

TP n° 12 : Bases de données : création

1. Pour éviter les redondances, il faut séparer :

- les clients (ne pas répéter les données du client pour les plusieurs factures éventuelles de chaque client)
- les produits (ne pas répéter les caractéristiques du produit pour chaque vente de ce produit)
- les factures vues globalement (ne pas répéter la date et le client pour chaque produit vendu dans une même facture)
- Le détail des factures (qui ne mentionnera que le numéro de facture, le numéro de produit et la quantité).

Nous obtenons les instructions de création suivantes :

```
import sqlite3

connection = sqlite3.connect('BDD-bricolage.db')
cur = connection.cursor()

cur.execute("""DROP TABLE IF EXISTS client""")
cur.execute("""DROP TABLE IF EXISTS facture""")
cur.execute("""DROP TABLE IF EXISTS detailfacture""")
cur.execute("""DROP TABLE IF EXISTS produit""")

cur.execute("""CREATE TABLE IF NOT EXISTS client (
    idclient int(5) NOT NULL PRIMARY KEY,
    nom varchar(128) NOT NULL,
    ville varchar(128) NOT NULL)""")

cur.execute("""CREATE TABLE IF NOT EXISTS facture (
    idfacture int(5) NOT NULL PRIMARY KEY,
    datefact date NOT NULL,
    idclient int(5) NOT NULL REFERENCES client )""")

cur.execute("""CREATE TABLE IF NOT EXISTS produit (
    idproduit int(5) NOT NULL PRIMARY KEY,
    nomproduit varchar(30) NOT NULL,
    categorie varchar(30) NOT NULL,
    prix decimal(10,2) NOT NULL)""")

cur.execute("""CREATE TABLE IF NOT EXISTS detailfacture (
    idfacture int(5) NOT NULL REFERENCES facture,
    idproduit int(5) NOT NULL REFERENCES produit,
    quantite int(5) NOT NULL,
    PRIMARY KEY(idfacture,idproduit))""")
```

