

Rapport Projet NF18

Système de gestion d'une bibliothèque



Groupe 2 TD1

EL MAHMOUD Yazan
DARTOIS Evan
THIAMENY Antoine
TROMBINI Gabriel
HOURMANT Pierre

Sommaire

| | |
|--------------------------------------|-----------|
| Introduction : | 3 |
| Relationnel | 4 |
| Modèle Conceptuel de Données | 4 |
| Modèle Logique de Données | 5 |
| Tables | 5 |
| Justifications et Explication du MLD | 7 |
| Application Python | 10 |
| Requetes | 12 |
| Non relationnel | 13 |
| Modèle non relationnel (JSON) | 13 |
| MLD | 13 |
| Application Python | 14 |

Introduction :

L'avènement des technologies de l'information offre à la bibliothèque municipale l'opportunité d'optimiser ses activités en les informatisant. Cette transformation vise à simplifier les processus liés au catalogage, à la consultation, à la gestion des utilisateurs et des prêts. Ce rapport détaille la conception d'un système complet, allant de la base de données à une application Python, afin de répondre aux besoins diversifiés de la bibliothèque.

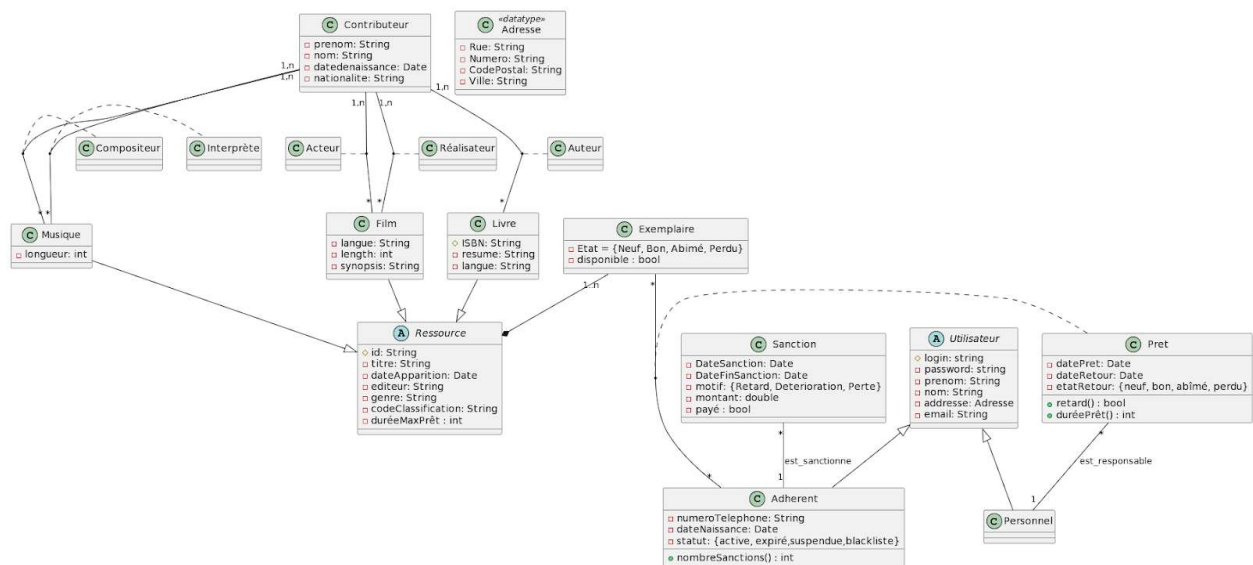
L'objectif principal de cette initiative est de faciliter tant aux adhérents qu'au personnel la gestion quotidienne des activités bibliothécaires. Des fonctionnalités telles que la recherche simplifiée de documents, la gestion intuitive des emprunts, le suivi des retards, et la génération de statistiques pertinentes sont au cœur de cette démarche d'informatisation.

