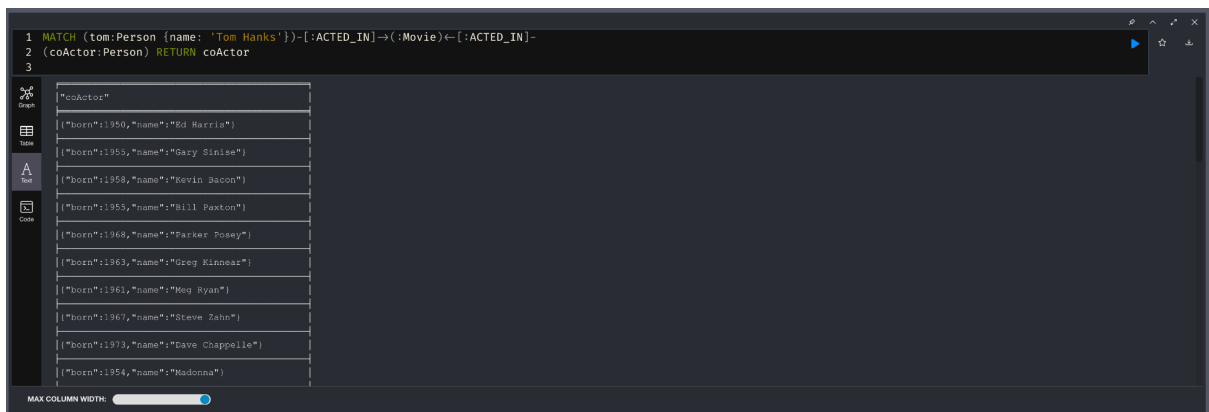
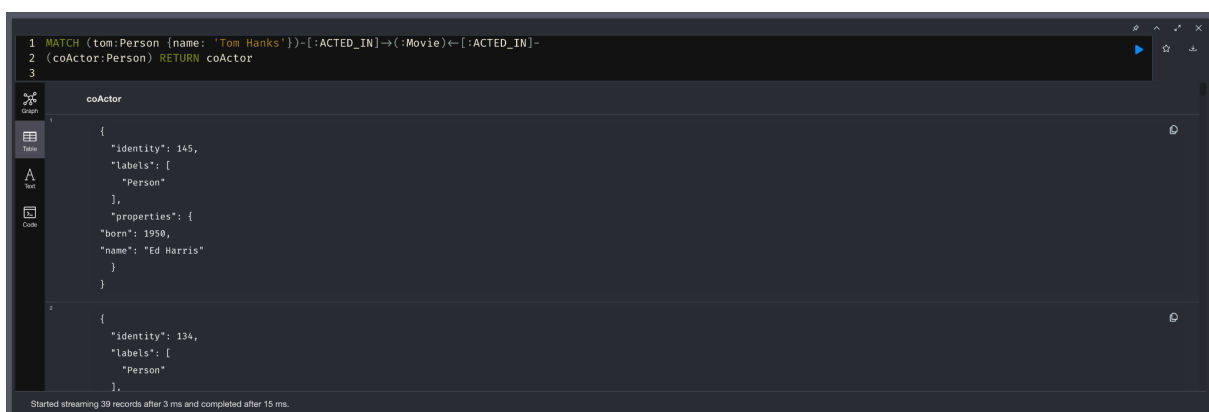


Teraz wyszukajmy w bazie aktorów którzy grali razem z Tomem w filmach.

`MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie:Movie)-[:ACTED_IN]->(coActor:Person) RETURN coActor.name`



Otrzymane wyniki możemy również przestawić w formie tekstowej i tabelarycznej.