2. Nous insérons alors les valeurs voulues. Un peu fastidieux, mais des copier/coller, et un peu de méthode permettent de s'en sortir en 15 à 20 minutes. Ceci est à mettre à la suite du programme précédent. Attention à bien fermer la connection à la base à la fin du programme, sinon vos valeurs ne seront pas enregistrées, et vous ne pourrez pas les réutiliser via un autre programme.

```
### Insertion des valeurs de la table client

cur.execute("""INSERT INTO client VALUES (1, "Chien", "La-Cour-Basse")""")
```

```

cur.execute("""INSERT INTO client VALUES (2, "Oie", "La-Cour-Basse")""")
cur.execute("""INSERT INTO client VALUES (3, "Cochon", "La-Cour-Basse")""")
cur.execute("""INSERT INTO client VALUES (4, "Lapin", "La-Cour-Basse")""")
cur.execute("""INSERT INTO client VALUES (5, "Poule", "La-Cour-Basse")""")
cur.execute("""INSERT INTO client VALUES (6, "Cheval", "Le-Vert-Pré")""")
cur.execute("""INSERT INTO client VALUES (7, "Vache", "Le-Vert-Pré")""")
cur.execute("""INSERT INTO client VALUES (8, "Renard", "Touffue-La-Forêt")""")
cur.execute("""INSERT INTO client VALUES (9, "Sanglier", "Touffue-La-Forêt")""")
cur.execute("""INSERT INTO client VALUES (10, "Cerf", "Touffue-La-Forêt")""")
cur.execute("""INSERT INTO client VALUES (11, "Lièvre", "Touffue-La-Forêt")""")

### Insertion des valeurs de la table produit

cur.execute("""INSERT INTO produit VALUES (1, "Vis 20mm", "Quincaillerie", 1)""")
cur.execute("""INSERT INTO produit VALUES (2, "Chevilles", "Quincaillerie", 2)""")
cur.execute("""INSERT INTO produit VALUES (3, "Tournevis", "Outil", 8)""")
cur.execute("""INSERT INTO produit VALUES (4, "Table d'extérieur", "Aménagement", 150)""")
cur.execute("""INSERT INTO produit VALUES (5, "Chaises d'extérieur", "Aménagement", 49)""")
cur.execute("""INSERT INTO produit VALUES (6, "Parasol", "Aménagement", 78)""")
cur.execute("""INSERT INTO produit VALUES (7, "Lampadaire", "Éclairage", 66)""")
cur.execute("""INSERT INTO produit VALUES (8, "Vis 10mm", "Quincaillerie", 0.5)""")
cur.execute("""INSERT INTO produit VALUES (9, "Clous 10mm", "Quincaillerie", 0.5)""")
cur.execute("""INSERT INTO produit VALUES (10, "Marteau", "Outil", 12)""")
cur.execute("""INSERT INTO produit VALUES (11, "Lampe de chevet", "Éclairage", 14)""")
cur.execute("""INSERT INTO produit VALUES (12, "Ampoule", "Éclairage", 3)""")
cur.execute("""INSERT INTO produit VALUES (13, "Interrupteur", "Électricité", 5)""")
cur.execute("""INSERT INTO produit VALUES (14, "Prise", "Électricité", 5)""")
cur.execute("""INSERT INTO produit VALUES (15, "Boîte d'encastrement", "Électricité", 5)""")
cur.execute("""INSERT INTO produit VALUES (16, "Perceuse", "Outil", 76)""")
cur.execute("""INSERT INTO produit VALUES (17, "Lave-vaisselle", "Électro-ménager", 999)""")
cur.execute("""INSERT INTO produit VALUES (18, "Cable", "Électricité", 5)""")
cur.execute("""INSERT INTO produit VALUES (19, "Lave-linge", "Électro-ménager", 1024)""")
cur.execute("""INSERT INTO produit VALUES (20, "Radiateur", "Électricité", 99)""")
cur.execute("""INSERT INTO produit VALUES (21, "Clé hexagonale", "Outil", 10)""")
cur.execute("""INSERT INTO produit VALUES (22, "Scie", "Outil", 15)""")
cur.execute("""INSERT INTO produit VALUES (23, "Barbecue", "Aménagement", 100)""")
cur.execute("""INSERT INTO produit VALUES (24, "Transat", "Aménagement", 56)""")

### Insertion des valeurs de la table facture

cur.execute("""INSERT INTO facture VALUES (1, '2014-05-12', 1)""")
cur.execute("""INSERT INTO facture VALUES (2, '2014-05-12', 6)""")
cur.execute("""INSERT INTO facture VALUES (3, '2014-05-13', 2)""")
cur.execute("""INSERT INTO facture VALUES (4, '2014-05-13', 8)""")
cur.execute("""INSERT INTO facture VALUES (5, '2014-05-14', 9)""")
cur.execute("""INSERT INTO facture VALUES (6, '2014-05-14', 1)""")
cur.execute("""INSERT INTO facture VALUES (7, '2014-05-14', 9)""")
cur.execute("""INSERT INTO facture VALUES (8, '2014-05-15', 7)""")
cur.execute("""INSERT INTO facture VALUES (9, '2014-05-15', 8)""")
cur.execute("""INSERT INTO facture VALUES (10, '2014-05-15', 5)""")
cur.execute("""INSERT INTO facture VALUES (11, '2014-05-16', 4)""")
cur.execute("""INSERT INTO facture VALUES (12, '2014-05-16', 10)""")
cur.execute("""INSERT INTO facture VALUES (13, '2014-05-16', 1)""")

