

MongoDB

Dr. Ali SKAF



MongoDB

Lancer la base de données:

Avant de pouvoir exécuter la moindre commande, il faut que la base de données soit lancée, cela se passe avec **mongod** (le d de dæmon) et se connecter au shell mongoDB, cela se passe avec **mongo** (le client).

- on lance la commande mongod (serveur)
- on lance la commande mongo (client) et on pourra saisir toutes nos commandes.

```
Microsoft Windows [version 10.0.19043.1526]
(c) Microsoft Corporation. Tous droits réservés.

C:\Users\ASUS>mongo
MongoDB shell version v5.0.6
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7d2a72b5-2f95-48ef-afb3-d175404f9930") }
MongoDB server version: 5.0.6
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
---
The server generated these startup warnings when booting:
  2022-02-25T14:17:33.043+01:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
---
---
  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()
---
```

MongoDB

Commandes essentielles :

afficher toutes les base de données

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
test     0.000GB
```

MongoDB

Commandes essentielles :

Créer ou utiliser la base de données « mydb »

```
> use mydb  
switched to db mydb
```

MongoDB

Commandes essentielles :

voir dans quelle base on est

```
> db  
mydb
```

MongoDB

Commandes essentielles :

insérer des données dans la collection « persons »

```
> db.persons.insert({"nom":"Thomas"})
WriteResult({ "nInserted" : 1 })
```

MongoDB

Commandes essentielles :

afficher les collections

```
> show collections
persons
```

MongoDB

Commandes essentielles :

afficher toutes les base de données

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
mydb     0.000GB
test     0.000GB
```


MongoDB

Commandes essentielles :

Vérifier si la base « mydb » a été créée

```
> show dbs
admin    0.000GB
config  0.000GB
local    0.000GB
mydb     0.000GB
test     0.000GB
```

MongoDB

Commandes essentielles :

Afficher les données dans la collection « persons »

```
> db.persons.find()  
{ "_id" : ObjectId("621ba67b19b4d816787f55b6"), "nom" : "Thomas" }
```

→ Ajout automatique d'un identifiant

MongoDB

Commandes essentielles :

Créer un utilisateur

```
> db.createUser({user:"Ali",  
... pwd:"1234",  
... roles:["readWrite","dbAdmin"]  
... });  
Successfully added user: { "user" : "Ali", "roles" : [ "readWrite", "dbAdmin" ]  
}
```

MongoDB

Commandes essentielles :

Afficher les utilisateurs

```
> show users
{
  "_id" : "mydb.Ali",
  "userId" : UUID("f53d6362-e8f9-458f-ae3-c5b49b79afe9"),
  "user" : "Ali",
  "db" : "mydb",
  "roles" : [
    {
      "role" : "readWrite",
      "db" : "mydb"
    },
    {
      "role" : "dbAdmin",
      "db" : "mydb"
    }
  ],
  "mechanisms" : [
    "SCRAM-SHA-1",
    "SCRAM-SHA-256"
  ]
}
```

MongoDB

Commandes essentielles :

Créer une deuxième collection « Customers »

```
> db.createCollection('Customers');  
{ "ok" : 1 }  
> show collections  
Customers  
persons
```

MongoDB

Commandes essentielles :

Insérer des données dans la collection « Customers »

```
> db.customers.insert({first_name:"Jhon",last_name:"Brun"});  
WriteResult({ "nInserted" : 1 })
```

Afficher les données de la collection « Customers »

```
> db.customers.find()  
{ "_id" : ObjectId("621bb77b19b4d816787f55b7"), "first_name" : "Jhon", "last_name" : "Brun" }
```

MongoDB

Commandes essentielles :

Insérer plusieurs documents documents (lignes) dans la collection « Customers »

```
> db.customers.find()
{ "_id" : ObjectId("621bb77b19b4d816787f55b7"), "first_name" : "Jhon", "last_name" : "Brun" }
{ "_id" : ObjectId("621bb8b019b4d816787f55b8"), "first_name" : "Steven", "last_name" : "Smith" }
> db.customers.insert([{first_name:"Aline",last_name:"Hugo"},
...                      {first_name:"George",last_name:"nemet"}]);
BulkWriteResult({
  "writeErrors" : [ ],
  "writeConcernErrors" : [ ],
  "nInserted" : 2,
  "nUpserted" : 0,
  "nMatched" : 0,
  "nModified" : 0,
  "nRemoved" : 0,
  "upserted" : [ ]
})
```

MongoDB

Commandes essentielles :

*# Afficher le contenu de la collection « Customers »
avec la commande pretty()*

```
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Brun"
}
{
  "_id" : ObjectId("621bb8b019b4d816787f55b8"),
  "first_name" : "Steven",
  "last_name" : "Smith"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet"
}
```


MongoDB

Commandes essentielles :

Mettre à jour un champs d'un document

```
> db.customers.update({first_name:"Steven"},{$set:{gender:"male"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.cudomers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb8b019b4d816787f55b8"),
  "first_name" : "Steven",
  "last_name" : "Smith",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet"
}
```

MongoDB

Commandes essentielles :

Trouver un document à l'aide d'un critère de recherche

Exp: First_name : « Aline »

```
> db.customers.find({first_name:"Aline"});  
{ "_id" : ObjectId("621bb95919b4d816787f55b9"), "first_name" : "Aline", "last_name" : "Hugo" }
```

MongoDB

Commandes essentielles :

Trouver un document avec plusieurs critères de recherche

Exp: First_name « Aline » ou First_name « George »

```
> db.customers.find({$or:[{first_name:"Aline"},  
...                       {first_name:"George"}]});  
{ "_id" : ObjectId("621bb95919b4d816787f55b9"), "first_name" : "Aline", "last_name" : "Hugo" }  
{ "_id" : ObjectId("621bb95919b4d816787f55ba"), "first_name" : "George", "last_name" : "nemet" }
```

