

DS NSI n°3 : Base de données

Dans ce devoir, nous allons analyser une base de données sur le thème du cinema. Vous préparerez quelques requêtes SQL dans un premier temps, puis vous utiliserez Python pour exécuter et exploiter les requêtes SQL.

Vous devez récupérer la base [ici \(https://github.com/sofaugeras/TNSI/blob/master/2_Base%20de%20donn%C3%A9es/cinema_BD\)](https://github.com/sofaugeras/TNSI/blob/master/2_Base%20de%20donn%C3%A9es/cinema_BD).

```
In [ ]: %load_ext sql

        %sql sqlite:///cinema_BD
```

Travail à faire :

Rédiger en langage SQL les requêtes suivantes :

1) Sélectionner tous les films dont les recettes sont supérieures à 1000.

```
In [ ]: %%sql
        SELECT titre,recettes
        FROM Film
        WHERE recettes > 1000 ;
```

2) Sélectionner tous les films de George Lucas

```
In [ ]: %%sql
        SELECT Titre
        FROM Film
        WHERE nom_realisateur = 'Lucas' AND prenom_realisateur='George' ;
```

4) Afficher les films entre 2000 et 2005

```
In [ ]: %%sql
        SELECT titre,annee
        FROM Film
        WHERE annee BETWEEN 2000 AND 2005;
```

```
In [ ]: %%sql
        SELECT titre,annee
        FROM Film
        WHERE annee >= 2000 AND annee <= 2005;
```

5) Afficher tous les films dont le titre commence par « T ».

```
In [ ]: %%sql
        SELECT Titre
        FROM Film
        WHERE titre LIKE 'T%';
```

6) Affichez pour chaque réalisateur le nombre de ses films présents dans la base

```
In [ ]: %%sql
        SELECT COUNT(*), nom_realisateur
        FROM Film
        GROUP BY nom_realisateur;
```

7) Afficher le nombre de film fantastique pour chaque année.

```
In [ ]: %%sql
        SELECT COUNT(*), nomGenre
        FROM Film INNER JOIN GENRE
        ON film.idgenre = genre.id
        GROUP BY nomGenre
```

8) Afficher la somme des recettes, le nombre et la moyenne des recettes des films Harry Potter

```
In [ ]: result = %%sql SELECT SUM(recettes) AS 'somme des recettes', COUNT(*)
        AS 'nb de film', AVG(recettes) AS 'moy des recettes' FROM Film WHERE
        titre LIKE 'Harry Potter%';
        result
```

9) Ecrire l'assert qui permet de valider la réponse ci dessus

```
In [ ]: assert result[0][0] == 7668
```

Partie Base de Données et Python

Pour rappel, on ouvre un curseur en indiquant les instructions suivantes.

```
In [ ]: import sqlite3
        bdd = sqlite3.connect("cinema_BD")
        curseur = bdd.cursor()
```

et pour l'exécuter

```
In [ ]: curseur.execute(requete)
resultats = curseur.fetchall()
#ou
#resultats = curseur.fetchone()
#resultats
```

et un exemple de fonction

```
In [ ]: def recupDonnees(c):
    requete = "SELECT nom_realisateur, annee, titre FROM FILM"
    c.execute(requete)
    r = c.fetchall()
    if len(r) == 0:
        return None
    elif len(r) >= 1:
        liste = []
        for i in range(len(r)) :
            liste.append(r[i][0])
        return liste
```

```
In [ ]: recup = recupDonnees(curseur)
recup
```

Travail à faire :

A l'aide de la base de données Cinéma, créer une fonction de création d'un Dictionnaire Python ayant pour clé le nom du réalisateur concaténé à l'année de sortie du film et ayant pour valeur le nom du film.

exemple :

```
In [ ]: liste_film = [{"cle" : "scott1982", "titre" : "1492"},
                    {"cle" : "Lynch1985", "titre" : "Dune"}
                    ]
```

```
In [ ]: def recupDonnees(c):
    requete = "SELECT nom_realisateur, annee, titre FROM FILM"
    c.execute(requete)
    r = c.fetchall()
    if len(r) == 0:
        return None
    elif len(r) >= 1:
        liste = []
        for i in range(len(r)) :
            dict = {"cle" : str(r[i][0])+str(r[i][1]), "titre" : r
[i][2]}
            liste.append(dict)
        return liste
```

```
In [ ]: liste_film = recupDonnees(curseur)
liste_film
```

```
In [ ]: bdd.close()
```