

Gestion de Données Avancées

TD : Neo4j

Résumé

Dans ce TD, nous allons travailler avec le système de gestion de base de données orienté graphes Neo4j. Neo4j fait partie des systèmes NoSQL, et comme son nom l'indique, il stocke les données sous forme de graphe (au sens mathématique). Vous allez utiliser la MV Neo4j que se trouve dans :

<http://distrimiage.dmiage.parisnanterre.fr/public/rukoz-castillo> .

1 Création des nœuds et des relations

Montrez les commandes pour créer les nœuds **person** suivants (donnez à chaque nœud la propriété *name*) :

1. "Dwalin".
2. "Bombur".
3. "Thorin".
4. "Nori".
5. "Bilbo".
6. "Bifur".
7. "Aragorn".
8. "Gandalf".
9. "Saruman".
10. "Sauron".
11. "Arwen".
12. "Galadriel".

Montrez les commandes pour créer les relations suivantes :

13. "Dwalin" KNOWS "Bombur".
14. "Dwalin" KNOWS "Bifur".
15. "Bombur" KNOWS "Thorin".
16. "Thorin" KNOWS "Nori".
17. "Nori" KNOWS "Gandalf".
18. "Nori" KNOWS "Bilbo".
19. "Bifur" KNOWS "Aragorn".
20. "Aragorn" KNOWS "Bombur".
21. "Aragorn" KNOWS "Gandalf".
22. "Aragorn" LOVES "Arwen".
23. "Gandalf" KNOWS "Saruman".
24. "Saruman" KNOWS "Sauron".
25. "Arwen" KNOWS "Gandalf".
26. "Arwen" KNOWS "Galadriel".

Après avoir créé les nœuds et ses relations :

- a. Montrez le graphe que vous avez construit. Pour le faire, allez sur <http://localhost:7474/browser/>, puis *Favorites, Get some data, Export PNG*.
- b. Maintenant, affichez le même graphe avec la valeur de la propriété *name* comme étiquette des nœuds au lieu de l'id de nœud.

2 Parcours de graphe

Montrez la commande pour chaque une des requêtes suivantes :

- c. Quels sont les nœuds à 2 degrés de séparation d'*Aragorn* en parcourant la relation KNOWS en toutes les directions ?
- d. Quels sont les nœuds à 2 degrés de séparation d'*Aragorn* en parcourant la relation KNOWS dans le même sens de l'orientation du graphe et la relation LOVES en toutes les directions ?

- e. Quel est le chemin le plus court à partir de *Dwalin* pour arriver à *Sauron* en parcourant la relation KNOWS dans le même sens de l'orientation du graphe ?

