

**Master Mathématiques, Informatique, Décision, Organisation (MIDO)  
2ème année - MIAGE-IF App.**

**ANNEE 2022/ 2023**

**Désignation de l'enseignement** : SQL-NoSQL-NewSQL

**Nom du document** : TP MongoDB

**Rédacteur** : Maude Manouvrier

La reproduction de ce document par tout moyen que ce soit est interdite conformément  
aux articles L111-1 et L122-4 du code de la propriété intellectuelle

## TABLE DES MATIERES

<b>I.</b>	<b>UTILISATION DE LA CONSOLE EN LIGNE DE MONGODB .....</b>	<b>3</b>
	1. <i>Insertions, requêtes d'interrogation et de mises à jour.....</i>	<i>3</i>
	2. <i>Création de schémas et validation de documents.....</i>	<i>5</i>
<b>II.</b>	<b>MANIPULER DES DOCUMENTS JSON SOUS POSTGRESQL .....</b>	<b>7</b>
<b>III.</b>	<b>LIENS OU TROUVER DES EXEMPLES DE REQUETES .....</b>	<b>8</b>
<b>IV.</b>	<b>UTILISATION DE MONGOATLAS .....</b>	<b>8</b>

## I. Utilisation de la console en ligne de MongoDB

La documentation de MongoDB dispose d'une console en ligne. Pour l'utiliser, il suffit d'aller sur la page : <https://docs.mongodb.com/manual/tutorial/insert-documents/> (attention de sélectionner, à gauche dans la liste déroulante, la version 4.2, la console en ligne ayant disparue de la documentation des versions suivantes).

Il existe également 2 autres consoles<sup>1</sup> *MongoDB* en ligne :

<https://www.mycompiler.io/new/mongodb> et <https://onecompiler.com/mongodb>

*Attention : ne quittez pas la fenêtre (pensez à faire ouvrir dans une autre fenêtre ou un autre onglet lorsque vous cliquez sur un lien, notamment la documentation en ligne), sinon vous perdez la connexion et toutes les instructions que vous avez faites. Pensez à copier les instructions dans un éditeur de texte, si vous souhaitez les conserver.*

### 1. Insertions, requêtes d'interrogation et de mises à jour

1. Exercez-vous dans un premier temps avec les commandes d'insertion et de requêtes disponibles dans la documentation.

- a. Cliquer dans la console (cf. Figure 1) apparaissant dans la documentation pour vous connecter.

You can run the operation in the web shell below:



Figure 1 - Console en ligne disponible dans la documentation de MongoDB

- b. Taper l'instruction `show collections` dans la console pour lister les collections définies dans la base.
- c. Copier l'instruction d'insertion située au-dessus de la console en ligne et coller dans la console, puis taper sur *Enter*.
- d. Taper à nouveau l'instruction `show collections`.

*ND : Dans la suite, si le résultat des requêtes n'apparaît pas clairement, n'hésitez pas à cliquer sur *Full* pour agrandir la fenêtre (*Esc* permet de revenir en mode normal). De même vous pouvez, dans la console, réafficher une commande précédemment saisie en utilisant les flèches haut et bas de votre clavier.*

---

<sup>1</sup> Attention ces consoles ne semblent pas apprécier les sauts de ligne en début de document. La console <https://onecompiler.com/mongodb> n'accepte pas les champs `url`.

2. Insérer le document ci-dessous dans la même collection ou dans une collection, dont vous choisirez le nom, (cf <https://www.mongodb.com/docs/mongodb-shell/crud/insert/#std-label-mongosh-insert>) :

```
{
  titre: 'TP MongoDB',
  description: 'Découverte de la base de données NoSQL
MongoDB',
  par: {nom:"Manouvrier", prenom:"Maude"},
  nbH : 3,
  tags: ['mongodb', 'document']
}
```

