UNIVERSITÉ DE MONTPELLIER
-
FACULTÉ DES SCIENCES

Session : 2	Durée de l'épreuve : 2 heures
Date : 31 mars 2022	Documents autorisés : tous sous format papier
Mention Informatique	Matériel utilisé : aucun
Master $2^{ème}$ année 2021/2022 : HAI914I	

1 Questions de cours (4 points)

CouchDB reprend à son avantage les principes de distribution proposés par Amazon dans le système DynamoDB, à savoir

1. architecture multi-maîtres
2. hachage cohérent

Vous expliquerez en quelques lignes en quoi consistent ces deux principes, ou tout au moins l'un de ces principes. Vous pouvez vous aider d'un schéma explicatif.

2 Partie Neo4J (8 points)

Le contexte estudiantin est exploité. Des étudiants sont inscrits annuellement dans des formations universitaires. Ces étudiants nouent par ailleurs des relations d'amitié avec d'autres étudiants (inscrits ou non dans la même formation).

2.1 Création du graphe

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

```
create (sb:Etudiant:Femme {numINE:'20203344',nom:'Bordet',prenom:'Salome',dob:'1-apr-1997'})
return sb ;

match (sb:Etudiant:Femme {numINE:'20203344'})
create (mm:Etudiant:Femme {numINE:'20142345',nom:'Martin',prenom:'Marie',dob:'19-apr-1996'})
-[:apprecie]-> (sb),
(sb) -[:apprecie]-> (mm),(mm) -[:inscritDans {annee:2021}]->
(f1:Formation {codeF:'HAM1FE-300', nomF:'ICo', niveau:'M1', responsable:'K. T.'}),
(sb) -[:inscritDans {annee:2021}]-> (f1), (mm)  -[:inscritDans {annee:2020}]-> (f1),
(pm:Etudiant:Homme {numINE:'20161234',nom:'Martin',prenom:'Paul',dob:'20-aug-1995'})
-[:apprecie]-> (mm), (pm)
-[:inscritDans {annee:2021}]->
(f2:Formation {codeF:'HAM1GEO', nomF:'Geomatique', niveau:'M1', responsable:'C. G.'})
return * ;
```

Vous choisirez le prénom des étudiants et le nomF des formations pour donner une étiquette aux nœuds visualisés et vous mentionnerez les labels de chacun des nœuds. Les autres propriétés des nœuds ne seront pas représentées. Vous indiquerez également le type des relations.

2.2 Consultation du graphe

2. Vous écrirez en langage Cypher, la requête : "retourner les nœuds d'étudiants que Paul et Salome apprécient tous les deux"

3. 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 (f1:Formation) <-[id1:inscritDans]- (e1:Etudiant) -[a:apprecie]-> (e2:Etudiant)
-[id2:inscritDans]-> (f1:Formation)
where id1.annee = id2.annee
return e1.prenom, e2.prenom, f1.nomF, id1.annee
```

4. Une nouvelle requête Cypher vous est donnée (toujours sur le graphe créé). Vous donnerez la signification de cette requête et et vous expliquerez le résultat obtenu que vous illustrerez en dessinant le graphe associé.

```
:POST /rdf/cypher { "cypher":"MATCH (c:Etudiant {prenom:'Marie'}) RETURN c" , "format" : "N3"}

@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix neovoc: <neo4j://vocabulary#> .
@prefix neoind: <neo4j://individuals#> .


neoind:76 a neovoc:Etudiant, neovoc:Femme;
  neovoc:dob "19-apr-1996";
  neovoc:nom "Martin";
  neovoc:numINE "20142345";
  neovoc:prenom "Marie" .
```

3 Partie CouchDB (8 points)

Un exemple de documents JSON décrivant des étudiants de l'université Montpellier, et gérés au sein d'une BD CouchDB, vous est donné.

```
{
      "docs":
[
{ "_id" :"20142345",
  "nom" : "Martin",
  "prenom" : "Marie",
  "type": "etudiant",
  "age" : 21,
  "genre" : "femme",
  "formations": [{"codeF":"HAM1FE-300","nomF":"ICo","niveau":"M1","annee":2021}]
},
{ "_id" :"20161234",
  "nom" : "Martin",
  "prenom" : "Paul",
  "type": "etudiant",
  "age" : 25,
  "genre" : "homme",
  "formations": [{"codeF":"HAM1FE-300","nomF":"ICo","niveau":"M1","annee":2021},
{"codeF":"HAM1GEO","nomF":"Geomatique","niveau":"M1","annee":2020}]
},
{ "_id" :"20203344",
  "nom" : "Bordet",
  "prenom" : "Salome",
  "type": "etudiant",
  "age" : 23,
  "genre" : "femme",
  "formations": [{"codeF":"HAM1FE-300","nomF":"ICo","niveau":"M1","annee":2021},
{"codeF":"HAM1FE-300","nomF":"ICo","niveau":"M1","annee":2020},
{"codeF":"HAM1GEO","nomF":"ICo","niveau":"M1","annee":2019}]
}
]
}
```

3.1 Structuration des documents

Vous donnerez votre perception sur l'organisation des documents retenue par le modélisateur. Quelle autre structuration vous aurait semblé pertinente ? Vous donnerez des exemples.

3.2 Consultation au travers de vues matérialisées

Vous donnerez votre compréhension des vues Map et Map/Reduce suivantes (écriture en javascript du corps des fonction map et 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.type ==’etudiant’)
  emit([doc._id, doc.genre, doc.prenom], doc.age);
}
```

2. Vue Map 2

```
function (doc) { if (doc.type ==’etudiant’ && doc.genre ==’femme’)
  {
    if (Array.isArray(doc.formations))
    {
      for (var f in doc.formations)
       emit([doc.formations[f].niveau, doc.formations[f].codeF,
doc.formations[f].annee], 1);
     }
   }
  }
```

3. Vue Map Reduce La vue Map 2 est enrichie d'une fonction reduce qui correspond à _sum

```
function (doc) { if (doc.type ==’etudiant’ && doc.genre ==’femme’)
  {
    if (Array.isArray(doc.formations))
    {
      for (var f in doc.formations)
       emit([doc.formations[f].niveau, doc.formations[f].codeF,
doc.formations[f].annee], 1);
     }
   }

  }

function(keys, values, rereduce) {
  return sum(values);
}
```