

Exercice 1 :

Nom	Date de naissance	Spé abandonnée	Nombre d'élèves abandonnant la spé
Tommy	2003	Maths	17
Timmy	2004	Anglais	25
Tammy	2003	SI	72
Robert	2002	Maths	17
Gérard	1994	Anglais	25

- 1) Quel(s) enregistrement(s) correspond(ent) à la date de naissance 2003 ? On utilisera la notation entre crochets.
- 2) En déduire ce qui manque à cette base de données.
- 3) **a.** À l'aide de clé(s) étrangère(s), décomposez cette base de données en deux bases de données.
b. Dans votre nouvelle base, donnez l'enregistrement correspondant à la spé abandonnée "Maths".

Exercice 2 :

On donne les différentes tables d'une base de données d'une bibliothèque :

- ❖ **Etudiant**(NumEtd, NomEtd, AdresseEtd)
- ❖ **Livre**(NumLivre, TitreLivre, NumAuteur, NumEditeur, NumTheme, AnneeEdition)
- ❖ **Auteur**(NumAuteur, NomAuteur, AdresseAuteur)
- ❖ **Editeur**(NumEditeur, NomEditeur, AdresseEditeur)
- ❖ **Theme**(NumTheme, IntituléTheme)
- ❖ **Prêt**(NumEtd, NumLivre, DatePret, DateRetour)

Un étudiant peut emprunter plusieurs livres à la fois. Chaque fois qu'un livre est retourné, la date de retour est mise à jour. Pour des raisons de statistique, on conserve dans une table **Prêt** toutes les informations sur les livres retournés.

- 1) Expliquer pourquoi "**NomAuteur**" n'est pas une bonne clé primaire pour la table **Auteur**.
- 2) Créez la table Prêt avec les conditions suivantes :
 - a. L'étudiant 1 prend le livre 3 le 12/12/2012
 - b. L'étudiant 1 prend le livre 2 le 12/12/2012
 - c. L'étudiant 3 prend le livre 3 le 15/12/2012
 - d. L'étudiant 1 reprend le livre 3 le 01/01/2013

- 3) Donner les clés primaires associées aux tables de cette bibliothèque. Pour la table **Prêt**, on s'aidera de la question 2) et on se rappellera qu'une clé primaire doit identifier de manière unique une entrée : une clé primaire peut donc être un couple (un triplet, etc...) d'identifiants.

Exercice 3 :

Réécrire le premier tableau de la page 2 sous format CSV.

Exercice 4 : réflexion autour de l'importation de fichier CSV

Dans la partie sur le CSV, nous avons utilisé la bibliothèque csv. Nous pouvons également faire l'importation des fichiers csv à la main et comprendre ce qu'il se passe sous le capot de la bibliothèque csv.

L'instruction `donnéesTexte.split(';')` permet de découper des données textuelles sous forme de liste selon un séparateur donné (ici, le ';').

Exemple :

`'les;marmottes;ont;le;covid'.split(';')` nous donne : `['les', 'marmottes', 'ont', 'le', 'covid']`

- 1) Créez une fonction : `def importCSVDuPauvre(fichier : str, separateur = ";")` qui va importer des fichiers CSV et renvoyer une liste de dictionnaire.
Pour l'instant, écrivez simplement `return` à la fin de votre fonction.
- 2) Dans votre fonction, ouvrez en lecture le fichier et chargez son contenu dans une variable appelée `donnéesDuFichier`.
- 3) Grâce à une boucle sur les lignes du fichier CSV, affichez toutes les lignes une à une puis essayez de les transformer en liste... On enregistrera chaque ligne dans une variable appelée `listLigne`.
- 4) Pour créer nos dictionnaires, nous avons besoin de deux choses : les en-têtes et les valeurs.
 - a. Quelle ligne du fichier CSV contient les en-têtes ? Imaginez alors une méthode (ou un test) qui vous permettrait d'obtenir l'en-tête du fichier CSV.
 - b. De la même manière, imaginez une manière d'organiser vos données pour que l'on obtienne des enregistrements du type ci-dessous. Pensez à la manière d'initialiser les dictionnaires :

```
Table = [  
    {'Nom': 'Erwann', 'Français': '16', 'Science': '12', 'Histoire': '15'},  
    {'Nom': 'Céline', 'Français': '14', 'Science': '16', 'Histoire': '13'}  
]
```

Exercice 5 : réflexion libre

En vous aidant de l'exercice 4, créez une fonction permettant de réaliser la même opération que exportCSV (p12) i.e. exporter une liste de dictionnaires vers un fichier CSV.