

Mini-projet : CDI



I. Description

Nous avons vu en cours que l'on peut utiliser Python pour manipuler une base de données à l'aide d'un module, notamment `sqlite3`.

Le but de ce mini-projet est de créer un programme Python qui permettra d'effectuer différentes opérations sur la base de donnée du CDI déjà manipulée en cours et en exercices cette année.

II. Instructions

La base de donnée CDI, déjà utilisée, est à reprendre dans ce mini-projet, qui ne nécessite donc pas la création d'une nouvelle base.

On pourra revoir les documents la concernant pour se rappeler sa structure.

Revoir le cours pour savoir comment importer le module et la base de données.

L'architecture globale du mini-projet devra contenir quelque chose sous la forme suivante (qui sera bien entendu à compléter) :

```
def main():
    stop=False
    while not stop:
        print("", "Choix :", "rechercher un document (d)",
              "emprunter (e)", "rendre (r)",
              "ajouter (a)", "supprimer (s)",
              "rechercher un prêt (p)", "quitter (q)", sep="\n")
        choix=input("Quel est votre choix ? ")
        if choix=="d":
            pass
        elif choix=="e":
            pass
        # ...
        elif choix=="q":
            stop=True
        else:
            print("Je n'ai pas compris...")
    print("\nSortie du programme. Au revoir !")

main()
```

Autrement dit, la fonction principale (**main**) exécutée est une boucle infinie de laquelle on sort uniquement lorsque l'on décide de quitter.

Les différentes actions exécutées pourront être définies dans des fonctions pour alléger le code global. Certaines nécessiteront elles-mêmes des sous-menus.

Les choses suivantes devront être rendues possibles :

- **rechercher un document** : dans cette partie, un sous-menu permettra de rechercher entre autres par **titre**, par **auteur**, par **année** ou par **ISBN**.
On devra préciser les informations sur le documents : titre, auteur, etc.
De plus, s'il est emprunté, le préciser et donner la date de retour.
- **emprunter** : dans cette partie, on permettra l'ajout dans la base de l'emprunt d'un livre par un élève.
On identifiera le livre et l'élève par leurs numéros.
- **rendre** : dans cette partie, on permettra la suppression dans la base de l'emprunt d'un livre par un élève.
Pour aider à cela, on pourra demander le numéro de l'élève, puis lister les livres empruntés et demander de choisir le livre rendu.
- **ajouter** : dans cette partie, un sous-menu permettra d'ajouter un auteur ou un livre. Dans chaque cas, les renseignements nécessaires seront demandés.
- **supprimer** : dans cette partie, un sous-menu permettra de supprimer de la base un livre. Si, ce faisant, son auteur n'apparaît plus comme auteur de livres du CDI, il faudra également supprimer cet auteur (en le précisant).
Le livre sera indiqué par son numéro (ISBN).
- **rechercher un prêt** : dans cette partie, un sous-menu permettra de chercher les prêts d'un élève donné (identifié par son numéro). On précisera pour chaque emprunt le numéro ISBN et la date de retour.

III. Indications supplémentaires

Il faudra penser à faire un code qui soit suffisamment robuste, autrement dit qu'il n'arrête pas sur une erreur à la moindre erreur de l'utilisateur.

Par exemple :

- Si l'utilisateur donne un identifiant erroné/inexistant, lui redemander (ou éventuellement lui permettre d'abandonner).
- Lors de l'ajout d'un livre, vérifier que l'auteur donné existe bien déjà (sinon on pourra éventuellement proposer qu'il soit ajouté en même temps à la base).

IV. Extensions possibles

La liste suivante est non exhaustive :

- Ajouter un système d'authentification, permettant des droits différents selon l'utilisateur (administrateur de la base de donnée ou simple élève) et la simplification de certaines actions.

Les bases de données sqlite ne bénéficient pas de la possibilité d'avoir des utilisateurs et mots de passe pour restreindre les accès.

On pourra cependant faire un système d'identification dans le programme Python (bien que le mot de passe de l'administrateur sera a priori en clair dans le code, à moins de trouver des méthodes plus complexes), et afficher les menus dépendant des droits de l'utilisateur connecté.

L'administrateur devra seulement mettre un mot de passe.

Pour les élèves, on pourra se contenter d'authentifier avec seulement l'identifiant (qui doit donc être dans la base de données).

Authentifier ainsi un élève permet de simplifier les actions d'emprunt et de rendu si on a au préalable enregistré son identifiant dans une variable.

- Pour toutes les identifications nécessaires d'auteurs, de livres ou d'élève, on pourra éventuellement permettre en plus de rechercher par nom ou titre plutôt que d'obliger à donner le numéro.
- Permettre de **modifier** : dans cette partie, un sous-menu permettra de modifier les données sur un auteur ou un livre (pour corriger l'orthographe du nom ou du titre).
Si l'on veut modifier d'autres choses que les noms et titres, attention à la difficulté.
En particulier, la correction d'un numéro (identifiant ou ISBN) entraîne des difficultés liées aux clés étrangères.
- On pourra permettre la recherche précise des livres dont la date de retour a été dépassée.
Toute autre recherche spécifique et pertinente pourra être ajoutée.
Les requêtes un peu complexes seront appréciées.
- Permettre de manipuler également les éditeurs des livres (recherche, modification, etc.).