Données structurées et leur traitement

Nous produisons de plus en plus de données numériques. Comment conserver "intelligemment" ces données pour ensuite les exploiter de manière efficace ?

1 Données numériques

Des données numériques sont produites de plus en plus ces dernières années.
Voici quelques exemples :

1. Météo-France a disposé à certains endroits à la surface des mers du Globe des sondes qui mesurent toutes les heures des grandeurs physiques (température ,pression ,hauteur de vague, etc...) qui sont ensuite **numérisées** et stockées dans des serveurs
2. Chaque utilisateur d'un smartphone du fait même qu'il soit géolocalisable produit des données numériques

Le terme "Open Data" signifie que certaines entreprises laissent "une partie " de leurs données numériques en libre accès par exemple Météo-France ou le ministère de l'Intérieur, sous condition d'une licence d'utilisation. (Voir TP)

Ces données consultables sont souvent des fichiers .csv (pour comma separated value) autrement dit un fichier texte brute où les données sont sommairement décrites par des descripteurs puis affichées sans mise en forme

Nous verrons en T.P des exemples de tels fichiers .csv par exemple les accidents de la route sur toute la France en 2018 (fichier fourni par le ministère de l'Intérieur)`https://www.data.gouv.fr/fr/datasets/base-de-donnees-accidents-corporels-de-la-circulation/`

On peut ouvrir de tels fichiers par un tableur mais il est préférable pour traiter efficacement les données utiliser un langage de programmation

2 Listes

Les données sont stockées sous forme de tables. Sommairement une table est un ensemble de lignes et chaque ligne a le même nombre de colonnes (tableur)
Voici par exemple les premières lignes de la table des accidents de la route en France en 2018 (Au moins 130 000 lignes) La première ligne contient les descripteurs de la table ainsi par exemple an_nais désigne l'année naissance de la personne accidentée. On observe aussi des données manquantes (cercle)

Python a une bibliothèque csv qui permet de lire les fichiers .csv pour pouvoir ensuite faire un traitement statistique. La table est transformée en Python dans une structure donnée appelée **liste**
Il est remarquable de "tenir" 5 Mo de données avec une seule étiquette la variable référençant ces données en mémoire

En connaissant quelques éléments sur les listes en Python on peut faire des calculs statistiques sur ses données

Num_Acc	place	catu	grav	sexe	trajet	secu	locp	actp	etatp	an_nais
201800000001	1	1	3	1	0	11	0	0	0	1928
201800000001	1	1	1	1	5	11	0	0	0	1960
201800000002	1	1	1	1	0	11	0	0	0	1947
201800000002		3	4	1	0	2	2	3	1	1959
201800000003	1	1	3	1	5	21	0	0	0	1987
201800000003	1	1	1	1	0	3	0	0	0	1977
201800000004	1	1	3	1	5	2	0	0	0	2013
201800000004	1	1	1	1	5	11	0	0	0	1982
201800000005	1	1	4	1	5	21	0	0	0	2001
201800000005	1	1	1	1	5	11	0	0	0	1946
201800000006	1	1	1	1	0	3	0	0	0	1984

```
['Num_Acc', 'place', 'catu', 'grav', 'sexe', 'trajet', 'secu', 'locp', 'actp', 'etatp', '
_nais', 'num_veh']
['201800000001', '1', '1', '3', '1', '0', '11', '0', '0', '0', '1928', 'B01']
['201800000001', '1', '1', '1', '1', '5', '11', '0', '0', '0', '1960', 'A01']
['201800000002', '1', '1', '1', '1', '0', '11', '0', '0', '0', '1947', 'A01']
['201800000002', '', '3', '4', '1', '0', '2', '2', '3', '1', '1959', 'A01']
['201800000003', '1', '1', '3', '1', '5', '21', '0', '0', '0', '1987', 'A01']
['201800000003', '1', '1', '1', '1', '0', '3', '0', '0', '0', '1977', 'C01']
['201800000004', '1', '1', '3', '1', '5', '2', '0', '0', '0', '2013', 'B01']
['201800000004', '1', '1', '1', '1', '5', '11', '0', '0', '0', '1982', 'A01']
['201800000005', '1', '1', '4', '1', '5', '21', '0', '0', '0', '2001', 'A01']
['201800000005', '1', '1', '1', '1', '5', '11', '0', '0', '0', '1946', 'B01']
['201800000006', '1', '1', '1', '1', '0', '3', '0', '0', '0', '1984', 'A01']
```

