Programme de gestion de bibliothèque

Jean Vanneste

Version colorisé : https://github.com/JeanVanneste/ORM/blob/master/rapport/rapport.md

Initialisation

1. Installer mysql ou mariadb sur le serveur

```
sudo apt install mysql
```

2. Créer un utilisateur *admin* avec comme mot de passe *password* (Si vous souhaitez utiliser un autre profil, il suffit de modifier *config_db.json*)

```
CREATE USER 'admin'@'localhost' IDENTIFIED BY 'password';
GRANT ALL ON *.* TO 'admin'@'localhost';
```

3. Créer la base de donnée *library* (ou autre à préciser dans *config_db.json*)

```
CREATE DATABASE library;
```

4. Installer les bibliothèques python requise

```
pip3 install -r requirement.txt
```

5. Créer le schéma avec *library.sql* ou le script *populate_db.py*

```
mysql -p library < library.sql
```

ou

```
python3 populate_db.py
```

Utilisation

Le script *interface.py* permet de rajouter facilement de nouveaux éléments à la base de données. Il faut cependant faire attention à bien rajouter les éléments dans l'ordre.

Pour créer un collection, son éditeur doit exister dans la db. Pour créer un livre, sa collection et au moins un auteur doivent exister dans la db.

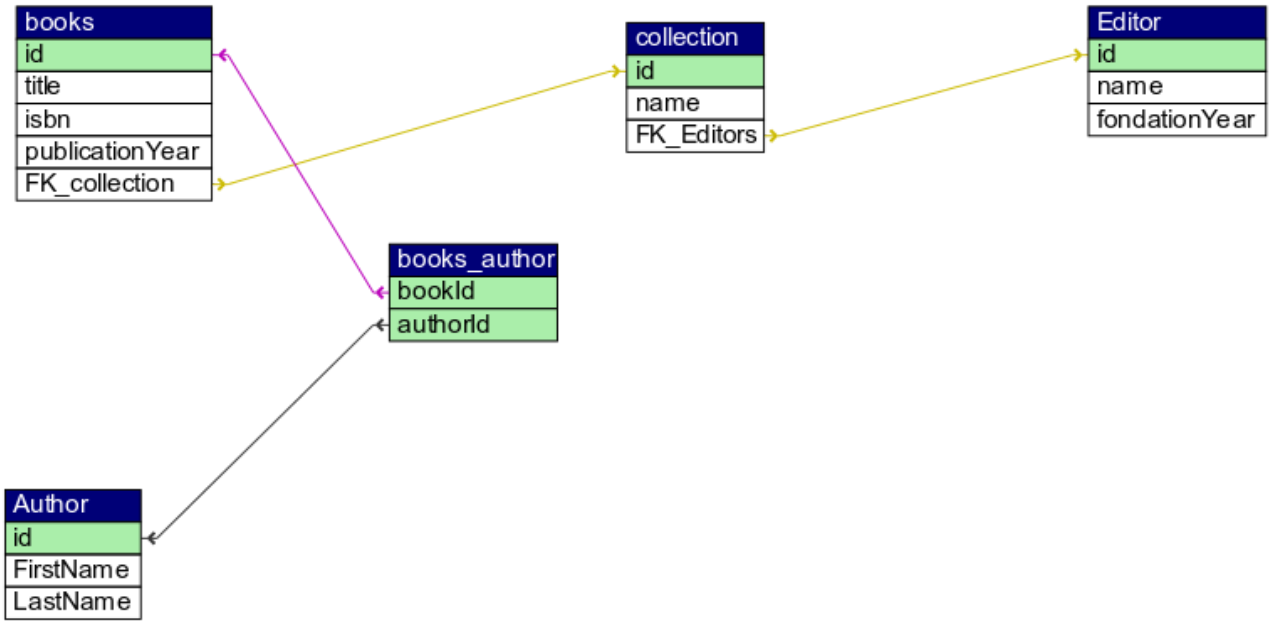
Un livre peut avoir plusieurs auteurs et un auteur peut avoir écrit plusieurs livres.

On ne peut indiquer qu'un seul auteur à la création d'un livre mais il est possible d'ajouter les auteurs restants par la suite.

