



Session : 1

Durée de l'épreuve : 1 heure 30

Date : 24 novembre 2021

Documents autorisés : tous sous format papier

Mention Informatique

Matériel utilisé : aucun

Master 2<sup>ème</sup> année 2021/2022 : HAI914I

## 1 Questions de cours (4 points)

Vous expliquerez très brièvement (moins de 10 lignes) le système QNRW abordé lors du cours autour du système CouchDB (mais qui dépasse le cadre de ce système), en exploitant au besoin les notions suivantes :

1. scalabilité
2. réplication
3. partitionnement
4. postulat CAP et système distribué

## 2 Partie Neo4J (8 points)

Le contexte des liens familiaux dans la famille de Camille Claudel (artiste reconnue dans le domaine de la sculpture du XIXe siècle) et de son frère Paul Claudel (écrivain tout aussi connu) est exploité.

### 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 (louis:Personne:Homme {nom:'Claudel', prenom:'Louis', naissance:1826})
-[:MARIE_A {annee_mariage:1866}]->
(louise:Personne:Femme {nom:'Cerveaux', prenom:'Louise', naissance:1840}),
(louis) <-[:A_PARENT]- (camille:Personne:Femme {nom:'Claudel', prenom:'Camille', naissance:1864}),
(camille) -[:A_PARENT]-> (louise), (louis) <-[:A_PARENT]-
(paul:Personne:Homme {nom:'Claudel', prenom:'Paul', naissance:1868})
RETURN louise, louis, camille, paul

MATCH (paul:Homme {nom:'Claudel', prenom:'Paul', naissance:1868})
CREATE (marie:Personne:Femme {nom:'Perrin', prenom:'Marie'}) -[:MARIE_A {annee_mariage:1906}]->
(paul),
(louise:Personne:Femme {nom:'Vetch', prenom:'Louise', naissance:1905}) -[:A_PARENT]->
(paul), (louise) -[:A_PARENT]->
(rosalie:Personne:Femme {nom:'Vetch', prenom:'Rosalie', naissance:1871}),
(henri:Personne:Homme {nom:'Vetch', prenom:'Henri', naissance:1898}) -[:A_PARENT]->
(rosalie), (pierre:Personne:Homme {nom:'Claudel', prenom:'Pierre', naissance:1908}) -[:A_PARENT]->
(paul), (pierre) -[:A_PARENT]-> (marie),
(reine:Personne:Femme {nom:'Paris', prenom:'Reine', naissance:1910})
-[:A_PARENT]-> (marie), (reine) -[:A_PARENT]-> (paul),
(reinemarie:Personne:Femme {nom:'Paris', prenom:'Reine-Marie'})
-[:A_PARENT]-> (reine), (renee:Personne:Femme {nom:'Nantet', prenom:'Renee', naissance:1917})
-[:A_PARENT]-> (marie), (renee) -[:A_PARENT]-> (paul),
(victoire:Personne:Femme {nom:'Nantet', prenom:'Marie-Victoire'})-[:A_PARENT]-> (renee),
RETURN *
```

Vous choisirez le prénom des personnes 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 : "donner les petits-enfants de Paul Claudel"
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 (cc:Femme {nom:'Claudel', prenom:'Camille'}) -[:A_PARENT]-> (pe:Personne) <-[:A_PARENT]-
(fs:Personne) <-[:A_PARENT]- (nn:Personne)
WHERE cc.prenom <> fs.prenom
RETURN *
```

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 (vous pouvez illustrer en dessinant le graphe associé).

```
:POST /rdf/cypher { "cypher":"MATCH (c:Personne {prenom:'Camille'}) 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:1069 a neovoc:Femme, neovoc:Personne;
  neovoc:naissance "1864"^^<http://www.w3.org/2001/XMLSchema#long>;
  neovoc:nom "Claudel";
  neovoc:prenom "Camille" .
```

## 3 Partie CouchDB (8 points)

Un exemple de documents JSON décrivant très partiellement trois bâtiments d'enseignement du campus Triolet, et gérés au sein d'une BD CouchDB, vous est donné.

```
{
  "docs":
  [
    { "_id" : "triolet_36",
      "code" : "t_b36",
      "campus" : "Triolet",
      "type": "batiment",
      "destination" : "enseignement",
      "annee_construction" : 2019,
      "salles": [{"designation":"A36.03","capacite":120,"type":"amphi","accessibilite":"oui","video":"oui"},
{"designation":"A36.02","capacite":120,"type":"amphi","accessibilite":"oui","video":"oui"},
{"designation":"A36.01","capacite":120,"type":"amphi","accessibilite":"oui","video":"oui"},
{"designation":"TD36.202","capacite":40,"type":"TDInfo","accessibilite":"oui","video":"oui"},
{"designation":"SC36.04","capacite":80,"type":"amphi","accessibilite":"oui","video":"oui"}]
    },
    { "_id" : "triolet_16",
      "code" : "t_b16",
      "campus" : "Triolet",
      "type": "batiment",
      "destination" : "enseignement",
      "annee_construction" : 1966,
      "salles": [{"designation":"SC16.03","capacite":120,"type":"amphi","accessibilite":"oui","video":"oui"},
{"designation":"TD16.02","capacite":18,"type":"TD","accessibilite":"oui","video":"oui"},
{"designation":"TPDeptInfo","capacite":40,"type":"TDInfo","accessibilite":"oui","video":"oui"},
{"designation":"TPBio","capacite":40,"type":"TP","accessibilite":"oui","video":"oui"}]
    },
    { "_id" : "triolet_05",
      "code" : "t_b05",
      "campus" : "Triolet",
      "type": "batiment",
```



```
"destination": "enseignement",
"annee_construction": 1964,
"salles": [{ "designation": "A.02", "capacite": 275, "type": "amphi", "accessibilite": "oui", "video": "oui" },
{ "designation": "TD5.125", "capacite": 20, "type": "TDInfo", "accessibilite": "oui", "video": "oui" },
{ "designation": "TD5.126", "capacite": 31, "type": "TDInfo", "accessibilite": "oui", "video": "oui" },
{ "designation": "TD5.210", "capacite": 40, "type": "TD", "accessibilite": "oui", "video": "oui" }
]
}
```

### 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 == 'batiment' && doc.destination == 'enseignement')
  emit(doc.code, doc.annee_construction);
}
```

#### 2. Vue Map 2

```
function (doc) { if (doc.type == 'batiment' && doc.destination == 'enseignement')
{
  if (Array.isArray(doc.salles))
  {
    for (var s in doc.salles)
      emit([doc.code, doc.salles[s].type], doc.salles[s].capacite);
  }
}
}
```

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

```
function (doc) { if (doc.type == 'batiment' && doc.destination == 'enseignement')
{
  if (Array.isArray(doc.salles))
  {
    for (var s in doc.salles)
      emit([doc.code, doc.salles[s].type], doc.salles[s].capacite);
  }
}

function(keys, values, rereduce) {
```

```
    return sum(values);  
}
```