Au fil des sections suivantes, nous explorerons en détail la conception de la base de données, les fonctionnalités clés de l'application, et comment celles-ci contribueront à une expérience utilisateur améliorée. L'objectif ultime est de créer un environnement numérique efficace, favorisant l'accès à la connaissance et simplifiant la gestion interne de la bibliothèque municipale.

Dans ce rapport, nous détaillerons toutes les étapes de la solution proposée ; de la conception de la base de données à la réalisation de l'application python permettant de la peupler et l'utiliser.

Relationnel

Modèle Conceptuel de Données



Modèle Logique de Données

Tables

Ressource(#id: VARCHAR, titre: VARCHAR, dateApparition: DATE, editeur: VARCHAR, genre: VARCHAR, codeClassification: VARCHAR)

Livre(#id_livre -> Ressource(id), ISBN: VARCHAR, resume: VARCHAR, langue: VARCHAR) avec ISBN UNIQUE NON NULL

Musique(#id_musique -> Ressource(id), longueur: INT) avec longueur NON NULL

Film(#id_film -> Ressource(id), langue: VARCHAR, length: INT, synopsis: VARCHAR) avec length NON NULL

Contributeur(#id : INT, prenom: VARCHAR, nom: VARCHAR, datedenaissance: DATE, nationalite: VARCHAR)

Auteur(#id_livre -> Livre(id), #id_contributeur -> Contributeur(id)) Contrainte : Projection(A_écrit, id_livre)) IN Projection(Livre, id))

Interprete(#id_musique -> Musique(id_musique), #id_contributeur -> Contributeur(id)) Contrainte : Projection(A_interprété, id_musique)) IN Projection(Musique, id_musique))

Compositeur(#id_musique -> Musique(id_musique), #id_contributeur -> Contributeur(id)) Contrainte : Projection(A_composé, id_musique)) IN Projection(Musique, id_musique))

Acteur(#id_film -> Film(id), #id_contributeur -> Contributeur(id)) Contrainte : Projection(A_joué, id_film)) IN Projection(Film, id_film))

Réalisateur(#id_film -> Film(id), #id_contributeur -> Contributeur(id)) Contrainte : Projection(A_réalisé, id_film)) IN Projection(Film, id_film))

Exemplaire(#id_exemplaire -> Ressource(id), Etat: ENUM ('Neuf', 'Bon', 'Abimé', 'Perdu'), Disponible : Bool)

<<datatype>> Adresse(Rue: VARCHAR, Numero: VARCHAR, CodePostal: VARCHAR, Ville: VARCHAR)

Utilisateur(#id : INT login: VARCHAR, password: VARCHAR, prenom: VARCHAR, adresse : Adresse, nom: VARCHAR, email: VARCHAR)

Personnel(#id_personnel -> utilisateur)

Adherent(#id_adherent -> Utilisateur(id), numeroTelephone: VARCHAR, dateNaissance: DATE, statut: {'active', 'expiré', 'suspendue', 'blacklisté'})

Pret(#id_exemplaire -> Exemplaire(id_exemplaire), #Id_adherent -> Adherent(id_adherent), datePret: DATE, durée: INT, dateRetour: DATE, etatRetour: ENUM ('neuf', 'bon', 'abîmé', 'perdu'), id_responsable -> Personnel(id_personnel) avec id_personnel UNIQUE NON NULL)

Sanction(#id_sanction: INT, DateSanction: DATE, DateFinSanction: DATE, motif: ENUM ('Retard', 'Déterioration', 'Perte'), montant: DOUBLE, Id_sanctionne -> Adherent(id_adherent) UNIQUE NON NULL)

Vue_LivreDetails(#id_livre: VARCHAR, Ressource.titre: VARCHAR, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Ressource.codeClassification: VARCHAR, Livre.ISBN: VARCHAR, Livre.resume: VARCHAR, Livre.langue: VARCHAR)

