**Développement d'application pour le Cloud**

Travaux Pratiques

1 Création d’une application ………………………………………………………………………………………….. 3

1.1 Jeux de données . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.2 Spécification des besoins . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3 Dénormalisation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3.1 Schéma . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.3.2 Transformation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.4 Vues . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3

1.4.1 Utilisateur standard . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.4.2 Analyste/Décisionnaire . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.4.3 Administrateur . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.5 Application . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

1.6 Évaluation du mini-projet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 4

**Table des matières**

**Table des matières**



**Chapitre**

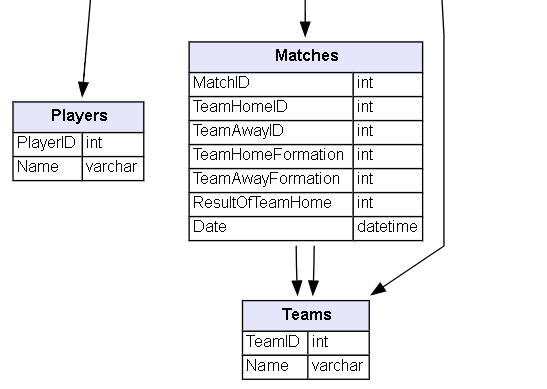
**Table des matières**

Le but de cet exercice est de développer votre propre application avec des données réelles. Vous allez définir les requêtes les plus pertinentes par rapport à ce jeu de données, les intégrer dans une application dédiée, et définir des vues en fonction des rôles.

# 1.1 Jeux de données

Nous avons choisi un jeu de donnée en rapport avec le Football. C’est une base de données contenant des informations sur les match de football de la Premier League anglaise.   
La liste des joueurs, les équipes, les matchs et les actions de chaque joueurs y sont représentés.  
C’est une base de donnée utilisé pour faire de la classification d’une taille de 11.3MB. Elle contient 4 tables, 10716 lignes pour 217 colonnes.





# 1.2 Spécification des besoins

Pour pouvoir orienter les choix des étapes suivantes, il va falloir étudier les besoins de l’application :

— Définir, en langage courant, 4 types d’interrogations sur votre jeu de données. On estimera que celles-ci sont effectuées très fréquemment. Positionnez-vous comme un utilisateur standard de l’application.

Voici 4 types d’interrogations fréquente :

* Nombre de but par joueur ( ou passe décisives par joueur)
* Nombre de cleansheet par gardien ( nombre de match sans se prendre de but)
* Victoire/nul/défaite par équipe sur les n derniers matchs
* Pénalités ( carton jaune/rouge, pénalty concédé)

— Définir, en langage courant, 2 types d’interrogations lourdes sur votre jeu de données. Ces requêtes seront effectuées moins fréquemment. Positionnez-vous comme un analyste ou un décisionnaire de l’application.

Voici 2 types d’interrogation lourde :

* Top n joueurs / top n équipe de la premier league
* Comparaison entre une équipe X ( bilan V/N/D) avec les équipe du top n

— Définir des indicateurs de statistiques pertinents sur votre jeu de données. Positionnez-vous comme administrateur de la base de données.

Utiliser des requetes avec explain !

Bien entendu, il faudra extrapoler l’évolution de la taille de la base de données pour un passage à l’échelle.

# 1.3 Dénormalisation

## 1.3.1 Schéma

**Chapitre 1. Création d'une application**



**Chapitre 1**

**Création d'une application**

Sur ce jeu de données, il va falloir effectuer un choix de dénormalisation pour intégrer les données dans une base de données MongoDB. Pour cela, reposez-vous sur les interrogations produites dans la section précédente pour orienter vos choix.

Vous présentrez le schéma produit, ainsi que l’argumentaire sur les choix de dénormalisation.

