

TD2
SQLITE ET PYTHON

❖ **Exercice 1 :**

Soit le schéma relationnel la base de données « **Entreprise.sqlite** » qui contient 3 tables:

- **Employe** (nume, nome, fonct, chef, Datemb, sal, comm, nums*)
- **Service** (nums, noms, loc)
- **Enfant** (numf, nomf, age, scol, nume*)

Exemple de données :**Employe**

nume	nome	Fonct	chef	datemb	sal	comm	nums
7369	Rmiki	Agent	7902	16/12/80	800		20
7499	El Faiz	Vendeur	7698	20/02/81	1600	300	30
7521	Alami	Vendeur	7698	21/02/81	1250	500	30
7566	Nasri	Directeur	7839	02/04/81	2975		20
7654	Bakari	Vendeur	7698	28/02/81	1250	1400	30
7698	Bensalah	Directeur	7839	01/05/81	2850		30
7782	Bennani	Directeur	7839	06/09/81	2450		10
7788	El Oifi	Analyste	7566	09/12/82	3000		20
7839	Karim	President		19/11/82	5000		10
7844	Sanhaji	Vendeur	7698	08/11/81	1500		30
7876	Sbai	Agent	7788	12/01/83	1100		20
7900	Benslimane	Agent	7698	03/12/81	950		30
7902	Boutayeb	Analyste	7566	02/12/81	3000		20
7934	Benbrahim	Agent	7782	23/01/82	1300		10

Enfant**Service**

nums	noms	loc
10	comptabilite	Fes
20	recherches	Rabat
30	ventes	Meknes
40	achat	Casa

Ecrire les fonctions python suivante :

1°/ executeRequete(r) : qui exécute la requête **r** et retourne une liste contenant le résultat de la requête. Tester cette fonction pour les requêtes suivantes :

- Liste des enfants : (numf, nomf, age)
- Nombre d'employés par service : (nums, noms, Nb_Emp)
- Insérer l'enfant (250,'Aziz',10, 'B' , 7566)

2°/ insererEnfant(num, nom ,age ,sc ,emp) qui :

- retourne **False** si le numéro d'enfant (paramètre **num**) existe déjà dans la table enfant ou si le numéro d'employé (paramètre **emp**) n'existe pas dans la table employé.
- insère l'enfant dont les valeurs d'attributs sont fournies en paramètre et retourne **True**.

3°/ NombreEnfants(num) : qui retourne le nombre d'enfants d'un employé dont le numéro est donné.

- 4°/ **ListeEnfant()** : qui retourne la liste des enfants (numf, nomf, age, nume, nome) triée par ordre croissant des nome et décroissant des âges.
- 5°/ **copierFichier(fich)** : qui copie vers le fichier texte fich la liste précédente, chaque ligne de fichier aura le format : numf, nomf, age, nume, nome
- 6°/ **afficheFichier(fich)** : qui affiche le fichier fich précédent.
- 7°/ **copierversFichiers()** : qui crée pour chaque employé un fichier nome_mume.txt contenant ses enfants.

❖ Exercice 2

Soit une base de données **poteries.sqlite** qui n'est constituée que d'une seule table **poteries** qui répertorie les proportions de différents métaux (**Al**, **Fe**, **Mg**, **Ca** et **Na**) présents dans des échantillons de poteries trouvés sur différents sites archéologiques désignés par une lettre (**L**, **C**, **I** ou **A**).

poteries

Al	Fe	Mg	Ca	Na	Site
14.4	7.0	4.3	0.15	0.51	L
13.8	7.08	3.43	0.12	0.17	L
11.5	6.37	5.64	0.16	0.14	L
11.8	5.44	3.94	0.3	0.04	C
11.6	5.39	3.77	0.29	0.06	C
18.3	1.28	0.67	0.03	0.03	I
15.8	2.39	0.63	0.01	0.04	I
18.0	1.5	0.67	0.01	0.06	I
18.3	1.14	0.67	0.06	0.05	A
16.7	0.92	0.53	0.01	0.05	A

A. Requêtes SQL

1. Écrire une requête SQL permettant de sélectionner les enregistrements qui correspondent au site **L** uniquement.
2. Écrire une requête SQL permettant de déterminer la moyenne des valeurs du champ **Al** pour le site **L**.

B. Interrogation d'une base de données dans un programme Python

3. Écrire une fonction **moyenne_métaux(s)** qui prend en paramètre un site **s** et qui retourne la liste des moyennes des proportions des différents métaux pour le site **s**
4. Écrire un programme qui permet d'obtenir les enregistrements pour lesquelles la valeur de **Al** est à $\pm 10\%$ de la moyenne des proportions de **Al** pour le site '**L**'

On désire savoir si les proportions de **Al** et **Fe** dans un échantillon de poterie peuvent permettre de retrouver de quel site il est issu.

5. Écrire une fonction **matrice_site(s)** qui prend en paramètre un site **s** et qui retourne une matrice de deux colonnes qui représentent les proportions de **Al** et **Fe** pour le site **s**
6. Écrire une fonction **trace_site(s)** qui prend en paramètre un site **s** et qui trace les coordonnées correspondant aux proportions de **Al** (en abscisses) et **Fe** (en ordonnées) pour le site **s**.

```
import sqlite3 as s3
import os
os.chdir('C:\\CPGE')
```

❖ Exercise 1:

