
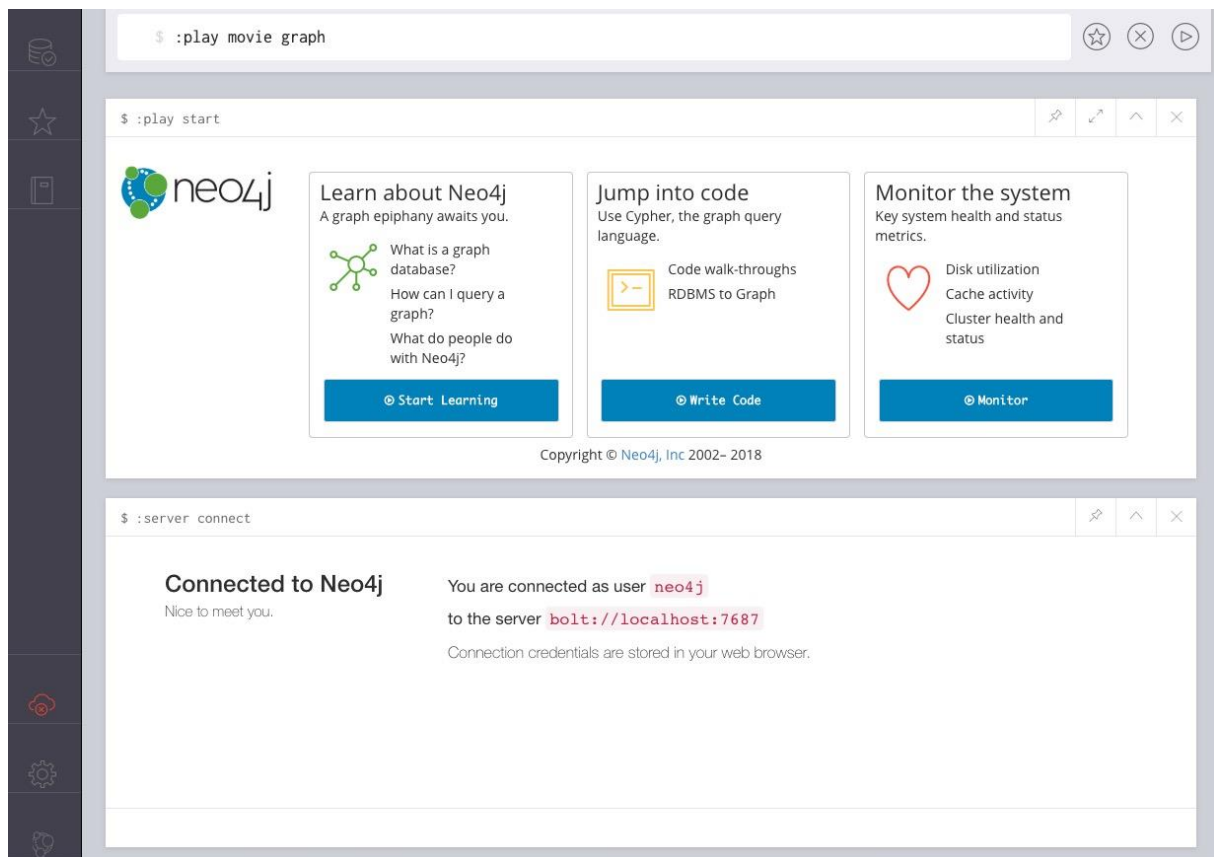


Neo4j – Tworzenie węzłów za pomocą Cypher

Gdy masz już uruchomioną instancję Neo4j, wypełnij zbiór danych filmu, wpisując `:play movie graph` do wiersza poleceń przeglądarki Neo4j i klikając przycisk odtwarzania ().