## 1.3.2 Transformation

Vous devrez produire une méthode pour effectuer la transformation du schéma relationnel vers la collection de documents JSON que vous utiliserez dans l’application. Plusieurs possibilités d’implémentation :

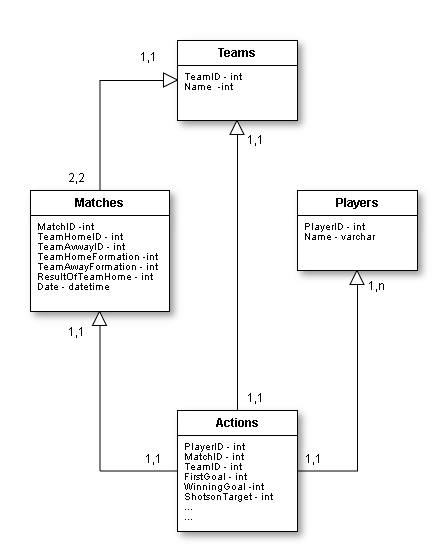
— Parcourir les fichiers séquentiellement et effectuer les jointures nécessaires pour produire les documents ;

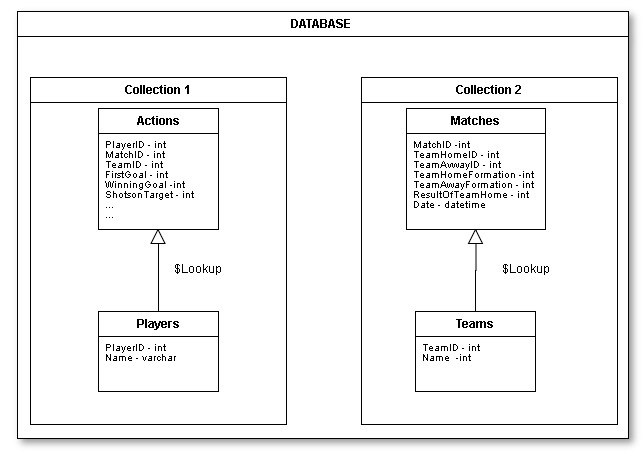
— Stocker les données dans une base de données relationnelle. Les documents destinations seront produits à l’aide de requêtes SQL (et un formattage) ;

— Stocker les données dans MongoDB dans différentes collections. Les documents destinations seront produits à l’aide de requêtes MongoDB en utilisant des $Lookup ;

Vous présentrez la méthode que vous avez choisie et la manière de le réaliser sur votre jeu de données.

Voici la méthode que nous avons choisie afin d’exploiter notre jeu de donnée :  
TODO : Expliquer comment nous avons construit le jeu de donnée





**Principales étapes de modélisation :**

1. Fréquence d'interrogation *(jointure : Action et Player)*

2. Données indépendantes *(jointure : Team et Matches)*

3. Relations 0-n des deux côtés : imbrication pour références

4. Même taux de mises à jour

5. Relation 1-1 des deux côtés

# 1.4 Vues

Grâce aux vues et requêtes définies dans la spécification des besoins, vous allez concevoir des vues pour les différents types d’utilisateur. Vous présenterez les choix d’implémentation effectués pour chaque vue.

## **Chapitre 1. Création d'une application**

**1.5**

**. Application**



### 1.4.1 Utilisateur standard

Fournir à un utilisateur standard un accès aux résultats des 4 requêtes proposées. La vue n’a pas forcément besoin de fournir la requête mais seulement une vision sur les données interrogées.

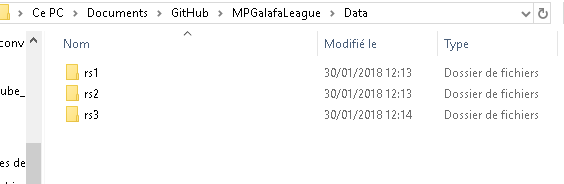
### 1.4.2 Analyste/Décisionnaire

Fournir à votre *Data Analyst* ou votre *Business User* une vue permettant de fournir les indicateurs proposés. L’interface devra permettre de paramétrer les requêtes avec des valeurs (soit à la main, soit par menu déroulant).

