

Référence d'objets avec Neo4J (BD orientée graphe)

Table des matières

I - Cours	3
1. Exemple de base de données orientée graphe avec Neo4J.....	3
1.1. Installation de Neo4J	3
1.2. Démarrage de Neo4J.....	3
1.3. Test de Neo4J	4
1.4. CREATE : Créer des nœuds et des relations.....	4
1.5. MATCH / RETURN : Trouver des nœuds et des relations	5
1.6. MATCH / DELETE : Supprimer des noeuds et des relations	5
II - Exercice	7
1. Exercice : De Neo4J à Game of Thrones	7



1. Exemple de base de données orientée graphe avec Neo4J

1.1. Installation de Neo4J

L'installation présentée ici est uniquement destinée à tester Neo4J localement.

L'installation de Neo4j peut être réalisée localement sans droit administrateur.

Documentation



Fondamental

<https://neo4j.com/docs/developer-manual/current/>

Installation de Neo4J

1. On utilise dans le cadre de cet exercice la version libre Neo4j Community Edition
2. Téléchargez cette version depuis : <https://neo4j.com/download/other-releases>
3. Suivez la procédure suivant votre système d'exploitation (Mac OSX, Linux/UNIX, Windows)

Copie de Neo4j Community Edition



Complément

<http://pic.crzt.fr/neo4j>

Pré-requis Java 8



Complément

<http://neo4j.com/docs/stable/deployment-requirements.html>

1.2. Démarrage de Neo4J

Lancer le serveur

Le démarrage du serveur dépend du système d'exploitation.

- Sous Windows et Mac OSX lancer l'application et cliquer sur le bouton Start
- Sous Linux exécuter : `$NEO4J_HOME/bin/neo4j console`

Lancer le client

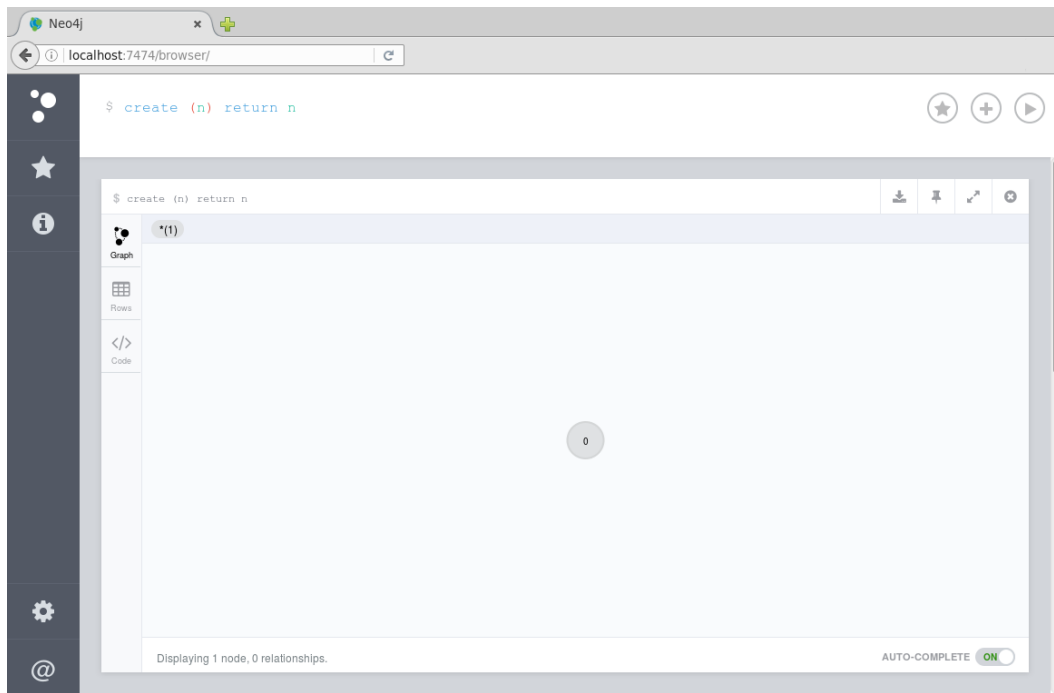
Ouvrir un navigateur Web à l'adresse : <http://localhost:7474>



Complément

- Pour arrêter le processus serveur sous Linux : `$NEO4J_HOME/bin/neo4j stop`
- Pour réinitialiser les bases de données : supprimer le dossier `$NEO4J_HOME/data/graph.db/`

1.3. Test de Neo4J



Interface de Neo4j (création d'un nœud)

- Les requêtes s'écrivent dans la partie haute.
- Les résultats se visualisent dans la partie basse.