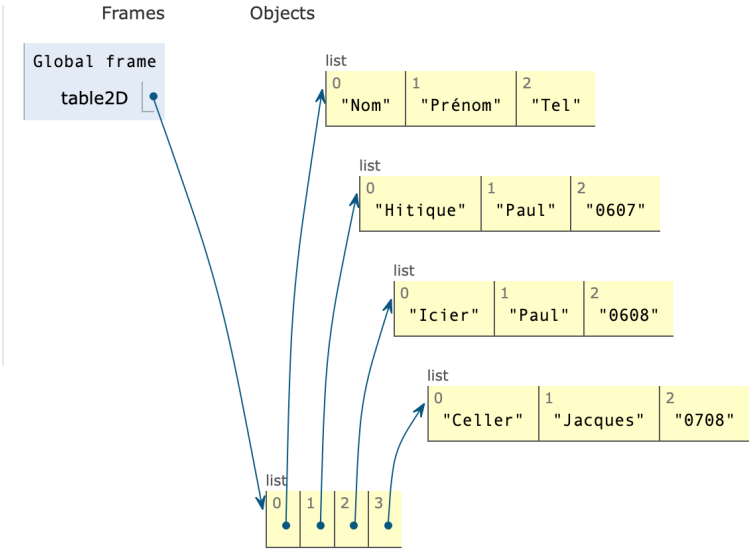
Pour mieux comprendre cette notion de liste on va créer une table plus simple contenant 3 descripteurs :
'Nom','Prénom','Tel' et 4 lignes

Nom	Prénom	Tel
Hitique	Paul	0607
Icier	Paul	0608
Celler	Jacques	0708

Voici comment on code ce tableau en Python

```
table2D = [['Nom','Prénom','Tel'],
['Hitique','Paul','0607'],
['Icier','Paul','0608'],
['Celler','Jacques','0708']]
```

Voici comment on visualise ce tableau en mémoire avec PythonTutor



`table2D` est référence (ou une adresse représentée par une flèche) vers d'autres références `table2D[0]`,`table2D[1]`,`table2D[2]` et `table2D[3]`.
`table2D[1]` est une référence de la liste ['Hitique',Paul','0607']
`table2D[1][2]` permet d'avoir accès au numéro de téléphone de la ligne 1, c'est à dire '0607'

En TP nous aurons à parcourir une colonne et compter le nombre de fois où apparaît une donnée précise, par exemple on pourrait vouloir compter le nombre de fois ou apparaît le prénom Paul :

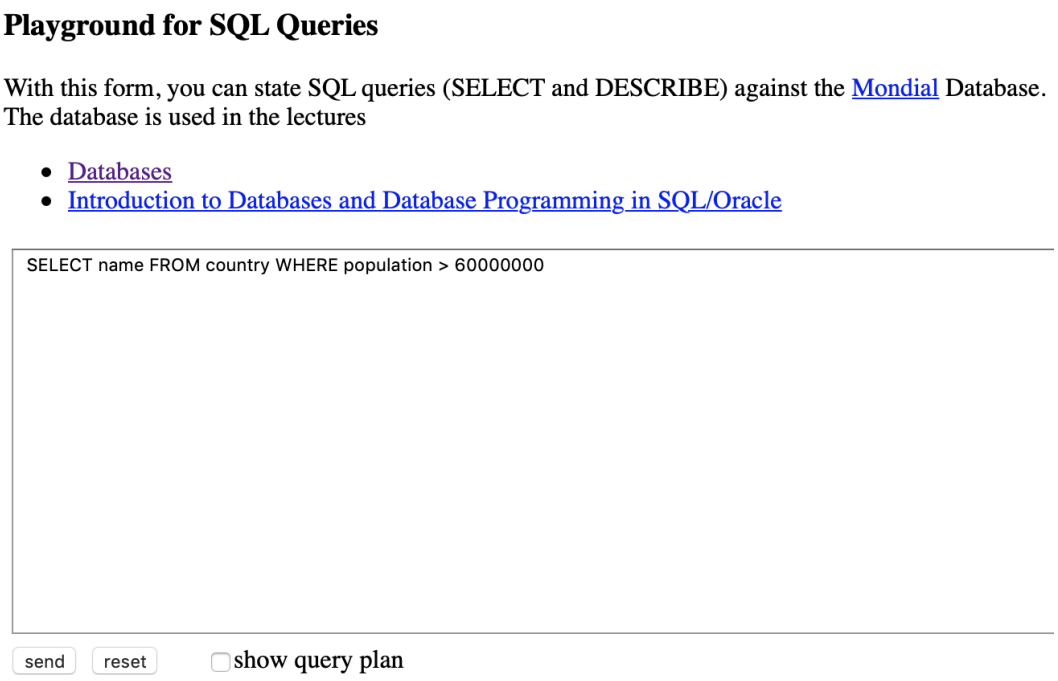
```
compteur = 0
for ligne in table2D:
```

```
    if ligne[1] == 'Paul':
        compteur = compteur + 1
```

