NoSQL Databases - Projet
Exploration et Interrogation de Bases de Données NoSQL avec Python

2024 - 2025

Objectifs du Projet :

Le but de ce TP / Projet est de vous familiariser avec deux types de systèmes de gestion de bases de données NoSQL : **MongoDB**, une base de données orientée document, et **Neo4j**, une base de données orientée graphe. Vous allez développer une application en Python capable d'interagir avec deux bases de données hébergées NOSQL dans le cloud, en répondant aux questions et en récupérant les données pertinentes.

Description du Projet :

Dans le cadre de ce projet, vous êtes chargés de développer une application Python en `streamlit` par exemple qui effectue les tâches suivantes :

1. Connexion aux Bases de Données
   - Établir une connexion sécurisée avec les instances cloud de MongoDB et de Neo4j.
2. Interrogation de MongoDB:
   - Effectuer des requêtes pour récupérer des documents spécifiques.
   - Implémenter des fonctions pour insérer, mettre à jour et supprimer des documents.
3. Interrogation de Neo4j:
   - Utiliser le langage de requête Cypher pour interroger la base de données Neo4j.
   - Créer des nœuds, des relations et des propriétés.
   - Effectuer des recherches pour trouver les chemins les plus courts, des motifs récurrents, ou pour effectuer des analyses de réseau.
4. Analyse et Visualisation:
   - Analyser les données récupérées à partir des requêtes.
   - Visualiser les résultats à l'aide d'une librairie Python appropriée, comme Matplotlib ou Seaborn pour MongoDB, et Neovis.js pour Neo4j.

Résultats Attendus

À la fin de ce projet, vous devez soumettre :

- Un rapport de projet détaillant votre démarche, les requêtes utilisées, les difficultés rencontrées et les solutions adoptées.
- Le code source de votre application, bien commenté pour expliquer votre logique.

Bonnes Pratiques de Développement

Pour assurer un développement structuré et reproductible, nous vous recommandons d'adopter les bonnes pratiques suivantes :

1. Utilisation d'un Environnement Virtuel

L'utilisation d'un environnement virtuel permet d'isoler les dépendances de votre projet et d'éviter les conflits avec d'autres projets.

- Création d'un environnement virtuel :

  ```
  python -m venv venv
  ```

- Activation de l'environnement :
  - Windows :

    ```
    venv\Scripts\activate
    ```

  - macOS/Linux :

    ```
    source venv/bin/activate
    ```

2. Gestion des Dépendances

Il est important de documenter toutes les bibliothèques utilisées dans un fichier `requirements.txt` :

```
pip freeze > requirements.txt
```

Pour installer les dépendances d'un projet :

```
pip install -r requirements.txt
```

3. Utilisation de GitHub

Le code source du projet est très fortement recommandé d'être versionné sur GitHub afin d'assurer un suivi des modifications et une collaboration efficace.

- Initialisation d'un dépôt Git :

```
git init
```

- Ajout et validation des fichiers :

```
git add .
git commit -m "Initial commit"
```

- Pousser les modifications vers GitHub :

```
git remote add origin https://github.com/votre-repo.git
git branch -M main
git push -u origin main
```

4. Organisation du Code

- Séparer les fichiers en modules logiques (ex: `database.py` pour la connexion aux bases, `queries.py` pour les requêtes, etc.).
- Utiliser des fichiers de configuration (`config.py` ou variables d'environnement) pour les informations sensibles (ex: clés API, identifiants de connexion).

5. Documentation et Bonnes Pratiques de Code

- Ajouter des commentaires explicatifs dans le code.
- Utiliser des docstrings pour documenter les fonctions et classes.
- Tester les fonctionnalités avec des jeux de données réduits avant d'exécuter des analyses complètes.

Base de données

Base de données MongoDB

Créer une base de données nommée `entertainment` sur mongodb contenant une collection nommée `films` et importer les données depuis le fichier movies.json déposé sur Moodle (Learning Esiea).

