École Polytechnique de Montréal
Département Génie Informatique et Génie
Logiciel INF3710 – Fichiers et Bases de données

Travail Pratique N° 4
NoSQL

I. Informations générales

Session	Automne 2017
Public cible	Étudiants de 1er cycle
Date de remise	Groupe 2 : Mardi 07 Novembre 2017 Groupe 1 : Mardi 14 Novembre 2017
Équipe de	2 étudiants
Pondération	8%
Directives particulières	1. ***Tout retard*** dans la remise du compte-rendu ***entraîne automatique-ment*** une pénalité aux étudiants concernés. 2. Aucun compte-rendu ne sera corrigé, s'il est soumis par une équipe dont la taille est différente de deux (2) étudiants sans l'approbation préalable du chargé de laboratoire. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés. 3. Soumission du compte rendu (au format PDF ou word) par ***moodle*** uniquement (https://moodle.polymtl.ca). 4. Aucune soumission "hors ***moodle***" ne sera corrigée. La note de zéro sur vingt (0/20) sera attribuée aux étudiants concernés.

II. Objectifs du laboratoire

Le but de ce TP est de permettre à l'étudiant(e) de se familiariser avec trois bases de données NoSQL. Ce TP se focalise sur :

1- Mongo DB (Orienté document)
2- Neo4j (Orienté graphe)
3- Cassandra (Orienté colonne)

III. Directives:

En utilisant Vmware 12player, veuillez ouvrir la nouvelle version du VM (inf3710) disponible sous C:\VM (Faudra comme toujours copier la VM dans un autre dossier autre que c:\ pour pouvoir l'exécuter).
Pour ce TP, je vous recommande fortement d'utiliser Putty pour vous faciliter la tâche, parceque qu'on va travailler sur plus ce qu'un terminal à la fois.

IV. Mongo DB

Définition:

MongoDB est un système de gestion de bases de données open source développé par MongoDB Inc depuis 2007. Il est orienté documents et ne nécessite pas de schéma prédéfini de données.
Dans MongoDB, les données sont modélisées sous forme de document sous un style JSON. Le JSON est un format de représentation textuelle des données dérivé de la notation des objets du langage JavaScript. Toutefois il est indépendant du JavaScript et de tout autre langage de programmation. Il permet de représenter l'information structurée comme le permet XML par exemple.
On ne parle plus de tables, ni d'enregistrements mais de collections (équivalent de table) et de documents(équivalent de ligne). Ce système de gestion de données nous évite ainsi de faire des jointures de tables car toutes les informations propres à une certaine donnée sont stockées dans un même document.
Dans MongoDB la clé primaire est automatiquement attribuée au champ _id.

Documentation :

https://docs.mongodb.com/
Un pdf 'MongoDB-Documentation.pdf' est disponible sur moodle dans le dossier mongodb sous TP4.

Avantage:

Plus proche des langages de programmations.
Très flexible grâce à l'orientation documents.
Accès aux données plus rapides que le SQL par l'absence de jointures.
Facile de gérer de très grandes quantités de données grâce au scaling (répartitions du moteur de base de données sur plusieurs ordinateurs).
Full-Text search.

Inconvénients:

Non existence de clés étrangères, du coup l'obligation à contrôler l'intégrité des données dans la couche application.
Dénormalisation qui cause la redondance.

Travail demandé:

Les travaux demandés pour la partie de MongoDB se déroulent sur la plate-forme mLab, située à l'adresse https://mlab.com pour la première question et sur la machine virtuelle pour la deuxième question.
Pour créer un compte et une base de données sur Mlab, vous suivrez les directives présentées dans le document *MongoDB_Connexion.docx* annexé au TP.

Question 1

Créer un project Java sous eclipse (disponible sur votre machine windows) et en vous basant sur le fragment de code fourni 'simple_exemple.java', développez dans votre projet java une méthode main qui permet d'importer dans votre base de données la collection **DBLP** contenue dans le fichier JSON qui vous a été fourni. Le fichier est un extrait d'une base de publications scientifiques, *The DBLP Computer Science Bibliography*. PS : Faudra fortement lire tout le projet Java avant de commencer et ne pas oublier d'ajouter les drivers fourni dans le dossier 'Drivers' à votre projet Java.