Attention ! au type des données la plupart du temps dans les fichiers .csv les données sont du type **chaîne de caractères** , attention aussi aux données manquantes !

3 Bases de données

Au lieu d'avoir une table avec de nombreux champs on s'est rendu compte qu'il est préférable d'avoir plusieurs tables que l'on peut, si besoin est ,fusionner.
Une **base de données** est pour simplifier un ensemble de tables que l'on peut consulter à l'aide d'un **système de gestion de base de données** (SGBD). Sans entrer dans les détails voir ici un exemple en ligne `http://www.semwebtech.org/sqlfrontend/` où dans un formulaire on peut entrer une requête SQL .



Par exemple on a entré :
SELECT name FROM country WHERE population > 60000000
Ce qui signifie dans la table country on sélectionne dans la colonne name les éléments dont la population est supérieure à 60 000 000
On a obtenu les 22 résultats suivants

Results: 22

NAME
France
Germany
Russia
Turkey
United Kingdom
China
Iran
Pakistan
Bangladesh
India
Thailand
Vietnam
Egypt
Indonesia
Japan
Philippines
Mexico
United States
Brazil
Zaire
Nigeria
Ethiopia

4 Enjeux sociétaux

"Nous vivons dans un monde de plus en plus dématérialisé. Nous payons nos impôts en ligne, regardons nos séries préférées en streaming, stockons nos milliers de photos dans le cloud... Dématérialisé, vraiment ? « Si l'on considère la totalité de son cycle de vie, le simple envoi d'un mail d'1 mégaoctet (1 Mo) équivaut à l'utilisation d'une ampoule de 60 watts pendant 25 minutes, soit l'équivalent de 20 grammes de CO2 émis », rappelle Françoise Berthoud, informaticienne au Gricad1 et fondatrice en 2006 du groupement de services EcoInfo – pour une informatique plus respectueuse de l'environnement. Car les mots des nouvelles technologies sont trompeurs : ils évoquent l'immatériel comme le mot « virtuel », l'éthéré comme le mot « cloud », ou encore la pureté comme l'expression de « salle blanche ». Et nous font oublier un peu vite les millions d'ordinateurs et de smartphones, les milliers de data centers et de kilomètres de réseaux utilisés pour traiter et acheminer ces données. Et la quantité considérable d'énergie qu'ils engloutissent. « Le secteur des nouvelles technologies représente à lui seul entre 6 et 10 % de la consommation mondiale d'électricité, selon les estimations – soit près de 4 % de nos émissions de gaz à effet de serre, assène Françoise Berthoud. Et la tendance est franchement à la hausse, à raison de 5 à 7 % d'augmentation tous les ans. »

Environ 30 % de cette consommation électrique est imputable aux équipements terminaux – ordinateurs, téléphones, objets connectés –, 30 % aux data centers qui hébergent nos données et, plus surprenant, 40 % de la consommation est liée aux réseaux, les fameuses « autoroutes de l'information ». « Beaucoup de gens pensent que les réseaux sont des tuyaux « passifs », mais ils sont constellés d'antennes et de routeurs, les aiguillages de l'Internet », explique Anne-Cécile Orgerie, chercheuse en informatique à l'Irisa (Institut de recherche en informatique et systèmes aléatoires). Tous ces équipements sont très gourmands en énergie : un simple routeur consomme 10 000 watts (10 kW), un très gros data center frise carrément les 100 millions de watts (100 MW), soit un dixième de la production d'une centrale thermique ! « Un processeur, c'est comme une résistance. Presque toute l'électricité qu'il consomme est dissipée en chaleur, détaille la chercheuse. C'est pourquoi, en plus de consommer de l'énergie pour faire tourner ses serveurs, un data center doit être climatisé afin de préserver l'intégrité des circuits électroniques. »(Laure Cailloce , Journal du CNRS - 2018)