



Session : 1

Durée de l'épreuve : 1h30

Date : 20 novembre 2024

Documents autorisés : tous sous format papier

Mention Informatique

Matériel utilisé : aucun

Master 2^{ème} année 2024/2025 : HAI914I

1 Questions de cours (4 points)

Vous citerez et expliquerez (en une vingtaine de lignes) les deux grands principes sous-tendant la distribution dans les systèmes NoSQL. Vous pourrez illustrer vos propos à partir du cluster CouchDB entrevu en cours, et des principes du système QNRW. Vous pourrez également rapprocher ces deux grands principes du postulat CAP.

2 Préalable)

Nous exploitons le contexte de la course en solitaire du Vendée Globe, qui est une course autour du monde de voiliers qui démarre des Sables d'Olonne et revient aux Sables d'Olonne, et qui se déroule tous les 4 ans. Une édition se déroule actuellement, les bateaux à voile d'un genre particulier nommé Imoca sont partis le 10 novembre et devraient revenir vers le mois de mars. Les Imoca sont des bateaux de course, dotés d'une seule coque (monocoque), d'une grande surface de voile, et qui se déplacent à la seule force du vent. Un seul marin, encore désigné par skipper, navigue à bord du bateau (on dit aussi que le bateau est barré par un skipper). Certains bateaux sont remaniés et participent au fil du temps à plusieurs éditions de la course du Vendée Globe avec un nom, un skipper, et des sponsors différents. Ce dernier point nous permet d'aborder la modélisation d'entités évolutives. En effet, les bateaux évoluent matériellement dans le temps, et ne sont plus tout à fait les mêmes qu'à leur construction. La modélisation d'entités évolutives est plus difficile que la modélisation d'entités plus statiques qui vont conserver leurs caractéristiques le long de leur cycle de vie. L'examen propose donc de traiter de cet aspect, avec des modèles graphe ou document qui restent perfectibles.

3 Partie Neo4J (8 points)

3.1 Création du graphe (3 points)

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

```
create (foncia:Imoca:Monocoque {nom:'FONCIA',nomBaptême:'FONCIA',architecte:'Farr Yacht Design',
chantier:'CDK Technologies', mise_eau:2007}),
(michelD:Skipper {nom:'Desjoyeaux',prenom:'Michel',nationalite:'France',
genre:'M',dateN:'1965-07-16'}), (michelD) -[p1:participeVG {annee:2009, rang:1}]-> (foncia)
return foncia, p1, michelD ;
```

```
match (foncia:Imoca {nom:'FONCIA'})
create (maitreCoq:Imoca:Monocoque {nom:'Maitre CoQ'}) <-[d:devient]- (foncia),
(jeremyB:Skipper {nom:'Beyou',prenom:'Jeremie',nationalite:'France',
genre:'M',dateN:'1976-06-26'}),
(jeremyB) -[p2:participeVG {annee:2013, rang:0}]-> (maitreCoq) ;
```

```
match (maitreCoq:Imoca {nom:'Maitre CoQ'})
create (fmv:Imoca:Monocoque {nom:'Finistere Mer Vent'}) <-[d:devient]- (maitreCoq),
(jeanKing:Skipper {nom:'Le Cam',prenom:'Jean',nationalite:'France',
```

```

genre:'M',dateN:'1959-04-17')),
(jeanKing) -[p2:participeVG {annee:2017, rang:6}]-> (fmv) ;

match (fmv:Imoca {nom:'Finistere Mer Vent'}), (king:Skipper {nom:'Le Cam'})
create (yesWeCam:Imoca:Monocoque {nom:'Yes We Cam !'}) <-[d:devient]- (fmv),
(king) -[p2:participeVG {annee:2021, rang:4}]-> (yesWeCam) ;

match (yesWeCam:Imoca {nom:'Yes We Cam !'})
create (dev:Imoca:Monocoque {nom:'Devenir'}) <-[d:devient]- (yesWeCam),
(violetteD:Skipper {nom:'Dorange',prenom:'Violette',nationalite:'France',
genre:'F',dateN:'2001-04-17'}), (violetteD)
-[p3:participeVG {annee:2025}]-> (dev) ;

```

Vous choisirez le nom du bateau, le nom des skippers 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. Vous ferez apparaître pour les relations de type *participeVG*, la propriété *annee* ainsi que sa valeur.

