

# Sprawozdanie z zadania nr 14

Wypełnienie zbioru danych filmu.

\$ :play movie graph

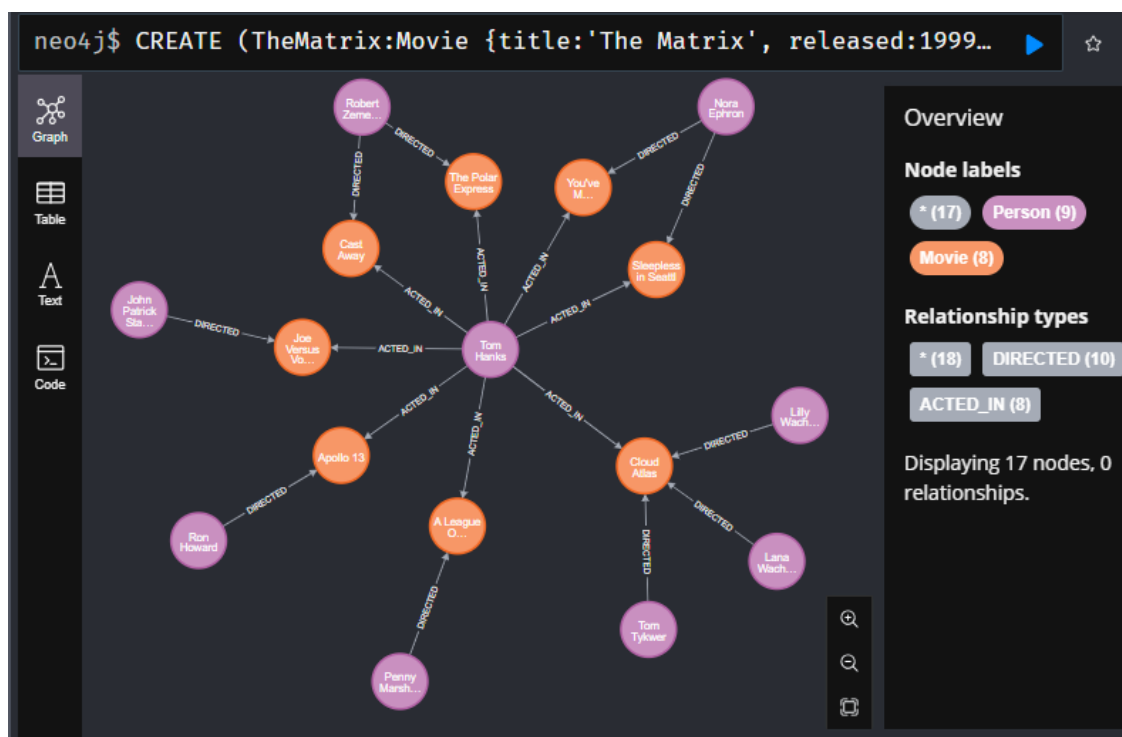
The Movie Graph

### Create

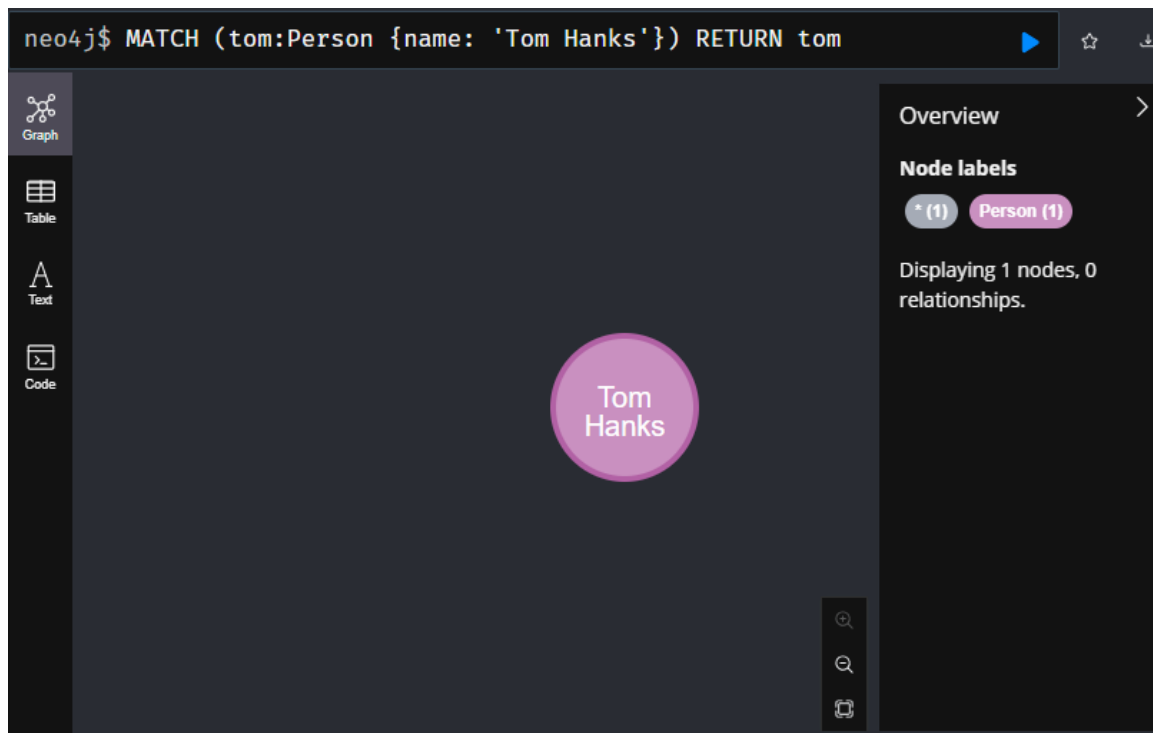
To the right is a giant code block containing a single Cypher query statement composed of multiple CREATE clauses. This will create the movie graph.

Click on the code block  
Notice it gets copied to the editor above ↑  
Click the editor's play button to execute  
Wait for the query to

```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
CREATE (LillyW:Person {name:'Lilly Wachowski', born:1967})
CREATE (LanaW:Person {name:'Lana Wachowski', born:1965})
```

