



Introduction à MongoDB

Support e-learning

Programme détaillé ou sommaire

Introduction

Installation

MongoDB Compass

La base de test

Création d'une base

Création d'une collection

Ajout d'un document à une collection

La commande find

Chapitre 1

Introduction

Chapitre 2

Installation

MongoDB Compass

- ❑ Installation avec le MSI.
- ❑ Interface visuelle pour dialoguer avec le serveur de base de données MongoDB

MongoDB Tools

- ❑ Permet d'installer les outils comme mongoimport qui permettent d'initialiser une collection
- ❑ Attention : pour que ces outils fonctionnent il est nécessaire de modifier la variable path sur windows.

Mongo shell

- ❑ Le MongoShell peut être démarré directement depuis MongoDB Compass
- ❑ On peut aussi le démarrer depuis cmd ou git bash en tapant mongosh
- ❑ Attention : sur le elearning le formateur utilise mongo au lieu de mongosh pour lancer MongoShell.
- ❑ Ensuite à partir du mongoshell on peut interagir avec la base de données.

Chapitre 3

MongoDB Compass

Chapitre 4

La base de test

Introduction

- ❑ Le MongoShell se lance par défaut sur une base nommée test
- ❑ Pour utiliser une autre base : use mabase

Chapitre 5

Création d'une base de données

Chapitre 6

Création d'une collection

Chapitre 7

Création d'un document

Pour créer un document

- ❑ use test : par défaut on est sur cette base de données
- ❑ `doc = { "title" : "Tacos", "cook_time": 20 }`
- ❑ `db.tacos.insertOne(doc)`
 - Création d'une collection appelée tacos et ajout du document à cette collection.
- ❑ `db.tacos.find()`

Chapitre 8

La commande find

Find

- ❑ use recipes
- ❑ db.recipes.find()
- ❑ db.recipes.find().pretty()
- ❑ db.recipes.find({"title":"Tacos"})
- ❑ db.recipes.find({"title":"Tacos", "cook_time":20}) => AND
- ❑ db.recipes.find({"title":"Tacos"})
- ❑ db.recipes.find({"title":"Tacos"}, {"title": 1})
 - Condition: titre=Tacos
 - Données à extraire : title:1 => on extrait que le titre, 1 signifie true
- ❑ db.recipes.find({"title":"Tacos"}, {"title": 0})
 - Condition: titre=Tacos
 - Données à extraire : title:0 => on extrait toutes les données sauf le titre, 0 signifie false

Find

❑ `db.recipes.find({}, {"title":1})`

- Condition: aucune
- Données à extraire : title:1 => on extrait que le titre, 1 signifie true

❑ `db.recipes.find({"title": { $regex: /taco/i}}, {"title":1})`

- Condition: le titre contient taco et /i signifie non sensible à la casse

Limitation et tri

- ❑ `db.recipes.find().count()`
 - Comptage

- ❑ `db.recipes.find().limit(2)`
 - Affichage des 2 premiers documents de la collection recipes

- ❑ `db.recipes.find().skip(2)`
 - Ignore les 2 premiers documents de la collection recipes

- ❑ `db.recipes.find({}, {"title":1}).sort({"title":1})`
 - Affichage des titres de tous les documents dans l'ordre alphabétique ascendant

- ❑ `db.recipes.find({}, {"title":1}).sort({"title":-1})`
 - Affichage des titres de tous les documents dans l'ordre alphabétique descendant

Opérateurs et tableaux

- ❑ `$gt : >`
- ❑ `$gte : >=`
- ❑ `$lt : <`
- ❑ `$lte : <=`
- ❑ `db.recipes.find({"cook_time": { $lte: 30}}, {"title": 1})`
- ❑ `db.recipes.find({$or : [{"cook_time": { $lte: 30}}, {"prep_time": { $lte: 30}}]}, {"title": 1})`
- ❑ `db.recipes.find({"tags": "easy"}, {"title": 1, "tags": 1})`
 - La recherche pour une propriété de type tableau (ici tags) se fait de la même manière que pour une propriété normale.
- ❑ `db.recipes.find({"tags": {$all :["easy", "mexican"]} }, {"title": 1, "tags": 1})`
 - Permet de rechercher les recettes qui ont à la fois le tag easy et le tag mexican.
- ❑ `db.recipes.find({"tags": {$in :["easy", "mexican"]} }, {"title": 1, "tags": 1})`
 - Permet de rechercher les recettes qui ont le tag easy **OU** le tag mexican.

Opérateurs et tableaux

- ❑ `db.recipes.find({"ingredients.name": "egg"}, {"title": 1})`
 - Recherche des recettes avec la propriété name de ingrédients qui vaut egg.
- ❑ `db.recipes.find({"ingredients" : { "name": "egg"}}, {"title": 1})`
 - Attention: ne fonctionne que si on trouve un ingrédient qui n'a que le name comme propriété et rien d'autre. Si les ingrédients ont une autre propriété comme qty on aura aucun match.

Modification : set, unset, inc

❑ `db.examples.updateOne({"title":"Pizza"}, {$set: {"title":"thin crust pizza"}})`

- L'opérateur \$set permet de modifier la propriété title
- Si la propriété n'existe pas elle est ajoutée au document

❑ `db.examples.updateOne({"title":"Pizza"}, {$unset: {"vegan": 1}})`

- L'opérateur \$unset permet de supprimer la propriété vegan
- Ici la valeur 1 signifie true

❑ `db.examples.updateOne({"title":"Tacos"}, {$inc: {"likes_count": 1}})`

- L'opérateur \$unset permet de supprimer la propriété vegan
- Ici la valeur 1 correspond à la valeur de l'incrément
- Si la valeur est négative, cela décrémente.

Modification : tableaux

- ❑ `db.examples.updateOne({"title":"Tacos"}, {$push: {"likes": 60}})`
 - Le \$push permet d'ajouter 60 à la propriété de type tableau nommée likes

- ❑ `db.examples.updateOne({"title":"Tacos"}, {$pull: {"likes": 60}})`
 - Le \$pull permet de supprimer la valeur 60 de la propriété de type tableau nommée likes

Suppression d'un document

- ❑ `db.examples.deleteOne({"title":"Delete me"})`
 - L'opérateur \$set permet de modifier la propriété title
 - Si la propriété n'existe pas elle est ajoutée au document

Chapitre 9

Modélisation

Bonnes pratiques

- ❑ Les données qui sont extraites ensemble doivent être stockées ensemble
- ❑ Le stockage dépend de la manière dont les données vont être exploitées.

Index

- ❑ Comme pour les bases SQL, ils permettent d'accélérer l'extraction de données sur une propriété donnée
- ❑ Evite un full scan
- ❑ `db.recipes.createIndex({"title":1})`
- ❑ `db.recipes.createIndex({"title":-1})`
- ❑ `db.recipes.getIndexes()`
- ❑ `db.recipes.dropIndex("title_1")`

Les collections cappées

- ❑ Permet de limiter le nb de documents dans une collection
- ❑ `db.createCollection("error_log", {"capped": true, "size": 10000, "max": 10000});`
- ❑ Size: taille max en octets (max : nombre de doc max)
- ❑ Une fois cappée, une collection vire les plus anciennes : first in, first out.
- ❑ Très utile pour les logs applicatives par exemple

E-learning

OBJECTIFS :

Comprendre les concepts utilisés par MongoDB

Savoir insérer, modifier, supprimer et lire des données

DESCRIPTION :

- Cf. fichier « Journée MongoDB en elearning.pdf »