Introduction aux bases de données

Activité 01

L'objectif de ce TP est de comprendre un peu ce que fait SQL sous le capot : on fera les calculs nous-mêmes en Python.

Partie 1 : statistiques tennistiques

Exercice 1

1. Sur le site `pierrebeaur.fr`, récupérer les fichiers `Athletes.csv` et `Victoires.csv`.
2. **Les placer dans le dossier `Musique`.**
3. Ouvrir le fichier `Athletes.csv` à l'aide d'un éditeur de texte : ce que vous avez est une base de données des tennismen et tenniswomen ayant remporté un tournoi majeur depuis 2000.
4. Ouvrir Pyzo, et dans votre nouveau fichier (fenêtre de gauche) recopier le script suivant :

```
# importation du fichier dans Python
import os
rep = os.getcwd()
fichier = open(rep+"\\Music\\Athletes.csv",encoding="utf-8")
# transformation du fichier en un tableau ligne à ligne
bdd_ath = fichier.readlines()

# transformation des lignes
for i in range(len(bdd_ath)):
    # découpage de chaque ligne selon les virgules
    bdd_ath[i] = bdd_ath[i][:-1].split(sep=",")

    # transtypage des données chiffrées depuis string vers int
    if i > 0:
        bdd_ath[i][0] = int(bdd_ath[i][0])
        bdd_ath[i][5] = int(bdd_ath[i][5])
fichier.close()
```

   Les commandes précédentes ont permis d'ouvrir le fichier CSV précédent et de le transformer en une matrice manipulable en Python.
5. Que vaut `bdd_ath[4]` ? De manière générale, que vaut `bdd_ath[i]` ? (attention, il y a un piège !)

Exercice 2 : questions statistiques simples

On répondra aux questions suivantes exclusivement à l'aide de Python.

1. Combien d'athlètes y a-t-il au total ?
2. Combien de tenniswomen y a-t-il ?
3. Combien y a-t-il d'athlètes américain·e·s ?
4. Combien y a-t-il d'athlètes américains masculins ?
5. Combien d'athlètes sont nés une année bissextile ?
   Rappel : une année bissextile est divisible par 4, mais pas par 100, sauf si divisible par 400.
6. Combien d'athlètes ont un prénom commençant par la lettre "R" ?
7. Combien d'athlètes ont un nom qui ne contient pas la lettre "a" (minuscule ou majuscule) ?
8. Combien d'athlètes ont un prénom dont la longueur divise l'année de naissance ?

Partie 2 : implémentations des fonctionnalités standard de SQL

Exercice 3

1. Écrire une fonction `select(attribut,bdd)` qui prend en paramètre un attribut (par exemple `"prenom"` ou `"nationalite"`) et une matrice, et renvoie un tableau correspondant à la colonne en question. Tester avec `select("prenom",bdd_ath)`.
2. Généralisons le principe précédent pour pouvoir gérer une quantité arbitraire d'attributs.
   a) Écrire une fonction `id_attribut(nom_attribut,bdd)` qui renvoie l'indice associé à `nom_attribut` dans la matrice `bdd`. Par exemple, `id_attribut("nationalite",bdd_ath)` doit renvoyer 4.
   b) Écrire une fonction qui permet de trier un tableau (choix parmi : tri par insertion, par sélection, tri fusion ou tri rapide).
   c) Écrire une fonction `liste_id_attributs(liste_attributs,bdd)` qui renvoie la liste des indices associés aux attributs de la liste (désignés par leurs noms), triés dans l'ordre croissant. Par exemple, `liste_id_attributs(["prenom","nationalite","nom"],bdd_ath)` doit renvoyer `[1,2,4]`.
   d) Écrire une fonction `select_liste(liste_attributs, bdd)` qui prend en paramètre une liste d'attributs et renvoie une matrice correspondant aux colonnes en question. Tester avec `select(["prenom",",nationalite","nom"],bdd_ath)`.
   On fera attention à garder une ligne décrivant les attributs.
3. Écrire une fonction `where(attribut, bdd, f)` qui prend en paramètre une fonction `f` : Dom(`attribut`) $\to$ `bool` qui à toute valeur possible pour `attribut` renvoie `True` ou `False`, et renvoie la matrice extraite de `bdd_vic` dont les lignes vérifient la condition de `f` selon la colonne `attribut`.
   Par exemple, si `f` : $n \mapsto \begin{cases} \texttt{True si } n \geqslant 1990 \\ \texttt{False sinon} \end{cases}$, `where("annee_naissance",bdd_ath,f)` renverra les lignes correspondant aux athlètes nés après 1990.
   On fera attention à garder la ligne des attributs.