```

```

cur.execute("""INSERT INTO facture VALUES (14, '2014-05-17', 9)""")
cur.execute("""INSERT INTO facture VALUES (15, '2014-05-17', 3)""")
cur.execute("""INSERT INTO facture VALUES (16, '2014-05-18', 1)""")
cur.execute("""INSERT INTO facture VALUES (17, '2014-05-18', 11)""")
cur.execute("""INSERT INTO facture VALUES (18, '2014-05-18', 6)""")

### Insertion des valeurs dans la table detailfacture

cur.execute("""INSERT INTO detailfacture VALUES (1, 1, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (1, 2, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (1, 3, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (2, 4, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (2, 5, 6)""")
cur.execute("""INSERT INTO detailfacture VALUES (2, 6, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (2, 7, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (3, 1, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (3, 8, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (3, 9, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (3, 10, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (4, 11, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (4, 12, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (5, 13, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (5, 14, 4)""")
cur.execute("""INSERT INTO detailfacture VALUES (5, 15, 6)""")
cur.execute("""INSERT INTO detailfacture VALUES (6, 16, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (6, 17, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (7, 18, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (8, 19, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (9, 4, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (9, 5, 4)""")
cur.execute("""INSERT INTO detailfacture VALUES (10, 1, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (10, 3, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (11, 20, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (11, 18, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (12, 10, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (12, 21, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (13, 6, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (14, 17, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (15, 1, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (15, 8, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (15, 12, 3)""")
cur.execute("""INSERT INTO detailfacture VALUES (15, 22, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (16, 23, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (16, 24, 2)""")
cur.execute("""INSERT INTO detailfacture VALUES (17, 1, 1)""")
cur.execute("""INSERT INTO detailfacture VALUES (18, 12, 4)""")

connection.commit()
connection.close()

```

3. Les modifications sont faites dans un autre programme :

```
import sqlite3
```

```

connection = sqlite3.connect('BDD-bricolage.db')
cur = connection.cursor()

cur.execute("""ALTER TABLE facture ADD COLUMN type_vente varchar(20) """)
cur.execute("""UPDATE facture SET type_vente = 'magasin'""")
cur.execute("""DROP TABLE IF EXISTS ville""")
cur.execute("""CREATE TABLE IF NOT EXISTS ville (
    nom_ville varchar(128) NOT NULL PRIMARY KEY,
    pays NOT NULL)""")

cur.execute("""INSERT INTO ville SELECT DISTINCT ville, 'France' FROM client""")
cur.execute("""INSERT INTO ville VALUES ('Chaude-La-Savane', 'Kenya')""")
cur.execute("""INSERT INTO ville VALUES ('Glagla-La-Banquise', 'Groenland')""")
cur.execute("""INSERT INTO client VALUES (12,'Lion', 'Chaude-La-Savane')""")
cur.execute("""INSERT INTO client VALUES (13,'Pingouin', 'Glagla-La-Banquise')""")
cur.execute("""INSERT INTO facture VALUES (19,'2014-05-22',12,'internet')""")
cur.execute("""INSERT INTO facture VALUES (20,'2014-05-23',13,'internet')""")
cur.execute("""INSERT INTO produit VALUES (25,'climatiseur','Aménagement',688)""")
cur.execute("""INSERT INTO detailfacture VALUES (19,6,1)""")
cur.execute("""INSERT INTO detailfacture VALUES (19,25,1)""")
cur.execute("""INSERT INTO detailfacture VALUES (20,20,3)""")

connection.commit()
connection.close()

```

4. Et enfin les différentes requêtes :

```

import sqlite3

connection = sqlite3.connect("BDD-bricolage.db")
cur = connection.cursor()

def affiche(cur):
    for L in cur:
        print(L)
    print()

# 4(a) La liste des clients de Touffue-La-Forêt

cur.execute("""SELECT nom FROM client
              WHERE ville = 'Touffue-La-Forêt'""")
affiche(cur)

#4(b) Le détail de la facture 12

cur.execute("""
    SELECT idfacture, nomproduit, quantite
    FROM detailfacture NATURAL JOIN produit WHERE idfacture = 12""")
affiche(cur)

#4(c) Les habitants de La-Cour-Basse ayant acheté des Vis 20mm

cur.execute("""
    SELECT DISTINCT nom FROM client WHERE idclient IN