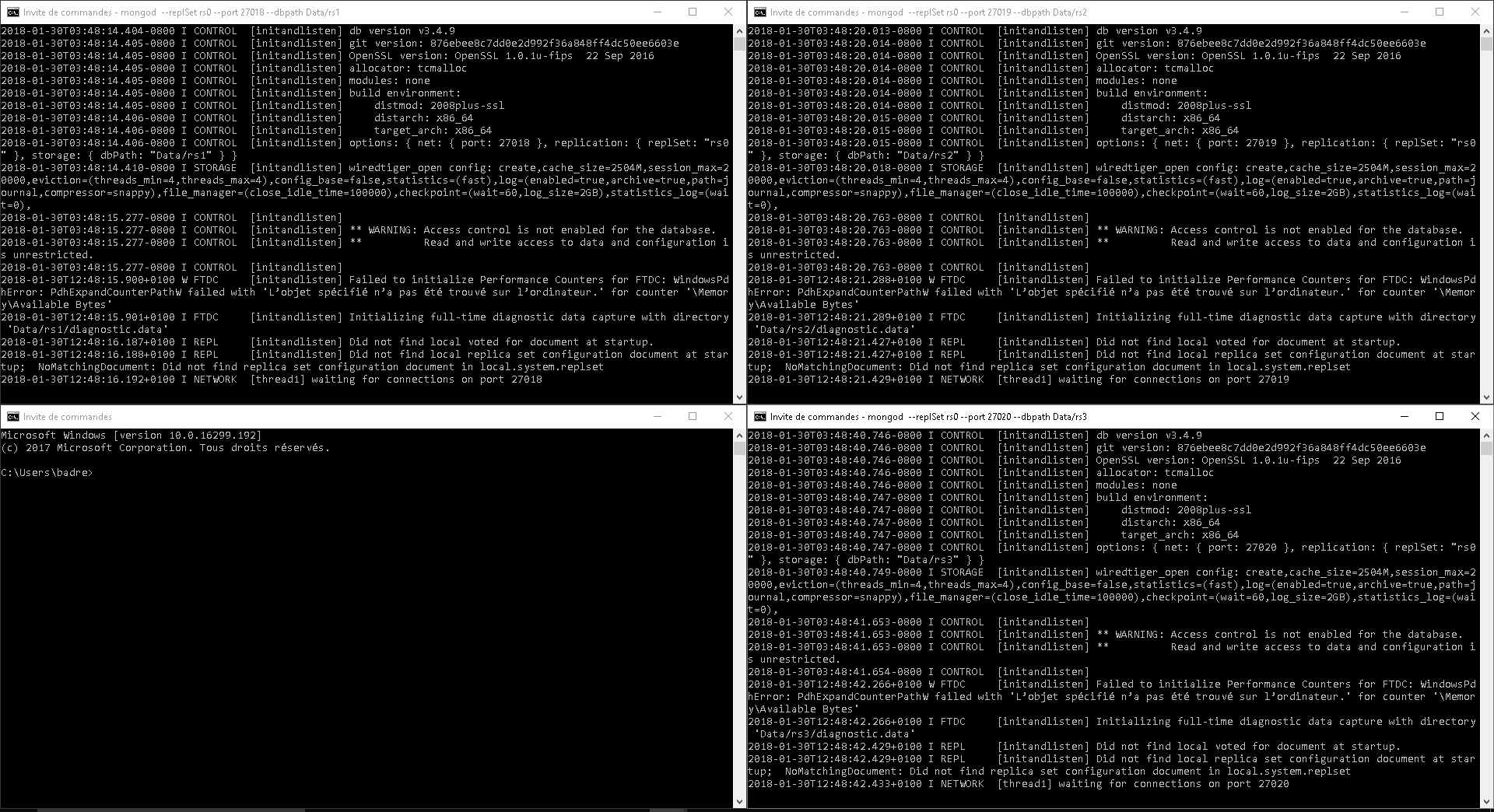
### 1.4.3 Administrateur

L’administrateur de la base MongoDB doit pouvoir récupérer différentes statistiques pour pouvoir faire évoluer le cluster en fonction de la charge. La vue doit pouvoir fournir les informations suivantes :

