Neo4J par la pratique

Objectif : comprendre les bases de données de type graphe ; savoir interroger une base de données de type graphe ; créer et mettra à jour une base de données de type graphe ; concevoir des bases de données de type graphe et importer des données.

Déroulement :
On propose une série de TP sur différents aspects sous la forme de tutoriels (pour la prise en main), puis de petits exercices, puis de cas d'utilisation.

Installation

1. Commencez par télécharger **Neo4J Desktop** version 1.5.9 depuis le site officiel Neo4j. Sauvegardez bien la clé de licence dans votre dossier permanent. Installez le également dans un répertoire (ex. neo4j) de votre espace permanent (pas dans C).
2. Au 1er lancement de Neo4J, allez dans « Software Key » (icone en bas à gauche) et copiez/collez la clé de licence.
3. Créez un nouveau Projet nommé par exemple « BUT3 » et une base de données locale dans ce projet (ex. « MaBD »). Puis démarrez là (avec start), puis ouvrez là (avec open).
4. Vous êtes prêts pour lancer des commandes cypher après l'invite « neo4j$ »

Création d'une BD graphe

5. Testez la commande *cypher* basique d'affichage de graphe : *match (n) return n*
6. Créez une BD Friends en suivant les commandes vues en cours. Créez les nœuds puis les arcs. Relancer la commande d'affichage de graphe.
7. Testez les 3 requêtes sur ce graphe vues en cours.
8. Sachant qu'on peut remplacer une condition par l'ajout de la propriété dans match (ex. *match (n:person {NAME: "Chandler" })*), simplifiez la 1ère requête.
9. **A savoir** : neo4j permet de déduire la structure du graphe et de la visualiser avec la commande (à tester) : *call db.schema.visualization*

Import de données CSV

10. Commencer par nettoyer la base en supprimant tout le graphe (cf. cours)
11. On part du fichier test.csv (fourni) utilisé en cours pour illustrer l'import. L'ajouter d'abord au projet (menu Add file to Project – puis rajouter les fichiers csv). Ensuite localisez le chemin (a priori dans dossier import de la hiérarchie .Neo4jDesktop dont le chemin à utiliser dans la commande *LOAD* est file:///test.csv. Importez comme vu en cours.
12. Entrainez-vous avec les requêtes fournies séparément sur ce graphe. Notez les résultats.

Cas d'utilisation BD Cinéma

13. Stoppez la BD précédente. Créez une BD « Cinema ». Puis insérer des nœuds et des arcs comme suit :

---------- les noeuds

```
-- Creation des noeuds producer
CREATE (clean : producer {Nom : "clean kill movies",
Address:"Versailles"}),
(sfm : producer {Nom : "sf movies", Address:"Velizy"}),
(clf : producer {Nom : "classique film", Address:"Paris"}),
(cof : producer {Nom : "constance film", Address:"Versailles"})

-- Creation des noeuds movie
CREATE (aliens : movie {Titre : "aliens", Duree:"137",
Annee:"1982"}),
(blade : movie {Titre : "blade runner", Duree:"117", Annee:"1982"}),
(casa : movie {Titre : "casablanca", Duree:"102", Annee:"1942"}),
(dances : movie {Titre : "dances with wolves", Duree:"180",
Annee:"1990"}),
(seven : movie {Titre : "seven pounds", Duree:"120", Annee:"2007"})

-- Creation des noeuds category
CREATE (a : category {Code : "a", Specification:"action"}),
(sf : category {Code : "sf", Specification:"science fiction"}),
(cd : category {Code : "cd", Specification:"comedy drama"}),
(w : category {Code : "w", Specification:"war"})
return a, sf, cd, w
```

---------- les arcs

```
-- Creation des liens (arcs) entre movies et producers

MATCH (m: movie), (p: producer)
WHERE m.Titre="aliens" AND p.Nom="clean kill movies"
CREATE (m)-[:has_Producer]->(p)
Return m, p

MATCH (m: movie), (p: producer)
WHERE m.Titre="blade runner" AND p.Nom="sf movies"
CREATE (m)-[:has_Producer]->(p)
Return m, p

MATCH (m: movie), (p: producer)
WHERE m.Titre="casablanca" AND p.Nom="classique film"
CREATE (m)-[:has_Producer]->(p)
Return m, p

MATCH (m: movie), (p: producer)
WHERE m.Titre="dances with wolves" AND p.Nom="constance film"
CREATE (m)-[:has_Producer]->(p)
Return m, p

MATCH (m: movie), (p: producer)
WHERE m.Titre="seven pounds" AND p.Nom="clean kill movies"
CREATE (m)-[:has_Producer]->(p)
Return m, p
```

```
---------------

-- Creation des liens entre movies et categories

MATCH (m: movie), (c: category)
WHERE m.Titre="aliens" AND c.Code="a"
CREATE (m)-[:has_Category]->(c)
Return m, c

MATCH (m: movie), (c: category)
WHERE m.Titre="aliens" AND c.Code="sf"
CREATE (m)-[:has_Category]->(c)
Return m, c

MATCH (m: movie), (c: category)
WHERE m.Titre="blade runner" AND c.Code="sf"
CREATE (m)-[:has_Category]->(c)
Return m, c

MATCH (m: movie), (c: category)
WHERE m.Titre="casablanca" AND c.Code="cd"
CREATE (m)-[:has_Category]->(c)
Return m, c

MATCH (m: movie), (c: category)
WHERE m.Titre="dances with wolves" AND c.Code="w"
CREATE (m)-[:has_Category]->(c)
Return m, c

MATCH (m: movie), (c: category)
WHERE m.Titre="seven pounds" AND c.Code="w"
CREATE (m)-[:has_Category]->(c)
Return m, c
```

14. Vérifiez le contenu de la base puis visualisez sa structure.

15. Répondez aux questions suivantes :
    a. Donner la liste de films
    b. Quels sont les producteurs ?
    c. Quel est l'adresse du producteur du film « casablanca » ?
    d. Quelles sont les categories du film « casablanca »?
    e. Quels sont les films produits par « clean kill movies »?
    f. Quelles sont les categories de films produits par « clean kill movies »?
    g. Quelles sont les films de guerre