**Projet**

# Une application de vente de produits en Python

**Objectif** : Écrire une petite application réaliste complète de base de données en Python en utilisant en mode programme un SGBD relationnel.

# Cahier des charges

LCER est une entreprise qui vend des produits de toutes sortes (vêtements, voitures, jouets, etc.). Elle dispose de données sur son catalogue de produits, ses clients et leurs achats. Ces données sont dans une base de données relationnelle gérée par un SGBD relationnel, organisées en trois tables, dont un extrait est donné ci- dessous. On suppose qu’il n’y a pas plusieurs produits avec le même nom. Une ligne de Achat représente l’achat par le client d’une unité du produit ; expedie vaut oui si le produit a déjà été expédié au client, non sinon. Le debit d’un client est le montant de ses achats récents pas encore payés.

Produit

|  |  |  |  |
| --- | --- | --- | --- |
| **Idp** | **nom** | **prix** | **stock** |
| 99 | peluche | 5 | 1000 |
| 95 | chemisier | 20 | 500 |
| 92 | cravate | 15 | 9900 |
| 97 | porsche | 1000 | 12 |

Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Idc** | **nom** | **age** | **adresse** | **debit** |
| 27 | Rita | 17 | Rio | 20 |
| 29 | Riton | 18 | Cayeux | 0 |
| 23 | Jeanne | 9 | New York | 300 |

Achat

|  |  |  |  |
| --- | --- | --- | --- |
| **Ida** | **idp** | **idc** | **expedie** |
| 1 | 99 | 29 | oui |
| 2 | 95 | 27 | non |
| 3 | 99 | 23 | non |
| 4 | 97 | 27 | oui |

LCER a besoin d’un programme Python permettant à son dirigeant et aux clients d’effectuer certains traitements. Il se composera d’un ensemble de fonctions qui pourront être appelées depuis un menu par le dirigeant ou les clients de LCER au fur et à mesure de leurs besoins. (Pour simplifier, on écrira un seul menu utilisé à la fois par le dirigeant et les clients, au lieu de deux menus séparés.)

Les traitements prioritaires sont les suivants. Pour le dirigeant : expédition d’un achat, relance pour le paiement d’un débit, analyse des ventes ; et pour les clients : achat, paiement du débit. Le but de ce projet est d’écrire ce programme.

Pour simplifier, les autres traitements seront faits en mode interactif. Pour le dirigeant : ajout, modification et suppression d’un produit, réapprovisionnement des stocks, consultation de toutes les tables ; et pour les clients : inscription, consultation de ses informations, consultation des produits limitée au nom et au prix, consultation de ses expéditions en attente.

# Menu interactif et accès aux tables de LCER

Le menu interactif en Python est fourni. Par ailleurs, on n’inclura pas la création des tables dans le programme Python, dans la mesure où elle n’a lieu qu’une fois

#########################################################################################

# Ce menu contient les 5 fonctions à écrire. Toutes les autres manipulations

# sont faites par l'utilisateur dans un client interactif : les fonctionnalités demandées

# et aussi les interrogations de la base (utiles aussi pour débuguer).

#########################################################################################

def relance():

print("Ici la fonction relance.")

def analyse():

print("Ici la fonction analyse.")

def expedition():

print("Ici la fonction expedition.")

def achat():

print("Ici la fonction achat.")

def paiement():

print("Ici la fonction paiement.")

def menu():

while True:

print("""\

------------------------------------------

Bienvenue chez LCER

Entrez un entier :

0 : Sortir du menu et du programme

Menu dirigeant :

1 : Relance pour le paiement d'un débit

2 : Analyse des ventes

3 : Expédition d'un achat

Menu clients :

4 : Achat d'un produit

5 : Paiement du débit

------------------------------------------

""")

n = int(input('Entrez votre choix : '))

if n == 0:

return

elif n == 1:

relance()

elif n == 2:

analyse()

elif n == 3:

expedition()

elif n == 4:

achat()

elif n == 5:

paiement()

else:

print('Choix non disponible')

menu()

#########################################################################################