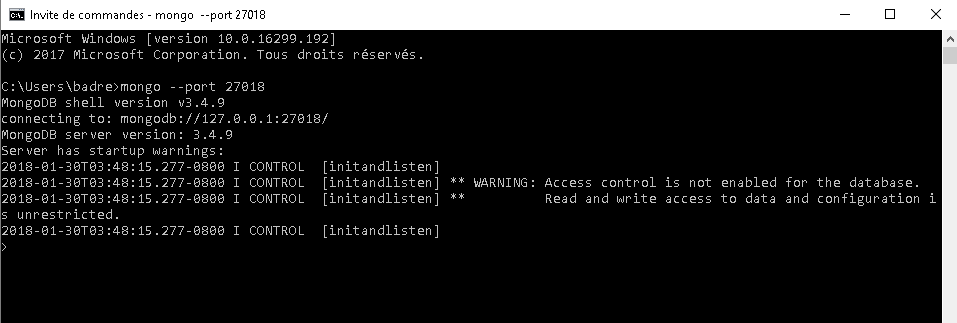
Etape 1 : Création de trois répertoires réplicas set (facteur de réplication = 3)

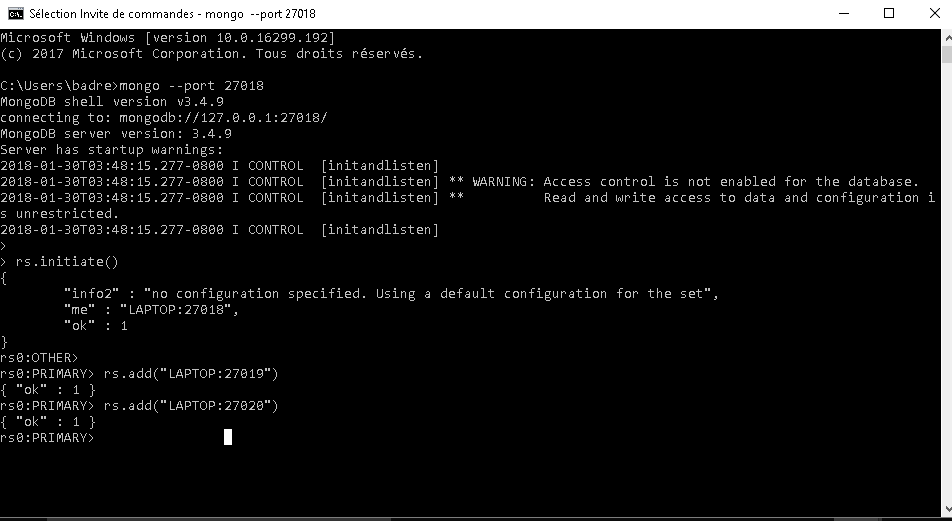


Etape 2 : Ouvrir Mongod dans les trois replicas ( port 27017 , 27018, 27019 et 27020)



Etape 3 : Connection au port 27018 ( premier replica set), initialiser et ajouter avec les autres replicas set





Etape 4 : récuperer le status des réplicas



Etape 5 : Test de la tolerance aux pannes, desactivation du port 27018, observation des status

