



Projet BDD Console: Banque

Ce projet peut être réalisé par un groupe de 2 à 3 personnes

Introduction

Le but de ce projet est de mettre en œuvre la gestion d'une base de données SQLite3 grâce à son API pour Python3. La création et le peuplement de la base de données n'est pas à faire avec python, vous pouvez utiliser DBBrowser. Ce que vous devez créer, c'est l'interface(en mode console) qui permettra à un utilisateur lambda de réaliser les opérations CRUD sur la base.

Cahier des charges - Présentation

Vous allez créer une interface utilisateur pour les conseillers d'une banque. Cette banque est composée de clients dont on connaît le nom, le prénom, la date d'inscription, l'adresse mail. Un client peut posséder aucun ou plusieurs comptes courant. Un client peut posséder aucun ou plusieurs comptes épargne. Un compte courant possède un numéro, un solde et un plafond de découvert. Un compte épargne possède un numéro, un solde et un taux d'intérêt.

Création du modèle relationnel et peuplement de la table.

À partir du cahier des charges, construisez un modèle relationnel le plus pertinent possible. Une fois créé, peuplez la table avec des cas imaginaires différents. Pensez à ce que les données soient variées (il faut des gens à découvert, des gens avec peu de comptes des gens aisés...). Si tous vos clients s'appellent Jean Dupont et ont les mêmes types de comptes et les mêmes soldes, ce ne sera pas intéressant. Une base de 8 clients différents possédant chacune entre 0 et 3 ou 4 comptes serait un bon début.

Création de l'interface.

Au lancement de votre programme, voilà ce que pourrait afficher votre terminal:

```
=====
Bienvenu sur l'interface de la banque la NSIBANK :
=====
Quelle action souhaitez vous effectuer ? répondre par 1,2,3 ou 4:
1 - C - Créer un nouvel utilisateur
2 - R - Consulter la liste des clients, des comptes...
3 - U - Effectue des opérations: virement, modifiez les autorisations de
    découvert,...
4 - D - Supprimer un compte, un utilisateur
5 - Quitter l'application
```

À vous d'imaginer la suite.

Ce que votre application doit permettre

- Rechercher un client par son nom et prénom
- Rechercher un client avec son numéro client
- Consulter les comptes courants d'un client
- Consulter les comptes épargnes d'un client
- Modifier l'autorisation de découvert d'un compte.
- Réaliser un virement entre 2 comptes
- Actualiser tous les soldes des comptes épargnes en fonction de leurs taux
- Récupérer la liste des mails des personnes à découvert
- Supprimer un compte ou un utilisateur.
- Votre interface devra être stylisée, exemple de réponse:

```
+=====+
| id | nom      | prenom    | mail                |
+=====+
| 1  | Bond     | James     | 007@mi6.com         |
+-----+-----+-----+
| 2  | Torreto  | Dominic   | dom@fastandfurious.us |
+-----+-----+-----+
```

Indices

Utiliser des fonctions : Prenons l'exemple d'un virement. Il nous faut le numéro du compte à débiter, le numéro du compte à créditer et le montant. on crée la fonction `virement(num_compte_debite, num_compte_credite, montant)` avant de débiter un compte, il faut être sûr qu'il existe, si c'est le cas, on vérifie que le montant est débitable sinon on indique que le solde est insuffisant. S'il est débitable, on le débite puis crédite le compte à créditer. On indique que tout c'est bien passé. Ne faites pas confiance à l'utilisateur ! Chaque opération sur la base peut utiliser une ou plusieurs fonction. Imaginez des fonctions réutilisables.