GADIO Abdourahmane
4A FISA TD 44
NoSQL – Projet

Sommaire

Introduction

Ce projet consiste à utiliser les systèmes de gestion de bases de données MongoDB et Neo4j, au travers d'une application Python utilisant Streamlit pour interagir avec nos donnees.

Ce projet nous permet d'effectuer des opérations CRUD (Create, Read, Update, Delete) et des analyses de réseau. Ce projet met l'accent sur la manipulation de données NoSQL, en programmation Python, et sur la visualisation de données afin de pouvoir analyser des réseaux complexes.

Ce projet est disponible sur Github a cette adresse :
https://github.com/AbdourahmaneGadio/NoSQL-Python-Project

GADIO Abdourahmane
4A FISA TD 44
NoSQL – Projet

Requêtes MongoDB

1. Afficher l'année où le plus grand nombre de films ont été sortis.

Stage 1: $group

{ "_id": "$year", "filmCount": { "$sum": 1 } }

Stage 2: $sort

{ "filmCount": -1 }

Stage 3: $limit

1

2. Quel est le nombre de films sortis après l'année 1999.

Stage 1: $match

{ "year": { "$gt": 1999 } }

Stage 2: $count

"totalFilms"

3. Quelle est la moyenne des votes des films sortis en 2007.

Stage 1: $match

{ "year": 2007 }

Stage 2: $group

{ "_id": null, "avgVotes": { "$avg": "$Votes" } }

avgVotes : 192.5

4. Affichez un histogramme qui permet de visualiser le nombres de films par année.

Stage 1: $group

{ "_id": "$year", "filmCount": { "$sum": 1 } }

Stage 2: $sort

{ "_id": 1 }

5. Quelles sont les genres de films disponibles dans la bases

Stage 1: $unwind

"$genre"

Stage 2: $group

{ "_id": "$genre" }

Stage 3: $sort

{ "_id": 1 }

6. Quel est le film qui a généré le plus de revenu.

Stage 1: $sort
{ "Revenue (Millions)": -1 }

Stage 2: $limit
1

film = Paris pieds nus

7. Quels sont les réalisateurs ayant réalisé plus de 5 films dans la base de données ?

Stage 1: $group

{ "_id": "$Director", "filmCount": { "$sum": 1 } }

Stage 2: $match

{ "filmCount": { "$gt": 5 } }

Stage 3: $sort

{ "filmCount": -1 }

8. Quel est le genre de film qui rapporte en moyenne le plus de revenus ?

Stage 1: $unwind

"$genre"

Stage 2: $group

{ "_id": "$genre", "avgRevenue": { "$avg": "$Revenue (Millions)" } }

Stage 3: $sort

{ "avgRevenue": -1 }

Stage 4: $limit

1

9. Quels sont les 3 films les mieux notés (rating) pour chaque décennie (1990-1999, 2000-2009, etc.) ?

Stage 1 : $addFields

{ "decade" : { "$subtract" : [{ "$multiply" : [{ "$floor" : { "$divide" : ["$year", 10] } }, 10] }, 0] }}

Stage 2 : $group

{ "_id" : { "decade" : "$decade", "title" : "$title" }, "rating" : { "$max" : "$rating" } }

Stage 3 : $sort

{"_id.decade" : -1, "rating" : -1}

Stage 4 : $limit

3

10. Quel est le film le plus long (Runtime) par genre ?

Stage 1 : $unwind

"$genre"

Stage 2 : $group

{"_id" : "$genre",

"longestFilmRuntime" : {"$max" :"$Runtime (Minutes)"},

"title" : {"$first":"$title"}

}

11. Créer une vue MongoDB affichant uniquement les films ayant une note supérieure à 80 (Metascore) et généré plus de 50 millions de dollars.

```
db.createView(
  "highRatedHighRevenueFilms",
  "movies",
  [
    {
      $match: {
        Metascore: { $gt: 80 },
        Revenue_Millions: { $gt: 50 }
      }
    }
  ]
);
```