Vue_LivreExemplaires(#id_livre: VARCHAR, Ressource.titre: VARCHAR, Exemplaire.etat: ENUM ('Neuf', 'Bon', 'Abîmé', 'Perdu'), Exemplaire.disponible: BOOLEAN, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Ressource.codeClassification: VARCHAR, Livre.ISBN: VARCHAR, Livre.resume: VARCHAR, Livre.langue: VARCHAR)

Vue_MusiqueDetails(#id_musique: VARCHAR, Ressource.titre: VARCHAR, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Ressource.codeClassification: VARCHAR, Musique.longueur: INT)

Vue_MusiqueExemplaires(#id_musique: VARCHAR, Ressource.titre: VARCHAR, Exemplaire.etat: ENUM ('Neuf', 'Bon', 'Abîmé', 'Perdu'), Exemplaire.disponible: BOOLEAN, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Ressource.codeClassification: VARCHAR, Musique.longueur: INT)

Vue_FilmDetails(#id_film: VARCHAR, Ressource.titre: VARCHAR, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Film.synopsis: VARCHAR, Ressource.codeClassification: VARCHAR, Film.length: INT, Film.langue: VARCHAR)

Vue_FilmExemplaires(#id_film: VARCHAR, Ressource.titre: VARCHAR, Exemplaire.etat: ENUM ('Neuf', 'Bon', 'Abîmé', 'Perdu'), Exemplaire.disponible: BOOLEAN, Ressource.dateApparition: DATE, Ressource.editeur: VARCHAR, Ressource.genre: VARCHAR, Film.synopsis: VARCHAR, Ressource.codeClassification: VARCHAR, Film.length: INT, Film.langue: VARCHAR)

Vue_PretDetails(#id: INT, Pret.datePret : DATE, Pret.duree, : INT Pret.dateRetour : DATE, Pret.etatRetour : ENUM, Adherent.numeroTelephone : VARCHAR, Adherent.dateNaissance : DATE, Adherent.statut : VARCHAR, Utilisateur.login : VARCHAR, Utilisateur.prenom : VARCHAR, Utilisateur.nom : VARCHAR, Utilisateur.email : VARCHAR, Utilisateur.adresse : Adresse, Exempleire.Etat : VARCHAR, Exempleire.Disponible : BOOLEAN, Ressource.titre : VARCHAR, Ressource.dateApparition : DATE, Ressource.editeur : VARCHAR, Ressource.genre : VARCHAR, Ressource.codeClassification : VARCHAR)

Justifications et Explication du MLD

Justification des Transformations par référence :

Intégrité des données : En choisissant la transformation par référence, on assure que les données de la base de données restent cohérentes et conformes aux règles de l'application.

Faciliter les requêtes complexes : En reliant les tables par des clés étrangères, on simplifie les requêtes qui nécessitent des informations de plusieurs entités.

Héritage par référence de "Livre" et "Film" à partir de "Ressource" :

Justification : L'utilisation de l'héritage par référence est privilégiée lorsque des entités partagent des attributs communs avec une autre entité. Dans le cas de "Livre" et "Film," ils partagent des attributs de base avec la table "Ressource" tels que le titre, la date d'apparition, l'éditeur, le genre et le code de classification. L'héritage par référence permet de lier ces deux entités à "Ressource" tout en conservant ces attributs communs dans une seule table. Cela simplifie la structure de la base de données et évite la duplication inutile de données. De plus, cela permet d'ajouter d'autres types de ressources sans avoir à créer de nouvelles tables.

Héritage par référence d'"Adherent" et "Personnel" à partir de "Utilisateur" :

Justification : L'héritage par référence est également pertinent pour "Adherent" et "Personnel" qui partagent des caractéristiques communes avec "Utilisateur" telles que le login, le mot de passe, le prénom, le nom, l'adresse et l'adresse e-mail. En utilisant l'héritage par référence, ces entités héritent des attributs communs de "Utilisateur" sans nécessiter la duplication des colonnes. Cela facilite la maintenance, la gestion et la cohérence des données des utilisateurs. De plus, cela permet de gérer les adhérents et le personnel de la bibliothèque de manière centralisée.

