

fiche 7

Langage SQL en python

Version avec preuves.

1 Création de la base

On souhaite créer une base de donnée sur les élèves (fictifs) de *NSI* :

Nom	prénom	Année de naissance
Lara	Croft	2002
Solid	Snake	2001
John	Marston	2001
Nathan	Drake	2003
Gordon	Freeman	2003
Zelda	Link	2002
Mario	Luigi	2003

Création de la base :

```
from sqlite3 import *
ma_base = connect("eleve_NSI.sql3")# creation ou ouverture du fichier
    de la base
curseur = ma_base.cursor() #utile pour la manipulation de la base
# creation de la base : Execution de la requete
curseur.execute("CREATE TABLE IF NOT EXISTS eleves (nom TEXT, prenom
    TEXT, annee INTEGER)")
# Envoyer la requete
ma_base.commit()
# Fermer la connexion
ma_base.close
```

Insertion de données :

```
# Insérer une ligne de donnees
curseur.execute ("INSERT INTO eleves (nom, prenom , annee) VALUES ( '
    Lara ' , ' Croft ' , 2002)")
# deuxieme methode : Insérer une ligne de donnees
curseur.execute("INSERT INTO eleves (nom, prenom , annee) VALUES
    (?, ?, ?)" , ( 'Solid ' , 'Snake ' , 2001))
```

Exercice 1: Construire la base de données associée à la liste d'élèves.

2 Lecture des résultats d'une requête

Exécution d'une requête :

```
# Une requette  
ma_requette = curseur.execute ("SELECT * FROM eleves")  
print(list (ma_requette))
```

Exercice 2: Écrire la fonction `requete` qui prend une requête et une base de donnée et retourne la liste des éléments répondants à la requête.

Correction

```
def requete(req,db):
    fichier=sql.connect(db)
    curseur=fichier.cursor()
    curseur.execute(req)
    a=list(curseur)
    fichier.close()
    return(a)
```

3 Lecture d'un fichier CSV

Enregistrer le fichier csv :

http://sebjaumaths.free.fr/NSI_term/td/liste_films.csv

Création d'une base de donnée à partir d'un fichier CSV :

```
import sqlite3 , csv

con = sqlite3.connect("films.db")
cur = con.cursor()
cur.execute("CREATE TABLE film (titre STR,genre,date_sortie STR,
    duree INT,note INT , PRIMARY KEY(titre))")

with open("liste_films.csv","r") as films:
    dr = csv.DictReader(films , delimiter=";")
    to_db = [(i["titre"], i["genre"], i["date_sortie"], i["duree"],
    i["note"]) for i in dr]

cur.executemany("INSERT INTO film VALUES (?,?,?,?,?);", to_db)
con.commit()

con.close()
```

Exercice 3: Écrire les requêtes suivantes pour obtenir les tableaux suivants :

1. La liste des titres et les notes des films. **Correction**

```
requete_1 = """ SELECT titre , note
                FROM film
                """
```

2. La liste des drames. **Correction**

```
requete_2 = """ SELECT titre
                FROM film
                WHERE genre='drame' """
```

```
"""
```

3. La liste des titres ayant une note supérieur à 4. **Correction**

```
requete_3 = """ SELECT titre
                FROM film
                WHERE note > 4 AND NOT ( note = 'NULL' )
                """
```

4. La liste des titres de films d'action sortis avant 2000. **Correction**

```
requete_4 = """ SELECT titre
                FROM film
                WHERE date_sortie <2000 AND genre ='Action'
                """
```

5. Augmenter la note du film "Casablanca" d'un point. **Correction**

```
requete_5 = """ UPDATE film
                SET note = note +1
                WHERE titre='Casablanca'
                """
```

6. La moyenne des notes par genre. **Correction**

```
requete_6 = """ SELECT genre , AVG(note) as moyenne
                FROM film
                GROUP BY genre
                """
```

4 Manipulation d'une base db3

Les fichiers contenant l'extension de fichier DB3 sont le plus souvent associés à l'application de base de données SQLite. On peut tout de même utiliser python pour travailler sur ces fichiers.

Création d'une base de donnée à partir d'un fichier CSV :

```
from sqlite3 import *

# creation de la base :
ma_base = connect("World.db3")
curseur = ma_base.cursor()

# pour obtenir la liste des tables dans la base de donnees
tables = curseur.execute("SELECT name FROM sqlite_master WHERE type
    ='table';").fetchall()
print(tables)

# on veut voir les colonnes de chaque table ainsi que la premiere
    ligne
for table in tables :
    print("Table :", table[0])
    schema = curseur.execute("PRAGMA table_info({})".format(table
    [0])).fetchall()
    print("Colonnes", [{"{}".format(x[1]) for x in schema])
    print("1ere ligne", curseur.execute('SELECT * FROM {} LIMIT 1'
    .format(table[0])).fetchall(), "\n")

ma_base.close
```

Exercice 4: Déterminer le schéma relationnel de la base de données.

Exercice 5: Déterminer :

1. La liste des langages utiliser en France. **Correction**

```
requete_1 = """
    SELECT CountryLanguage.Language
    FROM CountryLanguage
    JOIN Country
    ON CountryLanguage.CountryCode = Country.Code
    WHERE Country.Name = 'France'
    """
```

2. La liste des pays qui sont des monarchies. **Correction**

On commence par regarder les différentes formes de gouvernement.

```
requete_2_0 = """
    SELECT DISTINCT GovernmentForm
    FROM Country
```

```
"""
```

On se rend compte qu'il y a plusieurs formes de monarchies :

```
requete_2 = """
    SELECT name
    FROM Country
    WHERE GovernmentForm LIKE "%Monarchy%"
    """
```

3. Quels sont les 10 pays qui ont le plus de langues? **Correction**

```
requete_3 = """
    SELECT Country.name , count(CountryLanguage.language
    ) as nb
    FROM CountryLanguage
    JOIN Country
    ON CountryLanguage.CountryCode = Country.Code
    GROUP BY Country.name
    order by nb DESC
    LIMIT 10
    """
```

4. Quel langue est présent dans le plus de pays? **Correction**

```
requete_4 = """
    SELECT CountryLanguage.language , count(CountryCode)
    as nb
    FROM CountryLanguage
    GROUP BY CountryLanguage.language
    order by nb DESC
    LIMIT 1
    """
```

5. Quels sont les 3 gouvernements où la population est la plus importante? **Correction**

```
requete_5 = """
    SELECT GovernmentForm , SUM(Population) as Pop_Totale
    FROM Country
    GROUP BY GovernmentForm
    ORDER BY Pop_Totale DESC
    LIMIT 3
    """
```