Questions

Ecrire des requetes mongo permettant de répondre aux questions suivantes :

1. Afficher l'année où le plus grand nombre de films ont été sortis.
2. Quel est le nombre de films sortis après l'année 1999.
3. Quelle est la moyenne des votes des films sortis en 2007.
4. Affichez un histogramme qui permet de visualiser le nombres de films par année.
5. Quelles sont les genres de films disponibles dans la bases
6. Quel est le film qui a généré le plus de revenu.
7. Quels sont les réalisateurs ayant réalisé plus de 5 films dans la base de données ?
8. Quel est le genre de film qui rapporte en moyenne le plus de revenus ?
9. Quels sont les 3 films les mieux notés (`rating`) pour chaque décennie (1990-1999, 2000-2009, etc.) ?
10. Quel est le film le plus long (`Runtime`) par genre ?
11. Créer une vue MongoDB affichant uniquement les films ayant une note supérieure à 80 (`Metascore`) et généré plus de 50 millions de dollars.
12. Calculer la corrélation entre la durée des films (`Runtime`) et leur revenu (`Revenue`). (*réaliser une analyse statistique.*)
13. Y a-t-il une évolution de la durée moyenne des films par décennie ?

Base de données Neo4j

Intégrer les données de MongoDB à Neo4j en ajoutant des relations supplémentaires (ex : genres, relations entre réalisateurs et acteurs). (*Exporter les relations depuis MongoDB et les importer dans Neo4j.*)

- Créer des noeuds de type `films` contenant les uniquement les champs `_id`, `title`, `year`, `Votes` , `Revenue`, `rating` et `director`.
- Créer des noeuds de type `Actors` contenant uniquement de manières distinctes les acteurs.
- Créer des relations "A jouer" entre les acteurs et les films dont ils ont jouer. i.e.(requettes mongo suivi par requettes Neo4j)
- Créer des noeuds de type `Actors` contenant tout les membres du projet et attacher le a un film de votre choix.
- Créer des noeuds de type `Realisateur` depuis le champs `Director` de la base mongo.

Écrire une requête Cypher pour trouver

14. Quel est l'acteur ayant joué dans le plus grand nombre de films ?
15. Quels sont les acteurs ayant joué dans des films où l'actrice `Anne Hathaway` a également joué ?
16. Quel est l'acteur ayant joué dans des films totalisant le plus de revenus ?
17. Quelle est la moyenne des votes ?
18. Quel est le genre le plus représenté dans la base de données ?
19. Quels sont les films dans lesquels les acteurs ayant joué avec vous ont également joué ?
20. Quel réalisateur `Director` a travaillé avec le plus grand nombre d'acteurs distincts ?
21. Quels sont les films les plus "connectés", c'est-à-dire ceux qui ont le plus d'acteurs en commun avec d'autres films ?
22. Trouver les 5 acteurs ayant joué avec le plus de réalisateurs différents.
23. Recommander un film à un acteur en fonction des genres des films où il a déjà joué.
24. Créer une relation `INFLUENCE_PAR` entre les réalisateurs en se basant sur des similarités dans les genres de films qu'ils ont réalisés.
25. Quel est le "chemin" le plus court entre deux acteurs donnés (ex : Tom Hanks et Scarlett Johansson) ?
26. Analyser les communautés d'acteurs : Quels sont les groupes d'acteurs qui ont tendance à travailler ensemble ? (*Utilisation d'algorithmes de détection de communauté comme Louvain.*)

Questions Transversales

27. Quels sont les films qui ont des genres en commun mais qui ont des réalisateurs différents ?
28. Recommander des films aux utilisateurs en fonction des préférences d'un acteur donné.
29. Créer une relation de "concurrence" entre réalisateurs ayant réalisé des films similaires la même année.
30. Identifier les collaborations les plus fréquentes entre réalisateurs et acteurs, puis analyser si ces collaborations sont associées à un succès commercial ou critique