NoSQL Systems

MongoDB – Operations de la base

Paola Andrea Gómez Barreto

Sources. Tutoriels and documentation intégrés de MongoDB

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

OPÉRATIONS DE BASE DE DONNÉES

- ✓ **Interroger / Agréger**
- ✓ Insérer
- ✓ Supprimer
- ✓ Mettre à jour
- ✓ Indexer

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

TROUVER DES DONNÉES

Commande “find”

db.people.find()	Afficher tous les documents
db.people.find().pretty()	Afficher les données dans un format le plus clair
db.people.find().sort({“name”: 1})	Afficher tous les documents et le trier par nom croissant
db.people.find({"age": 34}).pretty()	Sélection. Filtrer par un ou plusieurs champs. Afficher tous les champs pour chaque élément trouvé.
db.people.find({"age": 34, "isActive": true} , {"name":1,"isActive": 1})	Projection Filtrer par un ou plusieurs champs (premier paramètre). Seuls les champs indiqués dans le second paramètre seront affichés
db.people.find({"age":{ **$gt** : 30} }, {"name": 1, "age": 1 })	Opérateur de comparaison $ gt (>) Afficher les personnes dont l'âge est supérieur à 30 Pour chaque personne, seulement le nom et l'âge seront affichés.
db.people.find({ "interests":{ $all: ["tennis" , "swimming"] }}, {"name": 1, "age": 1 })	Trouver des gens ayant "tennis" et "swimming" parmi leurs intérêts.

Note: Si le champ est imbriqué, il faut utiliser le chemin de navigation complet pour s'y référer. Rappelez-vous d'ajouter les guillemets.

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

TROUVER DES DONNÉES

MongoDB fournit de nombreux opérateurs pour faire:

- Opérations de comparaison:

  $gt, $gte, $lt, $lte, $ne (> , >=, < , <= , !=)

- Pour en savoir l'existence d'un champ particulier:
  - $exists: true (rechercher des documents avec un champ particulier)
  - $type: <BSON type>|<alias> (pour filtrer les documents avec un champ et un type particulier)

- Opérations logiques:

  $or, $and, $not, $nor

- Trouver dans les tableaux:

  $all, $in, $nin, $size, "array.$":X, $slice, "array.X"

(http://docs.mongodb.org/manual/reference/operator/)

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

AGRÉGATION

SQL	Mongo
WHERE	$match
GROUP BY	$group
HAVING	$match
SELECT	$project
ORDER BY	$sort
LIMIT	$limit
SUM	$sum
COUNT	$sum
JOIN	Pas d'opérateur correspondant direct; cependant, il existe $unwind pour gérer les champs incorporés dans un document et $lookup pour faire une correspondance d'égalité entre un champ des documents d'entrée et un champ provenant des documents de l'autre collection.

```
SELECT cust_id,
       SUM(price) as total
FROM orders
WHERE status = 'A'
GROUP BY cust_id
```

```
db.orders.aggregate (
    [
        { $match: { status: 'A' } },
        { $group: { _id: "$cust_id" , total: { $sum: "$price" } } }
    ]
)
```


(http://docs.mongodb.org/manual/reference/sql-aggregation-comparison/)

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

AGRÉGATION / UNWIND

Déconstruit un champ de tableau à partir des documents d'entrée pour produire un document pour chaque élément. Les attributs du document qui contient le tableau sont inclus dans chaque document des résultats.

```
{ $unwind: <field path> }
```

En considérant:

```
{ "_id" : 1, "item" : "ABC1", sizes: [ "S", "M", "L"] }
```

Interroger:

```
db.inventory.aggregate( [ { $unwind : "$sizes" } ] )
```

Résultats:

```
{ "_id" : 1, "item" : "ABC1", "sizes" : "S" }
{ "_id" : 1, "item" : "ABC1", "sizes" : "M" }
{ "_id" : 1, "item" : "ABC1", "sizes" : "L" }
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

UNWIND DES DOCUMENTS IMBRIQUÉS

Lorsqu'une recherche est effectuée sur les documents d'un tableau, la réponse apporte le document qui contient le tableau, y compris le tableau entier. Pour obtenir les documents du tableau souhaités, il est nécessaire d'aplatir le tableau.

En considérant la collection « companies »:

```
{"_id":1,"nameCom":"Company1",departaments: [ {"nameDept":"dept1", "numEmployees":4},
                                              {"nameDept":"dept2", "numEmployees":20}
                                            ]},
{"_id":2,"nameCom":"Company2",departaments: [ {"nameDept":"dept1", "numEmployees":6},
                                              {"nameDept":"dept1", "numEmployees":10}
                                            ]}
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

UNWIND DES DOCUMENTS IMBRIQUÉS

Interroger:

```
db.companies.aggregate ( [ { $match: { "departaments.numEmployees": { $gte: 15 } } } ] )
```

Résultats:

```
{"_id":1,"nameCom":"Company1",departaments: [ {"nameDept":"dept1", "numEmployees":4},
                                              {"nameDept":"dept2", "numEmployees":20} ]}
```

Maintenant Unwind:

```
db.companies.aggregate ( [
        { $match: { "departaments.numEmployees": { $gte: 15 } } },
        { $unwind: "departaments" }
] )
```

Résultats:

```
{"_id":1,"nameCom":"Company1",departaments: {"nameDept":"dept1", "numEmployees":4}  }

{"_id":1,"nameCom":"Company1",departaments: {"nameDept":"dept2", "numEmployees":20} }
```

Et maintenant? …

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

JOINTURE ET $LOOKUP

Il n'y a pas d'opérateur de jointure correspondant direct; Cependant, il existe dans l'opérateur $lookup.

```
db.< Nom de la collection >.aggregate(
        [ { $lookup: {
                from: < collection à rejoindre >,
                localField: < champ des documents d'entrée >,
                foreignField: < champ provenant des documents de la collection
                                "from" >,
                        as: < champ de tableau de sortie >
        }} ]
)
```

Ajoute à chaque document de la collection source un tableau contenant les documents de la collection cible correspondant au champ local et étranger.