Créer un nœud

? Exemple

```
create (n) return n
```

Voir tous les nœuds

? Exemple

```
match (n) return n
```

1.4. CREATE : Créer des nœuds et des relations

Créer un nœud avec une propriété

? Exemple

```
create (n {name:'Sun'}) return n
```

Créer un nœud avec un label

? Exemple

```
create (n:Astre {name:'Sun'}) return n
```

Créer des relations

§ Syntaxe

le formalisme pour les relations est :

- ()<-[]-()
- ()-[]->()
- ()<-[]->()
- ()-[]-()

où :

- on mettra nos nœuds à l'intérieur des parenthèses ()
- on mettra nos relations à l'intérieur des crochets []
- enfin on désignera le sens de la relation avec la flèche

Créer deux nœuds et une relation entre ces deux nœuds

? Exemple

```
create (e:Astre {name:'Earth'}) <-[:Satellite {distance:384000}]-
(m:Astre {name:'Moon'}) return e, m
```

Créer deux nœuds et une relation entre ces deux nœuds en plusieurs instructions

? Exemple

```
create (e:Astre {name:'Earth'})
create (m:Astre {name:'Moon'})
create (e)<-[:Satellite {distance:384000}]- (m)
return e, m
```

Q Remarque

- CTRL+ENTER
Lance la commande présente dans la ligne de commande.
- SHIFT+ENTER
Permet de faire un saut de ligne dans la ligne de commande (sans l'exécuter)

1.5. MATCH / RETURN : Trouver des nœuds et des relations

Trouver un nœud avec une propriété

? Exemple

```
match (n {name:'Sun'}) return n
```

Trouver un nœud avec un label

? Exemple

```
match (n:Astre) return n
```

Trouver des nœuds en fonction d'une relation

? Exemple

```
match (n1) <-[:Satellite]- (n2) return r,n1,n2
```

1.6. MATCH / DELETE : Supprimer des nœuds et des relations

Supprimer tous les liens :

```
match (n)-[l]-(m) delete l
```

Supprimer tous les nœuds :

```
match (n) delete n
```

supprimer un nœud

§ Syntaxe

Pour supprimer un nœud ou une relation on utilise la même syntaxe que pour la recherche, en remplaçant `return` par `delete`.

? Exemple

```
match (n {name:'Moon'}) delete n
```

Vider la base de données : supprimer tous les liens puis tous les nœuds

? Exemple

```
1 match (n)-[l]-(m) delete l
2 match (n) delete n
```

Exercice



1. Exercice : De Neo4J à Game of Thrones

[1h]

Un beau matin, vous vous réveillez en Westeros, le monde de Game Of Thrones, sans aucune connaissance historique. Afin d'y survivre, il va falloir intégrer les informations du royaume. Nous allons pour cela utiliser une base de données orientée graphe Neo4J.

Question 1

Avant toute chose vous allez vous allez devoir vous ajouter au monde que vous venez de rejoindre.

Créer un nouveau personnage avec votre nom et prénom.

Indice :

Créer des nœuds et des relations (cf. p.4)

Indice :

Le label est : `personnage`.

Indice :

Les propriétés sont :

```
{name : '...', nickname : '...' }
```

Question 2

Seconde chose, il faut apprendre à créer des relations, car le monde de Game Of Thrones est très complexe.

Créez deux personnages et exprimez une relation qui les relie.

Par exemple, un des combats les plus célèbres oppose Oberyn Martell, dit The Viper et Gregor Clegane, dit The Mountain. Gregor Clegane tue Oberyn Martell en duel.

