**MongoDB : Manipulation de bases de données**

1. **Introduction**

MongoDB fourni un stockage des données dans des documents, sous forme de clés associés à des valeurs. Ainsi, une clé peut avoir comme valeur, une paire de clé et de valeur.

Un document a une structure semblable à celle-ci-dessous :

{

sigle : "SMB214"   
 nom : "Réseaux, Système embarqué et système répartis"  
 nbcredits : 6

}

MongoDB stocke tous les documents dans des collections. Une collection est un ensemble de documents connexes qui partagent en commun un ensemble d'indexes. Les collections sont analogues à une table dans les bases de données relationnelles.

1. **Opérations sur la base de données**
   1. **Opération de lecture**

Les opérations de lecture, ou les requêtes, récupèrent les données existantes dans la base de données. Dans MongoDB, une requête sélectionne des documents d’après une seule collection. Une requête spécifie un certain critère ou des conditions afin d’identifier quels documents MongoDB va retourner pour le client. Une requête peut contenir une projection qui spécifie les attributs qu’on va faire retourner.

Pour rechercher des données, on utilise la méthode : db.collection.find().

Cette méthode accepte à la fois des critères et des projections et on peut encore spécifier l’ordre et nombre d’enregistrements à retourner.

Supposons qu’on a la requête suivante :

db.students.find (

{ age : {$gt 18 } },

{nom : 1, adresse : 1 } )

. limit(5)

)

db.students.find 🡪 spécifie la collection d’après laquelle on va faire notre requête.  
{age : {$gt18}} 🡪 spécifie la critère, qui est dans notre exemple, l’âge de l’étudiant qui doit être plus grande que 18 ans.  
{nom : 1, adresse : 1} 🡪 c’est la projection, qui spécifie que la requête va retourner les valeurs des attributs nom et adresse.  
limit(5) 🡪 spécifie que le nombre d’enregistrements qu’on veut c’est 5 parmi les documents qui satisfaisons la critère.  
Par default, la requête va retourner l’attribut \_id avec les autres attributs spécifiés.

On peut utiliser la méthode db.collection.findOne() pour retourner seulement le premier enregistrement.

Pour exclure un attribut de la réponse d’une requête, il suffit d’écrire le nom de l’attribut dans la section de projection, suivi de : 0. Par exemple, pour exclure \_id on écrit : \_id : 0.

* 1. **Opération d’écriture**

Une opération d’écriture, c’est toute opération qui crée ou modifie des données dans une instance. Dans MongoDB, une opération d’écriture est saisie sur une seule collection.

Les opérations d’écriture sont : insert, update et remove.  
L’opération « Insert » ajoute des données sur une collection.  
L’opération « Update » modifie des données existantes.  
L’opération « Remove » supprime des données dans une collection.  
Pour les opérations « Update » et « Remove » on peut spécifier des critères qui identifient les documents à modifier ou supprimer.

* + 1. **Création**

Les opérations de création ajoutent de nouveaux documents dans une collection. Dans MongoDB, la méthode db.collection.insert() nous permet de faire des opérations de création.

Supposons qu’on a la requête suivante :

db.course.insert {

sigle : "SMB214",

nom : "Réseaux, Système embarqué et système répartis",

nbcredits : 6

}

Cette requête va ajouter un nouveau document à la collection « course ». Ce document va avoir comme attributs « sigle » avec une valeur « SMB214 », « nom » avec une valeur « Réseaux, Système embarqué et système répartis » et « nbcredits » avec une valeur « 6 ». Par default , l’attribut \_id va se généré pour ce document avec une valeur unique.

* + 1. **Modification**

Pour réactualiser les valeurs des documents existants dans une collection on utilise la méthode « db.collection.update() ».

Cette méthode peut accepter des critères pour déterminer quels documents on a envie de réactualiser. Ainsi, on peut ajouter l’option « multi : true » pour pouvoir modifier plusieurs documents.

Supposons qu’on a la requête suivante :

db.students.update(

{ age : { $gt : 18}},

{ $set : { statut : "A"}},

{ multi : true}

)

db.students.update 🡪 spécifie la collection sur laquelle on va faire la modification.  
{age : {$gt : 18}} 🡪 spécifie la critère sur laquelle on va faire la modification. Dans cette requête on va mettre à jour seulement les étudiants qui ont un âge plus grand que 18 ans.  
{$set : { statut : "A"}} 🡪 met à jour le statut des étudiants pour les rendre actives. C’est ce qu’on appelle l’action.  
{multi : true} 🡪 cette option nous permet d’actualiser tous les documents qui répondent à la critère.

* + 1. **Suppression**

En utilisant la méthode « db.collection.remove() » on peut supprimer des documents dans une collection. Cette méthode peut utiliser des critères pour spécifier quels documents on va supprimer.

Supposons qu’on a la requête suivante :

db.collection.remove (

{ statut : " I"}

)

En spécifiant la critère « { statut : " I"} » cette requête va supprimer les documents ayant la valeur de « statut » égale à « I », qui sont donc relatifs à tous les étudiants inactifs.

1. **Indexe**

Les indexes, comme en base de données relationnelles, sont utilisés pour améliorer la performance des requêtes et des réactualisations.

1. **Agrégation**

En plus des requêtes de base, MongoDB offre plusieurs options d’agrégation. Par exemple, MongoDB peut retourner le nombre de documents qui répondent à une requête, ou bien le nombre de valeurs distinctes pour un attribut.