Afin de faciliter l'accès à la BDD, nous avons créer des vues, notamment les suivantes :

Vue_LivreDetails :

Justification : Cette vue est utile pour présenter en détail les informations spécifiques aux livres, telles que le titre, l'éditeur, la date d'apparition, le genre, le code de classification, l'ISBN, le résumé et la langue. Cela permet aux utilisateurs de voir toutes les informations essentielles concernant un livre en un seul coup d'œil, ce qui facilite la recherche et la sélection de livres.

Vue_LivreExemplaires :

Justification : Cette vue est importante pour les utilisateurs qui souhaitent savoir quels exemplaires de livres sont disponibles à la bibliothèque. Elle inclut des détails sur l'état de chaque exemplaire (neuf, bon, abîmé, perdu), ainsi que l'indication de leur disponibilité. Cela permet aux utilisateurs de voir rapidement quels exemplaires de livres ils peuvent emprunter.

Vue_MusiqueDetails :

Justification : Cette vue est conçue pour présenter de manière détaillée les informations spécifiques aux œuvres musicales, notamment le titre, l'éditeur, la date d'apparition, le genre, le code de classification et la longueur de la musique. Cela permet aux utilisateurs de mieux comprendre les détails des enregistrements musicaux.

Vue_MusiqueExemplaires :

Justification : Cette vue est essentielle pour les utilisateurs qui souhaitent connaître la disponibilité des exemplaires d'œuvres musicales. Elle indique l'état de chaque exemplaire (neuf, bon, abîmé, perdu) ainsi que leur disponibilité. Cela facilite la sélection et la réservation de musique par les adhérents.

Vue_FilmDetails :

Justification : Cette vue est utilisée pour présenter en détail les informations spécifiques aux films, notamment le titre, l'éditeur, la date d'apparition, le genre, le synopsis, le code de classification, la longueur du film et la langue. Elle permet aux utilisateurs de trouver des informations complètes sur les films.

Vue_FilmExemplaires :

Justification : Cette vue est utile pour les utilisateurs qui recherchent des exemplaires de films disponibles à la bibliothèque. Elle indique l'état de chaque exemplaire (neuf, bon, abîmé, perdu) ainsi que leur disponibilité. Cela facilite la sélection et la réservation de films.

Vue_PretDetails :

Justification : Cette vue rassemble les informations concernant les prêts, les adhérents, les exemplaires et les ressources empruntées. Elle fournit une vue d'ensemble des prêts actuels, y compris la date de prêt, la durée, la date de retour, l'état de retour, les coordonnées de l'adhérent, les informations sur l'utilisateur et les détails de la ressource empruntée. Cela simplifie la gestion des prêts et la communication avec les adhérents.

Application Python

Nous avons créer une application python qui permet de se connecter à la Base de données, puis de créer des tables et d'exécuter des commandes de type Select, Insert, ect.

Première interface Python depuis un terminal :

Connexion à un utilisateur :

Login: user1

Mot de passe: password1

On rentre les informations correspondant à un utilisateur afin d'avoir accès à l'ensemble de la base de données de la bibliothèque.

Vue du menu interactif au début :

===== Menu Personnel =====

1. Recherche Livre / Musique / Film
2. Ajouter Livre / Musique / Film
3. Gérer les prêts
4. Gérer les sanctions
5. Gérer les utilisateurs
6. Visualiser les tables (Administrateur)
7. Créer une nouvelle table (Administrateur)
8. Supprimer une table (Administrateur)
9. Statistiques
10. Quitter

=====

Quel opération voulez-vous effectuer ?

1.Rechercher un livre

A cette étape, nous pouvons choisir plusieurs options s'offrant à nous, nous allons procéder à quelques exemples et voir le rendu de cette application python.

En appuyant sur 1 afin de rechercher un livre, un film ou une musique.

