E1 : Intégration et Analyse des Données Littéraires

Introduction

L'objectif était de collecter des données sur des auteurs, leurs œuvres, et les distinctions qu'ils ont reçues à partir de trois sources principales : une API externe, des pages web par scraping, et une base de données existante. Une fois les données centralisées, elles ont été structurées dans une base relationnelle et exposées via une API sécurisée développée en FastAPI.

Les défis principaux comprenaient la manipulation de données hétérogènes, la sécurisation de l'accès à la base de données, et l'implémentation de requêtes analytiques pertinentes. Ce rapport décrit en détail les étapes suivies, les technologies choisies, et les résultats obtenus.

Objectifs du projet

Ce projet avait plusieurs objectifs précis, qui peuvent être résumés comme suit :

1. **Collecte de données** :
    - Récupérer des informations sur les auteurs et leurs œuvres depuis des sources diversifiées : une API (Google Books), des pages Wikipédia (scraping), et une base de données existante (distinctions littéraires).
2. **Organisation des données** :
    - Centraliser ces données dans une base de données relationnelle MySQL, avec une structure robuste permettant des jointures efficaces entre les tables.
3. **Création d'une API** :
    - Développer une API REST avec FastAPI pour permettre un accès simple et structuré aux données.
4. **Analyse des données** :
    - Implémenter des requêtes SQL complexes pour extraire des informations analytiques, comme les catégories les plus fréquentes des livres ou les distinctions par auteur.
5. **Sécurisation des données et de l'API**
    - Assurer la protection des informations sensibles comme les clés API et les identifiants de la base de données, ainsi que l'authentification de l'API finale.

Technologies choisies

Le choix des technologies a été guidé par leur pertinence et leur compatibilité avec les objectifs du projet :

- **Python** : Utilisé pour la collecte, le traitement, et la manipulation des données grâce à ses bibliothèques riches (requests, BeautifulSoup, pandas, SQLAlchemy).
- **FastAPI** : Framework rapide et léger pour créer une API RESTful, permettant une intégration simple avec Python et SQLAlchemy.
- **MySQL** : Système de gestion de base de données relationnelle pour stocker et organiser les données centralisées.
- **DBeaver** : Interface graphique utilisée pour gérer la base de données et exécuter des requêtes SQL.
- **dotenv** : Gestion des variables d'environnement pour sécuriser les informations sensibles comme les clés API.

Méthodologie et Étapes

1. Collecte des données

a) Extraction depuis l'API Google Books

Nous avons utilisé l'API Google Books pour collecter des informations sur les œuvres de dix auteurs célèbres. Les données extraites incluent les titres, les dates de publication, les descriptions, les catégories, et le nombre de pages. Une clé API a été utilisée pour authentifier les requêtes.

Exemple de requête :

```
def get_books_by_author(author_name):
    url = "https://www.googleapis.com/books/v1/volumes"
    params = {"q": f"inauthor:{author_name}", "key": API_KEY, "maxResults": 10}
    response = requests.get(url, params=params)
    return response.json() if response.status_code == 200 else None
```

Les résultats ont été nettoyés et convertis en un DataFrame avant d'être exportés au format CSV.

b) Scraping des données biographiques depuis Wikipédia

Pour enrichir nos données, nous avons récupéré les dates de naissance et de décès des auteurs depuis leurs pages Wikipédia respectives. Le scraping a été effectué avec BeautifulSoup.

Étapes principales :

1. Charger la page de chaque auteur.
2. Extraire les informations pertinentes (dates, biographie).
3. Nettoyer et formater les données.

