

Bases de données NoSQL

Révision Complète

Plan du cours

- Concepts Fondamentaux
- Types de Bases NoSQL
- MongoDB en détail
- Neo4j et bases de données graphes
- Cas pratiques
- Exercices type examen

Concepts Fondamentaux

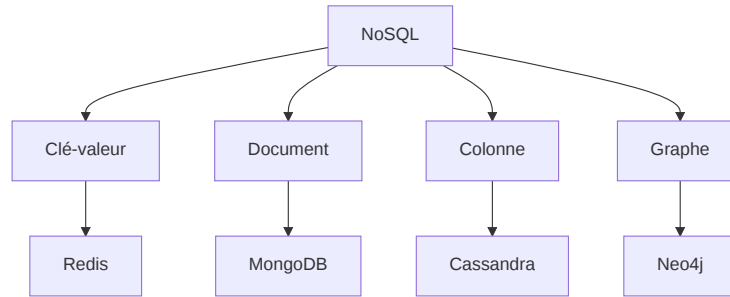
Propriétés BASE

- Basically Available
- Soft State
- Eventually Consistent

Théorème CAP

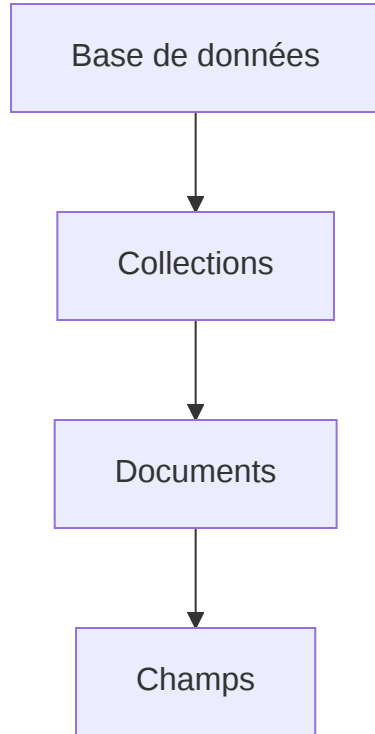
- Consistency
- Availability
- Partition Tolerance

Types de Bases NoSQL



MongoDB en détail

Structure hiérarchique



Requêtes MongoDB

Opérateurs de comparaison

```
// Égal
{field: {$eq: value}}
// Différent
{field: {$ne: value}}
// Supérieur
{field: {$gt: value}}
```

Exemple pratique

```
db.users.find({
  age: {$gte: 18},
  city: "Paris"
})
```

Write Concerns

w: 0

- Sans accusé
- Performance maximale
- Risque de perte

w: 1

- Primary only
- Compromis standard
- Bon pour dev

w: "majority"

- Majorité des nœuds
- Plus sécurisé
- Production

Neo4j

Éléments de base

```
// Création d'un nœud
CREATE (p:Person {name: 'John'})

// Création d'une relation
MATCH (a:Person), (b:Person)
WHERE a.name = 'John' AND b.name = 'Jane'
CREATE (a)-[:FOLLOWS]→(b)
```

Requêtes simples

```
// Trouver des amis d'amis
MATCH (p:Person)-[:FOLLOWS]→(:Person)-[:FOLLOWS]→(fof:Person)
WHERE p.name = 'John'
RETURN DISTINCT fof.name
```


Optimisation Neo4j



Bonnes pratiques

- Utiliser les labels
- Indexer les propriétés clés
- Patterns restrictifs d'abord

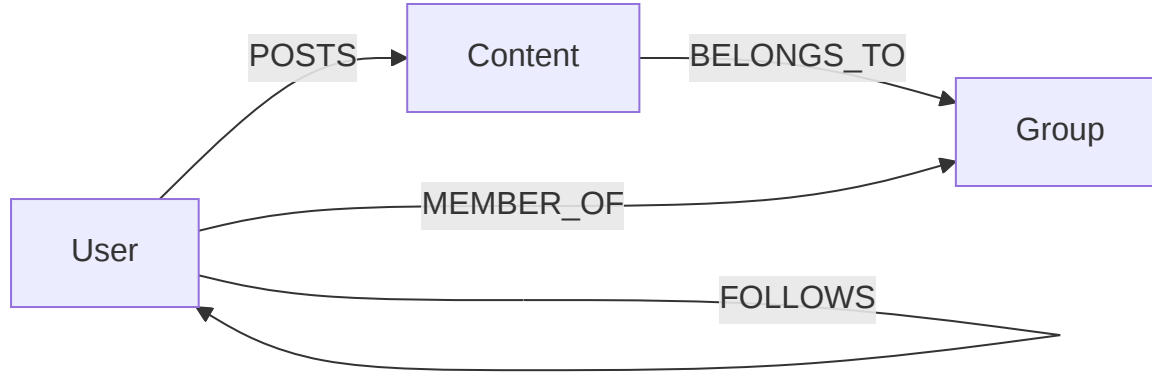


À éviter

- Parcours complet du graphe
- Requêtes sans labels
- Relations non indexées

Cas Pratique : Réseau Social

Modélisation



Considérations

- Choix du type de base
- Compromis CAP
- Stratégie de scaling
- Cohérence vs Disponibilité

Questions d'examen types

Théorique

- QCM sur les concepts
- Questions de cours

Pratique

- Exercice Mongo
- Exercice Neo4j

Merci !

Documentation MongoDB · Documentation Neo4j