

TP - BD orientées Documents Initiation à MongoDB

Dans la terminologie des BD *NoSQL*, il existe des BD *orientées documents* (ou *document store*) qui permettent de gérer des données semi-structurées. C'est-à-dire que les documents d'une collection n'ont pas tous la même structure. Ils sont donc difficilement mémorisable par des tables relationnelles. Ce type de BD est bien adapté pour des applications Web.

Les données sont toutefois gérées sur le principe du *CRUD*. Chaque document a un identifiant unique.

Dans le cas de MongoDB que nous allons découvrir, le format des documents est JSON¹.

Objectifs

- comprendre les spécificités des BD *orientées documents* ;
- savoir interroger une BD *orientées documents* au format JSON.
- comprendre les différences avec la commande SELECT du langage SQL ;

1 Premier pas avec MongoDB

Note: Lorsqu'une base de données est sélectionnée, MongoDB définit une variable globale "**db**" qui sert de référence vers celle-ci. C'est un objet possédant divers attributs et méthodes, notamment sa liste de collections, qui sont elles-mêmes des objets : **db.collectionName**.

Les collections possèdent des méthodes pour traiter des données, dont *.find()* qui permet de récupérer des documents. C'est cette méthode qui récupère les filtres de requête et de projection. Elle peut prendre jusqu'à 2 arguments et renvoie un curseur (qui n'est ni plus ni moins qu'une sous-collection) contenant les documents filtrés. Ce curseur est donc également un objet avec différentes méthodes permettant de manipuler les documents.

Exercice 1 (Requêtes sur les collections de la BD IUT)

Pour réaliser cet exercice, vous devez d'abord :

- téléchargez le script **iut.js** qui est sur Moodle ;
- ouvrez un terminal et déplacez vous à l'endroit où vous avez sauvegardé ce fichier ;
- puis, en ligne de commande, tapez *mongosh*

¹ *JavaScript Object Notation* (voir wikipedia.org/wiki/JSON)

- exécutez le script par la commande : `load("iut.js") ;`
- affichez le contenu de la BD par : `show collections.`

Normalement, le retour de MongoDB sera "true". Le prompt devrait être maintenant **iut>**. Si ce n'est pas le cas, tapez la commande `use iut.`
 Vous pouvez vérifier la liste des BD sur le serveur mongoDB en tapant la commande `show dbs.`

A) Requêtes sur documents "plats"

1. Donner la liste des enseignants.
2. Quel est le premier enseignant de la collection ?
3. Quel sont les cinq premiers enseignants de la collection ?
4. Quels sont les enseignants du département **INFO** ?
5. Quel est l'enseignant ayant l'identifiant 7 (`_id` équivalent clé primaire) ?
6. Quels sont les enseignants dont le nom commence par L ? Utiliser `$regex` : et pour comparer le début d'une chaîne de caractère utiliser le symbole `^`.

B) Requêtes sur documents intégrés (notion de champs multivalués)

1. Donner la liste des étudiants. Attention, la structure est différente ici du modèle relationnel utilisé sur Postgres en BUT1.
2. Quels sont les noms et prénoms des étudiants en S3 ?
3. Quels sont les noms et prénoms des étudiants en S4 triés selon leur prénom ?
4. Quels sont les noms et prénoms des étudiants en S2 ou S4 ? Trier selon la classe.
5. Quels sont les étudiants en S4 qui ont eu une note inférieure strictement à 6 ?
6. Quels sont les étudiants en S4 qui ont eu une note supérieure ou égale à 8 et inférieure strictement à 10 ?

2 BD Restaurants

Pour réaliser cet exercice, vous devez d'abord mettre en place la base de données à partir du fichier `restaurants.json` comme lors de l'exercice 1.

Cette BD est assez volumineuse donc il est assez difficile de se rappeler toutes les données sauvegardées... Il va falloir au moins connaître la structure de quelques documents dans les collections.

Exercice 2 (Requêtes simples)

Répondez aux questions suivantes :

1. Quels sont les restaurants situés dans Manhattan ? Combien y en a-t-il ?
2. Parmi eux, quels sont ceux qui proposent de la cuisine italienne ? Combien y en a-t-il ?
3. Parmi ces derniers, quels sont ceux qui sont sur la 5^e avenue ? Combien y en a-t-il ?
4. On ne souhaite retourner que le nom, c'est-à-dire faire une projection, comment faut-il s'y prendre ?
5. On cherche parmi ceux-ci ceux qui ont un nom qui comporte le mot pizza ou Pizza.

Exercice 3 (Filtrages)

Regardons les résultats des commissions d'hygiène : `grades.score`.

1. Documentez-vous sur le système de notation des restaurants new-yorkais.
2. Quel est le nom et l'adresse des restaurants de Manhattan qui ont obtenu le grade "A" lors de la dernière inspection ?
3. Parmi ces derniers, quels sont ceux qui ont eu une note (score) inférieure à 10 ?
4. On voudrait affiner un peu et trouver ceux qui n'ont jamais eu de score supérieur à 10.
5. Cherchez les noms des restaurants qui ont le grade "C" mais avec un score inférieur à 40, et ceci lors de la même évaluation.

Exercice 4 (Optionnel)

Proposez 5 questions en langage naturel sur les bases de données *iut* et *restaurant* et donnez la syntaxe Json correspondante.