Indice :

Créer des nœuds et des relations (cf. p.4)

Indice :

Créer d'abord les deux nœuds avec l'instruction CREATE.

```
1 CREATE(gregor:personnage { name : '...', nickname : '...' })
2 CREATE(oberyn:...)
3 ...
```

Indice :

Toujours avec CREATE, créer une relation de type « `:TUE` » entre les deux nœuds avec comme attribut : « `type : 'duel'` »

```

1 CREATE(gregor:personnage { name : 'Gregor Clegane', nickname : 'The Mountain' })
2 CREATE(oberyn:personnage { name : 'Oberyn Martell', nickname : 'The Viper' })
3 CREATE(...)-[...]->(...)
4 ...

```

Indice :

```

1 CREATE(gregor:personnage { name : 'Gregor Clegane', nickname : 'The Mountain' })
2 CREATE(oberyn:personnage { name : 'Oberyn Martell', nickname : 'The Viper' })
3 CREATE(gregor)-[...]->(oberyn)
4 RETURN gregor, oberyn

```

Indice :

```

1 CREATE(gregor:personnage { name : 'Gregor Clegane', nickname : 'The Mountain' })
2 CREATE(oberyn:personnage { name : 'Oberyn Martell', nickname : 'The Viper' })
3 CREATE(gregor)-[:TUE { type : ... }]->(oberyn)
4 RETURN gregor, oberyn

```

Question 3

Affichez l'ensemble des nœuds présents dans votre base de données.

Indice :

Test de Neo4J (cf. p.4)

Nous pouvons à présent alimenter notre base d'un ensemble de connaissance dont nous disposons. Pour cela copier et exécuter le code Cypher ci-après.

```

1 //GameOfThrones:
2
3 //sans clan :
4 CREATE (sansclan:clan { clanname : 'Clan des sans lien de sang'})
5
6 CREATE (hodor:personnage { name : 'Hodor', nickname : 'I am the only good actor here
!' })
7 CREATE (khal:personnage { name : 'Khal Drogo', nickname : 'Horse Man' })
8 CREATE (petyr:personnage { name : 'Petyr Baelish', nickname : 'LittleFinger' })
9 CREATE (shae:personnage { name : 'Shae for the Imp', nickname : 'Dead' })
10
11
12 //lanister :
13 CREATE (lanister:clan { clanname : 'The Lannisters'})
14
15 CREATE (tyrion:personnage { name : 'Tyrion Lannister', nickname : 'The Imp' })
16 CREATE (tywin:personnage { name : 'Tywin Lannister', nickname : 'Father of the
Lannisters' })
17 CREATE (jaime:personnage { name : 'Jaime Lannister', nickname : 'King Slayer' })
18 CREATE (cersei:personnage { name : 'Cersei Lannister', nickname : 'Brother Lover' })
19 CREATE (joffrey:personnage { name : 'Joffrey Lannister', nickname : 'Mad King 2' })
20
21
22 //baratheons :
23 CREATE (baratheon:clan { clanname : 'The Baratheons'})
24
25 CREATE (robert:personnage { name : 'Robert Baratheon', nickname : 'King Robert' })
26 CREATE (renly:personnage { name : 'Renly Baratheon', nickname : 'King Gay' })
27 CREATE (stannis:personnage { name : 'Stannis Baratheon', nickname : 'Stéphane' })
28 CREATE (gendry:personnage { name : 'Gendry ???Baratheon', nickname : 'What happened
to my story ?' })
29
30
31 //tyrells :
32 CREATE (tyrell:clan { clanname : 'The Tyrells'})
33

