

Première NSI
Chapitre XII - Les tables

I. Indexation de tables

I.1. Généralités

Définition

Une table est un tableau de p -uplets nommés qui partagent les mêmes **descripteurs** (ou **attributs**) : chaque champ de chaque p -uplet a le même nom.

Chaque p -uplet nommé est appelé **enregistrement** de la table.

Remarque

On peut stocker des tables dans des fichiers. Le format que nous utiliserons sera le format CSV (Comma Separated Values).

I.2. Récupérer une table à partir d'un fichier CSV

Un fichier CSV est un fichier dans lequel il y a une table.

En général, la première ligne de la table correspond aux descripteurs.

Les lignes suivantes correspondent aux données.

Les lignes sont séparés par le caractère `\n` (retour chariot).

Chaque champ est séparé par une tabulation, une virgule ou un point virgule (cela dépend du fichier).

```
1 le_fichier = open(nom_fichier, 'r')
2 les_donnees = le_fichier.read()
```

Ainsi, la variable `les_donnees` contient tous les objets contenus dans le fichier CSV, y compris les descripteurs, sous forme d'une chaîne de caractères.

Il faut donc créer une liste des différents enregistrements :

```
1 liste_enr = les_donnees.split('\n')
```

`liste_enr` est une liste de tous les enregistrements. Ces derniers sont des p -uplets. Le premier p -uplet est la collection des descripteurs. Il serait judicieux d'enregistrer dans une variable les descripteurs et de supprimer le premier élément de `liste_enr` ainsi que le dernier, qui est souvent un retour à la ligne.

```
1 descripteurs = tuple(liste_enr[0].split(';'))
2 liste_enr(0)
3 liste_enr(len(liste_enr)-1)
```

Il faut maintenant créer des p -uplets nommés.

```
1 listep_uplets_nommes = []
2 for puplet in liste_enr:
3     un_enr={}
4     les_valeurs= puplet.split(';')
5     for i in range(len(descripteurs)):
6         un_enr[descripteurs[i]] = les_valeurs[i]
7     listep_uplets_nommes.append(un_enr)
```

La liste `listep_uplets_nommes` contient des p -uplets nommés.

II. Recherche dans une table

Pour rechercher les lignes dans une table vérifiant que le champ est égal à une certaine valeur, on fait comme suit :

```
1 def rechercher(table, champ, valeur):
2     """
3     renvoie une table d'enregistrements
4     :param table: (list) une table (= liste de p-uplets nommés)
5     :return: (list) une table contenant uniquement des enregistrements (p-uplets nommés)
6     """
7     resultat = []
8     for element in table:
9         if element[champ] == valeur : #condition sur les champs
10            resultat.append(element)
11     return resultat
```

III. Tri d'une table

Définition

Une fonction lambda est une fonction qui s'écrit en une ligne. Le `return` est implicite. Elle s'écrit ainsi :

```
1 nom_fonction = lambda paramètre(s): renvoi
```

Exemple

```
1 >>> carre = lambda x: x * x
2 >>> carre(4)
3 16
```

Pour trier une table selon une colonne :

```
1 def trier(table, champ):
2     return sorted(table, key=lambda element: element[champ])
```

Remarque

La fonction `sorted` renvoie un tableau trié selon la clé fournie.
Pour notre exemple, la clé fournie est la valeur du champ, donnée par une fonction lambda.

IV. Fusion de tables

Définition

Fusionner deux tables, c'est regrouper les données de ces deux tables.
On peut distinguer deux types de fusion de tables :

- La concaténation de tables
- La jointure de tables

Définition

On appelle **domaine de valeur** d'un attribut l'ensemble dans lequel les valeurs de l'attribut sont prises.
Par exemple l'ensemble des nombres entiers, l'ensemble des chaînes de caractères, etc.

IV.1. Concaténation de deux tables

Propriété

Pour concaténer deux tables, il faut que les deux tables aient les mêmes descripteurs. De plus, ces descripteurs doivent avoir le même domaine de valeur. Ensuite, en Python, il suffit d'utiliser le symbole `+` :

```
1 table1 + table2
```

Exemple

On veut concaténer les deux tables suivantes :

id	jeux_video_prefere
1	Brawlstar
2	Minecraft
3	Spider-man : Miles Morales
4	Tales of Arise

 et

id	jeux_video_prefere
1	Elder Ring
2	Fifa 22
3	Minecraft
4	Ghostwire

Les deux tables ont les mêmes descripteurs donc on peut les concaténer :

id	jeux_video_prefere
1	Brawlstar
2	Minecraft
3	Spider-man : Miles Morales
4	Tales of Arise
1	Elder Ring
2	Fifa 22
3	Minecraft
4	Ghostwire

IV.2. Jointure de deux tables

Propriété

Pour joindre deux tables, il suffit qu'elles aient un descripteur commun qui soit unique pour chaque enregistrement. De plus, ces deux descripteurs doivent avoir le même domaine de valeur.

Exemple

On veut joindre les deux tables suivantes :

id	sexe	couleur_preferee
1	M	rouge
2	M	Vert
3	F	rouge
4	F	bleu

 et

id	jeux_video_prefere
1	Elder Ring
2	Fifa 22
3	Minecraft
4	Ghostwire

On remarque que le descripteur `id` prend des valeurs unique dans chaque table. Donc on peut joindre les deux tables grâce à ce descripteur pour obtenir la table :

id	sexe	couleur_preferee	jeux_video_prefere
1	M	rouge	Elder Ring
2	M	Vert	Fifa 22
3	F	rouge	Minecraft
4	F	bleu	Ghostwire

V. Exercices

V.1. Exercice 1

Le fichier `liste_villes_francaises_2015.csv` est un fichier CSV qui contient des informations sur les villes françaises. Dans tout cet exercice, une table, tableau de p -uplets nommés, sera implémentée par un tableau de dictionnaires.

1. Créer une fonction `renvoi_table(nom_fichier)` qui renvoie la table des informations sur les villes françaises.
2. Créer une fonction `rechercher(table, champ, valeur)` qui renvoie la table des enregistrements dont le descripteur `champ` prend la valeur `valeur`.
3. Créer une fonction `trier(table, champ)` qui renvoie la table `table`, triée selon les valeurs du descripteur `champ`.
4. Utiliser ces trois fonctions pour renvoyer le tableau des noms de villes triées par ordre alphabétique dont le code postal est 62223.

V.2. Exercice 2

On donne les deux tables contenues dans les fichiers `albums.csv` et `auteurs.csv`.

Dans tout cet exercice, une table, tableau de p -uplets nommés, sera implémentée par un tableau de dictionnaires.

1. Récupérer la table du fichier `albums.csv` dans la variable `albums` et celle du fichier `auteurs.csv` dans la variable `auteurs`.
2. Créer la fonction `supprime_doublets(table)` qui supprime les doublons (enregistrements identiques à un enregistrement existant) présents dans la table.
3. Créer la fonction `joindre(table1, table2, attribut1, attribut2)` qui joint les tables `table1` et `table2` selon les attributs `attribut1` de `table1` et `attribut2` de `table2`.
4. Créer la jointure des deux tables `albums` et `auteurs` : les descripteurs qui permettent de joindre les deux tables sont `auteur` de la table `albums` et `id` de la table `auteurs`.