3.2 Consultation du graphe (5 points)

2. Vous écrierez en langage Cypher, la requête : "retourner le nœud Imoca qui a pour nom de baptême FONCIA"
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 (i:Imoca {nomBaptême:'FONCIA'}) -[:devient*]-> (e:Imoca), (e) <-[p:participeVG]- (s:Skipper)
return i.nom, e.nom, s.nom, p.annee, p.rang

```

4. Le plugin Neosemantics est mobilisé et 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. Expliquez en quelques lignes ce qui rend difficile le passage d'un modèle à base de graphe attribué tel que celui de Neo4J à un modèle RDF.

```

:POST /rdf/cypher { "cypher":
"MATCH g = (i:Imoca {nom:'Yes We Cam !'}) <-[p:participeVG]-> (s:Skipper)
RETURN g", "format" : "N3"}

```

Le résultat de la requête au format N3

```

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

```

```

neoind:106 a neovoc:Skipper;
  neovoc:dateN "1959-04-17"; ✓
  neovoc:genre "M"; ✓
  neovoc:nationalite "France";
  neovoc:nom "Le Cam"; ✓
  neovoc:participeVG neoind:107;
  neovoc:prenom "Jean" .

```

```

neoind:107 a neovoc:Imoca, neovoc:Monocoque;
  neovoc:nom "Yes We Cam !" .

```

4 Partie CouchDB (8 points)

Deux exemples de documents JSON traitent de la modélisation des voiliers qui participent au Vendée Globe et de leurs skippers de manière différente. Deux instances de bases de données CouchDB, ont été construites à cet effet.



4.1 Première proposition de structuration

```
{
  "docs":
  [
    { "_id" : "baf26487ff27586e733fbfb51d000461",
      "nom_baptême" : "Apivia",
      "architecte" : "Guillaume Verdier",
      "chantier": "CDK Technologies",
      "mise_eau":2019,
      "courses": [{ "nom": "Apivia", "skipper": "Charlie Dalin", "rang": 2, "annee": 2021 },
                   { "nom": "Occitane en Provence", "skipper": "Clarisse Cremer", "annee": 2025 } ]
    },
    { "_id" : "baf26487ff27586e733fbfb51d00125b",
      "nom_baptême" : "FONCIA",
      "architecte" : "Farr Yacht Design",
      "chantier": "CDK Technologies",
      "mise_eau":2007,
      "courses": [{ "nom": "FONCIA", "skipper": "Michel Desjoyeaux", "rang": 1, "annee": 2009 },
                   { "nom": "Maitre CoQ", "skipper": "Jeremie Beyou", "annee": 2012 },
                   { "nom": "Finistere Mer Vent", "skipper": "Jean Le Cam", "rang": 6, "annee": 2017 },
                   { "nom": "Yes We Cam !", "skipper": "Jean Le Cam", "rang": 4, "annee": 2021 },
                   { "nom": "Devenir", "skipper": "Violette Dorange", "annee": 2025 } ]
    }
  ]
}
```

4.2 Seconde proposition de structuration (incomplet)

```
{
  "docs":
  [
    { "_id" : "Apivia",
      "type": "Imoca",
      "nom_baptême" : "Apivia",
      "architecte" : "Guillaume Verdier",
      "chantier": "CDK Technologies",
      "mise_eau":2019,
      "skipper": "Charlie Dalin",
      "rang": 2,
      "annee": 2021 },
    { "_id" : "OccitaneEnProvence",
      "type": "Imoca",
      "nom_baptême" : "Apivia",
      "nom" : "Occitane en Provence",
      "skipper": "Clarisse Cremer",
      "annee": 2025 }
  ]
}
```

+ } (2025)

4.3 Structuration des documents (5 points)

Vous donnerez votre perception sur l'organisation des documents retenue par le modélisateur. Vous ferez un choix entre les deux modélisations retenues, tout en expliquant ce que vous pourriez modifier dans la structuration retenue en vue d'amélioration. Vous pouvez illustrer au travers d'exemples.

4.4 Consultation au travers de vues matérialisées (3 points)

Seule la seconde proposition de structuration est traitée dans cet exercice. Vous donnerez votre compréhension de la vue Map/Reduce suivante (écriture en javascript du corps des fonctions). Quels sont les résultats renvoyés par la fonction map, puis par la fonction reduce et ce pour les différents niveaux de regroupement (illustrez les résultats obtenus avec les documents de la structuration incomplète présentée au dessus) ?

1. Vue Map

```
function (doc) {  
  if (doc.type == 'Imoca' && doc.nom)  
  { emit([doc.nom_baptême,1], doc.skipper); }  
  if (doc.type == 'Imoca' && doc.nom_baptême && !doc.nom)  
  { emit([doc.nom_baptême,0], doc.skipper); }  
}
```

2. Vue Map Reduce La vue Map est enrichie d'une fonction Reduce

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