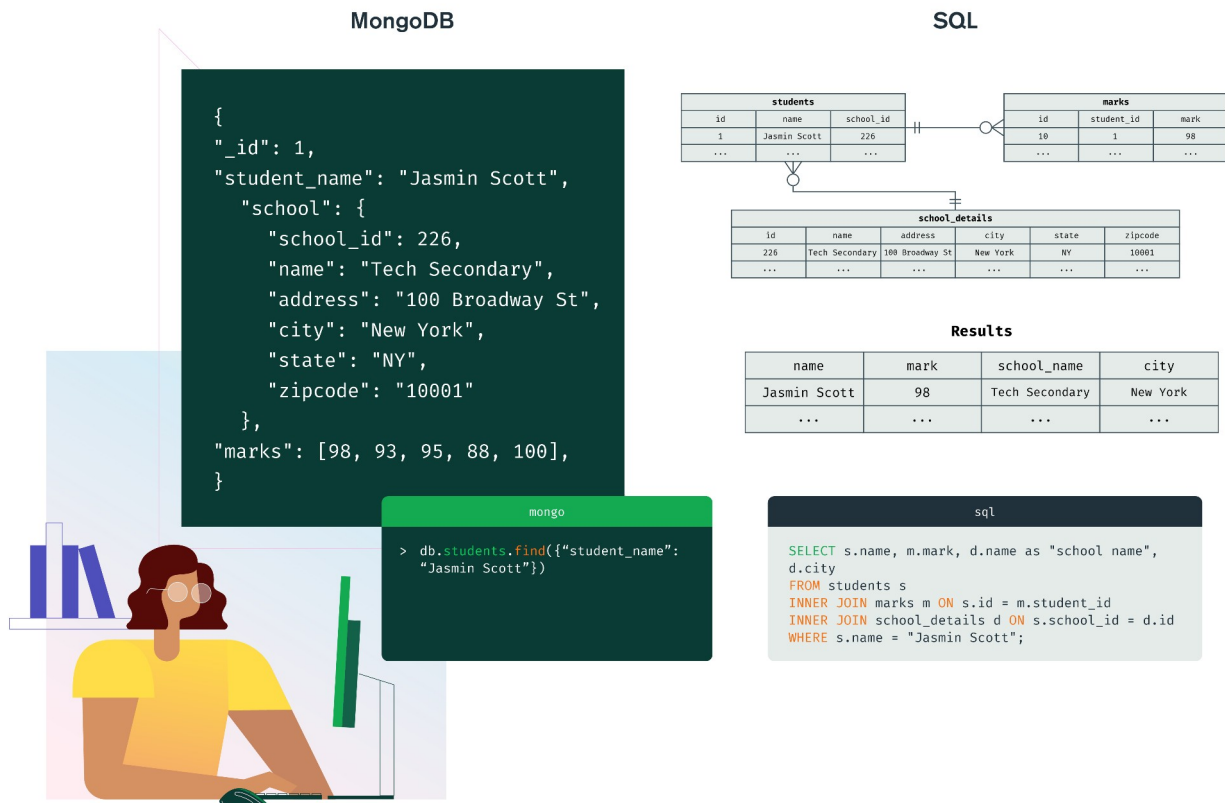


## 1] Procédure d'installation

MongoDB est un système de gestion de base de donnée basé sur des documents (on dit « orientée documents ») mémorisés dans des **classeurs** (ou collections) et non dans des tableaux ; au format **Bson** (ou Json binaire) suivant un ensemble de **paires valeur/clé**.



