RAPPORT DE MINI-PROJET : BDD DE BANDES DESSINÉES (BDD DE BD)

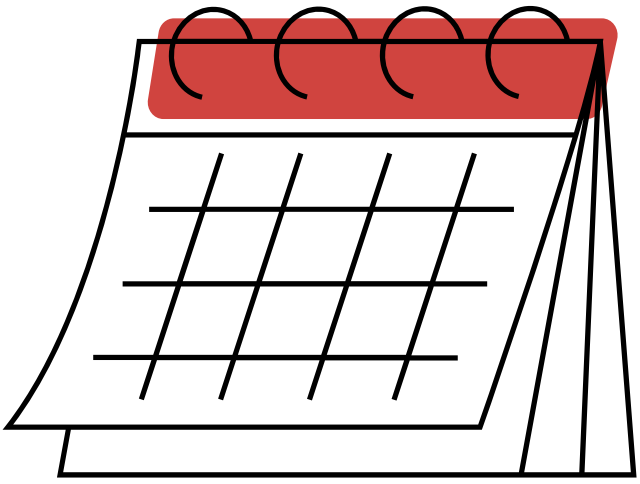
DIAKITE Diaby - SAKKAL Abdel-Waheb

2023-2024

PARIS PANTHÉON-ASSAS UNIVERSITÉ

INTRODUCTION

L'objectif de ce projet est de développer un système de gestion efficace pour organiser une collection de bandes dessinées. Cette démarche vise à résoudre le défi courant rencontré par les amateurs de bandes dessinées lors de l'organisation et de la classification de leurs ouvrages, en fournissant une solution pratique et élégante. En utilisant les concepts fondamentaux des bases de données et du langage SQL, nous cherchons à concevoir une structure optimisée permettant de répertorier et de classer les bandes dessinées en fonction de différents critères tels que le type, le genre, l'année de publication. En adoptant une approche méthodique, nous visons à fournir une solution qui garantit l'intégrité des données, tout en offrant une expérience utilisateur fluide et satisfaisante.



CALENDRIER DU PROJET

Séance 1 (3 heures): Analyse du cahier des charges, conception du modèle E/A

Séance 2 (3 heures): Création du schéma relationnel, mise en place de la base d
données, interrogation de la base de données

DESCRIPTION DE LA DÉMARCHE GÉNÉRALE DU PROJET

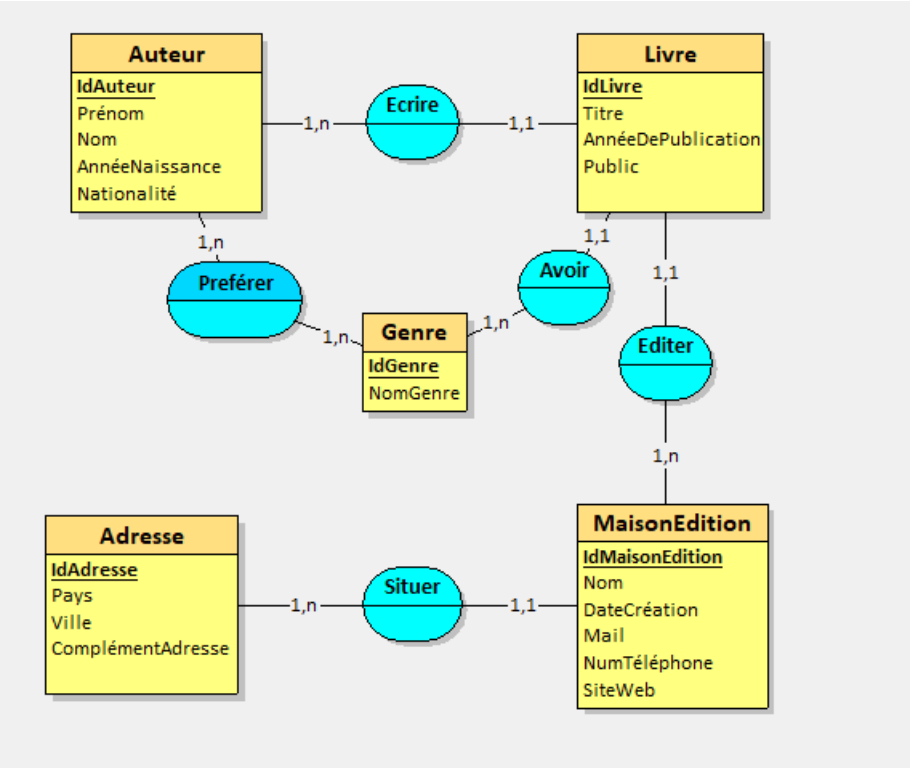
Nous avons commencé par analyser minutieusement le cahier des charges afin de bien comprendre les exigences du projet. Ensuite, nous avons listés les besoins en identifiant les entités principales ainsi que leurs attributs et les relations entre elles. Nous avons utilisé cette spécification pour concevoir le modèle Entité-Association (E/A) de notre base de données. En parallèle, nous avons également planifié la structure de notre base de données relationnelle et les contraintes qui devront être appliquées pour garantir l'intégrité des données. Enfin, nous avons implémenté la solution en utilisant SQL pour créer la base de données, y ajouter les données et interroger la base pour valider son fonctionnement.

