

TP n° 14 : Interrogation d'une BDD

Correction de l'exercice 1 – Pour commencer, l'entête Python, et une petite fonction qui lancera l'exécution d'une requête, et provoquera l'affichage ainsi que la sauvegarde dans un fichier.

```
import sqlite3

### Ouverture de la connection à la BDD:

connection = sqlite3.connect('py066-ma_banque.db')
cur = connection.cursor()

### Ouverture en écriture du fichier de sauvegarde des résultats

reponse = open('py066-rep.txt','w')

### Pour lancer une requête afficher et sauvegarder le résultat

def requete(R,n):
    cur.execute(R)
    print('Question_{}'.format(n))
    print()
    reponse.write('Question_{}\n\n'.format(n))
    for L in cur:
        reponse.write(str(L) + '\n')
        print(L)
    print()
    reponse.write('\n')

### INSÉRER LES DIFFÉRENTES REQUÊTES ICI ###

### Fermeture du fichier de sauvegarde et de la connection à la BDD

reponse.close()
connection.commit()
connection.close()
```

Et voici la liste des requêtes :

```
#### QUESTION 1
# Nom prénom de tous les clients

requete("""SELECT nom, prenom
          FROM client""",1)

#### QUESTION 2
# Nom prénom des clients de Paris
```

```

requete("""SELECT nom, prenom
        FROM client
        WHERE lower(ville) = 'paris'""",2)

#### QUESTION 3
# Identifiant des comptes Livret A

requete("""select idcompte
        from compte
        where type = 'Livret A'""",3)

#### QUESTION 4
# Identifiant et montant des opérations de débit sur le compte no 1

requete("""SELECT idop, montant
        FROM operation
        WHERE (idcompte = 1) AND (montant < 0)""",4)

#### QUESTION 5
# Identifiant des propriétaires de livret A (sans doublon),
# classé par ordre croissant d'identifiant.

requete("""SELECT DISTINCT idproprietaire
        FROM compte
        WHERE type = 'Livret A'
        ORDER BY idproprietaire""",5)

#### QUESTION 6
# Identifiant des clients n'ayant pas de livret A.
# Attention, certains n'ont pas de livret du tout, et ne sont pas dans la table compte.

requete("""SELECT idclient
        FROM client
        WHERE idclient NOT IN
        (SELECT idproprietaire
        FROM compte
        WHERE type = 'Livret A')
        ORDER BY idclient""",6)

#### QUESTION 7
# Numéros de comptes et types des clients de Paris

requete("""SELECT idcompte, type
        FROM compte
        WHERE idproprietaire IN
        (SELECT idclient
        FROM client
        WHERE ville = 'Paris')""",7)

#### QUESTION 8
# Liste des comptes et type de compte de Dumbledore

```

```
requete("""SELECT idproprietaire, idcompte, type
          FROM compte
          WHERE idproprietaire IN
                (SELECT idclient FROM client
                 WHERE nom = 'Dumbledore')""",8)
```

QUESTION 9

Nombre de client par ville, classé par ordre alphabétique de ville

```
requete("""SELECT ville, COUNT(idclient)
          FROM client
          GROUP BY ville
          ORDER BY ville""",9)
```

QUESTION 10

La ville dans laquelle se trouve le plus de client

Version 1, incorrecte on selectionne un champs non agrégé en même
temps qu'une fonction agrégative; cette syntaxe passe en SQLite,
mais pas en SQL, et renvoie une ville au hasard réalisant ce
maximum.

```
requete("""SELECT ville, max(nb)
          FROM (SELECT ville, COUNT(idclient) AS nb
                FROM client
                GROUP BY ville)""",10)
```

Version 2, renvoyant également une seule ville réalisant le maximum,
mais avec une syntaxe correcte

```
requete("""SELECT ville, COUNT(idclient)
          FROM client
          GROUP BY ville
          ORDER BY COUNT(idclient) DESC
          LIMIT 1""",10)
```

Version 3, renvoyant toutes les villes réalisant le maximum, s'il y
en a plusieurs

```
requete("""SELECT ville, COUNT(idclient)
          FROM client
          GROUP BY ville
          HAVING COUNT(idclient) =
                (SELECT COUNT(idclient)
                 FROM client
                 GROUP BY ville
                 ORDER BY COUNT(idclient) DESC
                 LIMIT 1) """,10)
```