- Complétez et exécutez votre projet afin de peupler la collection DBLP dans la base de données hébergée sur mLab. L'opération peut durer quelques minutes. Vous utiliserez le nom et le mot de passe de l'utilisateur que vous avez créé précédemment.

- Retournez à mLab et rafraîchissez la page. Produisez dans votre rapport, une capture d'écran montrant le nombre de documents ajoutés, le nom de votre base de données, le nom du déploiement ainsi que votre nom d'utilisateur.

Question 2

Exprimez des commandes simples pour les recherches et mises à jour suivantes :

a) Afficher la liste de tous les livres publiés depuis 2000 ;

b) Afficher la liste de tous les éditeurs (type "publisher") distincts ;

c) Trier les publications de "Ingrid Zukerman" ordonnés selon la date (le plus récent d'abord) ;

d) Compter le nombre d'articles de "Ingrid Zukerman" ;

e) Insérer un livre fictif dont vous êtes l'auteur unique ;

Directives : Afin de vérifier vos commandes, accéder à la VM que vous avez déjà ouvert, créer un dossier sur le chemin par défault /data/db. Puis à partir de ce dossier : /usr/local/mongodb-linux-x86_64-rhel70-3.4.9/bin, lancer le serveur mongo en utilisant cette commande : ./mongod (faut laisser le serveur en marche pour terminer l'exercice).

Dans un autre terminal, on va faire le lien entre Mlab et mongo shell, pour cela : lancer cette commande dans le même dossier /usr/local/mongodb-linux-x86_64-rhel70-3.4.9/bin :

Maintenant, vous êtes connecté. Vous pouvez commencer à exécuter vos commandes.

V. Cassandra

Définition :

Cassandra est un système de gestion de données à grande échelle conçu à l'origine (2007) par les ingénieurs de Facebook pour répondre à des problématiques liées au stockage et à l'utilisation de gros volumes de données. Cassandra est une base de données orientée colonne, qui permet de stocker une grande quantité de données grâce à sa scalabilité horizontale.
Dans les applications du monde du Web, type Facebook par exemple, les données ne sont pas décrites de la même façon. L'utilisation des bases de données relationnelles n'est pas toujours appropriée. En effet cela conduirait à stocker des données avec de nombreuses valeurs nulles. C'est dans cet objectif que des modèles de données proches de celui de Cassandra ont été développés.
CQL, est le langage de Cassandra.
Cqlsh, est la ligne de commande shell pour interagir avec Cassandra à travers CQL.

Avantage :

Tolérance aux pannes (grâce aux mécanismes de réplication de données)
La distribution de données
Décentralisé
Élastique
Haute disponibilité
Open Source

Inconvénients :

Lenteur
Pas d'interface graphique
Difficultés à l'utilisation
Limitation de la taille des données

Documentation :

http://cassandra.apache.org/doc/latest/

Travail demandé:

Pour pourvoir ouvrir Cassandra, veuillez accéder à votre VM et puis à partir du root taper cqlsh et vous seriez directement amener à la ligne de commande de Cassandra. (Le service est déjà ouvert sur vos VMs)

Question 1

- Créer une famille de colonne (Equivalent de table dans SQL) nomée "BDclass" comportant :

Un id (primary key), Un nom, Un prénom, et Age.
Indice 1 : Faut tout d'abord créer le keyspace qui va contenir les familles de colonnes:
CREATE KEYSPACE "KeySpace Name" WITH replication = {'class': 'SimpleStrategy', 'replication_factor' : 'No.Of replicas'};

- ➔ Replication_factor' : Nombre de noeuds sur lesquels les données doivent être répliquées.
- ➔ Class : Strétagie de gestion des clés primaires

Indice 2 : La commande Select est la même que SQL
Indice 3 : Faites attention avec le type de id, le primary key.

