



Session : 1

Durée de l'épreuve : 2 heures

Date : 14 janvier 2021

Documents autorisés : tous

Mention Informatique

Matériel utilisé : aucun

Master 2^{ème} année 2020/2021 : HMIN340

1 Questions de cours (6 points)

Vous expliquerez les mérites comparés des systèmes NOSQL face aux SGBD relationnels, en exploitant au besoin les notions suivantes :

1. scalabilité (verticale et horizontale)
2. réplication
3. partitionnement
4. persistance polyglotte
5. modèle semi-structuré
6. postulat CAP et système distribué

Vous pouvez vous aider des trois systèmes NOSQL étudiés et de votre connaissance du relationnel pour illustrer vos propos.

2 Le réseau de transport de la TAM et NoSQL

2.1 Enoncé

Les parties portant sur les systèmes de persistance (vus en cours) exploitent tous le contexte des transports urbains de l'agglomération de Montpellier (TAM)

2.2 Partie Neo4J (6 points)

1. Des ordres de création de nœuds et d'arêtes vous sont donnés dans la syntaxe Cypher. Vous construirez un premier graphe à partir des deux ordres de création donnés ci-dessous.

```
CREATE (s1:Station {nom:'Occitanie', latitude:43.63435811, longitude:3.84863696})
-[:STOP_DE]-> (la1:Liaison {distance:500}),
(la1) -[:APPARTIENT_A]->
(li1:Ligne {nom:'Mosson-Odysseum', num:1, type:'tram',longueur:15700, tempsTotal:50}),
(la2:Liaison {distance:1000}) -[:APPARTIENT_A]-> (li1), (s2:Station
{nom:'Hopital Lapeyronie',latitude:43.63166867, longitude:3.85260055}) -[:STOP_DE]->
(la1), (s1) -[:STOP_DE]-> (la2), (s3:Station
{nom:'Chateau d O',latitude:43.63131727, longitude:3.84299936}) -[:STOP_DE]-> (la2),
(s4:Station {nom:'Univ. Sciences et Lettres',latitude:43.63131727, longitude:3.84299936})
-[:STOP_DE]->
(la3:Liaison {distance:1200}),
(s2) -[:STOP_DE]-> (la3), (la3)-[:APPARTIENT_A]-> (li1)
RETURN s1, s2, s3, s4, li1, la1, la2, la3
```

```
MATCH (o:Station {nom:'Occitanie'})
CREATE (sp:Station {nom:'Saint-Priest', latitude:43.836699, longitude:4.360054})
-[:STOP_DE]-> (l:Liaison {distance:400}),
(l) -[:APPARTIENT_A]->(li:Ligne {nom:'Euromedecine-Pas du Loup', num:6, type:'bus'}),
(o) -[:STOP_DE]-> (l)
```

Vous choisirez le nom des stations, le numéro (num) des lignes, et la distance des liaisons pour donner une étiquette aux nœuds visualisés et vous mentionnerez les types de chacun des nœuds. Les autres propriétés des nœuds ne seront pas représentées.

2. Une requête de consultation en langage Cypher, vous est donnée qui porte sur le graphe qui vient d'être créé. Vous donnerez la signification de cette requête, ainsi que le résultat renvoyé par cette requête

```
MATCH (l:Ligne {type:'bus'}) <-[a:APPARTIENT_A]- ()
<-[st:STOP_DE]- (s:Station)
RETURN DISTINCT l.nom, s.nom
```

3. Une nouvelle requête Cypher vous est donnée (toujours sur le graphe créé). Vous donnerez également la signification de cette requête et et vous proposerez le sous-graphe qui est le résultat de la requête.

```
MATCH (l1:Liaison) <-[:STOP_DE]- (s:Station) -[:STOP_DE]-> (l2:Liaison)
RETURN l1, s , l2
```

4. Vous écrirez en langage Cypher, la requête : "donner le nombre de stations qui sont situées sur la ligne portant le nom Mosson-Odyseum"
5. Le choix de représentation arrêté pour représenter les lignes, les stations et leurs connexions manque d'efficacité, proposez une simplification du modèle pour le rendre plus performant.

2.3 Partie CouchDB (4 points)

Un exemple de documents JSON décrivant deux stations, et gérés au sein d'une BD CouchDB, vous est donné.

```
{
  "docs":
  [
    {
      "_id" : "Occitanie",
      "nom" : "Occitanie",
      "latitude" : 43.63435811,
      "longitude" : 3.84863696,
      "type" : "station",
      "correspondances": [
        {
          "libelle": "Mosson-Odyseum", "numLigne":1, "distance":471,"vitesseMax":50,
          "typeLigne": "tram", "destination":"Hopital Lapeyronie"}, {
          "libelle": "Mosson-Odyseum", "numLigne":1, "distance":799,
          "typeLigne": "tram", "destination":"Chateau d O"},
        {
          "libelle": "Euromedecine-Pas du Loup", "numLigne":6,
          "typeLigne": "bus", "destination":"Saint-Priest"}]
    },
  ],
}
```



```
{
  "_id" : "Hopital Lapeyronie",
  "nom" : "Hopital Lapeyronie",
  "type" : "station",
  "correspondances": [
    {
      "libelle": "Mosson-Odyseum", "numLigne":1, "distance":912, "vitesseMax":40,
      "typeLigne": "tram", "destination":"Universités Sciences et Lettres"}, {
      "libelle": "Mosson-Odyseum", "numLigne":1, "distance":471,"vitesseMax":50,
      "typeLigne": "tram", "destination":"Occitanie"}]
  }
}]}
```

Vous donnerez votre compréhension des vues Map et Map/Reduce suivantes (écriture en javascript du corps de la fonction map / des fonctions map/reduce). Quels sont les résultats renvoyés (illustrez les résultats obtenus avec les documents présentés au dessus).

1. Vue Map 1

```
function(doc) { if (doc.nom=='Occitanie')
  emit([doc._id, doc.latitude, doc.longitude], doc.correspondances);
}
```

2. Vue Map 2

```
function(doc) { if (doc.type=='station')
for(var idx in doc.correspondances) {
  correspondance = doc.correspondances[idx];
  emit(doc._id, correspondance.destination);
}
}
```

3. Vue Map Reduce

```
function(doc) { if (doc.type=='station')
for(var idx in doc.correspondances) {
  correspondance = doc.correspondances[idx];

  if (correspondance.numLigne==1)
    emit(doc._id, correspondance.distance);
}
}

function(keys, values) {
  return Math.max.apply(null, values);
}
```

2.4 Partie HBase (4 points)

Une proposition de table et familles de colonnes vous est faite. Des tuples sont également fournis au travers des ordres JRuby associés.

```
create 'station', 'general', 'ligne'
put 'station', 'Occitanie', 'general:nom', 'Occitanie'
put 'station', 'Occitanie', 'general:latitude', '43.63435811'
put 'station', 'Occitanie', 'general:longitude', '3.84863696'
put 'station', 'Occitanie', 'ligne:libelle', 'Mosson-Odyssseum'
put 'station', 'Occitanie', 'ligne:num', '1'
put 'station', 'Occitanie', 'ligne:typeLigne', 'tram'
put 'station', 'Occitanie', 'ligne:destination', 'Hopital Lapeyronie'
```

1. Vous reprendrez les ordres précédents sous une forme tabulaire
2. Cette représentation HBase vous semble valide? Si oui justifiez, si non, expliquez pourquoi et modifiez le schéma en conséquence (table ou ordres JRuby).