===== Menu Personnel =====

Entrez le titre de la ressource:

Nous devons choisir le titre et voici ce qui apparaît lorsqu'on met Book 1,

===== Menu Personnel =====

Livres

| Index | Titre | Résumé | Langue |
|-------|--------|----------------|---------|
| 0 | Book 1 | Book 1 Summary | English |

Ce sont les informations relatives au livre avec son titre, son index et sa langue. Puis appuyer sur -1 pour revenir au menu interactif.

2.Ajouter un film

Lorsque je veux ajouter un livre cette fois, je mets 2 dans le menu interactif.

===== Menu Personnel =====

Choisissez la table dans laquelle vous souhaitez insérer des données :

1. Livre
2. Musique
3. Film
4. Retour

Entrez le numéro de la table : 2

Entrez le titre de la ressource : musique contemporaine

Entrez la date d'apparition de la ressource (format YYYY-MM-DD) : 2023-05-06

Entrez l'éditeur de la ressource : nathan

Entrez le genre de la ressource : style

Entrez le code de classification de la ressource : 036

Longueur du film(en secondes) : 300

Je remplis les informations relatives au film et voici ce que j'ai

Musiques

| Index | Titre | Editeur | Longueur |
|-------|------------------------------|---------|----------|
| 1 | musique contemporaine nathan | | 300 |

3. Gérer les prêts

Nous retournons avec -1 à l'interface de base, maintenant j'appuie sur 3 pour gérer les prêts.

On nous demande avec login on veut se connecter en tant que utilisateur et cette interface ensuite apparaît :

1. Ajouter un nouveau prêt
 2. Gérer un prêt
 3. Retour
- Que voulez-vous faire ? :

Nous choisissons par la suite si nous voulons ajouter un prêt ou gérer le prêt.

4.Gérer les sanctions

Sanctions

| Index | Login | Date sanction | Date fin sanction | Motif | Montant | Payé |
|-------|-------|---------------|-------------------|--------|---------|------|
| 1 | user1 | 2023-01-25 | 2023-02-25 | Retard | 10.0 | Oui |

Sélectionnez un index de sanction (-1 retour):

Il est donc affiché les sanctions auxquelles est lié un utilisateur.

Sélectionnez un index de sanction (-1 retour): 1

1. Régler la sanction
 2. Ajouter date de fin à la sanction
 3. Supprimer la sanction
 4. Annuler
- Que voulez-vous faire ?:

Cette interface est ensuite affichée et nous choisissons l'option que nous voulons.

Par exemple, avec l'option "Régler la sanction", la sanction est relevée et l'utilisateur choisi n'a donc plus de sanction, exemple nous avons choisi user1 ici.

5. Gérer les utilisateurs

Cette interface permet de gérer les adhérents et le personnel. Par exemple nous pouvons supprimer ou ajouter un adhérent.

Que voulez vous faire ? : 1

Entrez login recherché: user1

Adhérents :

| | | | | | |
|---|-------|------|-----|-------------------|--------|
| 1 | user1 | User | One | user1@example.com | active |
|---|-------|------|-----|-------------------|--------|

1. Ajouter membre
2. Supprimer membre
3. Modifier statut adherent
4. Retour

Que voulez_vous faire ? :

Après avoir appuyé sur 2, l'utilisateur "user1" sera supprimé des adhérents et n'existera donc plus dans la base de données.

6. Visualiser les tables (Administrateur)

Après avoir entré le mot de passe administrateur "1", nous avons accès à l'ensemble des tables de la base de données. Nous choisissons ensuite la table que nous voulons visualiser.