- Afficher la structure de la famille de colonne crée en utilisant DESC.

Question 2

- Insérez dans votre famille de colonne 4 étudiants avec les données de votre choix. (Syntaxe est la même que SQL)

Faire attention à l'insertion de la clé primaire, c'est une clé qui est gérée par la base de données.

- Afficher le contenu de la famille de colonne.

Question 3
Sélectionner les étudiants ayant pour âge <Un âge de votre choix>. (Syntaxe est la même que SQL)
Indice : Dans le cas de Cassandra, faut créer un index sur la colonne sur laquelle le filtre sera appliqué avant d'exécuter le select.

Question 4
On souhaite stocker dans la base, les numéros de téléphones des étudiants, évidemment, les étudiants peuvent avoir plusieurs numéros sur lesquels ils peuvent être contactés. Étant donné que Cassandra ne permet pas la jointure entre différente tables, on va créer une collection pour chaque étudiant regroupant leurs numéros de téléphone.

- Modifiez la famille de colonnes pour ajouter une collection de numéros de téléphone sachant qu'un numéro de téléphone est constitué uniquement d'un varchar.

Indice 1 : Utiliser la commande Alter table
Indice 2 : Utilisez une collection de type SET

- Afficher la nouvelle structure de la famille de colonne.

VI. Neo4j

Définition :
Neo4j est un système de gestion de base de données au code source libre orienté graphes. La démarche utilisée par Neo4J en graphe NoSQL permet de traiter des données massivement interconnectées et très hétérogènes et aussi l'accélération de transfert de données.
Le language utilisé est "Cypher Query Language", il fournit des méthodes pour pouvoir matcher des patrons de nœuds et leurs relations.

Avantage :
Open source
Modélisation facilitée
Un moteur puissant accessible facilement
Langage Cypher "assez" visuel

Inconvénients :
Work In Progress : le système n'est pas forcément complet mais évolue vite !
Problème d'affichage de large graphe dans l'interface web.
Il n'est pas évident dans l'interface de base de rendre les graphes dynamiques.

Documentation :
https://neo4j.com/docs/
https://neo4j.com/developer/cypher/

Travail demandé:

Directives :
Pour pouvoir ouvrir le serveur Neo4j, faut accéder à la VM puis au dossier suivant: /usr/local/neo4j-community-3.2.6 mais avant de l'exécuter, on doit modifier le fichier de configuration pour autoriser les connections non local. Pour cela, accéder au dossier conf et puis faire (vi neo4j.conf)… Un fichier va s'ouvrir, faut dé-commenter la ligne comme le montre la capture d'écran ci-dessous et puis enregistrer et quitter par la commande (:wq)

```
#***************************************************************
# Network connector configuration
#***************************************************************

# With default configuration Neo4j only accepts local connections.
# To accept non-local connections, uncomment this line:
dbms.connectors.default_listen_address=0.0.0.0
```

Par la suite, faire (cd ..) pour sortir du dossier conf et puis (cd bin) pour accéder au fichiers binaires.
Exécuter la commande suivante pour lancer le serveur Neo4j : (./neo4j start) → maintenant le serveur est en marche.
Dans une fenêtre browser sur votre machine Windows, taper : http://@votreVM:7474
Veuillez se connecter avec neo4j / neo4j et puis créez votre propre password.

Exemple introductif (Pas obligatoire et pas nécessaire de l'ajouter dans votre rapport):

1- On va créer un noeud personnage ayant le formalisme suivant (commande CREATE) :

CREATE(gregor:personnage { name : 'Gregor Clegane', nickname : 'The Mountain' })
Veuillez exécuter la commande dans la fenêtre ouverte sur votre browser.