3. Insérer les éléments contenus dans le fichier<sup>2</sup> collectionDocs.json dans la collection de votre choix.
4. Recopier les 2 instructions d'insertion précédentes dans un fichier texte pour les avoir sous la main. En effet, en changeant de pages dans la documentation, la console se réinitialise. Il sera ainsi plus facile de réinsérer les documents.
5. Lister tous les documents de la collection où vous avez inséré les documents précédents (cf. <https://www.mongodb.com/docs/mongodb-shell/crud/read/#std-label-mongosh-read> ).
6. Compter les éléments de la collection  
(cf. <https://docs.mongodb.com/manual/reference/command/count/>)
7. Trouver tous les documents dont l'auteur est Manouvrier (cf. <http://www.w3big.com/fr/mongodb/mongodb-query.html>)
8. N'afficher que les titres des précédents documents.
9. Trouver tous les documents associés aux mots-clés mongodb.
10. Trouver tous les documents dont l'auteur est Manouvrier et associés aux mots-clés mongodb.
11. Trouver tous les documents dont le nombre de mots-clés est égale à 2 (cf. <https://www.mongodb.com/docs/manual/tutorial/query-arrays/> ).
12. Trouver les titres et le nombres d'heures (nbH) des documents ayant un nombre d'heures compris entre 2 et 3. (utiliser \$and puis une autre requête avec \$in).
13. Trouver les titres des documents n'ayant pas d'url.  
(cf <https://docs.mongodb.com/manual/reference/operator/query/exists/> )
14. Trouver les titres et les auteurs des documents n'ayant pas d'url ou de likes, triés par l'ordre alphabétique de titres.  
(cf. <https://docs.mongodb.com/manual/reference/method/cursor.sort/>)
15. Inverser l'ordre de tri de la requêtes précédentes.
16. Trouver les titres des documents n'ayant pas d'url et de likes.
17. Quelles est la liste distincte des mots-clés ?  
(cf. <https://docs.mongodb.com/manual/reference/command/distinct/>)
18. Trouver les titres des documents dont le titre contient le mot 'TP'.  
(cf. <https://docs.mongodb.com/manual/reference/operator/query/text/>)

---

<sup>2</sup> Disponible sur *moodle*.

19. Trouver les titres des documents dont le titre contient le mot 'MongoDB' mais pas le mot 'TP'.
20. Trouver les titres des documents dont la description contient le termes NoSQL en un seul mot ou plusieurs (avec ou sans majuscules).  
(cf. <https://docs.mongodb.com/manual/reference/operator/query/regex/>)
21. Compter combien il y a de tutorial par auteurs.  
(cf. [https://www.tutorialspoint.com/mongodb/mongodb\\_aggregation.htm](https://www.tutorialspoint.com/mongodb/mongodb_aggregation.htm))
22. Analyser ce que fait la commande suivante (*appuyer sur FULL pour avoir la fenêtre du tutoriel en grand sinon le résultat de la requête n'est pas affiché complètement*) :  

```
db.inventory.aggregate( [ { $unwind : "$tags" } ] )
```
23. Compter combien il a de documents par mots-clés.  
(cf. <https://docs.mongodb.com/manual/reference/operator/aggregation/unwind/#examples>)
24. Compter combien il a de documents par mots-clés, uniquement pour les documents dont le nombre d'heures (nbH) est supérieur ou égale à 2 (cf. <https://docs.mongodb.com/manual/reference/operator/aggregation/match/>).
25. Ajouter un champ likes valant 100 aux documents contenant le mot TP dans le titre et vérifier votre mise à jour (cf. <https://docs.mongodb.com/manual/tutorial/update-documents/>)

## **2. Création de schémas et validation de documents**

26. La commande suivante<sup>3</sup> permet de créer une nouvelle collection nommée doc avec l'option validator de \$jsonSchema permettant de stocker des documents avec un élément obligatoire author de types string.

```
db.createCollection("doc", { validator: { $jsonSchema: {  
  bsonType: "object", required: [ "author"], properties: {  
    author: { bsonType: "string", description: "must be an array  
of string and is required"}}}}})
```

