

HMIN313 : Introduction au système de gestion de données distribué Neo4J

1. Enoncé

Nous abordons dans ce TP, quelques principes de base de mise en œuvre au sein du système Neo4J. Vous exploiterez l'API Java de Neo4J. Pour ce faire, vous en téléchargerez une version récente (voir <http://www.neo4j.org/download>). Décompressez l'archive et lancez le serveur : `...bin/neo4j start` Une interface web est disponible à l'adresse `localhost:7474`

2. Appropriation

Vous ferez en sorte de construire un graphe portant sur les **Communes**. Pensez à bien poser votre modèle, vous pouvez vous aider d'un diagramme de classes à cet effet. Vous créerez à cet effet quelques nœuds avec pour labels `Commune`, `Departement` et `Region`. Vous doterez les nœuds de propriétés. Vous définirez également des relations entre les nœuds (à votre libre appréciation en fonction de votre connaissance des relations entre ces entités administratives).

2.1 Questions

- A partir d'un projet Java sous Eclipse et des API dédiées (requêtes cypher¹ ou navigation² dans le graphe). A cet effet, vous définirez une classe applicative dotée d'une méthode permettant d'afficher l'ensemble des communes.
- Définissez une méthode permettant d'afficher l'ensemble des communes pour un département donné.
- Définissez une méthode permettant d'afficher le chef-lieu de chaque commune de la base
- Définissez une méthode permettant d'afficher la population en 1975 de chaque commune de la base
- Définissez une méthode permettant d'afficher les informations sur la commune la plus peuplée en 2010

3. Import de données au format tabulé avec Cypher

Un exemple d'import de données dans la base de données Neo4J vous est fourni³. Cet exemple exploite la syntaxe cypher et un fichier de contrôle de chargement (extension `.cql`) pour insérer respectivement des données sur des ouvrages, des catégories d'ouvrage. Les correspondances entre les ouvrages

1. Doc Cypher : <http://neo4j.com/docs/stable/cypher-getting-started.html>

2. <https://github.com/neo4j/neo4j/tree/2.2.5/community/embedded-examples/src/main/java/org/neo4j/examples>

3. voir aussi <http://neo4j.com/docs/stable/cypherdoc-importing-csv-files-with-cypher.html>

et les catégories sont également posées à l'aide de la relation CATEGORISED_AS. Vous reprendrez cet exemple pour l'appliquer aux données commune-département-région. Vous définirez les relations APR_REGION (entre les nœuds de type DEPARTEMENT et REGION) et APR_DEPARTEMENT (entre les nœuds de type COMMUNE et DEPARTEMENT). Vous pouvez aussi travailler avec une ou plusieurs relations pour matérialiser le fait qu'une région possède une capitale administrative nommée chef-lieu de région, qu'un département a une capitale administrative nommée chef-lieu de département et qu'une commune peut dépendre d'un chef-lieu de région et d'un chef-lieu de département. Vous définirez la ou les relations qui vous semblent nécessaires à la prise en charge de cette réalité.

```
./bin/neo4j-shell -file importOuvCat.cql
```

Listing 1 – ordre de chargement

3.1 Questions

Vous proposerez une requête de votre choix qui exploite plusieurs relations (par exemple APR_REGION et APR_CHEFLIEU).

4. Annexe

4.1 Effacer le contenu de la base

```
MATCH (n)
OPTIONAL MATCH (n)-[r]-()
DELETE n,r
```

Listing 2 – Tout supprimer

4.2 Données Commune

CODE_I	NOM_COM	POP_1975	POP_2010
34248	SAINTE-CROIX-DE-QUINTILLARGUES	127.033553	561.720409
34249	SAINT-DREZERY	587.63955	2207.40611
34025	BASSAN	795.304324	1599.4016
34250	SAINT-ETIENNE-D'ALBAGNAN	253.31857	287.52352
34251	SAINT-ETIENNE-DE-GOURGAS	216.105575	489.314637
34252	SAINT-ETIENNE-ESTRECHOUX	406.570257	245.980104
34253	SAINT-FELIX-DE-L'HERAS	25.248617	36.8979349
34254	SAINT-FELIX-DE-LODEZ	519.740374	1209.95666
34255	SAINT-GELY-DU-FESC	2010.13406	8528.24947
34256	SAINT-GENIES-DES-MOURGUES	825.375932	1623.96003
34257	SAINT-GENIES-DE-VARENSAL	218.473421	209.311992
34167	MONTELS	127.655942	246.150356
34168	MONTESQUIEU	34.3551636	63.8836005
34169	MONTFERRIER-SUR-LEZ	1682.12888	3320.80271
34017	AUMES	307.185836	455.743575
34170	MONTOULIERS	209.369513	237.112177
34171	MONTOULIEU	38.7548087	161.147619

34172	MONTPELLIER	191767.053	260959.188
34173	MONTPEYROUX	774.815254	1206.2612
34174	MOULES-ET-BAUCELS	131.447644	845.71166
34175	MOUREZE	78.5494361	176.923015
34176	MUDAISON	735.409812	2506.47942

Listing 3 – Informations communes