**Sqlite**

create table Produit(

idp int primary key,

nom varchar(20) not null,

prix real not null,

stock int not null,

check(0<prix and prix<2000), -- valeurs possibles dans fourchette raisonnable

check(0<=stock and stock<10000) -- valeurs possibles dans fourchette raisonnable

);

insert into Produit values(99,'peluche',5,1000);

insert into Produit values(95,'chemisier',20,500);

insert into Produit values(92,'cravate',15,9900);

insert into Produit values(97,'porsche',1000,12);

create table Client(

idc int primary key,

nom varchar(20) not null,

age int,

adresse varchar(20) not null,

debit real not null,

check(0<age and age<=130), -- valeurs possibles dans fourchette raisonnable

check(0<=debit and debit<2000) -- valeurs possibles dans fourchette raisonnable

);

insert into Client values(27,'Rita',17,'Rio',20);

insert into Client values(29,'Riton',18,'Cayeux',0);

insert into Client values(23,'Jeanne',19,'New York',300);

create table Achat(

ida int primary key,

idp int not null,

idc int not null,

expedie varchar(3) not null,

foreign key (idp) references Produit(idp),

foreign key (idc) references Client(idc),

check(expedie='oui' or expedie='non')

);

insert into Achat values(1,99,29,'oui');

insert into Achat values(2,95,27,'non');

insert into Achat values(3,99,23,'non');

insert into Achat values(4,97,27,'oui');

select \* from Produit;

select \* from Client;

select \* from Achat;

Pour simplifier, chaque fonction ne prendra aucun paramètre depuis le menu, lira au clavier les éventuelles valeurs dont elle a besoin, ouvrira elle-même une connexion à un client en mode programme et la fermera.

# Relance pour le paiement d’un débit (dirigeant)

D’abord, trouver un client en débit non nul (choisir simplement le premier dans la liste renvoyée par l’exécution de l’ordre SQL d’interrogation), s’il en existe. Si oui, construire alors une petite lettre de relance d’une ou deux lignes faisant figurer en en-tête le nom et l’adresse du client et, dans le texte, son nom et son débit. Ce texte sera simplement affiché par le programme (sur l’écran du dirigeant, et non mailé au client, pour simplifier).

Écrire le corps de la fonction correspondante dans le menu. Tester, en vérifiant après la fin de l’exécution de la fonction, depuis un client en mode interactif, le contenu de la base.

# Analyse des ventes (dirigeant)

Donner le chiffre d’affaires de l’entreprise, c’est-à-dire la somme de toutes ses ventes.

Lire au clavier le nom d’un client (qu’on supposera unique dans cet exercice uniquement), donner la somme des achats de ce client et donner le nombre des produits achetés par ce client.

Écrire le corps de la fonction correspondante dans le menu. Tester, en vérifiant après la fin de l’exécution de la fonction, depuis un client en mode interactif, le contenu de la base.

# Achat d’un produit (clients)

Lire au clavier le nom et l’identifiant du client. Si ce couple nom et identifiant n’existe pas, afficher un message de refus et arrêter. S’il existe continue le traitement : lire au clavier le nom du produit à acheter, ajouter l’achat et ajouter le prix au debit du client. (Pour simplifier et éviter de gérer des AUTOINCREMENT, la fonction lira exceptionnellement au clavier la valeur de l’ida du nouvel achat à insérer.)

Écrire le corps de la fonction correspondante dans le menu. Tester, en vérifiant après la fin de l’exécution de la fonction, depuis un client en mode interactif, le contenu de la base.

# Expédition d’un achat (dirigeant)

Trouver un achat pas encore expédié (choisir simplement le premier dans la liste renvoyée par l’exécution de l’ordre SQL d’interrogation), s’il en existe. Si oui décrémenter le stock du produit correspondant et modifier sa valeur pour la colonne expedie.

Écrire le corps de la fonction correspondante dans le menu. Tester, en vérifiant après la fin de l’exécution de la fonction, depuis un client en mode interactif, le contenu de la base.

# Paiement du débit (clients)

Lire au clavier l’identifiant du client, vérifier que son débit n’est pas nul et l’afficher afin que le client puisse entrer un montant de paiement (il est en effet possible de régler le débit en plusieurs fois), vérifier que ce montant est inférieur ou égal au débit, décrémenter le débit du montant entré et afficher le nouveau débit.

Écrire le corps de la fonction correspondante dans le menu. Tester, en vérifiant après la fin de l’exécution de la fonction, depuis un client en mode interactif, le contenu de la base.