Fonctionnement

- Le fichier **database.py** décrit la structure de la base de donnée. Elle permet de recréer les tables si celles sont inexistante dans la base de donnée et de créer les contraintes de clés étrangère. Elle mappe aussi les objets de l'application au données contenues dans la base.
- Le fichier **ORM.py** contient les différentes fonctions pour créer et interagir avec la base de données.
- Le fichier **populate_db.py** utilise *ORM.py* pour remplir quelques lignes des tables.
- Le fichier **interface.py** est un script permettant un remplissage *user-friendly* de la base de données

Schéma de la base de données



Code

Création de la Base de données

```
-- phpMyAdmin SQL Dump
-- version 4.6.6deb5
-- https://www.phpmyadmin.net/
--
-- Client :  localhost:3306
-- Généré le :  Dim 07 Juillet 2019 à 19:25
-- Version du serveur :  5.7.26-0ubuntu0.18.04.1
-- Version de PHP :  7.2.19-0ubuntu0.18.04.1
```

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Base de données :  `library`
--

-- --------------------------------------------------------

--
-- Structure de la table `authors`
--

CREATE TABLE `authors` (
  `id` int(11) NOT NULL,
  `firstName` varchar(30) NOT NULL,
  `lastName` varchar(30) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Contenu de la table `authors`
--

INSERT INTO `authors` (`id`, `firstName`, `lastName`) VALUES
(1, 'Nicolas', 'Dupuy');

-- --------------------------------------------------------

--
-- Structure de la table `books`
--

CREATE TABLE `books` (
  `id` int(11) NOT NULL,
  `title` varchar(255) NOT NULL,
  `isbn` varchar(17) NOT NULL,
  `publicationYear` int(11) NOT NULL,
  `FK_collection` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Structure de la table `books_authors`
--

CREATE TABLE `books_authors` (
  `bookId` int(11) NOT NULL,
```

```
  `authorId` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Structure de la table `collections`
--

CREATE TABLE `collections` (
  `id` int(11) NOT NULL,
  `name` varchar(100) NOT NULL,
  `FK_editor` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Structure de la table `editors`
--

CREATE TABLE `editors` (
  `id` int(11) NOT NULL,
  `name` varchar(30) NOT NULL,
  `fondationYear` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--
-- Index pour les tables exportées
--

--
-- Index pour la table `authors`
--
ALTER TABLE `authors`
  ADD PRIMARY KEY (`id`);

--
-- Index pour la table `books`
--
ALTER TABLE `books`
  ADD PRIMARY KEY (`id`),
  ADD KEY `FK_collection` (`FK_collection`);

--
-- Index pour la table `books_authors`
--
ALTER TABLE `books_authors`
  ADD PRIMARY KEY (`bookId`,`authorId`),
  ADD KEY `authorId` (`authorId`);

--
-- Index pour la table `collections`
--
```

```
ALTER TABLE `collections`
  ADD PRIMARY KEY (`id`),
  ADD KEY `FK_editor` (`FK_editor`);

--
-- Index pour la table `editors`
--
ALTER TABLE `editors`
  ADD PRIMARY KEY (`id`);

--
-- AUTO_INCREMENT pour les tables exportées
--

--
-- AUTO_INCREMENT pour la table `authors`
--
ALTER TABLE `authors`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=2;
--
-- AUTO_INCREMENT pour la table `books`
--
ALTER TABLE `books`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT pour la table `collections`
--
ALTER TABLE `collections`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- AUTO_INCREMENT pour la table `editors`
--
ALTER TABLE `editors`
  MODIFY `id` int(11) NOT NULL AUTO_INCREMENT;
--
-- Contraintes pour les tables exportées
--

--
-- Contraintes pour la table `books`
--
ALTER TABLE `books`
  ADD CONSTRAINT `books_ibfk_1` FOREIGN KEY (`FK_collection`) REFERENCES
`collections` (`id`);