Węzły sąsiednie drugiego stopnia

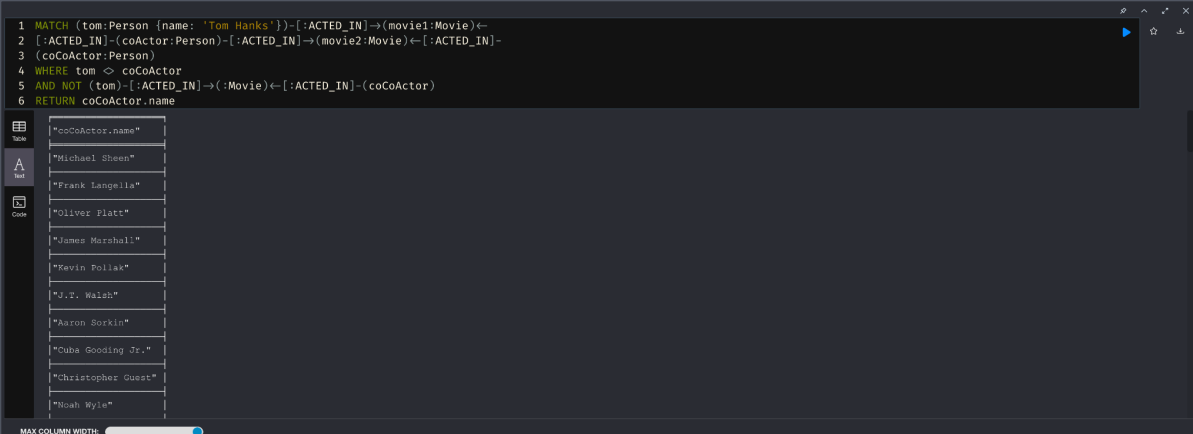
Możemy teraz przekształcić powyższe zapytanie w zapytanie o aktorów którzy współpracowali z aktorami z którymi współpracował Tom. Taką relację nazywamy sąsiadami drugiego stopnia.

```
MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie1:Movie)-[:ACTED_IN]-
(coActor:Person)-[:ACTED_IN]->(movie2:Movie)-[:ACTED_IN]-
(coCoActor:Person)
```

```

WHERE tom <> coCoActor
AND NOT (tom)-[:ACTED_IN]->(:Movie)<-[:ACTED_IN]-(coCoActor)
RETURN coCoActor.name

```



```

1 MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie1:Movie)<-
2 [:ACTED_IN]-(coActor:Person)-[:ACTED_IN]->(movie2:Movie)<-[:ACTED_IN]-(
3 coCoActor:Person)
4 WHERE tom <> coCoActor
5 AND NOT (tom)-[:ACTED_IN]->(:Movie)<-[:ACTED_IN]-(coCoActor)
6 RETURN coCoActor.name

```

coCoActor.name
"Michael Sheen"
"Frank Langella"
"Oliver Platt"
"James Marshall"
"Kevin Pollak"
"J.J. Walsh"
"Aaron Sorkin"
"Cuba Gooding Jr."
"Christopher Guest"
"Noah Wyle"

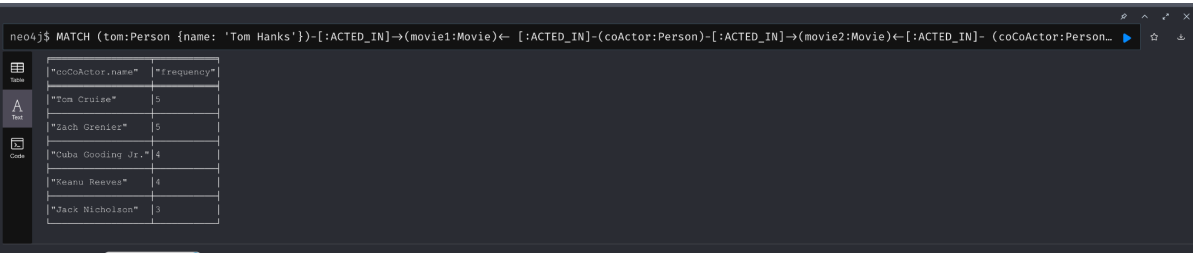
Prawdopodobnie zauważyłeś, że kilka nazw pojawia się wielokrotnie. Dzieje się tak, ponieważ istnieje wiele ścieżek, którymi można podążać od Toma Hanksa do tych aktorów.

