uc3m | Universidad Carlos III de Madrid

OpenCourseWare

Database

Lourdes Moreno López

Paloma Martínez Fernández

José Luis Martínez Fernández

Rodrigo Alarcón García

Lab demo 5 (Topic neo4j (3.4))

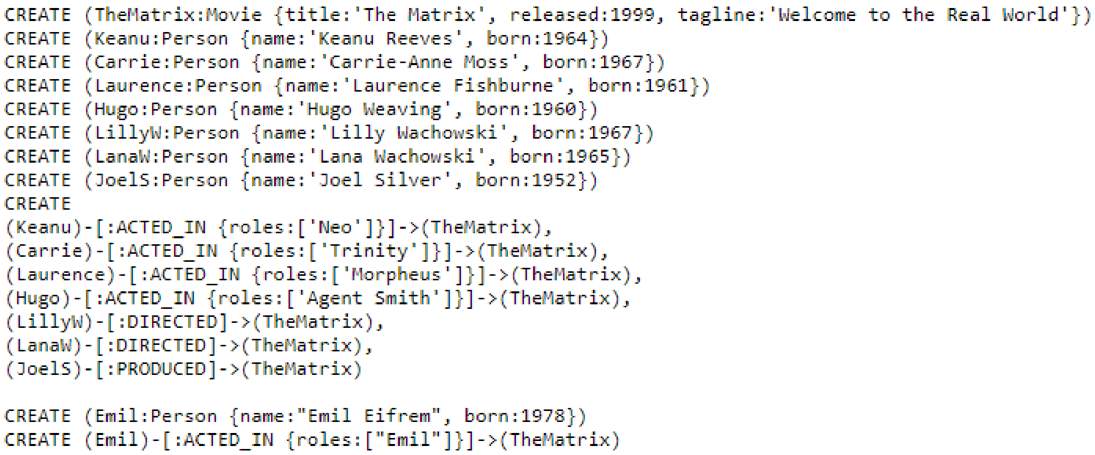
Download the script movie (see Figure 1) (source: neo4j official site)

The Movie Graph is a mini graph application containing actors and directors that are related through the movies they've collaborated on.

The Movie database contains 171 nodes and 253 relationships.

This guide will show you how to:

- Create: insert movie data into the graph
- Find: retrieve individual movies and actors
- Query: discover related actors and directors



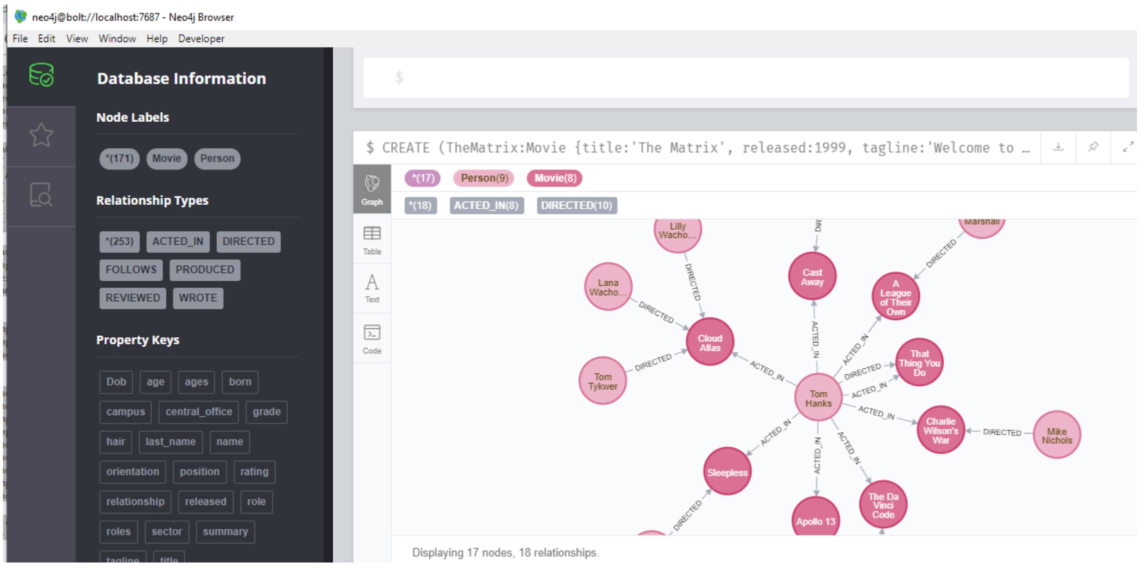
```
CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to the Real World'})
CREATE (Keanu:Person {name:'Keanu Reeves', born:1964})
CREATE (Carrie:Person {name:'Carrie-Anne Moss', born:1967})
CREATE (Laurence:Person {name:'Laurence Fishburne', born:1961})
CREATE (Hugo:Person {name:'Hugo Weaving', born:1960})
CREATE (LillyW:Person {name:'Lilly Wachowski', born:1967})
CREATE (LanaW:Person {name:'Lana Wachowski', born:1965})
CREATE (JoelS:Person {name:'Joel Silver', born:1952})
CREATE
(Keanu)-[:ACTED_IN {roles:['Neo']}]->(TheMatrix),
(Carrie)-[:ACTED_IN {roles:['Trinity']}]->(TheMatrix),
(Laurence)-[:ACTED_IN {roles:['Morpheus']}]->(TheMatrix),
(Hugo)-[:ACTED_IN {roles:['Agent Smith']}]->(TheMatrix),
(LillyW)-[:DIRECTED]->(TheMatrix),
(LanaW)-[:DIRECTED]->(TheMatrix),
(JoelS)-[:PRODUCED]->(TheMatrix)

CREATE (Emil:Person {name:"Emil Eifrem", born:1978})
CREATE (Emil)-[:ACTED_IN {roles:["Emil"]}]->(TheMatrix)
```