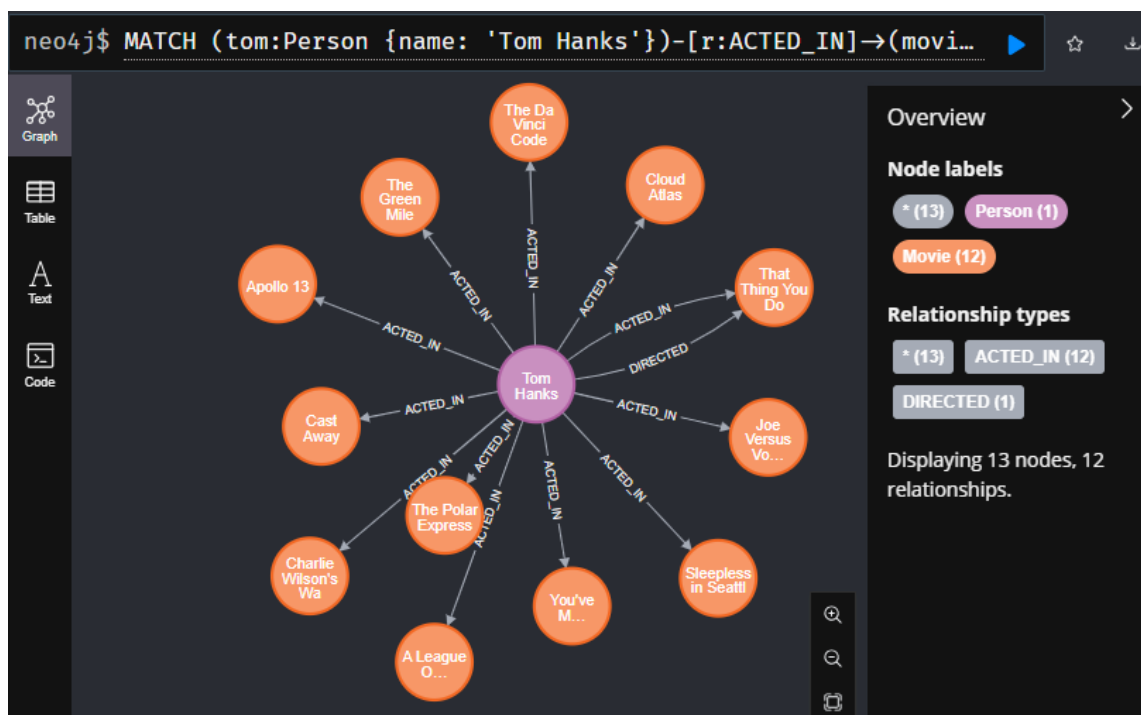


Znalezienie jednego aktora (w tym przypadku Tom Hanks).



Zapytanie zwracające wynik (w formie grafu) przedstawiający filmy, w których zagrał Tom Hanks.

*MATCH (tom:Person {name: 'Tom Hanks'})-[r:ACTED\_IN]->(movie:Movie) RETURN tom, r, movie*



**Zapytanie zwracające wynik (w formie tabeli) przedstawiający aktorów, którzy grali z nim w jego filmach. (39 rekordów)**

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



The screenshot shows a Neo4j query interface. At the top, a query bar contains the Cypher query: `neo4j$ MATCH (tom:Person {name: 'Tom Hanks'})-[:ACTED_IN]->(:Movie)-[:ACTED_IN] (coActor:Person) RETURN coActor.name`. Below the query bar, a sidebar on the left offers three view options: 'Table' (selected), 'Text', and 'Code'. The main area displays the results in a table format. The table has a single column header 'coActor.name'. The results are listed in six rows, numbered 1 to 6. The first row shows 'Meg Ryan', the second 'Greg Kinnear', the third 'Parker Posey', the fourth 'Dave Chappelle', the fifth 'Steve Zahn', and the sixth 'Meg Ryan'.

	coActor.name
1	"Meg Ryan"
2	"Greg Kinnear"
3	"Parker Posey"
4	"Dave Chappelle"
5	"Steve Zahn"
6	"Meg Ryan"

## Wszyscy aktorzy, z którymi Tom mógł jeszcze nie pracować. (81 rekordów)

```
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
```



The image shows a screenshot of the Neo4j query results interface. At the top, the Cypher query is entered in the command line. Below it, the results are displayed in a table view. The table has a single column labeled 'coCoActor.name'. The results list 81 records, with the first few being 'Zach Grenier', 'Jack Nicholson', and 'Cuba Gooding Jr.', followed by several more instances of 'Zach Grenier'. The interface includes a sidebar with icons for Table, Text, and Code views, and a status bar at the bottom indicating that 81 records were streamed after 32 ms and completed after 244 ms.

	coCoActor.name
76	"Zach Grenier"
77	"Jack Nicholson"
78	"Cuba Gooding Jr."
79	"Zach Grenier"
80	"Zach Grenier"
81	"Zach Grenier"

Started streaming 81 records after 32 ms and completed after 244 ms.

Można zauważyć, że kilku aktorów pojawia się kilkakrotnie. Jest to spowodowane tym, że istnieje wiele ścieżek, którymi można podążać od Toma Hanksa do tych aktorów.

Kolejne zapytanie jest podobne do poprzedniego, z tą różnicą, że tutaj zliczana jest częstotliwość występowania i na wyjściu widać 5 z największą jej wartością.

```
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]->(movie...

	coCoActor.name	frequency
1	"Zach Grenier"	5
2	"Tom Cruise"	5
3	"Keanu Reeves"	4
4	"Cuba Gooding Jr."	4
5	"Val Kilmer"	3

Started streaming 5 records after 119 ms and completed after 318 ms.

**Aktorzy, którzy mogliby „zapoznać Toma Hanksa i Toma Cruise’a”,  
tj. aktorzy, którzy grali zarówno w filmach Toma Hanksa, jak i Toma  
Cruise’a.**

*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*