(cf <https://docs.mongodb.com/manual/core/schema-validation/>)

27. Taper l'instruction show collections
28. Taper l'instruction : `db.getCollectionInfos( { name: " doc " } )` pour visualiser le schéma de la collection nouvellement créée.
29. Exécuter chacune des insertions suivante et analyser le résultat :

```
db.doc.insertOne({"title": "TP MongoDB", "author": "MaudeM"})  
db.doc.insertOne({"title": "TP Redis"})  
db.doc.insertOne({"author": "Maude Manouvrier"})
```

```
db.doc.insertOne({  
  "@mdate": "2017-05-28",  
  "@key": "journals/acta/VoglerS014",  
  "author": [  
    "Walter Vogler",  
    "Christian Stahl",  
    "Richard Muuml;ller 0001"  
  ],  
  },
```

---

<sup>3</sup> Que vous pouvez copier depuis MyCourse.

```

        "title": "Trace- and failure-based semantics for responsiveness.",
        "pages": "499-552",
        "year": "2014",
        "volume": "51",
        "journal": "Acta Inf.",
        "number": "8",
        "ee": "https://doi.org/10.1007/s00236-014-0205-y",
        "url": "db/journals/acta/acta51.html#VoglerS014"
    }
)

db.doc.insertOne({
    "@mdate": "2019-05-28",
    "@key": "journals/acta/Darlington78",
    "author": "John Darlington",
    "title": "A Synthesis of Several Sorting Algorithms.",
    "pages": "1-30",
    "year": "1978",
    "volume": "11",
    "journal": "Acta Inf.",
    "url": "db/journals/acta/acta11.html#Darlington78",
    "ee": "https://doi.org/10.1007/BF00264597"
})

```

30. Créer une nouvelle collection nommée `doc2`, ayant les mêmes éléments de *valid* mais avec l'option `validationAction: "warn"`, en utilisant la commande :

```

db.createCollection("doc2", { validator: { $jsonSchema: {
bsonType: "object", required: [ "author"], properties: {
author: { bsonType: "string", description: "must be an array
of string and is required" }}}}, validationAction:"warn"})

```

31. Répéter les insertions précédentes en remplaçant `doc` par `doc2` et analyser les résultats.

32. Créer un schéma en utilisant l'instruction suivante :

```

let mySchema = {bsonType: "object", required: [ "author"],
properties: {author: { bsonType: "string", description:
"must be an array of string and is required" }}}

```

33. Lister les éléments de la collection `doc2` vérifiant ce schéma en utilisant l'instruction :  
`db.doc2.find( { $jsonSchema: mySchema } )`

34. Analyser les règles de validation de la collection `contacts` suivante :

```

db.createCollection( "contacts",
  { validator: { $or:
    [
      { phone: { $type: "string" } },
      { email: { $regex: /@mongodb\.com$/ } },
      { status: { $in: [ "Unknown", "Incomplete" ] } }
    ]
  }
} )

