# TD n° 2
# Réplication et partitionnement en NoSQL

## Exercice 1 - Réplication

La ligue régionale de tennis souhaite à présent mettre en place un mécanisme de réplication des données de sa base de clubs et de rencontres inter-clubs.
Pour ce faire, la ligue s'équipe de trois serveurs pour y stocker ses données.

1. Utiliser les possibilités offertes par mongoDB pour permettre une réplication des données sur les trois serveurs.
Un serveur fera office de maître, il permettra de recevoir toutes les requêtes :
- d'insertion et de mise à jour des données,
- de lecture des données.

Les autres serveurs serviront de « sauvegarde » des données.

2. Devant l'accroissement considérable des accès en lecture sur les données de la base, on souhaite à présent permettre les accès en lecture sur n'importe lequel des trois serveurs.
Quels sont les avantages et les inconvénients d'un tel choix ?
Modifier la configuration de votre système pour permettre ces accès en lecture sur les trois serveurs.

3. Simuler la panne du serveur maitre. Que se passe-t-il alors ? Peut-on encore insérer des documents dans la base ?

4. Ajouter un nouveau nœud au système répliqué. Les données sont-elles alors accessibles sur ce nœud ?

5. Simuler des écritures sur le maître en faisant varier le nombre de réplications nécessaires avant l'envoi de l'acquittement. Comparer les performances obtenues.

## Exercice 2 – Partitionnement

Le volume de données reçues et le nombre de requêtes d'accès aux données étant en forte augmentation, la ligue décide de rendre encore plus performant son système en partitionnant les données sur plusieurs serveurs.

1. Configurer le système de telle sorte qu'il y ait :
- Un serveur de configuration,
- Un routeur,
- Deux serveurs de stockage qui recevront chacun un fragment de la collection des clubs.

2. Créer un partitionnement de la collection des clubs, avec pour clé de partitionnement l'identifiant généré par mongodb.

3. Créer un nouveau partitionnement sur le nom des clubs. Insérer un grand nombre de documents (faire une boucle sur l'insertion de documents dans la collection). Consulter le partitionnement réalisé par mongodb.

4. Faire des requêtes sur un club et observer les temps de réponse.

5. Mettre en place un partitionnement par hash. Reproduire la même insertion de documents que la précédente, et comparer les temps de réponses obtenus sur les requêtes d'accès aux résultats d'un club avec ceux obtenus dans la précédente configuration.

6. Quel est le meilleur partitionnement ? Pourquoi ?