3. En utilisant la bd Movies disponible sur Neo4j, répondez les questions suivantes :

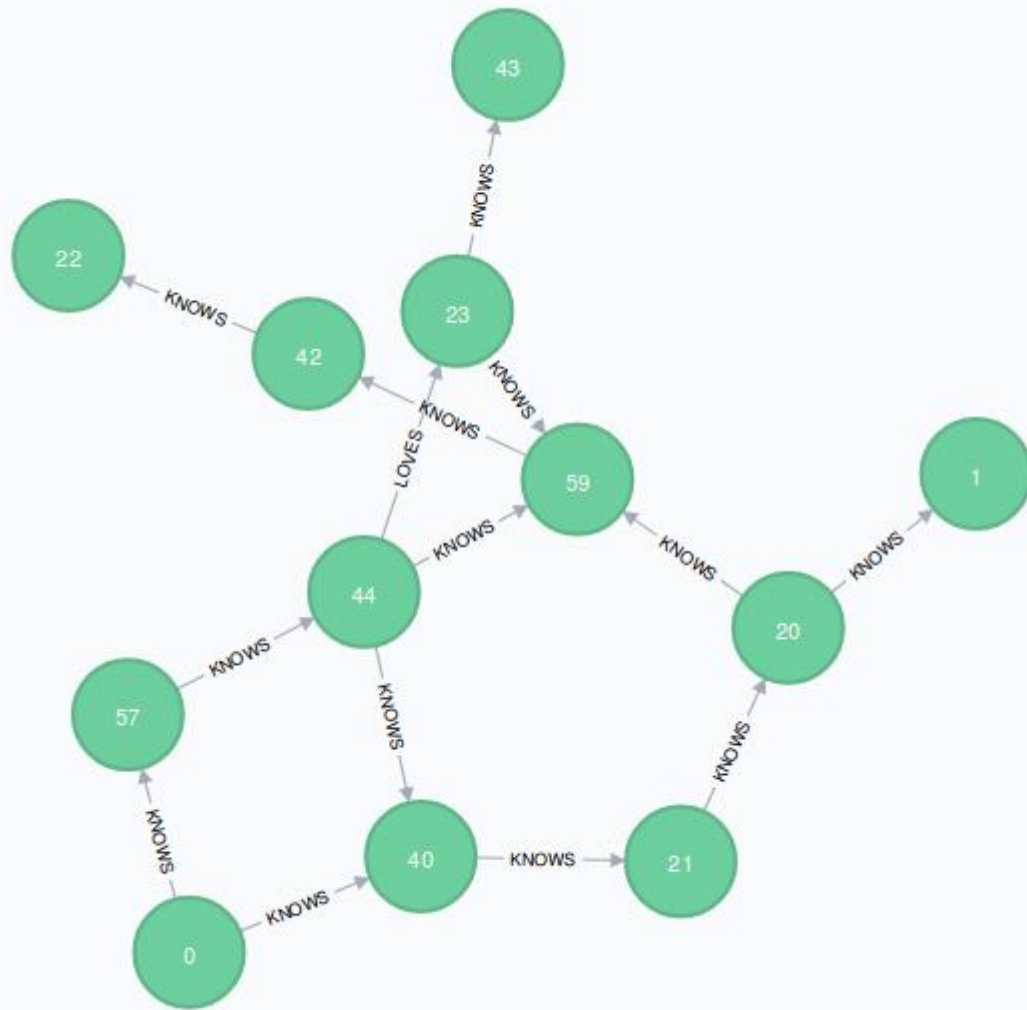
- a. Donnez le graph représentant les films où Meg Ryan a été acteur.
- b. Donnez la liste des 10 acteur plus jeunes (nom, année de naissance).
- c. Quel rôle a joué Meg Ryan dans le film "When Harry Met Sally"
- d. Donnez les acteurs qu'ont joué dans de films sortis depuis l'année 2000.