```

```

        (SELECT idclient FROM facture WHERE idfacture IN
            (SELECT idfacture FROM detailfacture WHERE idproduit =
                (SELECT idproduit FROM produit WHERE nomproduit='Vis 20mm'))))
        AND ville = 'La-Cour-Basse'""")
affiche(cur)

#Autre version: au lieu de faire des sous-requêtes, on peut s'en
#sortir par jointure multiple. A manipuler avec précaution cependant
#(il est conseillé de ne pas dépasser des jointures à 3 tables,
#consigne non respectée ici.) Pour cela, pour les requêtes suivants,
#sauf exception, nous reviendrons dans la mesure du possible à des
#sous-requêtes.

cur.execute("""
    SELECT DISTINCT nom FROM
        client NATURAL JOIN facture NATURAL JOIN detailfacture
        NATURAL JOIN produit
    WHERE nomproduit='Vis 20mm' AND ville = 'La-Cour-Basse'""")
affiche(cur)

#4(d) La liste des achats des habitants du Vert-Pré

cur.execute("""
    SELECT nomproduit, quantite FROM detailfacture NATURAL JOIN produit
    WHERE idfacture IN
        (SELECT idfacture FROM facture WHERE idclient IN
            (SELECT idclient FROM client WHERE ville = 'Le-Vert-Pré'))""")
affiche(cur)

#4(e) Le nom et le pays des clients ayant commandé par internet

cur.execute("""
    SELECT nom, pays FROM client JOIN ville ON ville = nom_ville
    WHERE idclient in
        (SELECT idclient FROM facture WHERE type_vente = 'internet')""")
affiche(cur)

# Ici, la version par jointure peut se faire avec une jointure triple:

cur.execute("""
    SELECT nom, pays
    FROM client JOIN ville ON ville = nom_ville NATURAL JOIN facture
    WHERE type_vente = 'internet'""")
affiche(cur)

#4(f) Le nom des clients ayant acheté en magasin un parasol

cur.execute("""
    SELECT nom FROM client WHERE idclient IN
        (SELECT idclient FROM facture WHERE idfacture in
            (SELECT idfacture FROM detailfacture WHERE idproduit =
                (SELECT idproduit FROM produit WHERE nomproduit = 'Parasol'))
            AND type_vente = 'magasin')""")

```

```

affiche(cur)

#4(g) La liste des factures internet et le nom des clients

cur.execute("""SELECT idfacture, nom FROM facture NATURAL JOIN client
              WHERE type_vente='internet'""")
affiche(cur)

#4(h) Les clients ayant acheté de l'électro-ménager en magasin

cur.execute("""
SELECT nom, datefact, nomproduit FROM client NATURAL JOIN
  (SELECT idclient, nomproduit, datefact
   FROM facture NATURAL JOIN
     (SELECT idfacture, nomproduit
      FROM detailfacture NATURAL JOIN produit
      WHERE categorie = 'Électro-ménager')
   WHERE type_vente = 'magasin')""")
affiche(cur)

#La encore, on peut faire une jointure globale des 4 tables au lieu de
#ces jointures 2 à 2, au risque cependant de faire exploser la taille
#des tables.

#4(i) La liste des factures des clients étrangers + nom et pays.

cur.execute("""
SELECT idfacture, nom, pays FROM facture NATURAL JOIN
  (SELECT idclient, nom, pays
   FROM client JOIN ville ON ville = nom_ville
   WHERE pays <> 'France')""")
affiche(cur)

#4(j) Total, facture par facture + date et nom client

cur.execute("""
SELECT idfacture, datefact, nom, total FROM client NATURAL JOIN
  (SELECT idfacture, datefact, idclient, total
   FROM facture NATURAL JOIN
     (SELECT idfacture, SUM(quantite*prix)AS total
      FROM detailfacture NATURAL JOIN produit
      GROUP BY idfacture))""")
affiche(cur)

#4(k) Le client qui a la plus grosse

cur.execute("""
SELECT nom, max(total) FROM client NATURAL JOIN
  (SELECT idclient, total FROM facture NATURAL JOIN
     (SELECT idfacture, SUM(quantite*prix)AS total
      FROM detailfacture NATURAL JOIN produit
      GROUP BY idfacture))""")
affiche(cur)