Aby zobaczyć, którzy sąsiedzi drugiego stopnia pojawiają się najczęściej w sieci Toma, możemy wziąć pod uwagę częstotliwość występowania, licząc ścieżki między Tomem Hanksem a każdym coCoActorem i ustawiając je według wartości od najwyższej do najniższej.

```

MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie1:Movie)<-
[:ACTED_IN]-(coActor:Person)-[:ACTED_IN]->(movie2:Movie)<-[:ACTED_IN]-(
coCoActor:Person)
WHERE tom <> coCoActor
AND NOT (tom)-[:ACTED_IN]->(:Movie)<-[:ACTED_IN]-(coCoActor)
RETURN coCoActor.name, count(coCoActor) as frequency
ORDER BY frequency DESC
LIMIT 5

```



```

neo4j$ MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie1:Movie)<- [:ACTED_IN]-(coActor:Person)-[:ACTED_IN]->(movie2:Movie)<-[:ACTED_IN]-( coCoActor:Person...

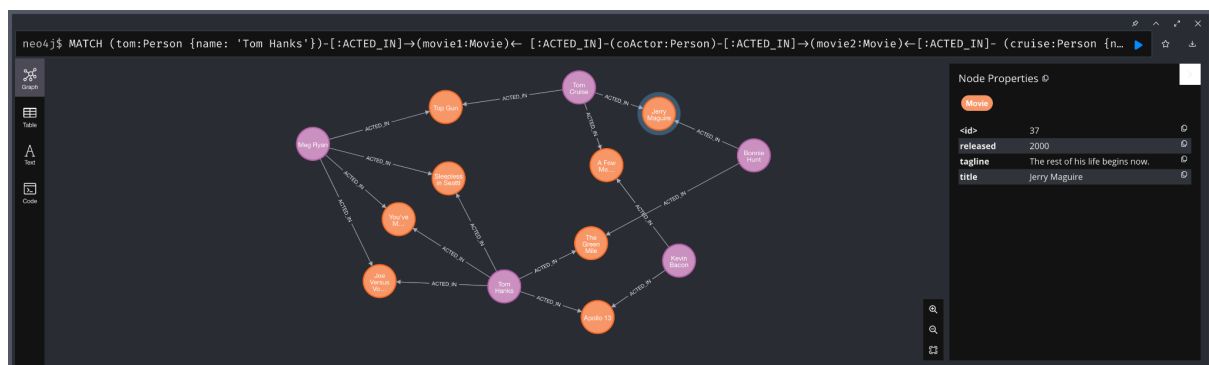
```

coCoActor.name	frequency
"Tom Cruise"	5
"Zach Grenier"	5
"Cuba Gooding Jr."	4
"Keanu Reeves"	4
"Jack Nicholson"	3

Jednym z tych sąsiadów jest Tom Cruise. Zobaczmy teraz, które filmy i aktorzy są między dwoma Tomami.

Eksploracja ścieżek/połączeń

```
MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(movie1:Movie)-[:ACTED_IN]-(coActor:Person)-[:ACTED_IN]->(movie2:Movie)-[:ACTED_IN]-(cruise:Person {name: 'Tom Cruise'})
WHERE NOT (tom)-[:ACTED_IN]->(:Movie)-[:ACTED_IN]-(cruise)
RETURN tom, movie1, coActor, movie2, cruise
```



Jak widać, zapytanie zwraca wiele ścieżek.

Jeśli kiedykolwiek słyszałeś o sześciu stopniach Kevina Bacona, to właśnie przedstawiliśmy ścieżkę między Tomem Hanksem a Kevinem Baconem. Koncepcja opiera się na fenomenie “małego świata” i zjawisku "sześciu stopni separacji".

Inne rekomendacje

Możesz zastosować te same pomysły, których się tu nauczyłeś, do wielu innych zastosowań, aby polecać produkty i usługi, znajdować restauracje lub zajęcia, które mogą Ci się spodobać.

Zadanie

Usuń istniejące dane w bazie poleceniem: `MATCH (n) DETACH DELETE n`

Odtwórz poniższą strukturę w bazie danych. Wcielając się w rolę błękitnego węzła przygotuj zapytanie w języku Cypher które zwórci nam odpowiedź na poniższe pytanie.

Jakie restauracje sushi w Nowym Jorku lubią moi przyjaciele?