TD Neo4J

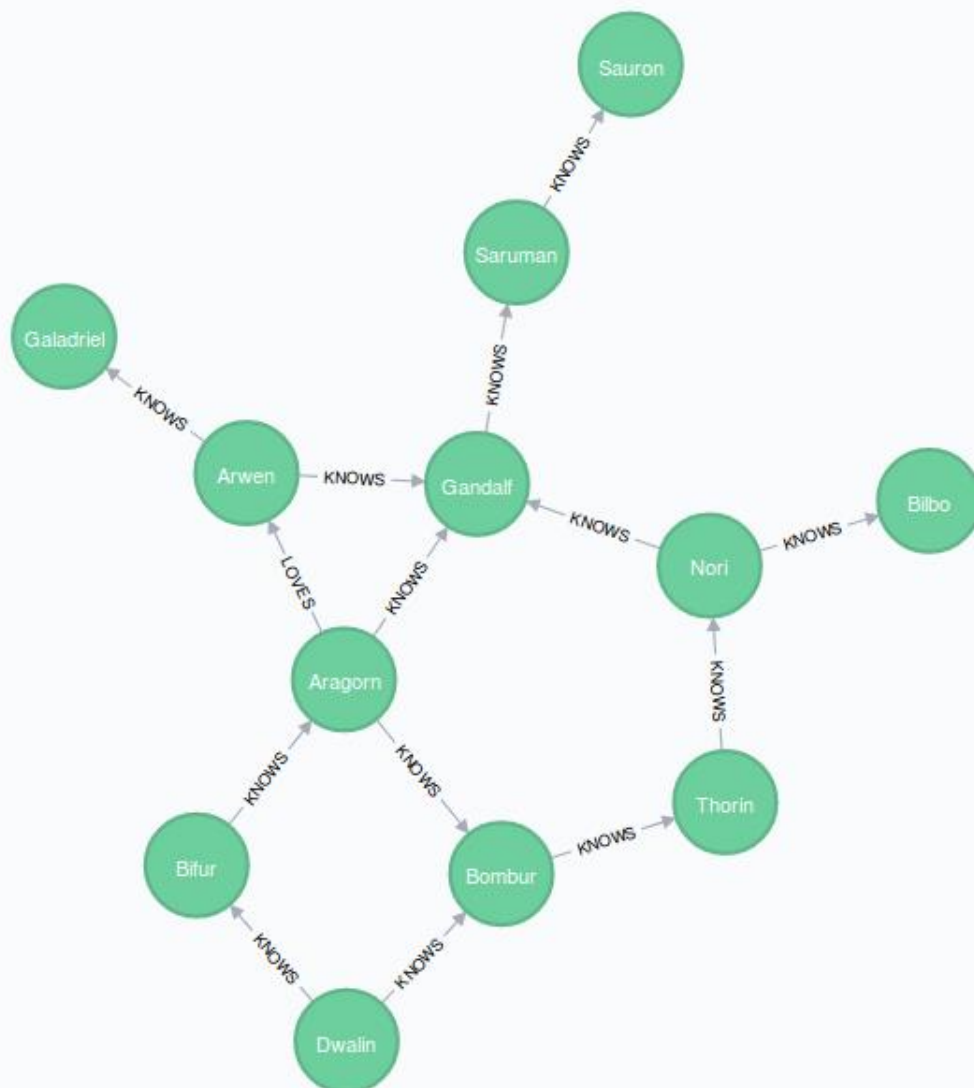
1- Création des noeuds et relations

1. CREATE (:person { name:"Dwalin" })
2. CREATE (:person { name:"Bombur" })
3. CREATE (:person { name:"Thorin" })
4. CREATE (:person { name:"Nori" })
5. CREATE (:person { name:"Bilbo" })
6. CREATE (:person { name:"Bifur" })
7. CREATE (:person { name:"Aragorn" })
8. CREATE (:person { name:"Gandalf" })
9. CREATE (:person { name:"Saruman" })
10. CREATE (:person { name:"Sauron" })
11. CREATE (:person { name:"Arwen" })
12. CREATE (:person { name:"Galadriel" })
13. MATCH (a:person),(b:person) WHERE a.name="Dwalin" and b.name="Bombur"
CREATE (a)-[r:KNOWS]->(b)
14. MATCH (a:person),(b:person) WHERE a.name="Dwalin" and b.name="Bifur"
CREATE (a)-[r:KNOWS]->(b)
15. MATCH (a:person),(b:person) WHERE a.name="Bombur" and b.name="Thorin"
CREATE (a)-[r:KNOWS]->(b)
16. MATCH (a:person),(b:person) WHERE a.name="Thorin" and b.name="Nori"
CREATE (a)-[r:KNOWS]->(b)
17. MATCH (a:person),(b:person) WHERE a.name="Nori" and b.name="Gandalf"
CREATE (a)-[r:KNOWS]->(b)
18. MATCH (a:person),(b:person) WHERE a.name="Nori" and b.name="Bilbo"
CREATE (a)-[r:KNOWS]->(b)
19. MATCH (a:person),(b:person) WHERE a.name="Bifur" and b.name="Aragorn"
CREATE (a)-[r:KNOWS]->(b)
20. MATCH (a:person),(b:person) WHERE a.name="Aragorn" and b.name="Bombur"
CREATE (a)-[r:KNOWS]->(b)
21. MATCH (a:person),(b:person) WHERE a.name="Aragorn" and b.name="Gandalf"
CREATE (a)-[r:KNOWS]->(b)
22. MATCH (a:person),(b:person) WHERE a.name="Aragorn" and b.name="Arwen"
CREATE (a)-[r:LOVES]->(b)
23. MATCH (a:person),(b:person) WHERE a.name="Gandalf" and b.name="Saruman"
CREATE (a)-[r:KNOWS]->(b)
24. MATCH (a:person),(b:person) WHERE a.name="Saruman" and b.name="Sauron"
CREATE (a)-[r:KNOWS]->(b)
25. MATCH (a:person),(b:person) WHERE a.name="Arwen" and b.name="Gandalf"
CREATE (a)-[r:KNOWS]->(b)
26. MATCH (a:person),(b:person) WHERE a.name="Arwen" and b.name="Galadriel"
CREATE (a)-[r:KNOWS]->(b)