```



```

34 CREATE (margeary:personnage { name : 'Margeary Tyrell', nickname : 'Tyrell asset' })
35 CREATE (loras:personnage { name : 'Loras Tyrell', nickname : 'King Gay Knight' })
36
37
38 //targaryens :
39 CREATE (targaryen:clan { clanname : 'The Targaryen'})
40
41 CREATE (daenerys:personnage { name : 'Daenerys Targaryen', nickname : 'Mother of
    Dragons' })
42 CREATE (viserys:personnage { name : 'Viserys Targaryen', nickname : 'Gold Head' })
43
44
45 //Stark :
46 CREATE (stark:clan { clanname : 'The Starks'})
47
48 CREATE (arya:personnage { name : 'Arya Stark', nickname : 'I am 20' })
49 CREATE (sansa:personnage { name : 'Sansa Stark', nickname : 'I am sorry I can t
    smile.' })
50 CREATE (rosse:personnage { name : 'Roose Bolton', nickname : 'Come by the house I ll
    kill you' })
51 CREATE (edward:personnage { name : 'Eddard Stark', nickname : 'Whhhhhy ???' })
52 CREATE (robb:personnage { name : 'Robb Stark', nickname : 'King of the North' })
53 CREATE (john:personnage { name : 'John Snow', nickname : 'The Bastard' })
54 CREATE (bran:personnage { name : 'Bran Stark', nickname : 'Stop boring me' })
55
56
57 //tullys :
58 CREATE (tully:clan { clanname : 'The Tullys'})
59
60 CREATE (catelyn:personnage { name : 'Catelyn Stark', nickname : 'Mother of Stark ?'
    })
61 CREATE (lysa:personnage { name : 'Lyse Tully', nickname : 'Crazy' })
62
63
64 //greyjoys :
65 CREATE (greyjoy:clan { clanname : 'The Greyjoys'})
66
67 CREATE (theon:personnage { name : 'Theon Greyjoy', nickname : 'The enuque' })
68 CREATE (balon:personnage { name : 'Balon Greyjoy', nickname : 'Nicest father ever' })
69
70
71 //actions :
72
73 CREATE (tyrion)-[:TUE{type : 'homicide'}]->(shae)
74 CREATE (tyrion)-[:TUE{type : 'homicide'}]->(tywin)
75 CREATE (petyr)-[:TUE{type : 'homicide'}]->(joffrey)
76 CREATE (stannis)-[:TUE{type : 'homicide'}]->(renly)
77 CREATE (khal)-[:TUE{type : 'homicide'}]->(khal)
78 CREATE (khal)-[:TUE{type : 'homicide'}]->(viserys)
79 CREATE (joffrey)-[:TUE{type : 'homicide'}]->(edward)
80 CREATE (rosse)-[:TUE{type : 'homicide'}]->(robb)
81 CREATE (rosse)-[:TUE{type : 'homicide'}]->(catelyn)
82 CREATE (petyr)-[:TUE{type : 'homicide'}]->(lysa)
83
84 CREATE (jaime)-[:AIME{type : 'amour'}]->(cersei)
85 CREATE (cersei)-[:AIME{type : 'amour'}]->(jaime)
86 CREATE (tyrion)-[:AIME{type : 'amour'}]->(shae)
87 CREATE (shae)-[:AIME{type : 'amour'}]->(tywin)
88 CREATE (robert)-[:AIME{type : 'amour'}]->(cersei)
89 CREATE (loras)-[:AIME{type : 'amour'}]->(renly)
90 CREATE (renly)-[:AIME{type : 'amour'}]->(loras)
91 CREATE (margeary)-[:AIME{type : 'amour'}]->(joffrey)
92 CREATE (joffrey)-[:AIME{type : 'amour'}]->(margeary)

