

TP NoSQL, MongoDB

Halim Djerroud (hdd@ai.univ-paris8.fr)

1 Installation

Pour lancer mongo pour la première fois :

```
sudo docker run --name mymongo -d mongo:4.4
sudo docker exec -it mymongo bash
mongo
```

Si le docker existe déjà :

```
sudo docker start mymongo
sudo docker exec -it mymongo bash
mongo
```

2 MongoSH

Affichez la liste de toutes les bases de données à l'aide de la commande suivante :

```
show dbs
```

Pour créer une base de données ou sélectionner une base de données que vous avez créée utiliser les commandes :

```
use my_first_database
```

Affichez la liste des collections de cette base de données :

```
show collections
```

Insérez un document dans la collection d'articles :

```
db.articles.insertOne({'name' : 'Introduction a NoSQL'})
```

The screenshot shows the MongoDB Compass interface. At the top, there's a navigation bar with tabs for 'Documents', 'Aggregations', 'Schema', 'Explain Plan', 'Indexes', and 'Validation'. Below the navigation bar, there's a search bar with a filter option ('field: "value"') and a 'FIND' button. A 'ADD DATA' button is also present. On the right side, there are buttons for 'OPTIONS', 'RESET', and '...'. Below the search bar, it says 'Displaying documents 1 - 3 of 3'. The main area shows a single document with the following fields:

```

_id: ObjectId("6188f13177a4a4edc252")
name: "Introduction to database"

```

FIGURE 1 – mongodb-compass

Vous pouvez remarquer dans la représentation graphique qu'un nouveau document, avec un élément généré "`_id`", a été inséré, ayant la structure illustrée dans la Figure 1 :

Vous pouvez insérer plusieurs documents dans vos collections en même temps :

```

db.articles.insertMany([
{
    title: 'MongoDB Overview',
    description: 'MongoDB is a no sql database',
    by: 'IT Students',
    url: 'http://it-students.com/mongo',
    tags: ['mongodb', 'database', 'NoSQL'],
    likes: 100
},
{
    title: 'NoSQL Database',
    description: "NoSQL database doesn't have tables",
    by: 'IT-Students',
    url: 'http://it-students.com/nosql',
    tags: ['mongodb', 'database', 'NoSQL'],
    likes: 20,
    comments: [
    {
        user:'user1',
        message: 'Mongo is the best!',
        dateCreated: new Date(2021,11,10,2,35),
        like: 0
    }]
})

```

Lorsque vous rafraîchissez votre collection, elle affiche maintenant trois documents. Attention, chaque document a une structure différente, et que nous n'avons pas défini un schéma prédéfini qui montre la structure autorisée à l'utilisateur. L'utilisateur est libre de faire toutes les erreurs à **ses risques et périls !**

Nous pouvons cependant détecter le schéma à partir des documents insérés (ces bases de données sont appelées "schema on read" pour des raisons de sécurité). (ce n'est pas pour rien que ces bases de données sont appelées "schema on read"...).

Cliquez sur l'anglet "Schema" dans Compass, puis sur Analyser. Vous devriez obtenir un résultat comme le illustré dans la figure 2 :

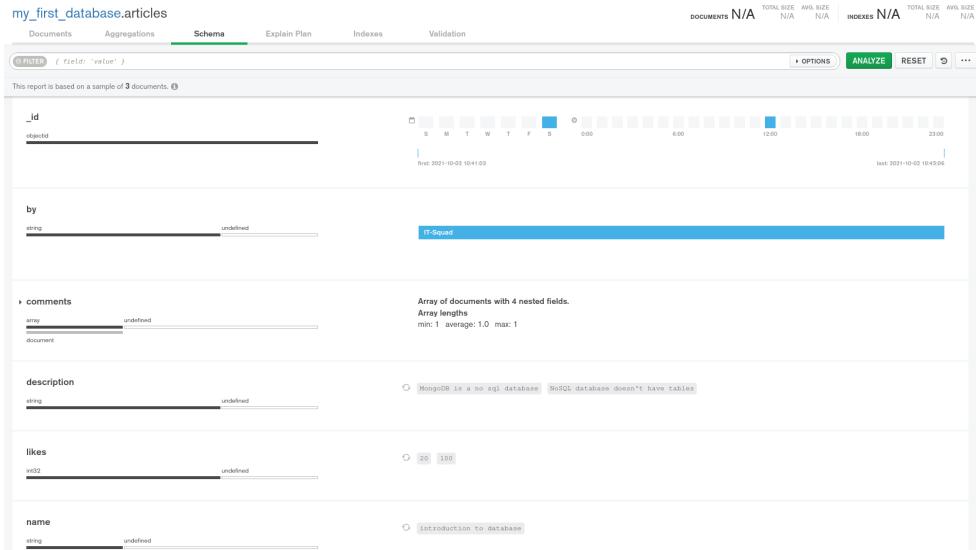


FIGURE 2 – mongodb-compass : Shemas

Pour interroger une collection, utilisez l'instruction suivante pour afficher tout les documents :

```
db.articles.find()
```

Vous pouvez également utiliser `findOne` pour ne retourner qu'un seul document (le premier).