Figure 1

Part 1: Create nodes and relationships with their properties

Execute the script (see figure 2)



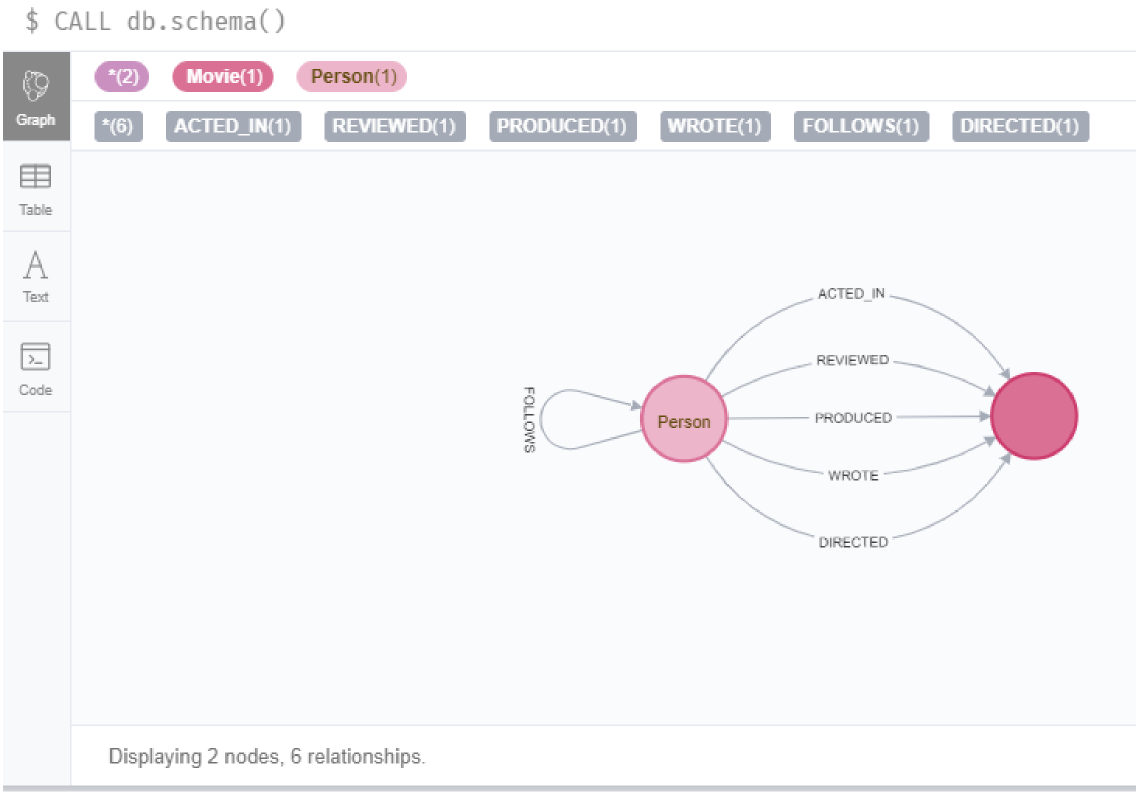
```
$ CREATE (TheMatrix:Movie {title:'The Matrix', released:1999, tagline:'Welcome to …
```

Displaying 17 nodes, 18 relationships.

Figure 2

Write a query to display the schema of your database. (see Figure 3)

```
CALL db.schema()
```



```
$ CALL db.schema()
```

Displaying 2 nodes, 6 relationships.