Exemple de code pour le scraping :

```
def scrape_wikipedia(url):
    response = requests.get(url)
    soup = BeautifulSoup(response.text, 'html.parser')
      birth_date = soup.find('span', {'class': 'bday'}).text if soup.find('span', {'class': 'bday'}) else "N/A"
    return {"BirthDate": birth_date}
```

c) Intégration des distinctions littéraires

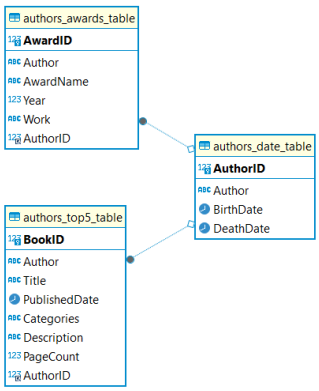
Une base de données existante a été utilisée pour collecter des informations sur les distinctions littéraires obtenues par les auteurs. Les données incluent le nom de l'auteur, le prix remporté, l'année, et l'œuvre primée.

2. Création de la base de données relationnelle

Les données collectées ont été organisées dans une base de données MySQL, composée de trois tables principales :

1. **`authors_date_table`** : Contient les informations biographiques des auteurs.
2. **`authors_top5_table`** : Liste les œuvres principales des auteurs.
3. **`authors_awards_table`** : Répertorie les distinctions reçues.

Des relations ont été définies entre ces tables pour permettre des jointures efficaces. Voici le MCD (Modèle Conceptuel des Données) illustrant la structure relationnelle mise en place :



Exemple de création d'une table :

```
CREATE TABLE authors_date_table (
    AuthorID INT AUTO_INCREMENT PRIMARY KEY,
    Author VARCHAR(255),
    BirthDate DATE,
    DeathDate DATE
);
```

3. Développement de l'API

Une API REST a été créée à l'aide de FastAPI pour permettre l'accès aux données et la réalisation de requêtes analytiques.

- **Endpoint : Catégories des livres**
    - Retourne le nombre de livres par catégorie.

Requête SQL :

```
 SELECT Categories, COUNT(BookID) AS NumberOfBooks
FROM authors_top5_table
GROUP BY Categories
ORDER BY NumberOfBooks DESC;
```

- **Endpoint : Distinctions des auteurs**
    - Retourne les distinctions reçues par auteur, avec leurs dates de naissance.

Requête SQL :

```
 SELECT ad.Author, ad.BirthDate, aa.AwardName, aa.Year, aa.Work
FROM authors_date_table ad
```

INNER JOIN authors_awards_table aa ON ad.AuthorID = aa.AuthorID
ORDER BY aa.Year DESC;

4. Sécurisation des données

Un fichier `.env` a été utilisé pour stocker les informations sensibles :

- Clé API pour Google Books.
- URL de connexion à la base de données.

- Token d'authentification à l'API

Le fichier `.env` a été exclu du dépôt Git via `.gitignore` pour éviter toute fuite d'informations sensibles.

Résultats

1. Analyse des données

- **Catégories des livres** :
    - La catégorie la plus
    - fréquente parmi les œuvres collectées est **Fiction**, suivie de **Literary Collections**. Ces données ont permis de mieux comprendre la répartition des œuvres des auteurs étudiés.
- **Distinctions littéraires** :
    - Une jointure SQL entre les tables a révélé les récompenses associées à chaque auteur. Les résultats montrent que des auteurs comme **George Orwell** et **Romain Gary** ont été récompensés pour des œuvres qui continuent d'avoir un impact significatif dans la littérature mondiale.

2. Fonctionnalités de l'API

- L'API FastAPI permet d'accéder efficacement aux données via des endpoints spécifiques. Par exemple :
    - **Liste des livres par catégorie** : Un endpoint retourne les catégories des livres et le nombre d'ouvrages dans chaque catégorie.
    - **Auteurs et leurs distinctions** : Un autre endpoint expose les récompenses reçues par chaque auteur, avec des informations biographiques.
- Les endpoints développés offrent une interaction directe avec les données stockées dans la base MySQL.

Les requêtes SQL exécutées dans le cadre de l'API ont été testées avec succès dans le code, et les résultats des endpoints sont retournés au format JSON, facilitant leur exploitation dans d'autres applications.