QUESTION 11

```

# Trouver le nombre d'opérations par compte

requete("""SELECT idcompte, COUNT(idop)
          FROM operation
          GROUP BY idcompte""",11)

#### QUESTION 12
# Trouver le nombre maximal d'opérations par compte

requete("""SELECT MAX(nbop)
          FROM (SELECT idcompte, COUNT(idop) AS nbop
                FROM operation
                GROUP BY idcompte)""",12)

#### QUESTION 13
# Afficher le ou les no de compte sur lesquels il y a eu le plus de mvt
# Version 1

requete("""SELECT compte.idcompte, nbop
          FROM compte,
          (SELECT idcompte, COUNT(idop) AS nbop
           FROM operation
           GROUP BY idcompte) AS cmp
          WHERE
            (cmp.idcompte = compte.idcompte)
          AND
            (nbop = (SELECT MAX(nb)
                     FROM (SELECT idcompte, COUNT(idop) AS nb
                            FROM operation
                            GROUP BY idcompte)))""", 13)

# Version 2, plus simple, illustrant l'intérêt de LIMIT permettant de
# récupérer facilement les extrêmes sans avoir à faire de selection
# supplémentaire:

requete("""SELECT idcompte, COUNT(idop) AS nbop
          FROM operation
          GROUP BY idcompte
          HAVING COUNT(idop) = (SELECT COUNT(*)
                               FROM operation
                               GROUP BY idcompte
                               ORDER BY COUNT(*) DESC
                               LIMIT 1)""", 13)

#### QUESTION 14
# Moyenne des soldes des comptes de chaque type

requete("""SELECT type, AVG(solde)
          FROM compte,

```

```

        (SELECT idcompte, SUM(montant) AS solde
         FROM operation
         GROUP BY idcompte) AS mo
WHERE mo.idcompte = compte.idcompte
GROUP BY type""",14)

```

ou avec JOIN:

```

requete("""SELECT type, AVG(solde)
         FROM   compte
         JOIN
             (SELECT idcompte, SUM(montant) AS solde
              FROM operation
              GROUP BY idcompte) AS mo
         ON mo.idcompte = compte.idcompte
         GROUP BY type""",14)

```

QUESTION 15

Nom prénom type-compte solde-compte

```

requete("""SELECT nom, prenom, type, SUM(montant)
         FROM client
         JOIN compte ON idclient = idproprietaire
         JOIN operation ON operation.idcompte = compte.idcompte
         GROUP BY operation.idcompte""",15)

```

ou bien

```

requete("""SELECT nom, prenom, type, SUM(montant)
         FROM operation AS op,
             (SELECT idcompte, idclient, nom, prenom, type
              FROM client, compte
              WHERE idclient = idproprietaire) AS cc
         WHERE
             op.idcompte = cc.idcompte
         GROUP BY op.idcompte""",15)

```

QUESTION 16

Solde de tous les comptes des clients qui commencent par K,L,M,N,

identifié par le nom prénom.

```

requete("""SELECT nom, prenom, type, SUM(montant)
         FROM client
         JOIN compte ON idclient = idproprietaire
         JOIN operation AS op ON op.idcompte = compte.idcompte
         WHERE nom BETWEEN 'K' AND 'O'
         GROUP BY op.idcompte
         ORDER BY NOM, PRENOM""",16)

```

QUESTION 17

Nom et prénom des personnes ayant fait au moins débité un chèque sur leur compte courant

```

requete("""SELECT DISTINCT nom, prenom
FROM compte
JOIN operation ON operation.idcompte = compte.idcompte
JOIN client ON idproprietaire = idclient
WHERE (lower(type) = 'compte courant')
AND
(lower(informations) = 'cheque')
AND
(montant < 0)
ORDER BY nom""",17)

```

QUESTION 18

Nom prénom ville de toutes les personnes ayant fait le plus d'opérations au guichet

```

requete("""SELECT nom, prenom, ville
FROM
(SELECT nom, prenom, ville, count(*) AS tot
FROM compte
JOIN operation ON operation.idcompte = compte.idcompte
JOIN client ON idclient = idproprietaire
WHERE lower(informations) = 'guichet'
GROUP BY informations, idproprietaire)
WHERE
tot IN (SELECT max(t) FROM
(SELECT count(*) AS t
FROM compte
JOIN operation ON operation.idcompte = compte.idcompte
WHERE lower(informations) = 'guichet'
GROUP BY informations, idproprietaire))""",18)

```

QUESTION 19

Moyenne par ville des sommes des soldes des comptes d'un client,
classé par valeur croissante.

```

requete("""SELECT ville, AVG(tot) AS mo
FROM (SELECT ville, idclient, sum(montant) as tot
FROM operation
JOIN compte ON operation.idcompte = compte.idcompte
JOIN client ON idclient = idproprietaire
GROUP BY idclient)
GROUP BY ville
ORDER BY mo""",19)

```