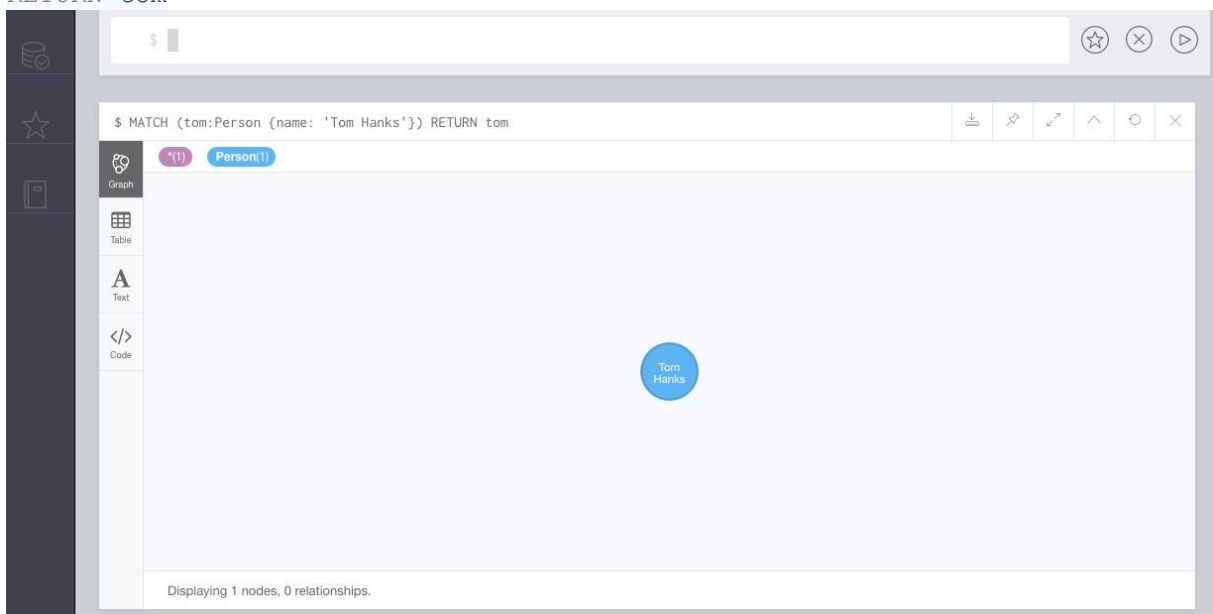
Przejdź do drugiego slajdu (za pomocą strzałki w prawo) okienka wyświetlanego poniżej wiersza polecenia, kliknij zapytanie i uruchom je.

Te dane zostały utworzone i przechowywane w bazie danych, abyśmy mogli je przesyłać. Następna sekcja pokaże, jak napisać kilka zapytań, aby zbadać dane, które właśnie utworzyłeś.

Podstawowe zapytania

Zanim zaczniemy polecać rzeczy, musimy dowiedzieć się, co jest interesujące w naszych danych, aby zobaczyć, jakie rzeczy możemy i chcemy polecać. Na początek musimy uruchomić takie zapytanie, aby znaleźć jednego aktora, takiego jak Tom Hanks.

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



Po znalezieniu interesującego nas aktora możemy pobrać wszystkie jego filmy, zaczynając od węzła Toma Hanksa i śledząc relacje `ACTED_IN`. Twoje wyniki powinny wyglądać jak sieć powyżej.

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



Oczywiście Tom ma kolegów, którzy grali z nim w jego filmach. Zapytanie dotyczące znalezienia współpracowników Toma wygląda następująco:

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

The image shows the Neo4j Cypher query interface with the same query as above. The results are displayed in a table with the column header `coActor.name`. The table contains 12 rows of co-actor names. The status bar at the bottom indicates "Started streaming 39 records after 43 ms and completed after 46 ms."

coActor.name
"Julia Roberts"
"Philip Seymour Hoffman"
"Geena Davis"
"Lori Petty"
"Rosie O'Donnell"
"Madonna"
"Bill Paxton"
"Helen Hunt"
"Kevin Bacon"
"Gary Sinise"
"Ed Harris"
"Bill Paxton"
"Michael Clarke Duncan"
"Patricia Clarkson"
"Gary Sinise"
"Bonnie Hunt"

Zalecenia dotyczące filtrowania opartego na współpracy

Możemy teraz przekształcić powyższe zapytanie o współtwórcę w zapytanie o rekomendację, wykonując kolejne kroki w celu znalezienia „współ-aktorów”, tj. Aktorów drugiego stopnia w sieci Toma. To pokaże nam wszystkich aktorów, z którymi Tom mógł jeszcze nie pracować, i możemy określić kryteria, aby mieć pewność, że nie działał bezpośrednio z tą osobą.

```
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
```

coCoActor.name
"Zach Grenier"
"Zach Grenier"
"Zach Grenier"
"Cuba Gooding Jr."
"Jack Nicholson"
"Oliver Platt"
"Michael Sheen"
"Frank Langella"
"Aaron Sorkin"
"Christopher Guest"
"Kevin Pollak"
"Cuba Gooding Jr."
"James Marshall"
"J.T. Walsh"
"Kiefer Sutherland"
"Noah Wyle"

Started streaming 81 records after 3 ms and completed after 24 ms.

