Giacomo Garbarino, Manuel Parmiggiani

# ASSIGNMENT IV: NEO4J

**EXERCISE 1**

1. MATCH (p) - [:ACTED_IN] -> (m) RETURN distinct(p.name)
2. MATCH ({name : 'Lana Wachowski'}) - [:DIRECTED] ->(m:Movie) <- [:ACTED_IN] – (p) WHERE m.released < 2005 RETURN p.name, m.title
3. MATCH (m1:Movie) <- [a1:ACTED_IN] - (), (m2:Movie) <- [a2:ACTED_IN] - () WHERE m1.title<>m2.title AND a1.roles = a2.roles RETURN a1.roles
4. MATCH (a) - [:ACTED_IN] -> (movie) WITH a, count(distinct(movie)) AS movies RETURN distinct(a.name), movies ORDER BY movies DESC
5. MATCH (person) - [relation] -> (m:Movie) WHERE m.title = 'The Matrix' RETURN person.name, type(relation)
6. MATCH (m:Movie) - [*..6] - ({title : 'The Matrix'}) RETURN distinct(m.title)
7. MATCH ({name : 'Tom Hanks'}) -[:ACTED_IN] ->(m:Movie) <- [:DIRECTED] - (director) WHERE director.name <> 'Tom Hanks' RETURN distinct(m.title)
8. MATCH p = shortestPath( ({name : 'Madonna'}) - [:ACTED_IN*] - ({name : 'Keanu Reeves'}) ) RETURN length(p)

**EXERCISE 2**

1. MATCH (actor) - [:ACTED_IN] -> () WHERE actor.born < 1960 RETURN distinct(actor.name)

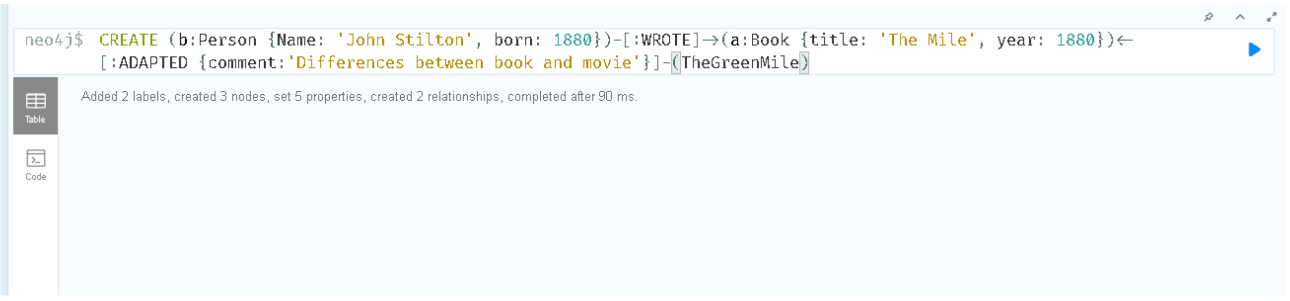
   The name of the actors who were born before 1960. By adding EXPLAIN before the MATCH keyword, it's possible to see that the query operates on a subgraph of 49 nodes instead of 171.

**EXERCISE 3**

1. We will add a new node type for books and use the existing Person node type for authors, we will add a new relationship of type "ADAPTED" to link books to movies and one new relationship of type "WROTE" to link authors to the books they have written. We will store the publishing information in the book node and the comment containing the differences between the book and the movie in the "ADAPTED" relationship.
2. CREATE (b:Person {Name: 'John Stilton', born: 1880})-[:WROTE]->(a:Book{title: 'The Mile', year: 1880})<-[:ADAPTED {comment:'Differences between book and movie'}]-(TheGreenMile)

   **NOTE: the above query creates three nodes instead of two (TheGreenMile node already exists in the graph) but neither of us nor the tutor know how it's possible. Below the screenshots to prove this:**