```

35. Insérer un document dans la collection `contacts` satisfaisant toutes les règles.

36. Exécuter l'instruction : `db.contacts.insertOne({status:'Incomplete'})`

37. Créer la requête d'insertion d'un document dans la collection `contacts` ne satisfaisant aucune les règles document.

38. Mettre à jour le `status` des documents de `status Unknown` à `Incomplete`.

39. Répéter les mêmes opérations (35 à 41) en modifiant le \$or par un \$and (vous pouvez réinitialiser la console).

## II. Manipuler des documents JSON sous PostgreSQL

1. Lancer <https://www.db-fiddle.com/>
2. Copier le script SQL<sup>4</sup> suivant pour créer une relation contenant des documents JSON :

```
CREATE TABLE docs (
    id serial NOT NULL PRIMARY KEY,
    contenu json NOT NULL
);
```

```
INSERT INTO docs(contenu) VALUES ('{"titre": "TP MongoDB", "nbH" :
3, "description": "Découverte de la base de données NoSQL
MongoDB", "par": {"nom": "Manouvrier", "prenom": "Maude"}, "tags":
["mongodb", "document"]}');
```

```
INSERT INTO docs(contenu) VALUES ('{"titre": "Découverte de
MongoDB", "description": "Installation et découverte de la base de
données MongoDB", "par": "Projets DIY", "nbH" :
1, "url": "https://www.projetsdiy.fr", "tags": ["mongodb", "document",
"NoSQL"]}'), ('{"titre": "MongoDB Overview", "description": "MongoDB
is no sql database", "par": "tutorials
point", "url": "http://www.tutorialspoint.com", "tags": ["mongodb",
"document", "NoSQL"], "nbH" : 2, "likes": 100, "comments": [{
"user": "user1", "message": "My first comment", "dateCreated": "2023-11-
07T08:53:22 +08:00", "like": 0}]}'), ('{"titre": "TP Redis",
"description": "Découverte de la base de données NoSQL Redis", "par":
{"nom": "Manouvrier", "prenom": "Maude"}, "tags": ["redis", "key-value",
"NoSQL"]}');
```

3. Ecrivez les requêtes suivantes en SQL : cf. documentation

<https://www.postgresql.org/docs/current/datatype-json.html> et  
[https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-  
json/](https://www.postgresqltutorial.com/postgresql-tutorial/postgresql-json/)

- a. Lister tous les documents de la table.
- b. Compter les documents de la table
- c. Trouver tous les documents dont l'auteur est Manouvrier.
- d. N'afficher que les titres des précédents documents.
- e. Trouver tous les documents associés aux mots-clés mongodb (vous pouvez utiliser json\_array\_elements<sup>5</sup> et CAST).
- f. Trouver tous les documents dont l'auteur est Manouvrier et associés aux mots-clés mongodb.
- g. Trouver tous les documents dont le nombre de mots-clés est égale à 2 (vous pouvez utiliser json\_array\_length).
- h. Trouver les titres et le nombres d'heures (nbH) des documents ayant un nombre d'heures compris entre 2 et 3 (vous pouvez utiliser CAST).

---

<sup>4</sup> Script disponible sous moodle.

<sup>5</sup> cf. [https://www.i3s.unice.fr/~rueher/Cours/BD/DocPostgresSQL9-5\\_HTML/fonctions-json.html](https://www.i3s.unice.fr/~rueher/Cours/BD/DocPostgresSQL9-5_HTML/fonctions-json.html)

- i. Trouver les titres des documents n'ayant pas d'url. (Vous pouvez utiliser CAST, EXCEPT et l'opérateur ? applicable au type jsonb).
- j. Ajouter un champ likes valant 100 aux documents contenant le mot TP dans le titre et vérifier votre mise à jour.

### III. Liens où trouver des exemples de requêtes

Vous avez dans les liens suivants d'autres exemples de requêtes :

- [https://www.tutorialspoint.com/mongodb/mongodb\\_aggregation.htm](https://www.tutorialspoint.com/mongodb/mongodb_aggregation.htm)
- <https://docs.mongodb.com/manual/tutorial/query-documents/>
- [https://rtavenar.github.io/mongo\\_book/content/01\\_find.html](https://rtavenar.github.io/mongo_book/content/01_find.html)
- <https://geekflare.com/fr/mongodb-queries-examples/>
- <https://fr.blog.businessdecision.com/tutoriel-mongodb-requetes/>

### IV. Utilisation de MongoAtlas

MongoDB Atlas est service cloud de base de données MongoDB disponible à l'adresse : <https://www.mongodb.com/cloud/atlas>

Vous pouvez utiliser une partie des fonctionnalités de ce service en cliquant sur **Start Free**. Une fois vos coordonnées renseignées, vous pouvez utiliser gratuitement la **version Shared Clusters** et en choisissant par exemple **AWS / Frankfurt (eu-central-1)** comme fournisseur.

Vous devez ensuite vous connecter au cluster créé et utiliser votre adresse IP, puis créer un utilisateur Mongo.