Données en table

fusion de tables

Nous avons vu comment chercher et trier des données sur un fichier CSV. Il est très courant d'avoir besoin de prendre les données provenant de plusieurs tables et de les regrouper pour former une nouvelle table. C'est ce qu'on appelle **fusionner** des tables. De plus, c'est une approche de la partie base de données du programme de terminale.

On va d'abord étudier un exemple à la main puis de l'implémenter par un programme.

I Fusion à la main

Une entreprise stocke dans une table ses clients et dans une autre table les commandes.

La table client

n_client	nom	prenom	ville
1212	Lacasse	Aubrey	Annecy
1343	Primeau	Angelette	Tours
2454	Gabriaux	Julie	Bordeaux
895	Gaulin	Élodie	Lyon
2324	Jobin	Dorene	Bourges
34	Boncoeur	Kari	Nantes
1221	Parizeau	Olympia	Metz
1114	Paiement	Inès	Bordeaux
3435	Chrétien	Adèle	Moulin
5565	Neufville	Ila	Toulouse
2221	Larivière	Alice	Tours

La table commande

n_commande	date	n_client
1111	22/09/18	1343
2323	20/03/19	34
987	12/09/15	5565
454	08/07/14	2324
1324	01/02/17	4444
1567	05/12/18	2221
45	02/02/12	2454
123	04/11/13	5565
2122	12/02/19	3435
1989	04/12/18	1212

On ne peut fusionner des tables que si elles ont un attribut commun.

À faire vous-même 1

Quel est l'attribut commun à ces deux tables ?

On peut fusionner les deux tables de deux façons.

- Soit en partant de la table client et ajouter les données de la table commande ce qui correspond à la liste des clients qui ont passé une commande.
- Soit en partant de la table commande et ajouter les données de la table client ce qui correspond à la liste des commandes à laquelle ont a ajouté les données client.

À faire vous-même 2

Compléter le tableau suivant qui devra correspondre à la liste obtenue par la fusion de la table client avec la table commande.

On suivra les règles suivantes :

- Si un client a une commande, on complète le tableau
- Si un client a plusieurs commandes, il y a autant de lignes que de commandes associées.

n_client	nom	prenom	ville	n_commande	date

À faire vous-même 2 bis

Compléter le tableau suivant qui devra correspondre à la liste obtenue par la fusion de la table commande avec la table client.

n_commande	date	n_client	nom	prenom	ville

À faire vous -même 3

Vous avez peut-être remarqué que Mme Élodie Gaulin (n° de client 895) bien que présente dans la table client, est absente des deux tableaux précédents. Pourquoi d'après vous ?

À faire vous – même 4

De même la commande n° 1324 du 01/02/2017 est absente dans les deux tableaux précédents, pourquoi d'après vous ?

II Fusion de tables en Python

À faire vous-même 5

Télécharger l'archive fichiers.zip. Extraire son contenu. Vous devez obtenir trois fichiers ; csv fiches_client.csv et fiches_com.csv et fusion_nsi.py. Copiez-les dans votre répertoire de travail. Par exemple dan un dossier nommé« NSI-fusion ».

Tester ce code en ajoutant les lignes suivantes au fichier Python :

```
clients = charger_fichier("fiches_client.csv")
commandes = charger_fichier("fiches_com.csv")
print("Tableau clients")
affiche_table(clients)
print("Tableau commandes")
affiche_table(commandes)
```

Il devrait afficher la table client et la table commande de la première page.

On veut maintenant écrire une fonction qui fusionne deux tables. Elle devra prendre en paramètre deux tables représentées par des listes de dictionnaires et une chaîne de caractères correspondant au nom de l'attribut commun. Comme il se peut que l'attribut n'ai pas le même nom dans le deux table (par exemple n_client pourrait, par exemple, être customer_id dans une autre table), un paramètre optionnel permet de donner les deux noms. Nous n'en aurons pas besoin dans notre exemple.

Voici la fonction qui permet de faire cette fusion, on parle de jointure de tables. Le joint étant l'attribut commun.

```
def jointure(table1,table2,cle1,cle2 = None):
    """
    list[dict]*list[dict]*string(*string) -> list[dict]
    """
    if cle2 is None:
        cle2 = cle1
    table = []
    for ligne1 in table1:
        for ligne2 in table2:
            if ligne1[cle1] == ligne2[cle2]:
                ligne= deepcopy(ligne1)
                for cle in ligne2:
                    if cle != cle2:
                        ligne[cle] = ligne2[cle]
```

```
                table.append(ligne)
    return table
```

Un peu d'aide pour comprendre cet algorithme :

Il y a deux boucles imbriquées.

On crée une table vide nommé table.

On lit les éléments de la table1 un par un.

Pour chacun d'entre eux, on lit tous les éléments de la table2.

Si la valeur de la clé dans table1 correspond à celle de la clé dans table2:

On copie dans ligne le contenu de l'élément courant de table1 auquel on ajoute les éléments de table2 qui n'y sont pas dans ligne

On ajoute ligne à table.

On retourne table.

À faire vous-même 6

Vérifier votre code en ajoutant les lignes suivantes, vous devriez retrouver les deux tableaux de la partie I.

```
clients_commandes = jointure(clients,commandes,'n_client')
print("Tableau jointure clients commandes")
affiche_table(clients_commandes)

commandes_clients = jointure(commandes,clients,'n_client')
print("Tableau jointure commandes clients")
affiche_table(commandes_clients)
```

```
À faire vous-même 7
Quel est l'intérêt selon-vous d'avoir utilisé deux tables plutôt qu'une qui
contiendrait les données fusionnées?
```

Sources :

https://pixees.fr/informatiquelycee/n_site/nsi_prem_traitCSV.html