



ÉCOLE  
**D'INGÉNIEURS**  
PARIS-LA DÉFENSE

## Développement d'application pour le Cloud

Travaux Pratiques

**ESILV**

nicolas.travers (at) devinci.fr

<b>1</b>	<b>Création d'une application</b>	<b>3</b>
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 . . . . .	4
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

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

Dans le cadre du développement de cette application, vous devrez choisir votre propre jeu de données. L'idée est de partir d'un jeu de données provenant d'une base de données relationnelle que vous pourrez trouver sur internet, comme celles présentes à cette adresse : <https://relational.fit.cvut.cz/search>. Un schéma à plusieurs tables est nécessaire (minimum 4).

Il est possible également de chercher un dataset sur Google : <https://toolbox.google.com/datasetsearch>. A vous de fournir un schéma pour comprendre le dataset.

### 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.
- Définir, en langage courant, 2 types d'interrogations lourdes sur votre jeu de données (au moins 1 Map/Reduce non exprimable en `aggregate()`). Ces requêtes seront effectuées moins fréquemment. Positionnez-vous comme un analyste ou un décideur de l'application.
- Définir des indicateurs de statistiques pertinents sur votre jeu de données. Positionnez-vous comme administrateur de la base de données.

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

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 présenterez 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 formatage) ;
- Stocker les données dans MongoDB dans différentes collections. Les documents destinations seront produits à l'aide de requêtes MongoDB en utilisant des `$Lookup` ;

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

## 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.

### 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 :

- 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 chacune 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