- "gregor" est l'identificateur du noeud
- "personnage" permet de lui donner un label pour le différentier des autres
- "name" et "nickname" désignent des attributs du noeud qui permettent d'enrichir le noeud en informations

2- On va maintenant chercher à créer une relation. Tout d'abord ajouter un nouveau noed :

CREATE(oberyn:personnage { name : 'Oberyn Martell', nickname : 'The Viper' })

Puis, il va falloir créer un lien entre les deux nœuds déjà créés : du coup créez une relation de type « :TUE » entre les deux noeuds avec comme paramètre : « type : 'duel' »

La commande est la suivante, veuillez la tester sur Neo4j: CREATE (gregor)-[:TUE{type : 'duel'}]->(oberyn)

Les relations sont définis par ces symboles : ()<-[]-() ou ()-[]->() ou ()<-[]->() ()-[]-()

- 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

Voila un exemple de relation :

Cypher

```
(a) -[:LIKES]-> (b)
```

➔ On passe maintenant à l'exercice GameofThrones.

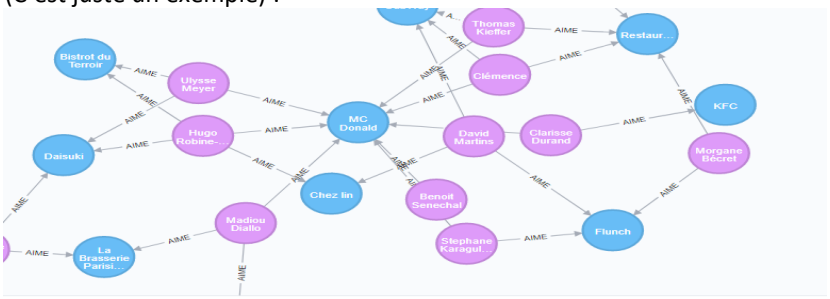
Question1 :

Ouvrez le fichier «GameofThrones.cypher », disponible sur moodle dans le dossier neo4j sous le tp4, dans un éditeur de texte et copier-coller son contenu dans la ligne de commande de Neo4J.

- Exécutez les commandes. (PS : CTRL+MAJ est un raccourci pour lancer les commandes dans la ligne de commande).

- Rapporter par une capture d'écrans le résultat. Détailler le nombre de nœuds, les différents noms de relations, etc.

PS : vous pouvez afficher le résultat sous forme graphe pour avoir une visualisation plus claire, il va gérer un graphe de ce genre (C'est juste un exemple) :



Pour les questions suivantes voici quelques astuces qui pourraient vous être utiles :
match (n) return n

- "match" agit presque comme un SELECT de SQL.
- "return" renvoie l'ensemble des matchs.
- Comme () désigne un noeud et que n est ici juste une variable, on va matcher avec l'ensemble des nœuds du graphes et donc récupérer l'intégralité des nœuds du graphe en retour ! On peut donc préférer sur un gros graphe rajouter un LIMIT 100 après le "return".
- Il est possible de compter les nœuds en remplaçant "n" par COUNT(n) comme paramètre de retour.

Question 2

Il existait des nœuds clan comme celui-ci : (stark:clan { clanname : 'The Starks'}) et qu'ils pouvaient être liés à des personnages à l'aide de cette relation : (arya)-[:LIER{type : 'liendeclan'}]->(stark), trouvez l'ensemble des Starks du graphe ?
Indice : match (*)-[result:LIER]->(:clan { *}) return * → (les * sont des trous à remplir)

Question 3

Maintenant qu'on sait qui sont les Starks et les Lanister, il est temps que quelqu'un meurt, trouver, qui a déjà tué qui ? (relation de label :TUE)

VI. Livrable

Le livrable à soumettre est une archive .zip ou .rar dont le nom est formé des numéros de matricules des membres de l'équipe, séparé par un trait de soulignement (_). Il doit comporter les éléments suivants :

a) Le projet ou el fichier .java. Le code doit être documenté en vue de faciliter sa lecture et sa compréhension.

b) Un rapport contenant les captures d'écrans ainsi que le nom, le mot de passe de L'utilisateur que vous avez créé. Les informations de l'utilisateur serviront lors de la correction.

Vous pouvez également inscrire dans le rapport, toute autre information que vous désirez porter à la connaissance de la personne en charge de la correction.

Évaluation

Rubrique	Points

Questions	19
Présentation du rapport	1
Total	**20**