**En cours** → analyse des différences SQL/MongoDB.

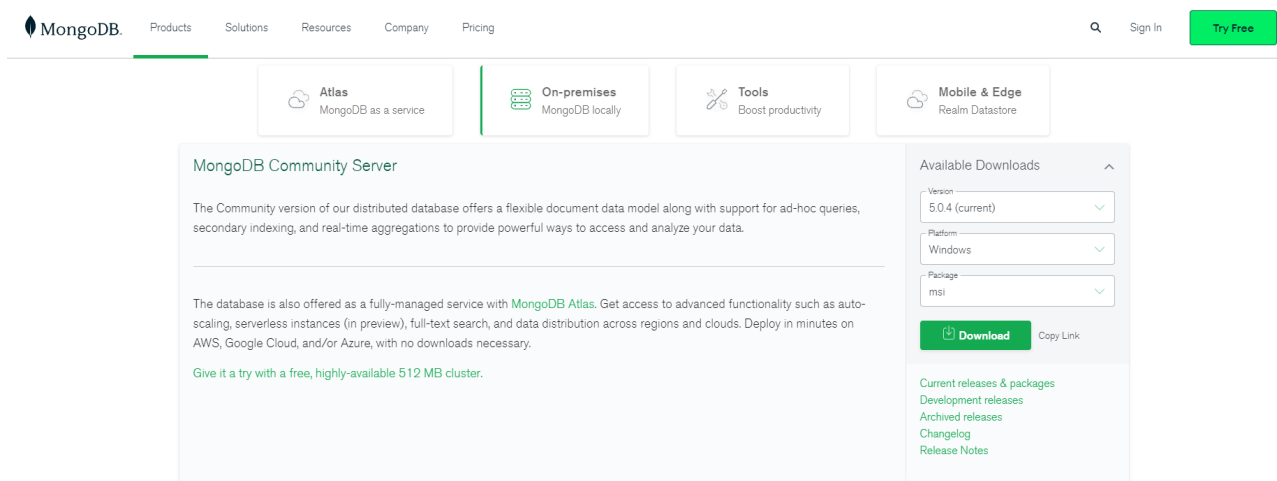
« Le travail orienté documents est prédestiné à de nombreux types de documents, notamment s'il ne suit pas de schéma prédéfini. Ces documents variés doivent être stockés et exploités avec rapidité. » ([www.ionos.fr](http://www.ionos.fr)).

La procédure d'installation générale pour une version « **Community Server** » plus conforme au monde OpenSource se trouve ici :

<https://docs.mongodb.com/manual/installation/#mongodb-community-edition-installation-tutorials>

- **Windows 10**

Les liens pour le téléchargement (**Etape 1** : lien MongoDB Download Center) pour ce système aboutissent à cette page :



La version concernée à la rédaction de ces lignes est la **5.0.4**, l'installation du paquetage msi (mongodb-windows-x86\_64-5.0.4-signed.msi) ne présente pas de difficultés.

### Important :

*Choisissez simplement l'installation **complète**, mais **décochez** la case installant MongoDB en temps que **service** et **décochez** ensuite aussi l'installation de MongoDB **Compass** car nous utiliserons une autre interface graphique pour la gestion de MongoDB.*

Comme MongoDB n'est pas installé en tant que service (pour préserver les ressources de votre système), il est nécessaire d'intégrer les binaires dans les **variables d'environnement du système** :

C:\Program Files\MongoDB\Server\5.0\bin dans (Path système).

**Dernière manipulation** : créez à la racine C : \ trois dossiers : C : \data, C : \data\db et C : \data\log.

- **Linux Ubuntu**

Pour le monde Linux, l'installation de MongoDB est un peu compliquée, cela est dû à des problèmes de licence... Si l'on suit le lien donné plus haut, la procédure pour Linux Ubuntu concerne la distribution **20.04** (Focal) avec pour l'installation de la clé du dépôt :

```
wget -qO - https://www.mongodb.org/static/pgp/server-4.4.asc | sudo apt-key add -
```

Pour la mise en place du dépôt :

```
echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu focal/mongodb-org/4.4 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-4.4.list
```

Il suffit ensuite de faire un **update** des sources suivi de l'installation du paquet mongodb-org.

Et enfin, comme pour Windows, il faut créer les dossiers à la racine (en **root**) : /data et /data/db.