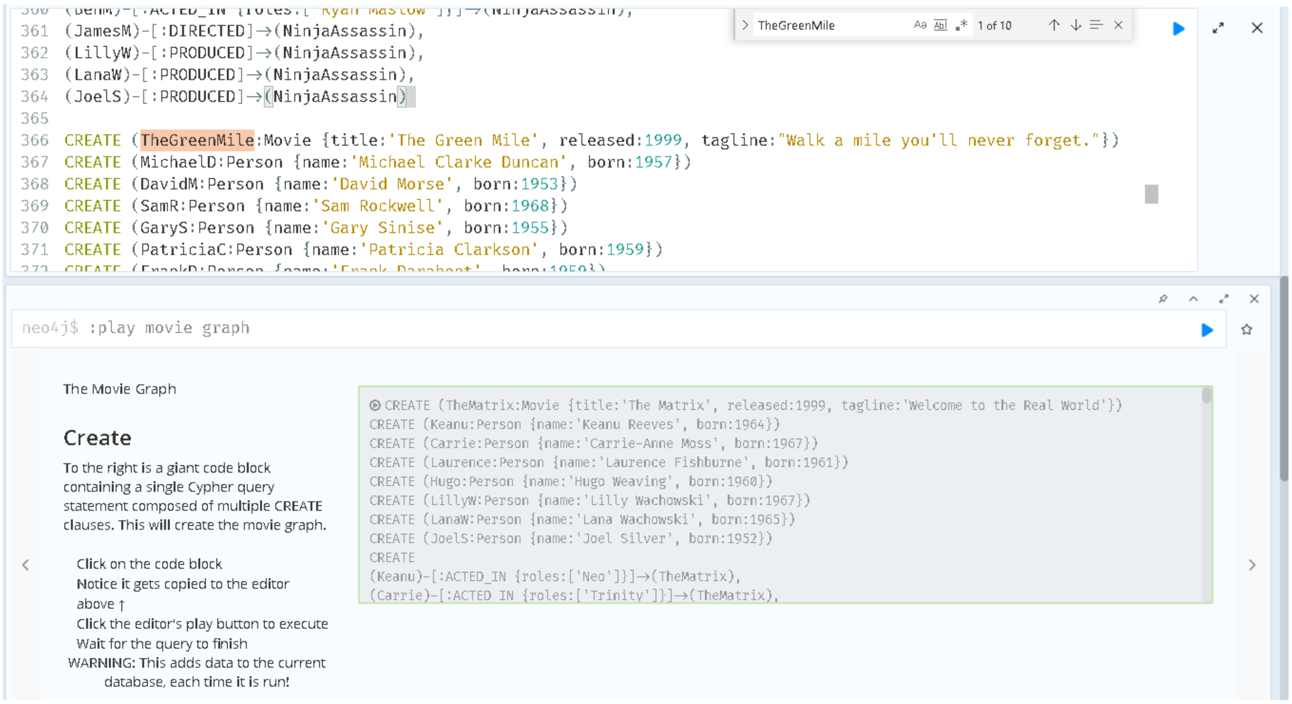
   Three nodes created instead of two:



```
neo4j$ CREATE (b:Person {Name: 'John Stilton', born: 1880})-[:WROTE]→(a:Book {title: 'The Mile', year: 1880})←
       [:ADAPTED {comment:'Differences between book and movie'}]-(TheGreenMile)
```

Added 2 labels, created 3 nodes, set 5 properties, created 2 relationships, completed after 90 ms.

TheGreenMile node already exists:



```
(BenM)-[:ACTED_IN {roles:['Ryan Maslow']}]→(NinjaAssassin),
(JamesM)-[:DIRECTED]→(NinjaAssassin),
(LillyW)-[:PRODUCED]→(NinjaAssassin),
(LanaW)-[:PRODUCED]→(NinjaAssassin),
(JoelS)-[:PRODUCED]→(NinjaAssassin)

CREATE (TheGreenMile:Movie {title:'The Green Mile', released:1999, tagline:"Walk a mile you'll never forget."})
CREATE (MichaelD:Person {name:'Michael Clarke Duncan', born:1957})
CREATE (DavidM:Person {name:'David Morse', born:1953})
CREATE (SamR:Person {name:'Sam Rockwell', born:1968})
CREATE (GaryS:Person {name:'Gary Sinise', born:1955})
CREATE (PatriciaC:Person {name:'Patricia Clarkson', born:1959})
CREATE (FrankD:Person {name:'Frank Darabont', born:1959})
```

```
neo4j$ :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 finish
WARNING: This adds data to the current database, each time it is run!

```
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),
```

3. Return the titles of the books written by John Stilton with the corresponding titles of the movies derived by them, together with the comments explaining the differences between books and movies.

   MATCH ({name : 'John Stilton'}) - [:WROTE] -> (b:Book) <- [a:ADAPTED] - (m:Movie)
   RETURN b.title, m.title, a.comment

**EXERCISE 4**

1. MATCH ({title : 'Cloud Atlas'}) <- [a:ACTED_IN] - (actor) WITH a.role as role, COLLECT(actor.name) as actors RETURN role, actors
2. MATCH ({name : 'Lana Wachowski'}) - [:DIRECTED] ->(movie) <- [a:ACTED_IN] - (actor) WITH a.role as role, COLLECT(actor.name) as actors, COLLECT(movie.title) as titles RETURN role, actors, titles