===== Menu Personnel =====

1. Recherche Livre / Musique / Film
2. Ajouter Livre / Musique / Film

Entrez le mot de passe : 1

Voici les tables :

1. ressource
2. livre
3. musique
4. film
5. auteur
6. contributeur
7. interprete
8. compositeur
9. acteur

- 10. realisateur
- 11. exemplaire
- 12. adresse
- 13. utilisateur
- 14. personnel
- 15. adherent
- 16. pret
- 17. sanction

Choisissez une table pour voir son contenu (0 pour retour) : 1

7. Créer une nouvelle table (Administrateur)

Après avoir entré le mot de passe "1", nous avons les accès administrateur, nous avons ensuite accès pour créer une table, nous avons ensuite cette interface là, où plusieurs détails peuvent nous être demandés, comme le nom de la colonne ou le type de la colonne.

Entrez le mot de passe : 1

Entrez le nom de la table : coucou

Entrez le nom de la colonne : moi

Choisissez le type de la colonne :

- 1. VARCHAR
- 2. INT
- 3. DATE
- 4. FLOAT
- 5. BOOLEAN

Entrez votre choix : 1

Choisissez le nombre de caractères : 50

Est-ce que c'est une clé primaire ? (O/N) : O

Ecrit le contrait s'il y a: (EX: NOT NULL, UNIQU

8. Supprimer une table(Administrateur)

Cette dernière fonction nous permet de supprimer une table, l'ensemble des tables s'affichent et alors nous pouvons choisir la table que nous supprimons.

9. Statistiques

Statistiques

| Titre | Nombre d'emprunts |
|--------|-------------------|
| Book 1 | 2 |

1. Retour

Que voulez-vous faire ? :

Après avoir appuyé sur 9, la partie “Statistiques”, nous voyons apparaître l’ensemble des emprunts qui ont été réalisés, ici nous voyons deux emprunts de livre.

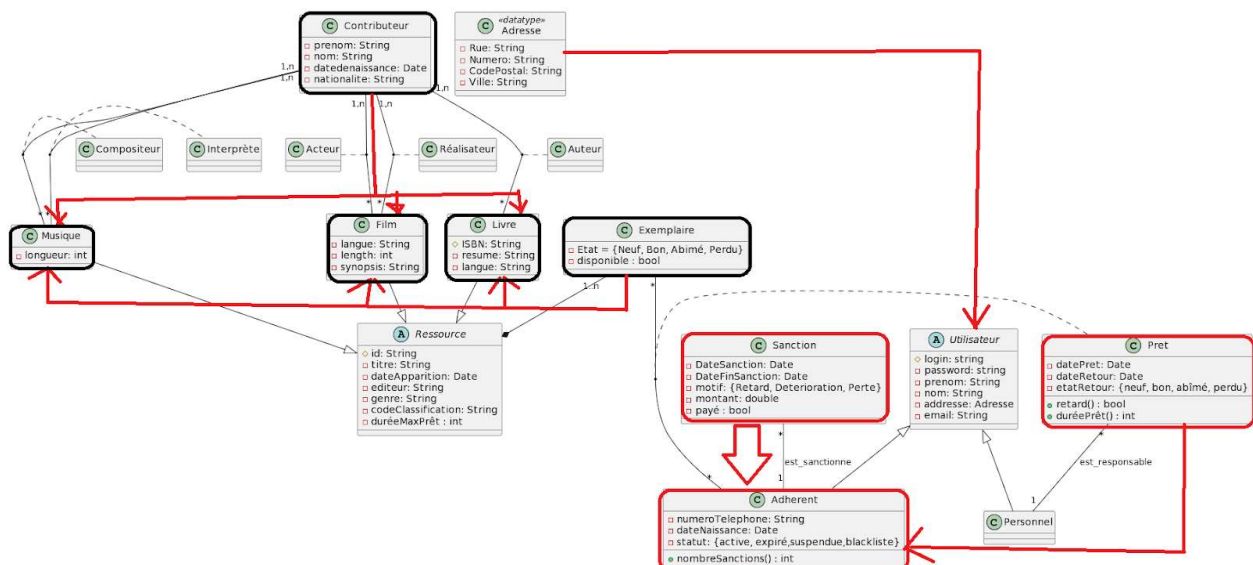
10. Quitter

Cette commande nous permet de quitter le programme.

