

TP Noté : Requêtes MongoDB

Ce TP doit être réalisé en équipe de 2 ou 3 élèves, et remis sur Campus. La date limite est indiquée dans l'espace de remise.

Sommaire

1	Sujet.....	1
2	Attention	1
2.1	Correction automatique	1
2.2	Exécution du script	2
2.3	Base de correction.....	2
3	Barème	3
4	Remise	3
4.1	Livrable	3
4.2	Dépôt.....	3

1 Sujet

Il s'agit de répondre aux questions listées dans le script `queries.js` sous forme de commentaires JavaScript. Ces requêtes portent sur la base de données créée et peuplée par le script `database.js`.

Pour faire le TP : (1) téléchargez et exécutez le script `database.js`, (2) téléchargez et complétez le script `queries.js`. Vous pouvez utiliser l'éditeur ou l'outil de votre choix pour compléter le script.

La population de la base fournie en exemple dans `database.js` est volontairement simple. Il se peut que les questions posées dans `queries.js` ne donnent aucune réponse avec cette population. Vous devez donc la compléter pour tester vos requêtes dans le plus de cas possibles.

L'archive `Exemple.zip` vous donne un exemple du fichier que vous devez remettre (répertoire « File to Submit ») en fonction des fichiers ressources qui vous sont donnés (répertoire « Resources »).

Si vous avez une question à propos de ce TP, postez-la svp sur le forum Questions et Réponses. Je ne répondrai pas aux courriels envoyés à mon adresse personnelle. Merci.

2 Attention

2.1 Correction automatique

Votre script sera corrigé de façon automatique. Les réponses qu'il affichera seront comparées, lignes à lignes, avec celles données par le script corrigé exécuté sur la même base de données. Vous devez donc veiller attentivement aux points suivants :

- Répondez précisément aux questions, comme en TP : une seule requête MongoDB par question (le plus souvent), respectant le schéma de sortie indiqué dans la question.

- Si (dans quelques rares cas) vous avez besoin de deux requêtes MongoDB pour répondre à une question, **ne passez pas** par une variable intermédiaire comme ceci :

```
temp = db.myCollection.find( { /* ... */ } ).foo
db.myCollection.find( { bar: temp } )
```

A la place, enchaenez directement les deux appels, comme ceci :

```
db.myCollection.find( { bar: db.myCollection.find( { /* ... */ } ).foo } )
```

- Pour certaines questions, les champs permettant de trouver ou trier les informations demandées ne doivent pas être affichés : c'est normal. Vous pouvez bien entendu afficher ces champs quand vous testez vos requêtes, mais pensez à les retirer avant de remettre votre script.
- Ne supprimez ou ne modifiez aucune des lignes du script fourni, notamment les instructions `print("Query xy")` : elles servent à identifier, dans la sortie, le numéro de question traitée.
- Votre script ne doit pas afficher autre chose (tests, commentaires, etc.) que le résultat des requêtes ; il ne doit pas appeler ou contenir tout ou partie du script `database.sql`.

2.2 Exécution du script

Votre script sera exécuté sur un serveur MongoDB 4.0, avec la commande shell Unix suivante :

```
mongo [paramètres de connexion] < queries.sql > queries.out 2>&1
```

ou, sous le powershell Windows :

```
get-content queries.js | mongo [paramètres de connexion] > queries.out
```

Avant de remettre votre script, vous devez vérifier qu'il s'exécute correctement dans votre environnement avec l'une des deux commandes ci-dessus.

2.3 Base de correction

Une requête ne doit pas faire d'hypothèse sur la population de la base : elle doit retourner un résultat correct quelle que soit cette population. Votre script sera testé sur une base de données ayant une population différente de celle de la base exemple.

De plus, le schéma de la base de correction pourra différer de celui de la base exemple sur les points suivants :

- Certains champs (par ex. *writers*) pourront ne pas exister ou être mis à null dans certains documents de la base de données.
- Les tableaux mono-éléments pourront être exprimés par d'un simple scalaire : `writers: ["Sergio Leone"]` and `writers: "Sergio Leone"` doivent être considérés comme équivalents.

3 Barème

Le barème indicatif est le suivant :

Rubrique	Points
Requêtes	18
Qualité du code	2
<i>Total</i>	<i>20</i>

Toutes les requêtes, simples ou complexes, valent les mêmes points. Pour la « Qualité du code », vos requêtes doivent : (1) être le plus simple possible ; par exemple, n'utilisez pas *aggregate()* si un simple *find()* suffit, (2) être proprement formatées.

4 Remise

4.1 Livrable

Le livrable est constitué du seul script `queries.js` que vous devez renommer `NOM1.Prenom1.NOM2.Prenom2.NOM3.Prenom3.js`, **sans aucun espace**, pour identifier les membres de l'équipe.

4.2 Dépôt

Déposez directement le fichier dans l'espace de remise, **sans constituer** d'archive. Vous pouvez le redéposer autant de fois que vous voulez jusqu'à la date limite, mais **toujours sous le même utilisateur Campus**.