- **Utilisation de RoboMongo**

**Robomongo** est un « front-end » graphique pour l'administration de MongoDB. Passer par une interface graphique est plus aisé mais cela ne vous dispense pas de connaître les principales commandes de MongoDB par la console... Vous trouverez le logiciel ici : <https://robomongo.org>

Notez aussi que Robomongo s'appelle dorénavant **Robo 3T**.

Choisissez le lien **Download Robo 3T Only** et non la version payante Studio 3T.

Pour Windows :

→ Téléchargez la version pour Windows, prenez la version `robo3t-1.4.4-windows-x86_64-e6ac9ec5.exe` et installez le logiciel.

Windows Mac Linux

Robo 3T 1.4.4

Download for Windows

[robo3t-1.4.4-windows-x86\\_64-e6ac9ec5.exe](#)

[robo3t-1.4.4-windows-x86\\_64-e6ac9ec5.zip](#)

Close

Pour Linux :

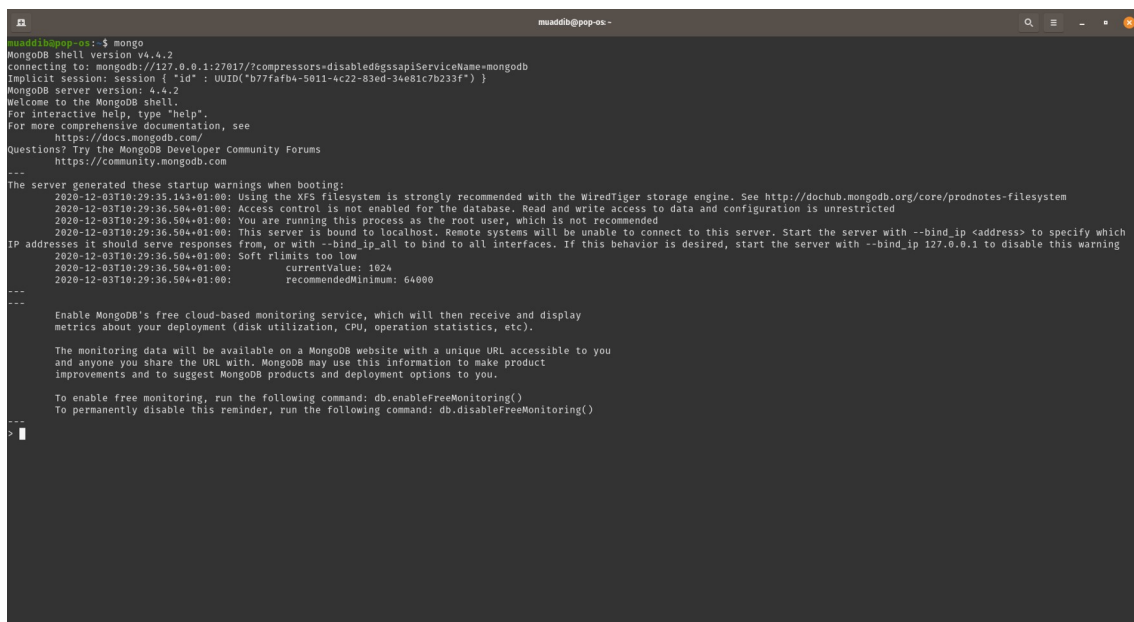
→ Téléchargez la version pour Linux, décompressez l'archive et placez le dossier dans votre répertoire utilisateur.

## 2] Prise en main rapide de MongoDB

Cette prise en main se fera en mode « commande » par une console.

**T2.1** | Lancez le serveur MongoDB par la commande `mongod`, dans un terminal administrateur pour Windows, en root pour Linux ; réduisez mais ne fermez pas le terminal.