OUTILS UTILISÉS :

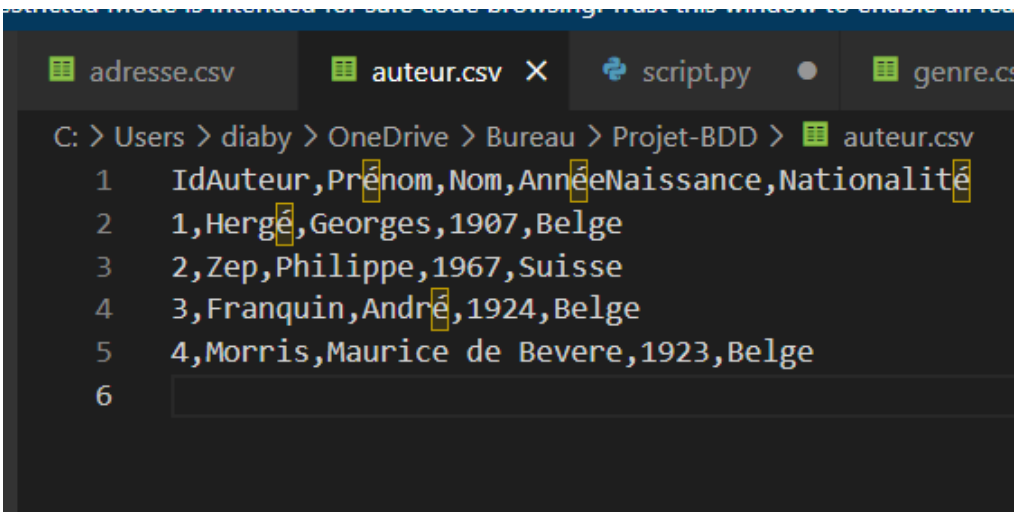
ETAPES ET MISE EN PLACE DE LA SOLUTION

1. Analyse du Cahier des Charges: Compréhension des besoins et des contraintes du projet.
2. Conception du Modèle E/A: Identification des entités, des attributs et des relations entre elles.
3. Création du Schéma Relationnel: Transformation du modèle E/A en schéma relationnel et application des règles de normalisation.
4. Création de la Base de Données: Utilisation de SQL et de Python pour créer la structure de la base de données et ajouter les contraintes d'intégrité.
5. Peuplement de la Base de Données: Insertion de données dans la base en respectant les contraintes définies.
6. Interrogation de la Base de Données: Utilisation de requêtes SQL pour interroger la base de données et répondre à des scénarios d'utilisation.