```

#4(l) Moyenne des factures, client par client

```
cur.execute("""
SELECT nom, moyenne FROM client NATURAL JOIN
  (SELECT idclient, AVG(total) AS moyenne FROM facture NATURAL JOIN
    (SELECT idfacture, SUM(quantite*prix)AS total
      FROM detailfacture NATURAL JOIN produit
      GROUP BY idfacture)
    GROUP BY idclient)""")
affiche(cur)
```

#4(m) Facture dépassant les 500

```
cur.execute("""
SELECT idfacture, datefact, nom, total FROM client NATURAL JOIN
  (SELECT idfacture, datefact, idclient, total
    FROM facture NATURAL JOIN
      (SELECT idfacture, SUM(quantite*prix)AS total
        FROM detailfacture NATURAL JOIN produit
        GROUP BY idfacture))
WHERE total >= 500""")
affiche(cur)
```

#4(n) Factures dépassant la moyenne

```
cur.execute("""
SELECT idfacture, datefact, nom, total FROM client NATURAL JOIN
  (SELECT idfacture, datefact, idclient, total
    FROM facture NATURAL JOIN
      (SELECT idfacture, SUM(quantite*prix)AS total
        FROM detailfacture NATURAL JOIN produit
        GROUP BY idfacture))
GROUP BY idfacture
HAVING total > AVG(total)""")
affiche(cur)
```

#4(o) Quantité vendue en magasin et par internet, produit par produit

Première version, très lourde: le problème est de gérer le fait que
la somme ne se calcule que sur les produits ayant été vendus en
magasin pour le première, ou par internet. Le problème est alors de
compléter la liste obtenue en ajoutant les produits non vendus
(avec quantité 0)

```
cur.execute("""
SELECT * FROM
  (SELECT idproduit, SUM(quantite) AS vente_magasin
    FROM detailfacture NATURAL JOIN facture WHERE type_vente = 'magasin'
    GROUP BY idproduit
  UNION
  SELECT idproduit, 0 from produit WHERE idproduit NOT IN
    (SELECT idproduit FROM detailfacture NATURAL JOIN facture
```

```

        WHERE type_vente = 'magasin'))
    NATURAL JOIN
    (SELECT idproduit, SUM(quantite) AS vente_internet
    FROM detailfacture NATURAL JOIN facture WHERE type_vente = 'internet'
    GROUP BY idproduit
    UNION
    SELECT idproduit, 0 from produit WHERE idproduit NOT IN
    (SELECT idproduit FROM detailfacture NATURAL JOIN facture
    WHERE type_vente = 'internet'))""")
affiche(cur)

# Deuxième version: L'utilisation de l'instruction CASE WHEN permet
# une structure conditionnelle interne à la somme, permettant de
# distinguer les produits vendus des produits non vendus,
# traités différemment.

cur.execute("""
    SELECT idproduit,
        SUM(CASE WHEN type_vente='magasin' THEN quantite ELSE 0 END) ,
        SUM(CASE WHEN type_vente='internet' THEN quantite ELSE 0 END)
    FROM detailfacture NATURAL JOIN facture
    GROUP BY idproduit""")

affiche(cur)

#4(p) Les clients ayant acheté un tournevis en même temps que des vis

cur.execute("""
    SELECT nom FROM client WHERE idclient IN
    (SELECT idclient FROM facture WHERE idfacture IN
    (SELECT idfacture FROM (detailfacture NATURAL JOIN produit)
    WHERE nomproduit = 'Tournevis'
    INTERSECT
    SELECT idfacture FROM (detailfacture NATURAL JOIN produit)
    WHERE nomproduit LIKE 'Vis%'))""")
affiche(cur)

connection.commit()
connection.close()

```