Manipulation de listes et de tuples en Python : application aux bases de données relationnelles

Dans ce projet vous devrez rédiger des fonctions Python afin d’interroger une base de données.

Cette base de données donne des informations à propos des villes et des pays de notre planète.

# Données

La base est fournie sous la forme de trois fichiers csv : villes.csv, pays.csv, langues.csv.

Le fichier villes.csv contient 4080 lignes, une ligne décrivant une ville avec dans l’ordre quatre informations :

* Identifiant (numérique) : l’identifiant unique de la ville
* Nom (chaine de caractères) : le nom de la ville
* CodePays (chaine de caractères) : l’identifiant du pays de la ville (qui se retrouve dans le fichier pays.csv)
* Etat (chaine de caractères) : le nom de l’état, du district ou du département dans lequel se trouve la ville

Le fichier pays.csv contient 239 lignes, une ligne décrivant un pays avec dans l’ordre quinze informations :

* Code (chaine de caractères) : le code identifiant de manière unique le pays
* Nom (chaine de caractères) : le nom du pays
* Continent (énuméré) : le nom du continent où se situe le pays. Les continents prennent leur nom dans la liste: Asia, Europe, North America, Africa, Oceania, Antarctica, South America
* Region (chaine de caractères) : le nom de la région dans laquelle se situe le pays
* Aire (reel) : l’aire du pays
* AnneeIndependance (entier) : l’année d’indépendance du pays
* Population (entier) : la taille de la population du pays
* EsperanceVie (reel) : l’espérance de vie dans le pays
* GNP (reel) : estimation de la production de produits et de services
* GNPAncien (reel) : idem, valeur ancienne, peut être de valeur NULL
* NomLocal (chaine de caractères) : nom local donné au pays
* FormeGouvernement (chaine de caractères) : forme de gouvernement du pays
* ChefEtat (chaine de caractères) : nom du chef de l’état (à l’époque de la constitution de la base)
* Capitale (entier) : identifiant de la capitale. Cet identifiant correspond à une ville dans le fichier villes.csv
* Code2 (chaine de caractères) : autre code pour le pays

Le fichier langues.csv décrit les langues parlées dans chaque pays. Le fichier comprend 984 lignes, chaque ligne donnant une information sur une langue parlée dans un pays. Chaque ligne contient 4 informations avec dans l’ordre :

* CodePays (chaine de caractères) : le code du pays concerné, qu’on retrouve en champ 1 du fichier pays.csv
* Langue (chaine de caractères) : la langue parlée dans le pays
* Officiel (énuméré valant T ou F) : information précisant si la langue concernée est la langue officielle (T) du pays ou non (F)
* Pourcentage (réel) : pourcentage de la population parlant cette langue

A noter que pour ce dernier fichier, une ligne est identifiée de manière unique par le couple (CodePays, Langue) et que dans un pays, plusieurs langues peuvent être parlées.

# Modélisation

Une ligne sera stockée dans un tuple.

Un fichier sera stocké dans une liste de tuples. Une telle liste sera appelée table par la suite.

# Chargement et visualisation de l’information

Créez une fonction qui prend en argument le nom d’un fichier et qui renvoie la liste de tuples correspondant au contenu du fichier.

Vous devrez peut-être revoir les problèmes liés au codage des caractères et utiliser la bibliothèque codecs (cela dépend en fait de votre système d’exploitation).

Utilisez trois fois cette fonction afin de charger les trois fichiers dans une liste chacun.

Pour votre culture personnelle, vous pouvez étudier le potentiel de la bibliothèque pprint pour pouvoir afficher ces longues listes de manière lisible. Ne rendez pas compte de cette étude dans votre rapport.

Indépendamment de pprint, créez une fonction afficher\_table qui affiche le contenu d’une table sous un format ASCII dont voici un modèle :

+---+---------------+----+--------------+

|1 |Kabul |AFG |Kabol |

+---+---------------+----+--------------+

|2 |Qandahar |AFG |Qandahar |

+---+---------------+----+--------------+

|3 |Herat |AFG |Herat |

+---+---------------+----+--------------+

|4 |Mazar-e-Sharif |AFG |Balkh |

+---+---------------+----+--------------+

|5 |Amsterdam |NLD |Noord-Holland |

+---+---------------+----+--------------+

|6 |Rotterdam |NLD |Zuid-Holland |

+---+---------------+----+--------------+

|7 |Haag |NLD |Zuid-Holland |

+---+---------------+----+--------------+

|8 |Utrecht |NLD |Utrecht |

+---+---------------+----+--------------+

|9 |Eindhoven |NLD |Noord-Brabant |

+---+---------------+----+--------------+

|10 |Tilburg |NLD |Noord-Brabant |

+---+---------------+----+--------------+

|11 |Groningen |NLD |Groningen |

+---+---------------+----+--------------+

La fonction (en fait, une procédure, à strictement parler), prend en argument une table et deux arguments optionnels : debut, qui vaut 0 par défaut et fin, qui vaut None par défaut. La table est affichée de la ligne debut à la ligne fin. Si fin vaut None, la table est affichée jusqu’à la fin. Donc par défaut, la table est entièrement affichée.