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 współ-aktorzy 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
```

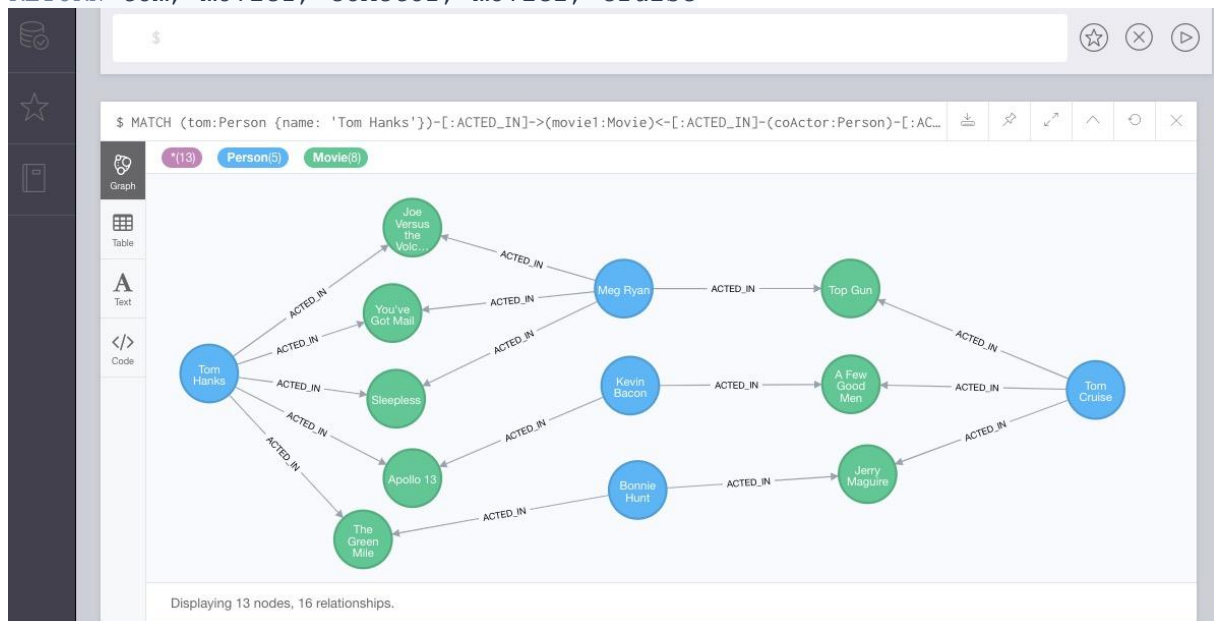
coCoActor.name	frequency
"Tom Cruise"	5
"Zach Grenier"	5
"Cuba Gooding Jr."	4
"Keanu Reeves"	4
"Kelly McGillis"	3

Started streaming 5 records after 29 ms and completed after 30 ms.

Jednym z tych „współaktorów” jest Tom Cruise. Zobaczmy teraz, które filmy i aktorzy są między dwoma Tomami, abyśmy mogli dowiedzieć się, kto może ich przedstawić.

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, zwraca to wiele ścieżek. Jeśli kiedykolwiek grałeś w grę o sześciu stopniach Kevina Bacona, ta koncepcja zobaczenia, ile przeskoków istnieje między ludźmi, jest dokładnie tym, co przedstawiają wykresy. Zauważysz, że nasze wyniki zwracają nawet ścieżkę z samym Kevinem Baconem.

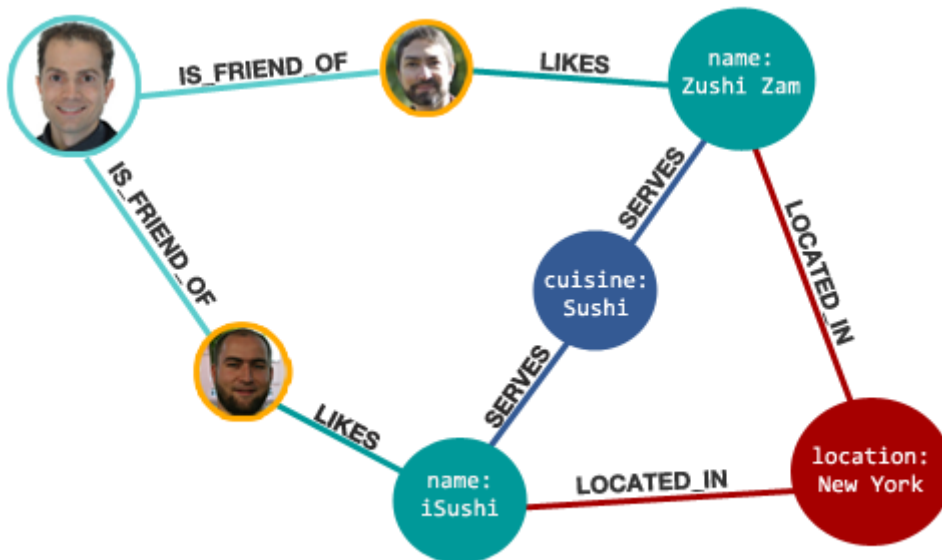
Za pomocą tych dwóch prostych instrukcji Cypher stworzyliśmy już dwa algorytmy rekomendacji - z kim się spotkać / z kim pracować i jak się z nimi spotkać..

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, lub nawiązywać kontakty z innymi kolegami, którzy mają podobne zainteresowania. Wspomnimy o kilku konkretnie tutaj, w których znajdziesz zasoby, z których możesz skorzystać, aby znaleźć więcej informacji.

Restauracja

Mamy wykres kilku znajomych z ich ulubionymi restauracjami, kuchnią i lokalizacjami.



Praktyczne pytanie, na które należy odpowiedzieć, sformułowane jako przeszukiwanie wykresu, to:

Jakie restauracje sushi w Nowym Jorku lubią moi przyjaciele?

Jak możemy przetłumaczyć to na odpowiednią instrukcję Cypher?