Nom: Draghia Denisa Iulia

Classe:MPSI 1

Thème: une société qui vend de la nourriture et les données financiers sur ses ventes et ses clients.

La base de données: lien site-https://info.blaisepascal.fr/

Structure de la base de données:

Il y a 4 tables:

```
Structure de la table `CLIENTS`
CREATE TABLE CLIENTS (
CODE CLIENT char(5) NOT NULL,
 SOCIETE varchar(40) NOT NULL.
 ADRESSE varchar(60) NOT NULL,
 VILLE varchar(30) NOT NULL,
 CODE POSTAL varchar(10) NOT NULL.
PAYS varchar(15) NOT NULL,
CATEGORIE int(2) NOT NULL,
PRIMARY KEY (CODE CLIENT))
-- Structure de la table `PRODUITS`
CREATE TABLE PRODUITS (
REF PRODUIT int(6) NOT NULL,
NOM PRODUIT varchar(40) UNIQUE NOT NULL,
PRIX UNITAIRE float(8,2) NOT NULL CHECK
(PRIX UNITAIRE>=0),
PRIMARY KEY (REF PRODUIT)
);
-- Structure de la table `COMMANDES`
CREATE TABLE COMMANDES (
 NO COMMANDE int(6) NOT NULL,
 DATE COMMANDE date NOT NULL,
CODE CLIENT char(5) NOT NULL,
```

```
PRIMARY KEY (NO COMMANDE),
 FOREIGN KEY (CODE_CLIENT) REFERENCES CLIENTS
(CODE CLIENT)
);
-- Structure de la table `DETAILS COMMANDES`
CREATE TABLE DETAILS COMMANDES (
 NO_COMMANDE int(6) NOT NULL,
 REF PRODUIT int(6) NOT NULL,
 QUANTITE int(5) NOT NULL DEFAULT 1,
 PRIMARY KEY (NO COMMANDE, REF PRODUIT),
 FOREIGN KEY (NO COMMANDE) REFERENCES
COMMANDES (NO COMMANDE),
 FOREIGN KEY (REF PRODUIT) REFERENCES PRODUITS
(REF PRODUIT)
);
On utilise python 3 pour ouvrir la base de données. On tape:
import sqlite3
db=sqlite3.connect('facturation.sqlite')
pour faire ça.
Après, on tape le nom de la fonction qui on veut l'utiliser et ses
arguments s'il y a le cas. Pour chaque fonction est indiquée entre " "
qu'est qu'elle fait.
Par example, pour la fonction:
def cherche_numero_comm(prod):
"""input:nom d'un produit(char).On renvoie les numero des commandes qui contient le
produit introduit"""
req="SELECT NO_COMMANDE FROM DETAILS_COMMANDES,PRODUITS WHERE
PRODUITS.REF_PRODUIT=DETAILS_COMMANDES.REF_PRODUIT GROUP by
PRODUITS.REF PRODUIT HAVING NOM PRODUIT=?"
cursor.execute(req,(prod,))
liste=cursor.fetchall()
return liste
Si on tape 'Tofu', on va recevoir la liste:
[('Finlande',), ('Autriche',), ('Irlande',)]
```

On tape cherche_numero_comm(X) où à la place de X, on tape le produit qu'on veut et python va envoyer une liste avec tous les numéro des commandes qui ont contenu le produit qu'on a choisi.

Dans autre cas, il sera suffi de taper juste la fonction, sans aucun argument pour obtenir. Par exemple, pour la fonction :

def produit():

"""input:NONE.Renvoie la liste de produit de notre societe"""

req='SELECT NOM_PRODUIT FROM PRODUITS'

cursor.execute(req)

liste=cursor.fetchall()

return liste

On va juste taper produit() et on va recevoir une liste avec tout les produits vendues par la société:

[('Alice Mutton',), ('Aniseed Syrup',), ('Boston Crab Meat',), ('Camembert Pierr ot',), ('Carnarvon Tigers',), ('Chai',), ('Chang',), ('Chartreuse verte',), ("Ch ef Anton's Cajun Seasoning",), ("Chef Anton's Gumbo Mix",), ('Chocolade',), ('Es cargots de Bourgogne',), ('Filo Mix',), ('Geitost',), ('Genen Shouyu',), ('Gnocc hi di nonna Alice',), ('Gorgonzola Telino',), ("Grandma's Boysenberry Spread",), ('Gravad lax',), ('Gudbrandsdalsost',), ('Gula Malacca',), ('Ikura',), ('Inlagd Sill',), ('Ipoh Coffee',), ("Jack's New England Clam Chowder",), ('Konbu',), (' Laughing Lumberjack Lager',), ('Louglife Tofu',), ('Louisiana Fiery Hot Pepper S auce',), ('Louisiana Hot Spiced Okra',), ('Manjimup Dried Apples',), ('Mascarpon e Fabioli',), ('Maxilaku',), ('Mishi Kobe Niku',), ('Mozzarella di Giovanni',), ('Nord-Ost Matjeshering',), ('Northwoods Cranberry Sauce',), ('Outback Lager',), ('Pavlova',), ('Perth Pasties',), ('Queso Cabrales',), ('Queso Manchego La Past ora',), ('Raclette Courdavault',), ('Ravioli Angelo',), ('Sasquatch Ale',), ('Sc hoggi Schokolade',), ('Scottish Longbreads',), ('Singaporean Hokkien Fried Mee',), ("Sir Rodney's Marmalade",), ("Sir Rodney's Scones",), ('Spegesild',), ('Stee leye Stout',), ('Tarte au sucre',), ('Teatime Chocolate Biscuits',), ('Tofu',), ("Uncle Bob's Organic Dried Pears",), ('Valkoinen suklaa',), ('Vegie-spread',), ('Zaanse koeken',)]

Quelques exemples de requête complexe:

def repart_command():

"""input:NONE.Renvoie chaque societe et son nombre de commandes""" req="SELECT SOCIETE, count(*) as nb_comm FROM CLIENTS,COMMANDES WHERE CLIENTS.CODE_CLIENT=COMMANDES.CODE_CLIENT GROUP by CLIENTS.CODE CLIENT"

cursor.execute(req)

liste=cursor.fetchall()

return liste

Cette requête va envoyer une liste avec chaque société et le nombre de commandes qu'elle a fait.

```
def revenu_par_pays(pays): """input:pays(char)- on calcule la somme d'argent obtenu pour le pays qu'on introduit""" l=[] v=[]
```

req1="SELECT QUANTITE FROM DETAILS_COMMANDES JOIN COMMANDES on DETAILS_COMMANDES.NO_COMMANDE=COMMANDES.NO_COMMANDE JOIN CLIENTS on COMMANDES.CODE_CLIENT=CLIENTS.CODE_CLIENT WHERE PAYS=?"

cursor.execute(req1,(pays,))

for row in cursor:

l.append(row[0])

req2= "SELECT PRIX_UNITAIRE FROM PRODUITS JOIN DETAILS_COMMANDES on PRODUITS.REF_PRODUIT=DETAILS_COMMANDES.REF_PRODUIT JOIN COMMANDES on

DETAILS_COMMANDES.NO_COMMANDE=COMMANDES.NO_COMMANDE JOIN CLIENTS on COMMANDES.CODE_CLIENT=CLIENTS.CODE_CLIENT WHERE PAYS=?"

cursor.execute(req2,(pays,))

for row in cursor:

v.append(row[0])

somme=0

for i in range (len(l)):

somme=somme+l[i]*v[i]

return somme

Cette fonction envoie les revenus obtenus pour chaque pays où la société a vendu ces produits.