A)



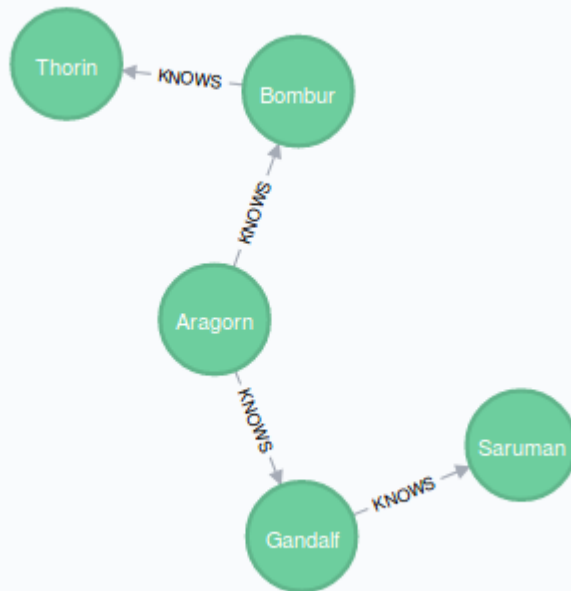
B)



2- Parcours de Graphe

C)

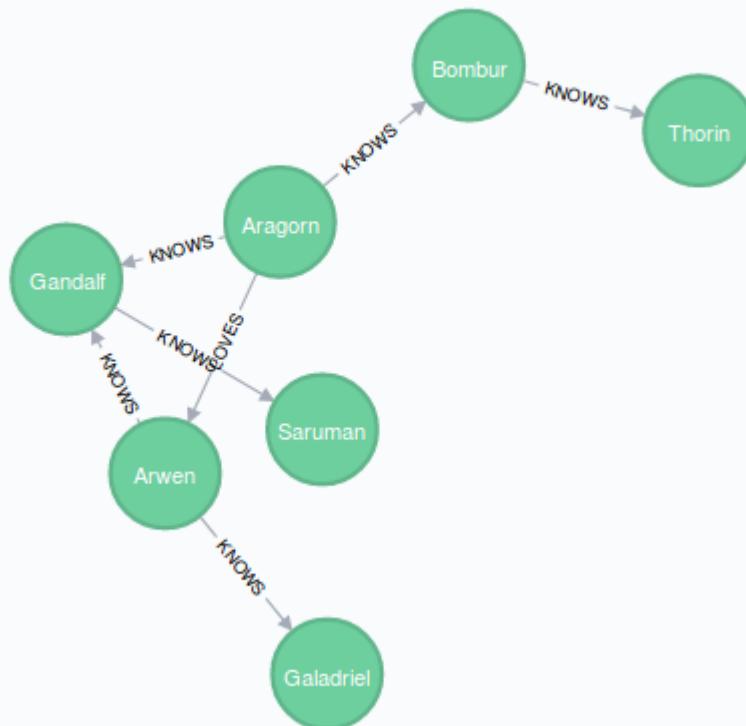
Match(a:person{name:"Aragorn"}) <-[r:KNOWS]->(t:person)-[s:KNOWS]->(u:person) return a,t,u



D)

Match(a:person{name:"Aragorn"})
>(u:person) return a,t,u

<-[r:KNOWS|:LOVES]->(t:person)-[s:KNOWS|:LOVES]-



E)

3- Questions