Créez une fonction projection\_table qui prend en argument une table et une suite d’arguments correspondant à des numéros de colonnes. Cette fonction renvoie une table correspondant à la table en entrée, mais dont on ne garde que les colonnes spécifiées par la liste des numéros. Ainsi afficher\_table(projection\_table(villes,0,10),1,3) affiche :

+---------------+--------------+

|Kabul |Kabol |

+---------------+--------------+

|Qandahar |Qandahar |

+---------------+--------------+

|Herat |Herat |

+---------------+--------------+

|Mazar-e-Sharif |Balkh |

+---------------+--------------+

|Amsterdam |Noord-Holland |

+---------------+--------------+

|Rotterdam |Zuid-Holland |

+---------------+--------------+

|Haag |Zuid-Holland |

+---------------+--------------+

|Utrecht |Utrecht |

+---------------+--------------+

|Eindhoven |Noord-Brabant |

+---------------+--------------+

|Tilburg |Noord-Brabant |

+---------------+--------------+

|Groningen |Groningen |

+---------------+--------------+

Créez une fonction produit\_cartesien qui prend en argument deux tables et renvoie le produit cartésien de ces deux tables. Cette fonction sera utile par la suite. Ainsi, si vous opérez le produit cartésien d’une table de 3 lignes de 4 champs et d’une table de 5 lignes de 2 champs, vous obtenez une table de 15 lignes de 6 champs (donc une liste de tuples de taille 6). Vous aurez donc noté qu’en termes mathématiques, ce n’est pas à strictement parler un produit cartésien !

# Interroger l’information

Créez les fonctions qui renvoient la table contenant les informations demandées :

1. Donnez toutes les villes dont le nom commence par « pa » (sans tenir compte de la casse)
2. Donnez tous les pays d'Amérique du Sud.
3. Donnez toutes les villes d'Europe dont le nom commence par pa
4. Donnez toutes les villes de plus de 100 000 habitants d'Europe.
5. Donnez le nombre de formes de gouvernements présentes dans la base.
6. Donnez le nombre de pays répertoriés dans la base.
7. Donnez tous pays où l'on parle français.
8. Donnez tous les pays où le français est langue officielle.
9. Donnez toutes les villes de moins de 100 000 habitants, d'Afrique, ayant le français pour langue officielle.
10. Quels sont les pays d'Amérique du Sud de plus de 10 000 000 d'habitants ayant un régime républicain ?
11. Quelles sont les villes de plus de 100000 habitants de pays nord-américains où l'on parle espagnol ?
12. Donnez la surface de l'Europe.
13. Donnez la surface de la Polynésie.
14. Combien y-a-t’il de pays en Océanie de plus 10000 km² ?
15. Quelles sont les langues officielles des pays d'Europe de l'est ?
16. Quelle est la population moyenne dans les pays d'Asie ?
17. Quelle est la population moyenne dans les villes des pays d'Asie ?
18. Quelles sont les capitales d'Europe, ordonnées par ordre alphabétique ?
19. Donnez les villes des pays d'Afrique où la capitale a plus de 3 000 000 d'habitants ?
20. Quels sont les pays d’Amérique du Nord ayant accédé à l’indépendance avant 1912, où l’on parle portugais et pour lesquels sont répertoriées dans la base plus de 49 villes?
21. Quels sont les pays dont toutes les villes répertoriées dans la base ont plus de 100000 habitants ?
22. Quels sont les pays dont toutes les villes ont plus d’habitants que la ville la plus peuplée du Népal ?
23. Quels sont les pays où l’on parle français mais pas anglais ?
24. Quels sont les pays pour lesquels au moins une ville est répertoriée dans la base ?
25. Quels sont les pays pour lesquels aucune langue n’est répertoriée ?
26. Donnez les pays pour lesquels la somme du nombre d’habitants de ses villes est supérieure à 10000000. La France en fait-elle partie ?
27. Donnez le pays asiatique ayant l’espérance de vie la plus courte.
28. Quel est le nombre de villes répertoriées dans les pays qui parlent au moins trois langues, dont le français est langue officielle et dont l’espérance de vie des habitants est supérieure à l’espérance de vie de tous les habitants d’Amérique du Sud ?

# Compte-rendu du travail

Le rendu du travail sera constitué

* du code Python, qui devra être commenté
* un affichage des tables produites (utilisation de afficher\_table). Il faudra donner à chaque fois le nombre de lignes de la table, ainsi que l’affichage de seulement les 10 premières lignes.
* Une page listant les difficultés Python que vous avez (re)découvertes et comment vous les avez surmontées.
* Une page décrivant comment vous feriez évoluer la modélisation si les fichiers csv contenaient une première ligne donnant le nom de chaque colonne, et si on voulait accéder à une colonne via son nom (et non plus son numéro).
* Une page analysant le potentiel de ce projet pour des lycéens et donnant des pistes d’adaptation nécessaires pour ce public. Notamment, sur le plan technique Python, ce projet nécessite-il un apport lourd en terme de notion, ou est-il possible de demander aux élèves de lire la documentation (par exemple, sur la manipulation des fichiers csv) ? Le projet est-il un bon moyen de préparer à la manipulation de bases de données relationnelles, ou bien, finalement, demande-t-il trop de travail pour un élève par rapport à une approche plus classique (apports notionnels, exercices d’application)

# Critères d’évaluation du travail

La partie programmation sera évaluée sur les critères suivants :

* Correction des fonctions : le résultat donné est-il correct ?
* Lisibilité des fonctions : le code est-il commenté ? Les variables, le nom des fonctions ont-elles un nom lisible ?
* Généricité du code : le prototype des fonctions vous permettent-ils de les réutiliser d’une question à l’autre ?
* Modularité : avez-vous pensé à créer des fonctions supplémentaires pour les parties de code répétitives ?

Les autres parties seront évaluées sur la pertinence de votre réflexion, de votre argumentation, sachant qu’il n’y a pas de bonne réponse.

**Pour les Villes**

id NomVille CodePays Etat Nbrha

1 Kabul AFG Kabol 1780000

2 Qandahar AFG Qandahar 237500

3 Herat AFG Herat 186800

4 Mazar-e-Sharif AFG Balkh 127800

5 Amsterdam NLD Noord-Holland 731200

6 Rotterdam NLD Zuid-Holland 593321