

TP Neo4j

Bases de données orientées graphe

Halim Djerroud (hdd@ai.univ-paris8.fr)

1 Installation

Pour lancer mongo pour la première fois :

```
sudo docker run --rm --name neo4j -p7474:7474 -p7687:7687  
-e NEO4J_AUTH=neo4j/pass neo4j:latest
```

Puis dans un navigateur connecter vous à l'adresse suivante :

`http://localhost:7474/`

Connectez vous grâce aux identifiants suivants :

```
http://localhost:7474/  
login : neo4j  
mot de passe : pass
```

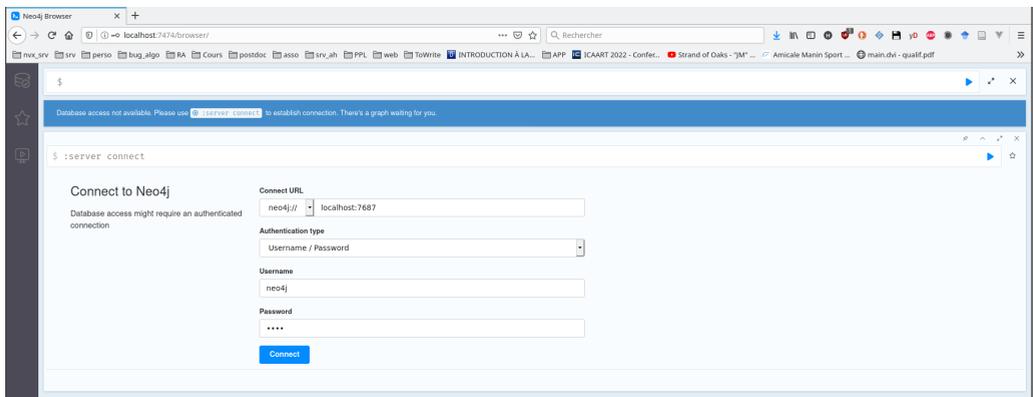


FIGURE 1 – Interface de connexion *neo4j*

Charger le script `cinema.cypher` qui se trouve sur moodle. Il suffi de copier/coller le contenu du fichier dans le champ d'exécution des commandes sur l'interface web, comme illustré dans la Figure 4.

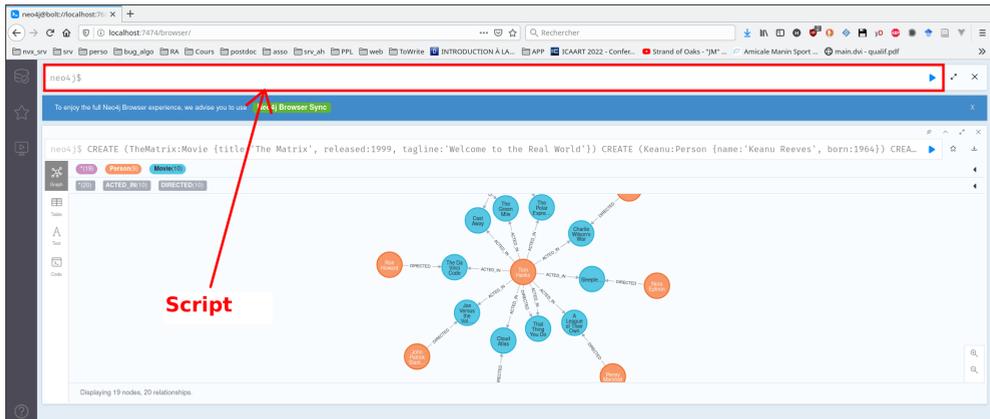


FIGURE 2 – Chargement du script

Les graphes, c'est quoi ?

Formellement, un graphe est une collection de nœuds et d'arcs qui les relient. Les graphes représentent les entités sous forme de nœuds, et les façons dont ces entités sont reliées sous forme d'arcs. Les nœuds sont connectés entre eux avec des arcs. Cette structure a inspiré les bases de données orientées graphe, car elles permettent de modéliser tous types de scénarios. Par exemple, les relations entre des acteurs et films dans lesquels ils ont joué, peut être représentées de façon simplifiée par le graphe suivant :