--
-- Contraintes pour la table `books_authors`
--
ALTER TABLE `books_authors`
  ADD CONSTRAINT `books_authors_ibfk_1` FOREIGN KEY (`bookId`) REFERENCES
`books` (`id`),
  ADD CONSTRAINT `books_authors_ibfk_2` FOREIGN KEY (`authorId`) REFERENCES
`authors` (`id`);
```

```
--
-- Contraintes pour la table `collections`
--
ALTER TABLE `collections`
  ADD CONSTRAINT `collections_ibfk_1` FOREIGN KEY (`FK_editor`) REFERENCES
`editors` (`id`);

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

database.py

```
from sqlalchemy import Column, Integer, String, ForeignKey, Table,
create_engine
from sqlalchemy.ext.declarative import declarative_base
from sqlalchemy.orm import relationship
import json

with open ('config_db.json') as json_data_file:
    data = json.load(json_data_file)
    engine= 'mysql+pymysql://' + data['user'] + ':' + data['password'] +
'@' + data['host'] + '/' + data['db']

Base = declarative_base()

books_authors = Table('books_authors', Base.metadata,
                      Column('bookId', Integer, ForeignKey('books.id'),
primary_key = True),
                      Column('authorId', Integer, ForeignKey('authors.id'),
primary_key = True)
                     )


class Book(Base):
    __tablename__ = 'books'

    id = Column(Integer, primary_key = True)
    title = Column(String(255))
    isbn = Column(String(17))
    publicationYear = Column(Integer)
    FK_collection = Column(Integer, ForeignKey('collections.id'),
                           nullable = False)

    collection = relationship("Collection", back_populates =
"booksPublished")

    authors = relationship(
                           "Author",
                           secondary=books_authors,
                           back_populates="books")
```

```
    def __repr__(self):
        return ("<Book(title='%s')>" % self.title)

class Editor(Base):
    __tablename__ = 'editors'

    id = Column(Integer, primary_key=True)
    name = Column(String(30), nullable=False)
    fondationYear = Column(Integer, nullable=False)

    collectionsPossessed = relationship("Collection",
                                        back_populates = "editor")

    def __repr__(self):
        return "<Editor(name='%s', year of fondation = '%d')" % (
            self.name, self.fondationYear)

class Collection(Base):
    __tablename__ = 'collections'

    id = Column(Integer, primary_key=True)
    name = Column(String(100), nullable=False)
    FK_editor = Column(Integer, ForeignKey('editors.id'), nullable=False)

    editor = relationship("Editor", back_populates =
"collectionsPossessed")

    booksPublished = relationship("Book", back_populates = "collection")

    def __repr__(self):
        return "<Collection(name='%s', editor='%s')" % (self.name,
            self.FK_editor)

class Author(Base):
    __tablename__ = 'authors'

    id = Column(Integer, primary_key=True)
    firstName = Column(String(30), nullable=False)
    lastName = Column(String(30), nullable=False)

    books = relationship(
                           "Book",
                           secondary=books_authors,
                           back_populates="authors")

engine = create_engine(engine, echo = False)
Base.metadata.create_all(engine)
```

ORM.py

```
from sqlalchemy import create_engine, and_
import database
from sqlalchemy.orm import sessionmaker
from sqlalchemy.ext.declarative import declarative_base
import json

with open ('config_db.json') as json_data_file:
    data = json.load(json_data_file)
    engine= 'mysql+pymysql://' + data['user'] + ':' + data['password'] +
'@' + data['host'] + '/' + data['db']

engine = create_engine('mysql+pymysql://admin:password@localhost/library',
echo = False)
Session = sessionmaker(engine)


def editor_create(editorName, fondationYear):
    session = Session()

    editor = database.Editor()

    editor.name = editorName
    editor.fondationYear = fondationYear

    session.add(editor)
    session.commit()

def author_create(authorFirstName, authorLastName):
    session = Session()

    author = database.Author()

    author.firstName = authorFirstName
    author.lastName = authorLastName

    session.add(author)
    session.commit()

def collection_create(name, editorName):

    session = Session()

    collection = database.Collection()

    editor =
session.query(database.Editor).filter(database.Editor.name==editorName).fir
st()

    collection.name = name
    collection.editor = editor

    session.add(collection)
    session.commit()
```

```
def book_create(title, isbn, publicationYear, collectionName,
authorFirstName, authorLastName):
    session = Session()

    book = database.Book()

    authors = session.query(database.Author).\
                    filter(and_(database.Author.firstName==authorFirstName,
database.Author.lastName == authorLastName)).\
                    all()

    collection = session.query(database.Collection).\
                        filter_by(name=collectionName).\
                        one()
    book.authors = authors
    book.title = title
    book.isbn = isbn
    book.publicationYear = publicationYear
    book.collection = collection

    session.add(book)
    session.commit()


def book_add_author(title, authorFirstName, authorLastName):
    session = Session()

    book = session.query(database.Book).\
        filter(database.Book.title==title).\
        one()

    author = session.query(database.Author).\
        filter(and_(database.Author.firstName==authorFirstName,
database.Author.lastName==authorLastName)).\
        one()

    book.authors.append(author)

    session.commit()

if __name__ == "__main__":
    pass
```