Non relationnel

Modèle non relationnel (JSON)

A ce niveau d’implémentation de notre conception d’une base de données, nous nous intéresserons à l’implémentation d’attributs de type JSON dans nos tables déjà existantes. Pour ce faire, la première étape consiste à s’intéresser de plus près notre diagramme UML et de voir les changements possibles :



Ici nous avons fait le choix de mettre comme attribut JSON la table “**contributeur**” et “**exemplaire**” dans les tables filles “**Musique**”, “**Film**” et “**Livre**” qui héritent de la table Ressource. Nous avons de même pour les tables “**Sanction**” et “**Pret**” qui deviennent des attributs JSON dans la table “**Adhérent**”

MLD

Dans cette partie, nous allons présenter les modifications apportées au MLD après l'implémentation de JSON.

Voici les tables modifier :

Livre (#id: INT, titre: VARCHAR, dateApparition: DATE, editeur: VARCHAR, genre: VARCHAR, codeClassification: VARCHAR, dureeMaxPret INT, ISBN VARCHAR, resume TEXT, langue VARCHAR, contributeur JSON, exemplaires JSON)

Musique (#id: INT, titre: VARCHAR, dateApparition: DATE, editeur: VARCHAR, genre: VARCHAR, codeClassification: VARCHAR, dureeMaxPret INT, longueur INT, langue VARCHAR, contributeur JSON, exemplaires JSON)

Film (#id: INT, titre: VARCHAR, dateApparition: DATE, editeur: VARCHAR, genre: VARCHAR, codeClassification: VARCHAR, dureeMaxPret INT, langue VARCHAR, longueur INT, synopsis TEXT, langue VARCHAR, contributeur JSON, exemplaires JSON)

Utilisateur (#id : INT, login: VARCHAR, password: VARCHAR, prenom: VARCHAR, nom: VARCHAR, email: VARCHAR, adresse : JSON)

Personnel (#id_personnel -> utilisateur)

Adherent (#id_adherent -> Utilisateur(id), numeroTelephone: VARCHAR, dateNaissance: DATE, statut: {'active', 'expiré', 'suspendue', 'blacklisté'}, sanctions JSON, prets JSON)

Application Python

De la même manière que nous avons procédé pour l'application python sans le Json, nous avons conçu une application python avec le Json.

Voici le menu interactif :

```
1. Recherche Livre / Musique / Film
2. Ajouter Livre / Musique / Film
3. Gérer les prêts
4. Gérer les sanctions
5. Gérer les utilisateurs
6. Visualiser les tables (Administrateur)
7. Créer une nouvelle table (Administrateur)
8. Supprimer une table (Administrateur)
9. Statistiques
10. Quitter
=====
Quel opération voulez-vous effectuer ? 
```

Afin de rechercher une ressource :

```
Entrez le titre de la ressource: Inception 
```

Voici le résultats affiché :

| Films | | | |
|-------|-----------|---|----------|
| Index | Titre | Synopsis | Langue |
| 0 | Inception | Dom Cobb, un voleur expérimenté, est capable d'en | Français |

| Musiques | | | |
|----------|-------|---------|----------|
| Index | Titre | Editeur | Longueur |

Index de la ressource à consulter (-1 pour retour):

Acteurs :
 DiCaprio Leonardo 11/11/1974 américaine
 Realisateurs :
 Nolan Christopher 30/07/1970 britannique

1. Modifier la ressource
2. Supprimer la ressource
3. Consulter les exemplaires
4. Retour

Que voulez-vous faire ? :

noSQL ↺ ⊗ 0 ⚠ 1 🔊 0 📁 Connect 📄 Select Postgres Server

Pour insérer des données :

Choisissez la table dans laquelle vous souhaitez insérer des données :

1. Livre
2. Musique
3. Film
4. Retour

Entrez le numéro de la table :