Shéma relationnel :



Exemple d'application



Fichier auteur.csv :

- Ce fichier contient des informations sur les auteurs de bandes dessinées telles que leur identifiant, prénom, nom, année de naissance et nationalité. Chaque ligne représente un auteur avec ses données correspondantes, séparées par des virgules.
- Les données sont organisées en colonnes, chaque colonne étant séparée par une virgule, ce qui en fait un fichier CSV (Comma-Separated Values).
- Les colonnes comprennent : **IdAuteur**, **Prénom**, **Nom**, **AnnéeNaissance**, et **Nationalité**.

adresse.csv | auteur.csv | script.py | genre.csv | data.csv | preferer.csv | Untitled-1

C: > Users > diaby > OneDrive > Bureau > Projet-BDD > script.py

```python
import csv
import mysql.connector


def load_auteur():
    # Lecture du fichier CSV et insertion des données dans la base de données
    with open('auteur.csv', 'r', encoding='utf-8') as file:
        reader = csv.DictReader(file)
        for row in reader:
            id_auteur = row['IdAuteur']
            prenom = row['Prénom']
            nom = row['Nom']
            annee_naissance = row['AnnéeNaissance']
            nationalite = row['Nationalité']

            # Requête SQL pour insérer les données dans la table
            sql = "INSERT INTO auteur (IdAuteur, Prénom, Nom, AnnéeNaissance, Nationalité) VALUES (%s, %s, %s, %s, %s)"
            values = (id_auteur, prenom, nom, annee_naissance, nationalite)

            # Exécution de la requête SQL
            cursor.execute(sql, values)
            conn.commit()

```

- **Script Python script.py** :
  - Ce script Python utilise la bibliothèque csv pour lire les données du fichier CSV auteur.csv et la bibliothèque mysql.connector pour se connecter à une base de données MySQL.
  - La fonction load_auteur() est définie pour charger les données du fichier CSV dans la table auteur de la base de données.
  - À chaque itération sur une ligne du fichier CSV, les données sont extraites et utilisées pour construire une requête SQL d'insertion dans la table auteur.
  - Après l'exécution de la requête d'insertion pour chaque ligne, les modifications sont validées pour enregistrer les données dans la base de données.

Ce script permet donc de peupler la table auteur de la base de données avec les informations contenues dans le fichier CSV auteur.csv.

JEUX DE TESTS

Une fois le schéma crée et les différentes tables peuplées, il est nécessaire de voir si l'implémentation que nous avons fait est fonctionnelle.

Pour mettre en évidence le bon fonctionnement de la base de données, nous avons effectué une série de requêtes en SQL afin de vérifier chaque aspect des tables du schéma.

Ainsi vous retrouverez dans le fichier "tests.sql" toutes les commandes. Nous pouvons prendre une commande en guise d'exemple:

```
-- On affiche le titre des livres qui ont étés publiés en Suisse
SELECT L.Titre FROM Livre L JOIN MaisonEdition M USING(IdMaisonEdition) JOIN Adresse Ad USING(IdAdresse) WHERE Ad.Pays = "Suisse";
```

Cette commande est une double jointure, d'abord entre Livre et MaisonEdition pour récupérer les livres en fonctions de leurs maisons d'édition, puis entre MaisonEdition et Adresse, pour récupérer l'adresse de ces maisons d'édition, avant de les filtrer en fonction du pays de l'adresse qui doit être la Suisse.

CONCLUSION

Notre projet offre une manière efficace d'organiser une collection de bandes dessinées dans sa structure bien pensée, qui permet de gérer les différentes informations relatives aux bandes dessinées de manière cohérente. Cependant, des améliorations pourraient être apportées en ajoutant des nouvelles relations entités/associations, ou encore en ajoutant plus de données. Ce projet nous a permis de mettre en pratique les concepts théoriques étudiés en cours, tout en développant nos compétences en conception de bases de données et en langage SQL.