**Maîtrisez les bases de données NoSQL**

**Au fond, c'est quoi le NoSQL ?**

Depuis les années 70, la base de données relationnelle était l'incontournable référence pour gérer les données d'un système d'information. Toutefois, face aux 3V (Volume, Velocity, Variety), le relationnel peut difficilement lutter contre cette vague de données. Le NoSQL s'est naturellement imposé dans ce contexte en proposant une nouvelle façon de gérer les données, sans reposer sur le paradigme relationnel, d'où le "**Not Only SQL**". Cette approche propose de relâcher certaines contraintes lourdes du relationnel pour favoriser la distribution (structure des données, langage d'interrogation ou la cohérence).

Dans un contexte bases de données, il est préférable d'avoir un langage de haut niveau pour interroger les données plutôt que tout exprimer en [Map/Reduce](https://fr.wikipedia.org/wiki/MapReduce). Toutefois, avoir un langage de trop haut niveau comme SQL ne facilite pas la manipulation. Et c'est en ce sens que l'on peut parler de "Not Only SQL", d'autres solutions peuvent être proposées pour résoudre le problème de distribution. Ainsi, le NoSQL est à la fois une autre manière d'interroger les données, mais aussi de les stocker.

**Etudiez le fonctionnement d’Elasticsearch**

Elasticsearch est une base de données **NoSQL** dont la particularité est de pouvoir indexer des documents fortement orientés textes. On pourrait le comparer à un moteur de recherche, mais que vous pourriez paramétrer pour qu’il colle exactement à vos besoins de recherche. Elasticsearch, c’est donc un moteur de recherche capable de stocker une grande quantité de documents et que l’on peut interroger en temps réel. De plus, son langage de requête apporte des possibilités d’interrogation intéressantes que l’on pourra exploiter pour extraire des statistiques en temps réel, mais gardons cela pour le dernier chapitre.

**Le moteur de recherche Lucene**

Les moteurs de recherche reposent sur le domaine de la "**Recherche d’information**" principalement utilisé dans les moteurs de recherche tels que Google, Bing, Yahoo!, DuckDuckGo, Qwant… Parmi eux, nous pouvons trouver le moteur de recherche **[Lucene](https://lucene.apache.org/)**, un logiciel OpenSource Apache présent dans de nombreux site web pour créer des moteurs de recherche dédiés.

Elasticsearch utilise Lucene pour ses requêtes.

Le principe est assez simple : tous les mots d’un texte ont leur importance et on peut effectuer des recherches sur ces mots. Mais cela ne repose pas simplement sur le fait que le mot est présent dans un document pour qu’il réponde ; il faut être capable de déterminer sa **pertinence**, sinon comment ferions-nous pour trouver celui qui nous intéresse le plus ?

Pour cela, nous allons utiliser la fréquence des mots dans le document (de quoi on parle dans ce document), la fréquence des mots dans l’ensemble des documents (est-ce que ce mot est vraiment important), la taille du document (document court vs roman). Au final, on va obtenir pour chaque mot un **poids** qui va nous servir pour établir un **score**pour chaque requête avec un modèle mathématique reposant sur des *vecteurs* et la fonction *cosinus*.

Le poids d’un mot dépend principalement de :

1. Sa fréquence dans le texte (plus le mot est présent, plus le sujet est ciblé),
2. La taille du texte (si l’on parle de tout, chaque mot n’a plus d’importance),
3. Sa fréquence dans toute la collection de documents (un mot fréquent est moins pertinent).

Pour fonctionner, Elasticsearch aura donc besoin de savoir quels mots sont employés dans chaque document. Pour cela, Elasticsearch intègre un moteur Lucene qui va s’occuper d’extraire les mots d’une collection de documents et de préparer des colonnes de mots. Car en effet, elasticsearch est une base de données NoSQL orientée colonnes : un mot = une colonne de documents avec pour valeur le poids du mot dans chacun des documents.

La capacité maximale d’un serveur Lucene est de **2 147 483 519** documents. Au-delà, un nettoyage est effectué pour libérer de la place. D’où l’utilité de créer une architecture distribuée.

**Installation d’un serveur elasticsearch**

Pour installer un serveur Elasticsearch, il est nécessaire de télécharger les [fichiers binaires](https://elastic.co/downloads), disponibles pour chaque systèmes d’exploitation.

Une fois téléchargée, ouvrez l’archive, puis rendez-vous dans le répertoire « bin/ ». Celui-ci contient les fichiers exécutables de démarrage du serveur.

Maintenant que cela fonctionne, laissons tourner cette console et ouvrons une page Web : [http://localhost:9200](http://localhost:9200/). Celle-ci nous retourne un document JSON donnant les propriétés du cluster :



**Importation de données**

Toutefois, notre base de données est vide, il faut insérer les données dans un **index**(l’équivalent d’une table). Chaque index peut être raffiné avec la notion de **type**, correspondant à une sous-catégorie de l’index que l’on pourra spécifier au besoin. Tous les types d’un index partagent le même schéma de documents JSON.

Pour ce qui est de l’import, le fichier doit respecter certaines caractéristiques. En effet, chaque document JSON à importer doit avoir la même structure (mêmes clés, mêmes types de données, et le même ordre). D’autre part, chacun des documents doit avoir un document JSON

Téléchargez l’archive suivante : [movies](https://drive.google.com/file/d/1FjHSOK97aBtPmHtBaaWUhaeIqp9f_tGp). Après décompression, vous pourrez constater que chaque document du fichier « movies\_elastic.json » est bien prefixé par un index que nous appellerons « movies », avec un type « movie » et un identifiant. Le document qui suit donne un extrait des informations qui seront indexées.

Pour pouvoir importer ce fichier, il faut l’envoyer sur l’**API REST** d’Elasticsearch. Pour cela, nous utiliserons le petit utilitaire « **curl** » qui permet d’envoyer des requêtes HTTP. Nous pourrons ainsi interagir avec la base de de données sur l’adresse : [http://localhost:9200](http://localhost:9200/)

curl -XPUT localhost:9200/\_bulk -H"Content-Type: application/json" --data-binary @movies\_elastic.json

Pour vérifier le contenu de la base, il suffit d’ouvrir un navigateur Web, et d’utiliser l’URL suivante : <http://localhost:9200/movies/movie/_search>. La liste des films est alors produite en sortie.

L’URL est composée de :

1. L’index : **movies**
2. Le type : **movie**
3. Le service utilisé : **\_search**

pour le dernier chapitre.