À des fins de correspondance, si localfield n'existe pas dans la collection source, elle est traitée comme ayant une valeur nulle.

Si localField est un tableau, une opération $ unwind doit être appliquée avant.

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

EXEMPLE $LOOKUP

orders

```
{ "_id" : 1, "item" : "MON1003", "price" : 350, "quantity" : 2, "specs" :
[ "27 inch", "Retina display", "1920x1080" ], "type" : "Monitor" }
```

inventory

```
{ "_id" : 1, "sku" : "MON1003", "type" : "Monitor", "instock" : 120,
"size" : "27 inch", "resolution" : "1920x1080" }
{ "_id" : 2, "sku" : "MON1012", "type" : "Monitor", "instock" : 85,
"size" : "23 inch", "resolution" : "1280x800" }
{ "_id" : 3, "sku" : "MON1031", "type" : "Monitor", "instock" : 60,
"size" : "23 inch", "display_type" : "LED" }
```

```
db.orders.aggregate([
        { $lookup: {
                from: "inventory",
        localField: "item",
        foreignField: "sku",
                as: "inventory_docs"
        } }
 ])
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

EXEMPLE $LOOKUP

```
{ "_id" : 1, "item" : "MON1003", "price" : 350, "quantity" : 2, "specs" :
[ "27 inch", "Retina display", "1920x1080" ], "type" : "Monitor" }
```

```
{ "_id" : 1, "sku" : "MON1003", "type" : "Monitor", "instock" : 120,
"size" : "27 inch", "resolution" : "1920x1080" }
{ "_id" : 2, "sku" : "MON1012", "type" : "Monitor", "instock" : 85,
"size" : "23 inch", "resolution" : "1280x800" }
{ "_id" : 3, "sku" : "MON1031", "type" : "Monitor", "instock" : 60,
"size" : "23 inch", "display_type" : "LED" }
```

Orders'

```
{
  "_id" : 1,
  "item" : "abc",
  "price" : 12,
  "quantity" : 2,
  "inventory_docs" : [
    { "_id" : 1, "sku" : "abc", description: "product 1", "instock" : 120 }
  ]
}
{
  "_id" : 2,
  "item" : "jkl",
  "price" : 20,
  "quantity" : 1,
  "inventory_docs" : [
    { "_id" : 4, "sku" : "jkl", "description" : "product 4", "instock" : 70 }
  ]
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

INSÉRER DES DONNÉES

Commandes “insert” et “save”

Le **_id** est un champ unique attribué manuellement ou automatiquement..

Save: si l'identifiant existe déjà, l'information est remplacée.

Insert: si l'identifiant existe déjà, insert renvoie une erreur.

```
db.people.insert(
    {
        "_id": ObjectId ("51c4218"),
        "name": "Claudia",
        "NumberKids": 3,
        "isActive": true,
        "interests": ["swimming", "tennis"]
        "favoriteCountries":
      [
        {
            "name" : "France",
            "capital" : ”Paris"
        },
        {
            "name": "Japan"
        }
      ]
    }
)
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

SUPPRIMER DES DONNÉES

Commande “remove”

Il faut une requête qui filtre les documents à supprimer.
S'il n'y a pas de requête, tous les documents seront supprimés.

db.people.remove({"_id"=2})

Il supprime le document dont _id est 2.

db.people.remove({})

Il supprime tous les documents de la collection "people"

METTRE À JOUR DES DONNÉES

Commande "update"

Le premier paramètre filtre les documents à modifier. Le second paramètre contient les informations à modifier. Par défaut, le premier document trouvé sera le document mis à jour.

db.people.update({"name": "Claudia"}, {"kids":4})

Il efface toutes les informations du premier document trouvé (sauf l'identifiant), et insère les informations fournies par le deuxième paramètre.

db.people.update({"name": "Claudia"}, { $set: {"kids":4} })

Il remplace sur le premier document trouvé seulement les informations fournies dans le deuxième paramètre, et il préserve les informations des autres champs.

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

METTRE À JOUR DES DONNÉES

Commande “update”

Pour mettre à jour tous les documents trouvés, un troisième paramètre doit être fourni avec l'option "multi" activée:

```
db.people.update({"name": "Claudia"}, { $set: {"kids":4} } , { multi : true })
```

S'il n'y a pas de documents correspondant au premier paramètre, un nouveau document peut être créé. Pour cela, un troisième paramètre doit être fourni avec l'option "upsert" activée:

```
db.people.update({"name": "ClaudiaRoncancio"}, { $set: {"kids":4} } , { upsert : true })
```

UFR IM²AG

UNIVERSITÉ Grenoble Alpes

QUESTIONS?