

Données en table et programmation

L'année dernière en SNT, vous avez déjà rencontré les données structurées en table mais les avez principalement utilisées avec un tableur. Cette année, nous allons nous concentrer sur leur utilisation à travers à un programme informatique.

I. DONNÉES EN TABLE

Une **table** contient des données composées construites suivant le même modèle. Chaque ligne s'appelle un **enregistrement**. Chaque colonne possède un titre que l'on appellera **descripteur** (ou propriété) et est associée à un type de données simples (booléen, entier, flottant, texte, date ou heure).

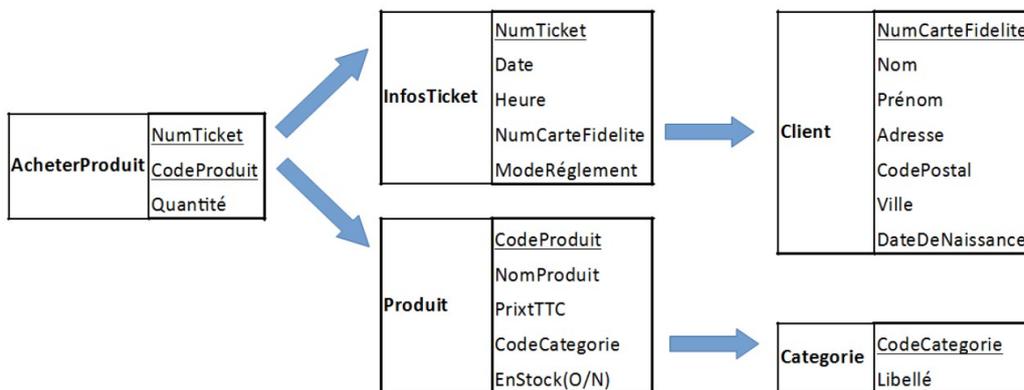
Une **clé primaire** est un descripteur suffisant pour identifier un enregistrement dans une table.

Exemple 1: la table ci-contre permet de stocker les informations de plusieurs tickets de caisse (un par ligne).

- 1) Quel est le type du descripteur NumCarteFidelite
-
- 2) Quelle est la clé primaire ?
-

NumTicket	Date	Heure	NumCarteFidelite	ModeRéglement
2132	19/10/2019	14 :32	900000001	CB
2133	19/10/2019	14:40	900000002	ESPECE
2134	19/10/2019	15:10	900000024	CHEQUE
3143	20/10/2019	09 :22		ESPECE
3144	20/10/2019	14:40	900000142	CB
6192	21/10/2019	11 :15	900000023	CHEQUE
7193	22/10/2019	12:12	900000142	CB

Remarque : L'utilisation de plusieurs tables permet d'éviter des répétitions.



Lorsqu'il y a beaucoup de données, les entreprises utilisent un **SGBD** (Système de Gestion de Base de Données). Chaque SGBD ayant son propre mode de fonctionnement, il existe des standards pour exporter et importer les données. Le plus courant est le format CSV.

Un **fichier CSV** représente une seule table. La première ligne liste les descripteurs. Chaque autre ligne représente un enregistrement. Le séparateur le plus souvent utilisé est la virgule mais on rencontre fréquemment le point virgule.

```
CodeCategorie;Libellé
1;Alimentaire
2;Hors Alimentaire
```

Exemple 2: Écrire les deux premières lignes du fichier CSV de la table NumTicket

.....

.....

L'union européenne a mis en place le RGPD (Règlement Général sur la Protection des Données) pour nous protéger contre les risques liés à un usage abusif des données collectées.

II. PROGRAMMATION

Dans un programme informatique, le contenu d'un fichier CSV pourra être chargé dans **une liste de p-uplets nommés dont les clés sont les descripteurs de la table.**

En Python, on utilise le module csv.

Liste dans laquelle sera chargée la table

```
import csv
films = []
with open("films.csv", 'r') as data:
    lecteur = csv.DictReader(data, delimiter=";")
    films = [dict(row) for row in lecteur]
```

Nom du fichier contenant la table à charger

La méthode DictReader permet de préparer la conversion des lignes en dictionnaires

Variable de boucle (la même)

II.1. FILTRE

Appliquer un **filtre** sur une table revient à rechercher les lignes d'une table vérifiant des critères exprimés en logique propositionnelle.

En python, la construction par compréhension des listes simplifie grandement l'écriture des filtres.

```
filmsLongs = [ ligne for ligne in films if int(ligne['Length']) > 120 ]
```

Variable qui contiendra la table filtrée

Variable de boucle (la même tant que l'on ne fait ni projection ni fusion)

Liste de dictionnaires dans laquelle est chargée la table

Critère de sélection des lignes. S'il y en a plusieurs, les séparer par AND ou OR suivant l'effet voulu.

II.2. PROJECTION

On dit que l'on fait une **projection** sur certaines colonnes d'une table lorsque l'on ne garde que celles-ci.

En Python, nous utiliserons là-encore la construction par compréhension des listes.

```
title_genre = [ {'Title':lg['Title'], 'Subject':lg['Subject']} for lg in films ]
```

Variable qui contiendra la projection

Définition des couples clé-valeurs à conserver pour chaque ligne avec la variable de boucle

Variable de boucle

Liste de dictionnaires dans laquelle est chargée la table

II.3. TRI

Il arrive souvent que l'on veuille **trier les enregistrements d'une table** par rapport aux valeurs d'une des colonnes de cette table.

En Python, nous utiliserons la fonction `sorted` avec l'argument `key` qui doit être une fonction renvoyant la valeur utilisée pour le tri.

```
def pop( film : dict ) -> int :
    return int( film['Popularity'] )

filmsTries = sorted( films, key = pop, reverse = True )
```

Variable qui contiendra la table triée suivant le critère choisi

Liste de dictionnaires dans laquelle est chargée la table

Uniquement pour un tri décroissant

Définition de la fonction permettant d'extraire la valeur qui est à comparer pour le tri

II.4. FUSION

Parfois, la recherche d'informations que l'on souhaite faire nécessite de croiser des tables ayant des descripteurs communs. On dit que l'on fait une **fusion**.

En Python, nous utiliserons là-encore la construction par compréhension des listes.

```
Ticket82 = [ {'Nom':p['NomProduit'], 'Q':ap['Quantite']} for ap in acheterProduit for p in produit if ap['CodeProduit'] == p['CodeProduit'] ]
```

Variable qui contiendra la fusion

Définition des couples clé-valeur à conserver

Les deux colonnes qui doivent être égales pour les lignes à fusionner

Variables contenant les tables à fusionner