**T2.2** | Lancez ensuite une deuxième console, cette fois-ci en tant qu'utilisateur et tapez la commande `mongo` pour lancer le « front end » en mode commandes :



```

muaddib@pop-os: ~$ mongo
MongoDB shell version v4.4.2
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("b77fafb4-5011-4c22-83ed-34e81c7b233f") }
MongoDB server version: 4.4.2
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  https://docs.mongodb.com/
Questions? Try the MongoDB Developer Community Forums
  https://community.mongodb.com
---
The server generated these startup warnings when booting:
2020-12-03T18:29:35.143+01:00: Using the XFS filesystem is strongly recommended with the WiredTiger storage engine. See http://dochub.mongodb.org/core/prodnotes-filesystem
2020-12-03T18:29:36.504+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2020-12-03T18:29:36.504+01:00: You are running this process as the root user, which is not recommended
2020-12-03T18:29:36.504+01:00: This server is bound to localhost. Remote systems will be unable to connect to this server. Start the server with --bind_ip <address> to specify which IP addresses it should serve responses from, or with --bind_ip_all to bind to all interfaces. If this behavior is desired, start the server with --bind_ip 127.0.0.1 to disable this warning
2020-12-03T18:29:36.504+01:00: Soft rlimits too low
2020-12-03T18:29:36.504+01:00:      currentValue: 1024
2020-12-03T18:29:36.504+01:00:      recommendedMinimum: 64000
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
>

```

**Note :** MongoDB respecte la « casse », c'est à dire la différence entre les minuscules et les majuscules. Pour des commandes plus complètes, une simple recherche sur Internet suffira.

### T2.3 | Commande pour voir les bases existantes :

```
> show dbs
```

MongoDB crée « à la volée » les bases de données. Autrement dit, dire qu'on en utilise une **revient à la créer** si elle n'existe pas.

### T2.4 | Commande pour utiliser et donc créer si elle n'existe pas, une base mais effectif quand on y met quelque chose :

```
> use personnages
```

Le retour indique : switched to db personnages.

**Note :** il existe la complétion des commandes, ceci vous aidera parfois...

- **Commandes usuelles**

#### 1) Voir le nom de la base en cours :

```
db.getName()
```

#### 2) Voir les collections de la base :

```
show collections
```

ou

```
db.getCollectionNames()
```

#### 3) Insertion d'un élément (pour un Quizz) :

```
db.personnages.insert({ "nom":"McClane", "prenom":"John", "acteur":"Bruce Willis", "films":["Die Hart","Terminator","Mad Max"], "reponse":1, "genre":"Action"})
```

#### 4) Voir les éléments d'une collection :

```
db.personnages.find()
```

Le retour montre en plus pour chaque élément un `ObjectId`, référence numérique de l'élément dans la collection ("`_id`"), élément réservé MongoDB.

#### 5) Voir un élément en fonction d'un critère :

```
db.personnages.find({"nom":"McClane"})
```

Vous avez aussi les possibilités d'une recherche conditionnelle avec :

```
$lt pour <, $lte pour ≤, $gt pour >, $gte pour ≥  
$ne - '!='  
$in - 'is in array'  
$nin - '! in array'
```

Exemple :

```
db.personnages.find({"reponse":{"$lte:2}})
```

**Note :** il existe d'autres possibilités de construire un critère avec les expressions rationnelles.

6) Modifier un élément :

D'abord on en ajoute un avec une erreur dans l'orthographe de l'actrice :

```
db.personnages.insert({ "nom":"Galadriel", "prenom":"","acteur":"Kate Blanchet", "films":
["Le Seigneur des Anneaux","Heaven","Pulp Fiction"], "reponse":1, "genre":"Fantastique"})
```

La modification :

```
db.personnages.update({ "nom":"Galadriel" }, {'$set':{'acteur':"Cate Blanchett"}})
```

**Note :** en cas d'erreur, l'indication de la colonne est donnée, il suffit de compter...

7) Supprimer un élément :

```
db.personnages.remove({ "nom":"Galadriel" })
```

8) Vider une collection :

```
db.personnages.remove({})
```

7) Supprimer une collection (après un use):

```
db.dropDatabase()
```

- **Utilisation de Robo3T**

→ Lancez le logiciel, une fenêtre s'ouvre, cliquez sur **Create** et validez simplement la connexion car elle se base sur un système en local ce qui est notre cas (vous pouvez changer le nom) ; ensuite sélectionnez la ligne et appuyez sur le bouton **Connect**