Afin de rechercher une valeur définie, utilisez la requête suivante :

```
db.articles.find({ 'by' : 'IT-Students' })
```

Le résultat est illustré dans la Figure 3

Nous remarquons ici qu'il y a eu une erreur dans la valeur "by". Nous voulions que les deuxième et troisième articles et le troisième article devaient être écrits par le même sujet, mais nous avons fait une erreur de "name". Parce que la base de données n'effectue aucun contrôle sur les valeurs (pas de clés étrangères automatiquement définies automatiquement), tout s'est bien passé, même avec l'erreur.

Nous pouvons changer la valeur de l'autre article en utilisant l'instruction suivante :

```
db.articles.updateOne({ 'by' : 'IT Students' }, { $set: { 'by' : 'IT-Students' } })
```

```

my_first_database> db.articles.find({'by': 'IT-Students'})
[ {
    _id: ObjectId("616f17490715bbe8362d13ec"),
    title: 'NoSQL Database',
    description: "NoSQL database doesn't have tables",
    by: 'IT-Students',
    url: 'http://it-students.com/nosql',
    tags: [ 'mongodb', 'database', 'NoSQL' ],
    likes: 20,
    comments: [
        {
            user: 'user1',
            message: 'Mongo is the best!',
            dateCreated: ISODate("2021-12-10T01:35:00.000Z"),
            like: 0
        }
    ]
}
]

```

FIGURE 3 – Resultat commande `find`

Le document sera alors mis à jour. Si vous devez mettre à jour plusieurs documents, utilisez `findMany` au lieu de `findOne`.

On peut aussi effectuer des recherches plus sophistiquées :

- Cherchez une chaîne qui contient le mot Mongo :

```
db.articles.find({"title": /.*Mongo.*/})
```

- Inférieur :

```
db.articles.find({"likes":{$lt:50}})
```

- Supérieur :

```
db.articles.find({"likes":{$gt:20}})
```

```
db.articles.find({"likes":{$gte:20}})
```

- Différent :

```
db.articles.find({"likes":{$ne:20}})
```

- AND ou OR

```
db.articles.find({$and:[{"by": "IT-Students"}, {"title": "MongoDB Overview"}]})
```

```
db.articles.find({$or:[{"by": "IT-Students"}, {"title": "MongoDB Overview"}]})
```

- AND et OR

```
db.articles.find({$and:  
  [{"by": "IT-Students"},  
   {$or: [{"title": "MongoDB Overview"}, {"title": "NoSQL Database"}]}]  
}]  
)
```

Nous pouvons aussi effectuer plusieurs mises à jour à la fois :

```
db.articles.updateMany({ 'by': 'IT-Students' }, { $set: { 'by': 'IT-Squad' } })
```

Il est également possible d'effectuer une recherche avec un attribut (équivalent d'un select A,B au lieu de select *) :

```
db.articles.find({}, {"title": 1, "_id": 0})
```

Pour afficher une colonne, il faut l'ajouter à la liste, avec la valeur 1. L'`_id` étant le seul attribut qui est affiché par défaut, vous devez le désactiver explicitement pour ne pas l'afficher.

Pour afficher les documents triés par titre descendant, utilisez l'instruction suivante :

```
db.articles.find({}, {"title": 1, "_id": 0}).sort({"title": -1})
```

Vous pouvez également ajouter un index pour accélérer la recherche par une certaine colonne.

```
db.articles.ensureIndex({"title": 1})
```

Après avoir indexé le titre, si vous exécutez à nouveau la dernière instruction `find`, vous remarquez que, pour le premier document qui n'avait pas de titre, une nouvelle colonne `titre` est ajoutée, avec la valeur `NULL`. Pour le deuxième document qui n'avait pas de titre, une nouvelle colonne `titre` est ajoutée, avec la valeur `NULL`.

3 Exercie

Modélez la base de données `Mission` utilisée dans les TP précédents avec MongoDB. Insérer les données, réaliser les requêtes avec MongoDB et montrez leurs résultats dans un rapport PDF à rendre en binôme.