

Neo4j – TP – Correction

Chafik NOUIRA, LesFurets.com
tp-bigdata@lesfurets.com

I. Prise en main

1. Au démarrage de la VM, exécuter la commande suivante afin de rediriger le flux de l'interface graphique vers le système hôte :

```
ssh -L 7474:127.0.0.1:7474 -L 7687:127.0.0.1:7687 bigdata@xxx.xxx.xxx.xxx
```

Remplacer xxx.xxx.xxx.xxx par l'adresse ip de la VM affichée au démarrage. 2.

Accéder au dossier [neo4j]/conf

3. Ouvrir le fichier neo4j.conf et dé-commenter la ligne
dbms.security.auth_enabled=false

4. Accéder au dossier [neo4j]/bin

5. Exécuter la commande suivante pour lancer Neo4j
./neo4j console

6. Accéder à l'interface utilisateur via votre navigateur préféré
<http://localhost:7474/>

II. Lecture

Question 0 – Prise en main de la base Movies

- Exécuter la requête suivante :
- Observer les différents types de nœuds et de relations.
- Observer les différentes propriétés de chaque type de nœud et de relation. •

Extraire le modèle de données.

```
MATCH (start)-[r]->(end)  
RETURN distinct type(r), labels(start) as startNode, labels(end) as endNode
```

Question 1 – Récupérer le nœud de l'acteur Tom Hanks

- Solution 1 – Très gourmande en ressources
MATCH (tom)
WHERE tom.name = "Tom Hanks"
RETURN tom
- Solution 2 – Beaucoup plus performante
MATCH (tom:Person)
WHERE tom.name = "Tom Hanks"
RETURN tom
- Solution 3 – Équivalente à la 2^{ème} mais compacte
MATCH (tom:Person {name : "Tom Hanks"})
RETURN tom

Question 2 – Afficher les noms de 10 personnes quelconques

- Solution

```
MATCH (p:Person)
RETURN p.name
LIMIT 10
```

Question 3 – Afficher les titres des films sortis dans les années

- 90** • Solution

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

- Solution équivalente

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

Question 4 – Afficher les titres des films de Tom Hanks ainsi que les rôles qu'il a joué • Solution

```
MATCH (tom:Person {name : "Tom Hanks"})-[r:ACTED_IN]->(movie)
RETURN movie.title AS Movie, r.roles AS Roles
```

Question 5 – Afficher la liste des co-acteurs de Tom Hanks

- Solution

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

Question 6 – Qui a réalisé le film « Cloud Atlas » ?

- Solution

```
MATCH (m:Movie {title : "Cloud Atlas"})<-[:DIRECTED]-(director:Person) RETURN director.name
```

Question 7 – Comment les gens sont-ils reliés au film « Cloud Atlas »

- ? • Solution

```
MATCH (p:Person)-[:relatedTo]-(:Movie {title: "Cloud Atlas"})
RETURN p.name, Type(relatedTo), relatedTo
```

Question 8 – Qui sont les acteurs qui n'ont pas travaillé avec Tom Hanks

- ? • Solution

```
MATCH (tom:Person {name:"Tom Hanks"})-[ACTED_IN]->(m)<-[:ACTED_IN]-(coActors),
(coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cocoActors)
WHERE NOT (tom)-[:ACTED_IN]->()<-[:ACTED_IN]-(cocoActors) AND tom <>
cocoActors RETURN cocoActors.name AS Recommended, count(*) AS Strength
ORDER BY Strength DESC
```

Question 9 – Qui pourrait introduire Tom Cruise à Tom Hanks ?

- Solution

```
MATCH (tom:Person {name:"Tom  
Hanks"})-[:ACTED_IN]->(m)<-[:ACTED_IN]-(coActors),  
(coActors)-[:ACTED_IN]->(m2)<-[:ACTED_IN]-(cruise:Person {name:"Tom Cruise"})  
RETURN tom, m, coActors, m2, cruise
```

Question 10 – Afficher les noms des acteurs avec le nombre de films pour chacun. Trier les résultats par nombre de films dans l'ordre décroissant

- Solution

```
MATCH (p:Person)-[r:ACTED_IN]-()  
RETURN p.name, count(DISTINCT r) AS nbFilms  
ORDER BY nbFilms DESC
```

III. Optimisation de requêtes

Question 11 – Retourner le nombre d'acteurs

- Solution

```
MATCH (a:Person)  
WHERE (a)-[:ACTED_IN]->()  
RETURN count(DISTINCT a)
```

Question 12 – Ajouter le label `Actor` aux acteurs ensuite refaire la Question 11 en plus simple

- Solution

```
MATCH (a:Person)  
WHERE (a)-[:ACTED_IN]->()  
SET a:Actor
```

Exécuter les requêtes suivantes pour comparer :

1. MATCH (p:Person)
WITH count(p) AS nbPerson
MATCH (a:Actor)
RETURN count(a) AS nbActor, nbPerson
2. MATCH (p:Person), (a:Actor)
RETURN count(p), count(a)

Question 13 – Faire la même chose pour les Producteurs, les réalisateurs et scénaristes et les reviewers

Question 14 – Pour chaque nœud, compter le nombre de relations par type et stocker les résultats dans des propriétés. (Les noms des propriétés doivent être assez parlants)

- Solution

```
MATCH (p:Actor)-[r:ACTED_IN]-()  
WITH p, count(r) AS nbFilms  
SET p.nbFilms = nbFilms
```