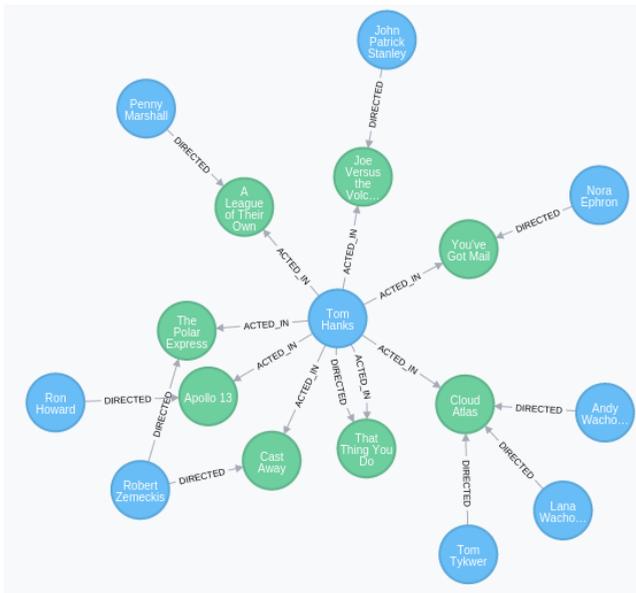


FIGURE 3 – Exemple d'un graphe

Un graphe peut représenter tout type de données.

- Un noeud (**node**) est l'équivalent d'un enregistrement
- Une relation (**relation**) permet de connecter des noeuds et peut être typée.
- Les noeuds et les relations peuvent avoir des propriétés (**properties**), représentant des attributs nommés
- Une étiquette (**label**) est un nom permettant d'organiser les noeuds en groupes.

La base de données que nous allons utiliser dans ce TP est Neo4j, cette base utilise le langage **Cypher** pour représenter les requêtes.

Qu'est-ce que Cypher ?

Cypher est un langage d'interrogation de graphes qui est utilisé pour interroger la base de données Neo4j. Tout comme vous utilisez SQL pour interroger une base de données MySQL, vous utilisez Cypher pour interroger la base de données Neo4j.

Les nœuds :

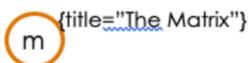
(a)	Actors	
(m)	Films	
()	Anonymous Node	

Les relations :

-[r]->	A relation called R	
(a) -[r]-> (m)	Actors having a relation r with movies	
-[:ACTED_IN]->	The relation is of type ACTED_IN	
(a)-[:ACTED_IN]->(m)	Actors who play in a movie	
(d)-[:DIRECTED]->(m)	The directors of a film	

Propriétés des nœuds

(m {title : "The Matrix"})	Movie with a property <i>title</i>
(a {name: "Keanu Reeves", born : 1964})	Actor with properties <i>name</i> and <i>born</i>



Propriétés des relations

(a)-[:ACTED_IN {roles:['« Neo »']}]>(m)

Relation *ACTED_IN* with the property *roles*



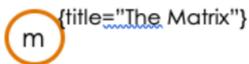
Les labels

(m {title : "The Matrix"})

Movie with a property *title*

(a {name: "Keanu Reeves", born : 1964})

Actor with properties *name* and *born*



Les requêtes Cypher

Voici quelques exemples de requêtes Cypher :

```
MATCH (node) RETURN node.property
```

```
MATCH (node1) --> (node2)
RETURN node2.propertyA, node2.propertyB
```

La première instruction renvoie la valeur de la propriété pour tous les nœuds. La deuxième, renvoie les valeurs de la *propertyA* et de la *propertyB* de chaque paire de nœuds *node2* et *node1* qui sont reliés par une relation (quelle que soit sa nature).

Pour retourner uniquement les nœuds connectés par une relation de type *REL_TYPE* :

```
MATCH (node1) -[:REL_TYPE] ->(node2)
```

Si la relation en question contient des propriétés que l'on souhaite lire, on lui donne un nom (ici *rel*) :

```
MATCH (node1) -[rel:TYPE] ->(node2)
RETURN rel.property
```

Pour indiquer un type particulier de nœuds, utilisez les étiquettes :

```
MATCH (node:Label1) RETURN node;
```

```
MATCH (node1:Label1) -[:REL_TYPE] ->(node2:Label2)
RETURN node1, node2;
```

Il est possible d'attribuer un identifiant à une relation, puis de retourner le type de la relation, ou ses propriétés :

```
MATCH (node)-[rel]->()
      RETURN node, rel.property;
```

```
MATCH (node)-[rel]->()
      RETURN node, type(rel);
```