Exercice 4

1. Adapter le code fourni précédemment pour créer une matrice `bdd_vic` à partir du fichier `Victoires.csv`.
2. Écrire une fonction `id(nom)` qui prend en paramètre le nom d'un athlète et renvoie son identifiant.
3. Votre fonction précédente a-t-elle un défaut ?
4. Écrire une fonction `nb_victoires(id_ath)` qui prend l'identifiant d'un athlète et renvoie son nombre de victoires. Par exemple : Andy Murray, dont l'id est 9, a remporté 3 tournois majeurs.

Exercice 5

1. Trier la matrice `bdd_ath` en fonction du nombre de victoires de chaque athlète. Plus précisément :
   a) La première ligne doit rester intacte ;
   b) Les athlètes sont triés par ordre décroissant de nombre de victoires (avec toutes leurs informations).
2. Écrire une fonction `limit(n)` qui renvoie les n athlètes les plus médaillés.
3. Écrire une fonction `offset(n,k)` qui renvoie les n athlètes les plus médaillés, en excluant le top k.

Exercice 6

1. Écrire une fonction `max(bdd,attribut)` qui renvoie le maximum de la colonne `attribut` dans la matrice `bdd`.
2. En déduire l'âge minimal d'un athlète de la BDD.
3. Déterminer, à l'aide de la fonction `where`, l'athlète australien·ne la plus jeune de la BDD.

Exercice 7

1. Tester la commande `select_liste(["nationalite"],bdd)` : vous devriez observer des répétitions.
2. Écrire une fonction `sans_doublons(bdd)` qui renvoie `bdd` sans doublons.

Partie 3 : Pour aller plus loin

Exercice 8

L'objectif de cette section est d'observer en direct ce qu'est un tri stable.

1. Écrire une fonction `tri_selection(attribut,bdd)` qui trie dans l'ordre croissant une matrice selon la colonne `attribut` en utilisant le tri par sélection.
2. Écrire une fonction `tri_rapide(attribut,bdd)` qui trie dans l'ordre croissant une matrice selon la colonne `attribut` en utilisant le tri rapide.
3. Tester de trier par les prénoms, puis de trier par les noms en utilisant les deux méthodes (important : il faut recompiler entre deux tests, car le tri par sélection est en place). Qu'observez-vous sur la situation des sœurs Williams ?

On dit que le tri par sélection est un tri stable : une fois qu'on a trié selon un attribut A, en triant selon un autre attribut B, deux enregistrements égaux selon B resteront triés selon A. Le tri rapide n'est pas un tri stable.

4. Expliquer la situation de Juan Martin del Potro.

Exercice 9 : pour les 5/2

1. Écrire une fonction `join(categorie)` qui renvoie la jointure de `bdd_ath` et `bdd_vic` selon `id_ath` = `categorie` (par exemple, on pourrait avoir `categorie` = `rg_f` ou `wim_h`).
2. Supposons qu'on veuille avoir une table réunissant toutes les informations : combien de colonnes disposeraient-elles ? Proposer une requête SQL qui permettrait d'obtenir une telle table.

Exercice 10

Le sujet de TP d'aujourd'hui est inspiré du sujet X-ENS MP-PC-PSI Informatique B 2018. Si vous êtes arrivés jusqu'ici, vous pouvez commencer à faire ce sujet.

Pour la semaine prochaine :

— Déterminer les complexités des différentes fonctions étudiées en fonction des dimensions de la base de données.

— Dans l'exercice 2, question 1 à 5 : proposer des requêtes SQL correspondant.
Pour obtenir un modulo en SQL, la syntaxe est `MOD(dividende,diviseur)`.

— Proposer une requête SQL correspondant à l'exercice 6 question 3.