Figure 3

Part 2 Queries

Write a query to retrieve all nodes from the database.

```
MATCH (n) RETURN n
```

Write a query to retrieve all Person nodes.

```
MATCH (p:Person) RETURN p
```

Write a query to retrieve all Movie nodes.

```
MATCH (p:Movie) RETURN p
```

Find the actor named "Tom Hanks"

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

Find the movie with title "Cloud Atlas"

```
MATCH (cloudAtlas {title: "Cloud Atlas"}) RETURN cloudAtlas
```

Find movies released in the 1990s...

```
MATCH (nineties:Movie) WHERE nineties.released >= 1990 AND
nineties.released < 2000 RETURN nineties.title
```

List all Tom Hanks movies

```
MATCH (tom:Person {name: "Tom Hanks"})-[:ACTED_IN]->(tomHanksMovies)
RETURN tom,tomHanksMovies
```

Who directed "Cloud Atlas"?

```
MATCH (cloudAtlas {title: "Cloud Atlas"})<-[:DIRECTED]-(directors)
RETURN directors.name
```

Tom Hanks' co-actors...

```
MATCH (tom:Person {name:"Tom Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-
(coActors) RETURN coActors.name
```

Retrieve all Movie nodes from the database and return the title, released, and tagline values.

```
MATCH (m:Movie) RETURN m.title, m.released, m.tagline
```

Retrieve all people name who wrote the movie Speed Racer.

```
MATCH (p:Person)-[:WROTE]->(:Movie {title: 'Speed Racer'}) RETURN
p.name
```

Retrieve all movies that Tom Cruise acted in and return their titles.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE a.name = 'Tom Cruise'
RETURN m.title as Movie
```

Retrieve all people that were born in the 70’s and return their names and year born.

```
MATCH (a:Person)
WHERE a.born >= 1970 AND a.born < 1980
RETURN a.name as Name, a.born as `Year Born`
```

Retrieve the actors who acted in the movie The Matrix who were born after 1960, and return their names and year born.

```
MATCH (a:Person)-[:ACTED_IN]->(m:Movie)
WHERE a.born > 1960 AND m.title = 'The Matrix'
RETURN a.name as Name, a.born as `Year Born`
```

Retrieve all actors whose name begins with James, returning their names.

```
MATCH (a:Person)-[:ACTED_IN]->(:Movie)
WHERE a.name STARTS WITH 'James'
RETURN a.name
```

Retrieve all people who have produced a movie, but have not directed a movie, returning their names and the movie titles.

```
MATCH (a:Person)-[:PRODUCED]->(m:Movie)
WHERE NOT ((a)-[:DIRECTED]->(:Movie))
RETURN a.name, m.title
```

Retrieve all movies that were released in the years 2000, 2004, and 2008, returning their titles and release years.

```
MATCH (m:Movie)
WHERE m.released in [2000, 2004, 2008]
RETURN m.title, m.released
```

Retrieve actors and the movies they have acted in, returning each actor's name and the list of movies they acted in.

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)
RETURN p.name as actor, collect(m.title) AS `movie list`
```

Retrieve all movies that Tom Cruise has acted in and the co-actors that acted in the same movie, returning the movie title and the list of co-actors that Tom Cruise worked with.

```
MATCH (p:Person)-[:ACTED_IN]->(m:Movie)<-[:ACTED_IN]-(p2:Person)
WHERE p.name ='Tom Cruise'
RETURN m.title as movie, collect(p2.name) AS `co-actors`
```

Retrieve all directors, their movies, and people who acted in the movies, returning the name of the director, the number of actors the director has worked with, and the list of actors.

```
MATCH (d:Person)-[:DIRECTED]->(m:Movie)<-[:ACTED_IN]-(a:Person)
RETURN d.name AS director, count(a) AS `number actors` ,
collect(a.name) AS `actors worked with`
```

Retrieve the top 5 ratings and their associated movies, returning the movie title and the rating.

```
MATCH (:Person)-[r:REVIEWED]->(m:Movie)
RETURN  m.title AS movie, r.rating AS rating
ORDER BY r.rating DESC LIMIT 5
```

Query the graph to display Forrest Gump and all of its relationships.

```
MATCH (p:Person)-[rel]-(m:Movie)
WHERE m.title = 'Forrest Gump'
RETURN p, rel, m
```

Part 3 Create, delete and update nodes and relationships with their properties

Create a Movie node for the movie with the title, Forrest Gump.

```
CREATE (:Movie {title: 'Forrest Gump'})
```

Create a Person node for the person with the name, Robin Wright.

```
CREATE (:Person {name: 'Robin Wright'})
```

Add the label OlderMovie to any Movie node that was released before 2010.

```
MATCH (m:Movie)
WHERE m.released < 2010
SET m:OlderMovie
RETURN DISTINCT labels(m)
```

Add the label Female to all Person nodes that has a person whose name starts with Robin.

```
MATCH (p:Person)
WHERE p.name STARTS WITH 'Robin'
SET p:Female
```

Add the following properties to the movie, Forrest Gump:

- released: 1994
- tagline: Life is like a box of chocolates…you never know what you’re gonna get.
- lengthInMinutes: 142

```
MATCH (m:Movie)
WHERE m.title = 'Forrest Gump'
SET m:OlderMovie,
    m.released = 1994,
    m.tagline = "Life is like a box of chocolates...you never know 
what you're gonna get.",
    m.lengthInMinutes = 142
```

Delete Forrest Gump, along with its relationships in the graph.

```
MATCH (m:Movie)
WHERE m.title = 'Forrest Gump'
DETACH DELETE m
```