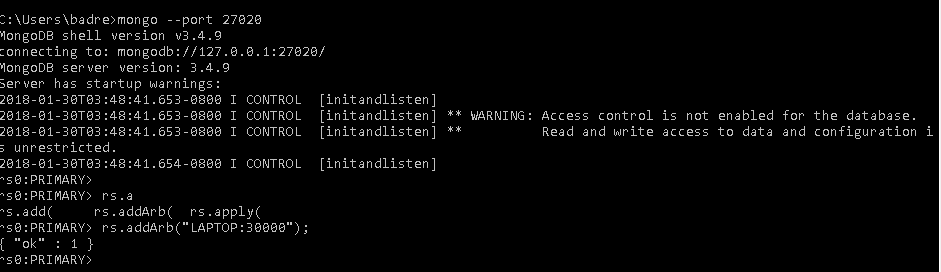


Nous remarquons que le serveur connecté au port 27020 est devenu primary

Etape 6 : tout est ok, réactivons le serveur du port 27018

Nous remarquons qu’il reste secondary car le primary est déjà fixé à 27020

Etape 7 : ajout d’un arbitre

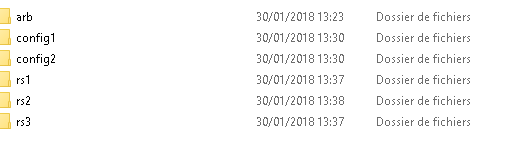




Nos données sont maintenant disponibles en permanence. Notre système est tolérant aux pannes

SHARDING :

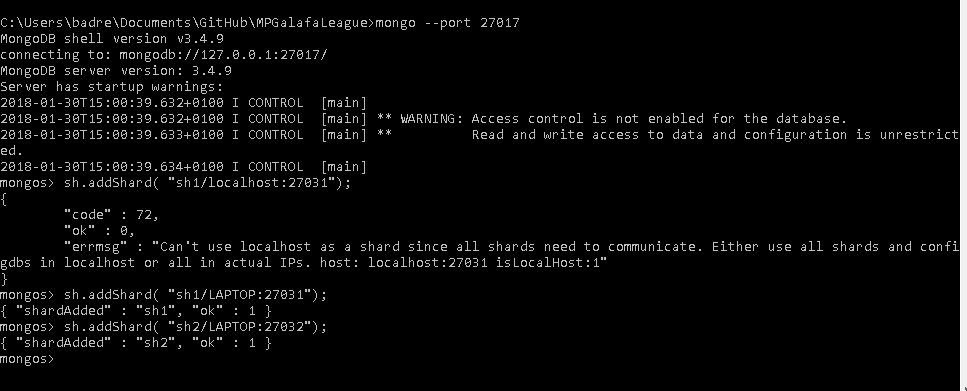
Etape 1 création des répertoires de configuration



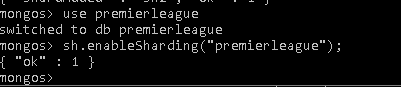
Etape 2

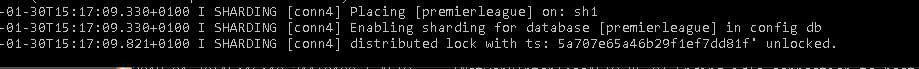
Comment distribuer les données dans les différents serveurs.

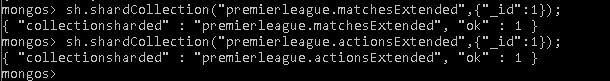
Creation de shard et mise en place d’un serveur de routage



Activation du sharding







Recuperation du status des shards :



— Les statistiques sur les données définies dans la spécification des besoins

— Consulter l’état du cluster : nombre de shards, nombre de réplicats par shard

— Répartition des données sur les shards (nombre de documents)

— Indexes existants sur les données

# 1.5 Application

Vous utiliserez le langage de votre choix. Le code source doit être clair et commenté. Différentes parties doivent être clairement identifiées :

— La connexion à MongoDB ;

— Chaque vue

# 1.6 Évaluation du mini-projet

Pour l’évaluation de ce mini-projet, vous serez noté sur :

— Un rapport détaillant les différentes étapes de cette application

— Détailler la spécification des besoins, la dénormalisation et les choix effectués. Expliquer comment l’exportation au format JSON a été effectué

— Expliquer les différentes vues réalisées et les choix effectués

— Donner les requêtes MongoDB intégrées à chacunes des vues

— Qualité du code source

Vous devrez envoyer une archive contenant :

— Un rapport

— Jeu de données original

— Jeu de données transformé

— Code source de l’application

— README expliquant l’installation