```

```

93 CREATE (khal)-[:AIME{type : 'amour'}]->(daenerys)
94 CREATE (daenerys)-[:AIME{type : 'amour'}]->(khal)
95 CREATE (petyr)-[:AIME{type : 'amour'}]->(catelyn)
96 CREATE (eddard)-[:AIME{type : 'amour'}]->(catelyn)
97 CREATE (catelyn)-[:AIME{type : 'amour'}]->(eddard)
98
99
100 //liens de clan :
101
102 CREATE (theon)-[:LIER{type : 'liendecan'}]->(greyjoy)
103 CREATE (balon)-[:LIER{type : 'liendecan'}]->(greyjoy)
104
105 CREATE (khal)-[:LIER{type : 'liendecan'}]->(sansclan)
106 CREATE (john)-[:LIER{type : 'liendecan'}]->(sansclan)
107 CREATE (petyr)-[:LIER{type : 'liendecan'}]->(sansclan)
108 CREATE (gendry)-[:LIER{type : 'liendecan'}]->(sansclan)
109 CREATE (hodor)-[:LIER{type : 'liendecan'}]->(sansclan)
110
111 CREATE (gendry)-[:LIER{type : 'liendecan'}]->(baratheon)
112 CREATE (joffrey)-[:LIER{type : 'liendecan'}]->(baratheon)
113 CREATE (robert)-[:LIER{type : 'liendecan'}]->(baratheon)
114 CREATE (renly)-[:LIER{type : 'liendecan'}]->(baratheon)
115 CREATE (stannis)-[:LIER{type : 'liendecan'}]->(baratheon)
116
117 CREATE (margeary)-[:LIER{type : 'liendecan'}]->(tyrell)
118 CREATE (loras)-[:LIER{type : 'liendecan'}]->(tyrell)
119
120 CREATE (daenerys)-[:LIER{type : 'liendecan'}]->(targaryen)
121 CREATE (viserys)-[:LIER{type : 'liendecan'}]->(targaryen)
122
123 CREATE (lysa)-[:LIER{type : 'liendecan'}]->(tully)
124 CREATE (catelyn)-[:LIER{type : 'liendecan'}]->(tully)
125
126 CREATE (arya)-[:LIER{type : 'liendecan'}]->(stark)
127 CREATE (hodor)-[:LIER{type : 'liendecan'}]->(stark)
128 CREATE (rosse)-[:LIER{type : 'liendecan'}]->(stark)
129 CREATE (sansa)-[:LIER{type : 'liendecan'}]->(stark)
130 CREATE (petyr)-[:LIER{type : 'liendecan'}]->(stark)
131 CREATE (eddard)-[:LIER{type : 'liendecan'}]->(stark)
132 CREATE (robb)-[:LIER{type : 'liendecan'}]->(stark)
133 CREATE (john)-[:LIER{type : 'liendecan'}]->(stark)
134 CREATE (bran)-[:LIER{type : 'liendecan'}]->(stark)
135 CREATE (catelyn)-[:LIER{type : 'liendecan'}]->(stark)
136 CREATE (theon)-[:LIER{type : 'liendecan'}]->(stark)
137
138 CREATE (shae)-[:LIER{type : 'liendecan'}]->(lanister)
139 CREATE (rosse)-[:LIER{type : 'liendecan'}]->(lanister)
140 CREATE (tyrion)-[:LIER{type : 'liendecan'}]->(lanister)
141 CREATE (tywin)-[:LIER{type : 'liendecan'}]->(lanister)
142 CREATE (jaime)-[:LIER{type : 'liendecan'}]->(lanister)
143 CREATE (cersei)-[:LIER{type : 'liendecan'}]->(lanister)
144 CREATE (joffrey)-[:LIER{type : 'liendecan'}]->(lanister)

```

Question 4

Affichez l'ensemble des nœuds présents dans votre base de données.

Manipulez le graphe quelques minutes afin de vous faire une idée des données.

Question 5

Affichez à présent la liste des clans.

Indice :

On utilise le label : `clan` après l'alias de nœud `n`.

Trouver des nœuds et des relations (cf. p.5)

Question 6

Afficher le nœud qui possède comme *nickname* The Imp.

Question 7

Affichez le clan qui a pour *clanname* The Starks.

Question 8

Affichez tous les personnages qui ont une relation :*LIER* avec le clan The Starks.

Indice :

```
1 match (c:clan {clanname:'The Starks'})<-[...]-(...) return ...
```

Indice :

```
1 match (c:clan {clanname:'The Starks'})<-[l:...]- (p) return l,c,p
```

Question 9

Affichez toutes les relations de type :*TUE*, pour savoir qui a tué qui.

Indice :

```
1 match (:personnage)-[...] -> (:personnage)
2 return result
```