Exemple :

```

1. Livre
2. Musique
3. Film
4. Retour
Entrez le numéro de la table : 1
Entrez le titre de la ressource : L'enfant noir
Entrez la date d'apparition de la ressource (format YYYY-MM-DD) : 2023-12-21
Entrez l'éditeur de la ressource : Gamard
Entrez le genre de la ressource : Madingue
Entrez le code de classification de la ressource : 106
Entrez le code ISBN du livre : ISBN-88
Entrez le résumé du livre : L'histoire de l'empire du mandingue
  
```

Gestion des prêts :

```
Entrez un login: utilisateur2
```

Pour visualiser un prêt :

```

Prêts en cours des logins commençant par utilisateur2
Index Date prêt Durée Date retour Titre          PreNom          Nom          Login
=====
0 2023-12-01 2023-12-10 Bon Le Seigneur de Fabrice      Durand      utilisateur2
=====
  
```

```

1. Ajouter un nouveau prêt
2. Gérer un prêt
3. Retour
Que voulez-vous faire ? : 
  
```

Entrez un login Adhérent: utilisateur1

Entrez le titre de la ressource: Inception

```

Films
Index      Titre      Synopsis
=====
0          Inception  Dom Cobb, un voleur expérimenté, est capable d entrer dans les rêves des autres pour voler leurs secrets les plus précieux. Français
=====

Musiques
Index      Titre      Editeur      Longueur
=====
=====
Index de la ressource d'intérêt (-1 pour annuler):
  
```

```

Prêts en cours des logins commençant par utilisateur2
Index Date prêt Durée Date retour Titre      Prenom      Nom      Login
=====
0 2023-12-01 2023-12-10 Bon Le Seigneur de Fabrice      Durand      utilisateur2
=====
  
```

1. Ajouter un nouveau prêt
 2. Gérer un prêt
 3. Retour
 Que voulez-vous faire ? : 2

Gestion sanctions

```
Sanctions
Index  Login      Date sanction  Date fin sanction  Motif      Montant  Payé
=====
1      utilisateur1 deterioration  2023-12-15      null        10        Non
=====
```

Sélectionnez un index de sanction (-1 retour):

```
Index  Login      Date sanction  Date fin sanction  Motif      Montant  Payé
=====
1      utilisateur1 deterioration  2023-12-15      null        10        Non
=====
```

Sélectionnez un index de sanction (-1 retour): 1

1. Régler la sanction
2. Ajouter date de fin à la sanction
3. Supprimer la sanction
4. Annuler

Que voulez-vous faire ? :

Gestion utilisateurs

1. Gérer les Adhérents
2. Gérer le Personnel
3. Retour

Que voulez vous faire ? :

Visualisons les Statistiques :

Pour un livre

Entrez votre choix : 1
 [('Le Petit Prince', 3), ('Le Seigneur des Anneaux : La Communauté de l Anneau', 3), ('Les Misérables', 3)]
 Statistiques

```
Titre      Nombre d'emprunts
Le Petit Prince      3
Le Seigneur des Anneaux : La Communauté de l Anneau      3
Les Misérables      3
```

1. Retour

Que voulez_vous faire ? :

Musique

```
Entrez votre choix : 2
[('Daft Punk - Random Access Memories', 3), ('La La Land - Original Motion Picture Soundtrack', 3)]
Statistiques
```

| Titre | Nombre d'emprunts |
|---|-------------------|
| Daft Punk - Random Access Memories | 3 |
| La La Land - Original Motion Picture Soundtrack | 3 |

1. Retour
Que voulez-vous faire ? :

```
Entrez votre choix : 3
[('Inception', 3), ('Le Seigneur des Anneaux : La Communauté de l Anneau', 3), ('Amélie Poulain', 3)]
Statistiques
```

| Titre | Nombre d'emprunts |
|---|-------------------|
| Inception | 3 |
| Le Seigneur des Anneaux : La Communauté de l Anneau | 3 |
| Amélie Poulain | 3 |

1. Retour
Que voulez-vous faire ? :