1°/

```
def executeRequete(r):
    con=s3.connect("Entreprise.sqlite")
    cur=con.cursor()
    cur.execute(r)
    L=cur.fetchall()
    con.commit()
    con.close()
    return L
```

```
>>> executeRequete('select numf,nomf, age from enfant')
```

```
[(123, 'Loubna', 17), (124, 'Zohra', 8), (235, 'Abdelkrim', 13), (236,
'Abdellah', 8), (237, 'Samah', 2), (238, 'Mohamed', 5), (239, 'Said',
12), (240, 'Khalid', 4)]
```

```
>>> executeRequete('select service.numf,noms,count(nume) from
service,employe where employe.numf =service.numf group by service.numf')
```

```
[(10, 'comptabilite', 3), (20, 'recherches', 5), (30, 'ventes', 6)]
```

```
>>> executeRequete("insert into enfant values (250,'Aziz',10,'B',7566)")
[]
```

2°/

```
def insererEnfant(num,nom,age,sc,emp):
    con=s3.connect("Entreprise.sqlite")
    cur=con.cursor()

    #Tester si le numéro d'enfant existe déjà
    req="select numf from enfant where numf=?"
    cur.execute(req,[num])
    if cur.fetchone()!=None:
        valeur_retour = False
    else:
        #Tester si le numéro d'employé existe dan la table employé
        req="select nume from employe where nume=?"
        cur.execute(req,[emp])
        if cur.fetchone()==None:
            valeur_retour = False
        else:
            req="insert into enfant values(?,?,?,?)"
            cur.execute(req,[num,nom,age,sc,emp])
            con.commit()
            valeur_retour = True
    con.close()
    return valeur_retour
```

3°/

```
def NombreEnfant(num) :
    con=s3.connect("Entreprise.sqlite")
    cur=con.cursor()
    req="select count(num) from enfant where num=?"
    cur.execute(req,[num])
    t=cur.fetchone() #t est un tuple
    con.close()
    return t[0]
```

4°/

```
def ListeEnfant() :
    return executeRequete("select numf, nomf, age,employe.num, nom
from enfant,employe where enfant.num=employe.num order by employe.num
asc,age desc")
```

5°/

```
def copierFichier(fich) :
    #récupérer la liste des enfants
    L=ListeEnfant()

    #ouverture du fichier en création
    f=open(fich,"w")

    #parcourir la liste L et écrire dans le fichier f
    for l in L:
        ligne=str(l[0])+", "+l[1]+", "+str(l[2])+", "+str(l[3])+", "+l[4]+"\\n"
        f.write(ligne)

    #fermer le fichier
    f.close()
```

6°/

```
def AfficherFichier(fich) :
    #ouverture du fichier en lecture
    f=open(fich,"r")

    #parcourir le fichier et afficher ligne par ligne
    for ligne in f:
        print(ligne.rstrip())

    #fermer le fichier
    f.close()
```

7°/

```
def copierFichiers() :
    #récupérer la liste des enfants
    L=ListeEnfant()

    for l in L:
        fich=l[4]+'_'+str(l[3])
        f=open(fich+".txt","a")
        f.write(str(l[0])+", "+l[1]+", "+str(l[2])+"\\n")

    #fermer le fichier
    f.close()
```

❖ Exercice 2 :

A. Requêtes SQL

3. Écrire une requête SQL permettant de sélectionner les enregistrements qui correspondent au site **L**.

```
SELECT * FROM POTERIES WHERE Site LIKE 'L' ;
```

4. Écrire une requête SQL permettant de déterminer la moyenne des valeurs du champ **Al** pour le site **L**.

```
SELECT AVG(Al) FROM POTERIES WHERE Site LIKE 'L'
```

ou

```
SELECT AVG(Al) FROM POTERIES GROUP BY site HAVING site='L'
```

C. Interrogation d'une base de données dans un programme Python

Pour les questions suivantes on doit importer les modules suivants :

```
import sqlite3 as s3
import numpy as np
import matplotlib.pyplot as plt
```

7. Écrire une fonction **moyenne_métaux(s)** qui prend en paramètre un site **s** et qui retourne la liste des moyennes des proportions des différents métaux pour le site **s**

```
def moyenne_métaux(s):
    con=s3.connect('c:/poteries.sqlite')
    cur=con.cursor()
    req='select avg(Al),avg(Fe),avg(Mg),avg(Ca),avg(Na) from poteries where site LIKE ?'
    cur.execute(req,[s])
    L= cur.fetchone()
    con.close()
    return L
```

8. Ecrire un programme qui permet d'obtenir les enregistrements pour lesquelles la valeur de **Al** est à $\pm 10\%$ de la moyenne des proportions de **Al** pour le site **'L'**

```
L=moyenne_métaux('L')
moy=L[0]
con=s3.connect('c:/poteries.sqlite')
cur=con.cursor()
req='select* from poteries where Al between ? and ? '
cur.execute(req,[moy*0.9,moy*1.1])
for e in cur.fetchall():
    print(e)
con.close()
```

On désire savoir si les proportions de **Al** et **Fe** dans un échantillon de poterie peuvent permettre de retrouver de quel site il est issu.

9. Écrire une fonction **matrice_site(s)** qui prend en paramètre un site **s** et qui retourne une matrice de deux colonnes qui représentent les proportions de **Al** et **Fe** pour le site **s**

```
def matrice_site(s):
    con=s3.connect('c:/poteries.sqlite')
    cur=con.cursor()
    req='select Al, Fe from poteries where site=? '
    cur.execute(req , [s])
    L= cur.fetchall() #ici L est une liste de tuples chacun avec 2 valeurs
    return np.matrix(L)
```

10. Écrire une fonction **trace_site(s)** qui prend en paramètre un site **s** et qui trace les coordonnées correspondant aux proportions de **Al** (en abscisses) et **Fe** (en ordonnées) pour le site **s**.

```
def trace_site(s):
    M=matrice_site(s)
    plt.plot(M[:,0],M[:,1])
    plt.grid()
    plt.xlabel('Al')
    plt.ylabel('Fe')
    plt.show()
```