12. Calculer la corrélation entre la durée des films (Runtime) et leur revenu (Revenue). (réaliser une analyse statistique.)

```
db.movies.aggregate([
  {
    $project: {
      Runtime_Minutes: true,
      Revenue_Millions: true,
      _id: false // Exclure l'ID pour simplifier l'analyse externe.
    }
  }
]);
```

13. Y a-t-il une évolution de la durée moyenne des films par décennie ?

Stage 1 :$addFields

{ "decade" : { "$subtract" : [{ "$multiply" : [{ "$floor" : { "$divide" : ["$year",10] } },10] },0] }}

Stage 2 :$group

{"_id":"$decade","avgRuntime":{"$avg":"$Runtime (Minutes)"}}

Stage 3:$sort

{"_id":1}

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. (requêtes mongo suivi par requêtes 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.**

Requêtes Cypher pour Neo4J

14. Quel est l'acteur ayant joué dans le plus grand nombre de films ?

```
MATCH (act:Acteur)-[:A_JOUE_DANS]->(flm:Film)
RETURN act.nom, COUNT(flm) AS nb_films
ORDER BY nb_films DESC LIMIT 1
```

15. Quels sont les acteurs ayant joué dans des films où l'actrice Anne Hathaway a également joué ?

```
MATCH (anne:Acteur {nom: "Anne Hathaway"})-[:A_JOUE_DANS]->(flm:Film)<-[:A_JOUE_DANS]-(coact:Acteur)
WHERE anne <> coact
RETURN DISTINCT coact.nom
```

16. Quel est l'acteur ayant joué dans des films totalisant le plus de revenus ?

```
MATCH (act:Acteur)-[:A_JOUE_DANS]->(flm:Film)
RETURN act.nom, SUM(toFloat(flm.revenue)) AS revenus_totaux
ORDER BY revenus_totaux DESC LIMIT 1
```

17. Quelle est la moyenne des votes ?

```
MATCH (flm:Film)
RETURN AVG(toInteger(flm.Votes)) AS moyenne_votes
```

18. Quel est le genre le plus représenté dans la base de données ?

```
MATCH (flm:Film)
UNWIND split(flm.genre, ",") AS gnr
RETURN trim(gnr) AS genre, COUNT(*) AS nb_films
ORDER BY nb_films DESC LIMIT 1
```

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 ?

```
MATCH (real:Réalisateur)-[:A_REALISE]->(flm:Film)<-[:A_JOUE_DANS]-(act:Acteur)
RETURN real.nom, COUNT(DISTINCT act) AS nb_acteurs
ORDER BY nb_acteurs DESC LIMIT 1
```

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 ?

```
MATCH (flm1:Film)<-[:A_JOUE_DANS]-(act:Acteur)-[:A_JOUE_DANS]->(flm2:Film)
WHERE flm1 <> flm2
RETURN flm1.titre, COUNT(DISTINCT flm2) AS connexions
ORDER BY connexions DESC LIMIT 10
```

22. Trouver les 5 acteurs ayant joué avec le plus de réalisateurs différents.

```
MATCH (act:Acteur)-[:A_JOUE_DANS]->(flm:Film)<-[:A_REALISE]-(real:Réalisateur)
RETURN act.nom, COUNT(DISTINCT real) AS nb_realisateurs
ORDER BY nb_realisateurs DESC LIMIT 5
```

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) ?

```
MATCH (act1:Acteur {nom: "Tom Hanks"}), (act2:Acteur {nom: "Scarlett Johansson"})
CALL apoc.algo.shortestPath(act1, act2, "A_JOUE_DANS>|<A_JOUE_DANS", "COSTAR")
YIELD chemin
RETURN chemin
```

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.)

Difficultés et solutions

Lors de ce projet, plusieurs difficultés ont été rencontrées.
Notamment :

- L'apprentissage et la maîtrise du langage de requête Cypher pour Neo4j
- L'implémentation d'analyses de réseau efficaces dans Neo4j
- L'intégration des bases de données avec Streamlit en Python
  C'est la documentation Streamlit qui nous a guidé pour simplifier l'intégration.