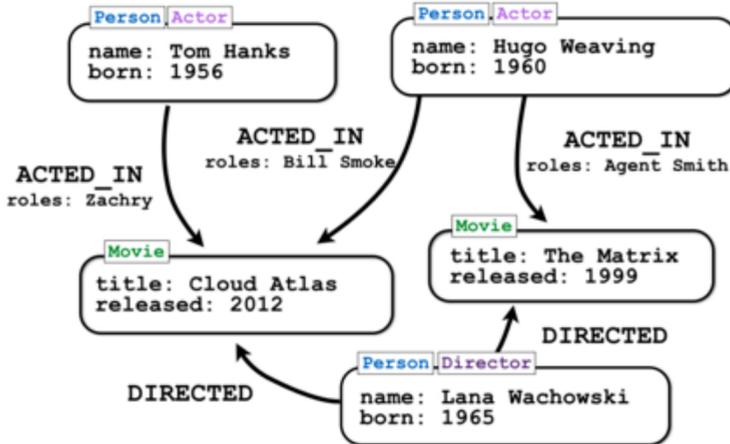


FIGURE 4 – Exemple d'un graphe

La requête pour trouver tous les personnages du film The Matrix est :

```
MATCH (movie :Mivie)<-[role :ACTED_IN]-(actor :Person)
      WHERE movie.title="The Matrix"
      RETURN role.roles, actor.name
```

Une autre façon est également possible pour filtrer les données (au lieu du traditionnel WHERE), en remplaçant la requête précédente par :

```
MATCH (movie :Movie (title : "The Matrix"))<-
      [role :ACTED_ID]-(actor :Persone)
      RETURN role.roles, actor.name
```

Exercice 1

Donnez les résultats des requêtes suivantes :

1. Donner tous les films où Tom Hanks a joué
2. Donne la liste de tous les personnages de Matrix
3. Donnez, pour chaque film, sa date de sortie

Les chemins (paths)

Un chemin est un ensemble de nœuds et de relations connectés, qui peut être un modèle. Par exemple :

```
(a)->(b)->(c)
(a)->(b)<-(c)
```

Exemple : Afficher tous les acteurs et réalisateurs de tous les films :

```
MATCH (a)-[:ACTED_IN]->(m)<[:DIRECTED]-(d)
RETURN a.name, m.title, d.name
```

Il est également possible de diviser la requête en plusieurs chemins, comme suit :

```
MATCH (a)-[:ACTED_IN]->(m),
      (m)<[:DIRECTED]-(d)
RETURN a.name, m.title, d.name
```

OU :

```
MATCH (a)-[:ACTED_IN]->(m),
      (d)-[:DIRECTED]->(m)
RETURN a.name, m.title, d.name
```

Les résultats des requêtes précédentes sont des tableaux. Il est possible de retourner le chemin entier, avec toutes les propriétés de tous les nœuds et relations concernées :

```
MATCH p=(a)-[:ACTED_IN]->(m)<[:DIRECTED]-(d)
RETURN p
```

Si cela représente trop d'informations, vous pouvez simplement sélectionner les nœuds du chemin :

```
MATCH p=(a)-[:DIRECTED_ID]->(m)<[:DIRECTED]-(d)
RETURN nodes(p)
```

... ou seulement des relations :

```
MATCH p=(a)-[:ACTED_IN]->(m)<[:DIRECTED]-(d)
RETURN rels(p)
```

Exercice 2

Activité 2. Exécutez les requêtes suivantes :

1. Quels réalisateurs ont joué dans leurs propres films ?
2. Donnez les films dans lesquels Keanu Reeves a joué le rôle de Neo.

3. Quels acteurs ont déjà joué dans le même film que Tom Hanks, et dans quel film ?
4. Quels acteurs ont joué dans le même film avec Tom Hanks, et avec Keanu Reeves ?
5. Qui sont les personnages du film "The Matrix" ?

Modifier les graphes avec le Cyphe

Pour ajouter un nœud au graphe :

```
CREATE (me:Persone {name: "My Name"})
RETURN me;
```

Pour ajouter (ou modifier) des propriétés à un nœud existant

```
MATCH (movie:Movie)
WHERE movie.title="Mystic River"
SET Movie.tagline = " We bury our sine here, Dave.
                    We wash them clean."
RETURN movie;
```

Pour ajouter une relation entre deux nœuds existants :

```
MATCH (me:Person), (movie:Movie)
WHERE me.name="My Name"
AND movie.title = "Mystic River"
CREATE (m)-[:REVIEWED {rating:80, summary:"tragic character movie"}]
->(movie);
```

Exercice 3

Réaliser les requêtes suivantes :

1. Créez un film appelé Mystic River
2. Ajouter Kevin Bacon comme acteur dans le film "Mystic River" avec le rôle "Sean".
3. Changez le rôle pour devenir "Sean Devine" ?

Exercice 4

Modéliser la base de données *Cinema* avec Neo4J. Choisissez un ou deux enregistrements dans chaque table.