populate_db.py

```
import ORM

ORM.author_create('Nicolas', 'Dupuy')
ORM.editor_create('First', 2001)
ORM.collection_create('First', 'First')
```

```
ORM.book_create('Les 100 meilleurs albums de rock', '978-2-7540-1144-0',
2010, 'First', 'Nicolas', 'Dupuy')
ORM.collection_create('For Dummies', 'First')
ORM.author_create('Mark', 'Phillips')
ORM.author_create('Jon', 'Chappell')
ORM.book_create("La guitare pour les nuls", '2-75400-124-7', 2005, 'For
Dummies', 'Mark', 'Phillips')
ORM.book_add_author('La guitare pour les nuls', 'Jon', 'Chappell')
ORM.book_create('Le Rock pour les nuls', '978-2-7540-0819-8', 2009, 'For
Dummies', 'Nicolas', 'Dupuy')
```

interface.py

```
import ORM

def create_editor():
    editorName = input("Quel est le nom de l'éditeur ?\n")
    fondationYear = input("En quelle année l'éditeur a-t-il était créé ?
\n")
    print(editorName + " " + fondationYear + " ajouté")
    ORM.editor_create(editorName, fondationYear)

def create_collection():
    collectionName = input("Comment s'appelle la collection ?\n")
    editor = input("Quel est l'éditeur de la collection ?\n")
    print (collectionName + " créé par " + editor + " ajoutée")
    ORM.collection_create(collectionName, editor)

def create_author():
    authorFirstName = input("Quel est le prénom de l'auteur/autrice ?\n")
    authorLastName = input("Quel est le nom de l'auteur/autrice ?\n")
    print(authorFirstName + " " + authorLastName + " ajouté·e")
    ORM.author_create(authorFirstName, authorLastName)

def create_book():
    title = input("Quel est le titre du livre?\n")
    isbn = input("Quel est le code ISBN du livre ?\n")
    authorFirstName = input("Quel est le prénom du premier auteur ?\n")
    authorLastName = input("Quel est le nom du premier auteur ?\n")
    publicationYear = input("En quelle année le livre a-t-il été publié ?
\n")
    collection = input("À quelle collection le livre appartient-il ?\n")
    print(title + " ajouté")
    ORM.book_create(title, isbn, publicationYear, collection,
authorFirstName, authorLastName)

def add_author():
    title = input("Titre du livre :\n")
    firstName = input("Prénom de l'auteur :\n")
    lastName = input("Nom de l'auteur :\n")
    ORM.book_add_author(title, firstName, lastName)
```

```
if __name__ == "__main__":

    while (True):
        print("1. Créer un éditeur")
        print("2. Créer une collection")
        print("3. Créer un auteur")
        print("4. Créer un livre")
        print("5. Ajouter un auteur à un livre")
        option = input()

        if (option == "1") : create_editor()
        elif (option == "2") : create_collection()
        elif (option == "3") : create_author()
        elif (option == "4") : create_book()
        elif (option == "5") : add_author()
        else:
            print("Option invalide")
            print("Au revoir")
            break
```

config_db.json

```
{
        "host": "localhost",
        "user": "admin",
        "password": "password",
        "db": "library"
}
```