MongoDB

Commandes essentielles :

Mettre à jour un champs d'un document

Exp: ajouter le champs « Age » pour Aline avec la valeur 45

```
> db.customers.update({first_name:"Aline"},
...                   {$set:{age:45}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet"
}
```

MongoDB

Commandes essentielles :

Mettre à jour un champs d'un document

Exp: ajouter le champs « Age » pour George avec la valeur 25

```
> db.customers.update({first_name:"George"},
...                   {$set:{age:25}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet",
  "age" : 25
}
```

MongoDB

Commandes essentielles :

Chercher les personnes ayant un age <40

```
> db.customers.find({age:{$lt:40}});  
{ "_id" : ObjectId("621bb95919b4d816787f55ba"), "first_name" : "George", "last_name" : "nemet", "age" : 25 }
```

Opérateurs conditionnels :

(>) greater than - \$gt

(<) less than - \$lt

(>=) greater than equal to - \$gte

(<=) less than equal to - \$lte

MongoDB

Commandes essentielles :

Afficher le nombre des clients dans la collection

```
> db.customers.find().count();  
3
```

MongoDB

Commandes essentielles :

Afficher les documents triés par un critère

```
> db.customers.find().sort({last_name:1});  
{ "_id" : ObjectId("621bb77b19b4d816787f55b7"), "first_name" : "Jhon", "last_name" : "Doe", "gender" : "male" }  
{ "_id" : ObjectId("621bb95919b4d816787f55b9"), "first_name" : "Aline", "last_name" : "Hugo", "age" : 45 }  
{ "_id" : ObjectId("621bb95919b4d816787f55ba"), "first_name" : "George", "last_name" : "nemet", "age" : 25 }
```

Sort ({champs: 1}) → Ordre croissant

Sort ({champs: -1}) → Ordre décroissant

MongoDB

Commandes essentielles :

Afficher le nombre des clients « male »

```
> db.customers.find({gender:"male"}).count();  
1
```

MongoDB

Commandes essentielles :

*# Ajouter un champs Membership pour le document d'Aline,
Ce nouveau champs est un tableau : ["mem1", "mem2"]*

```
> db.customers.update({first_name:"Aline"},
...                   {$set:{membership:["mem1","mem2"]}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45,
  "membership" : [
    "mem1",
    "mem2"
  ]
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet",
  "age" : 25
}
```

MongoDB

Commandes essentielles :

Afficher les deux premiers clients de la collection

```
> db.customers.find().limit(2);
{ "_id" : ObjectId("621bb77b19b4d816787f55b7"), "first_name" : "Jhon", "last_name" : "Doe", "gender" : "male" }
{ "_id" : ObjectId("621bb95919b4d816787f55b9"), "first_name" : "Aline", "last_name" : "Hugo", "age" : 45, "membership" : [ "mem1", "mem2" ] }
```

MongoDB

Commandes essentielles :

Supprimer l'âge de George avec la commande \$unset

```
> db.customers.update({first_name:"George"},{$unset:{age:25}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "gender" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45,
  "membership" : [
    "mem1",
    "mem2"
  ]
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet"
}
```

MongoDB

Commandes essentielles :

Resaisir l'âge de George avec la commande \$set

```
> db.customers.update({first_name:"George"},  
...                   {$set:{age:25}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

MongoDB

Commandes essentielles :

Incrémenter l'âge de George de 5 avec la commande \$inc

```
> db.customers.update({first_name:"George"},
...                   {$inc:{age:5}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "sex" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45,
  "membership" : [
    "mem1",
    "mem2"
  ]
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet",
  "age" : 30
}
```

MongoDB

Commandes essentielles :

Renommer un champs

```
> db.customers.update({first_name:"Jhon"},
...                   {$rename:{"gender":"sex"}});
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.custmers.find().pretty();
> db.customers.find().pretty();
{
  "_id" : ObjectId("621bb77b19b4d816787f55b7"),
  "first_name" : "Jhon",
  "last_name" : "Doe",
  "sex" : "male"
}
{
  "_id" : ObjectId("621bb95919b4d816787f55b9"),
  "first_name" : "Aline",
  "last_name" : "Hugo",
  "age" : 45,
  "membership" : [
    "mem1",
    "mem2"
  ]
}
{
  "_id" : ObjectId("621bb95919b4d816787f55ba"),
  "first_name" : "George",
  "last_name" : "nemet",
  "age" : 25
}
>
```

MongoDB

Commandes essentielles :

recherche par première lettre

```
> db.test.find({first_name:/^A/})
{ "_id" : ObjectId("622a13e40a12c485aa55b96b"), "first_name" : "ALine", "last_name" : "Brun", "time" : ISODate("1995-02-03T00:00:00Z") }
{ "_id" : ObjectId("622a734e33ad4726269dde88"), "first_name" : "Armelle", "last_name" : "Bonin", "time" : ISODate("1994-03-03T00:00:00Z") }
>
```


MongoDB

Commandes essentielles :

ajouter date

```
> db.test.insert({first_name:"Armelle", last_name:"Bonin", time:new Date("1994-03-03")})
WriteResult({ "nInserted" : 1 })
> db.test.find().pretty()
{
  "_id" : ObjectId("622a13e40a12c485aa55b96b"),
  "first_name" : "Aline",
  "last_name" : "Brun",
  "time" : ISODate("1995-02-03T00:00:00Z")
}
{
  "_id" : ObjectId("622a734e33ad4726269dde88"),
  "first_name" : "Armelle",
  "last_name" : "Bonin",
  "time" : ISODate("1994-03-03T00:00:00Z")
}
```