REPUBLIQUE DU SENEGAL



# MINISTERE DE L'ENSEIGNEMENT SUPERIEUR DE LA RECHERCHE ET DE L'INNOVATION UNIVERSITE IBA DER THIAM DE THIES



## U.F.R DES SCIENCES ET TECHNOLOGIES- U.F.R DES SCIENCES ECONOMIQUES ET SOCIALES

<u>Projet</u>: création d'une application qui permet de faire l'interfaçage entre **PYTHON** et **MSQL** 

Master 1 en Sciences des Données et Application option (MSDA/IIA, SEM, CAA)

Matière : Base De Données relationnelle

Présenté par :

Kalidou Alassane Diao <u>Professeur</u>: Dr Youssoufa Ly

Gora Diallo

Khona Fall

Jean Marcel Waly Ndiogoye

Année Académique 2023 /2024

Dans le cadre de notre projet, nous avons réussi à faire l'interfaçage entre << Python>> et << MySQL>> avec deux applications différentes et proposer des requêtes qui pourront être exécuter correctement. Au cours de notre réalisation, nous songerons à faire des captures du travail fait tout en expliquant les lignes de codes.

#### Objectifs:

- . Montrer le fichier de la base de données sur MySQL Workbench,
- . Montrer le fichier python
- . Expliquer étape par étape les requêtes souhaitées

### **Explication des lignes de codes MySQL:**

Nous avons créé notre base de données nommée « Electroménager »

```
# Projet_base_de_donnees-----
# CREATION de la base de donnees-----
Create Database Electromenager;
Use Electromenager;
```

#### Création de la table « Produit »

```
# creation de la table Produit----
create table Produit(
   idproduit INT NOT NULL,
   nomproduit VARCHAR(25),
   prixUnitaire FLOAT NOT NULL,
   quantite INT NOT NULL,
   PRIMARY KEY (idproduit));
```

Insertion des valeurs dans la table « Produit »

Création de la table « Client »

```
create table Client(
    idClient INT NOT NULL,
    nom VARCHAR(30),
    prenom VARCHAR(40),
    ville VARCHAR(25),
    telephone INT NOT NULL,
    email VARCHAR(30),
    PRIMARY KEY(idClient));
Insertion des valeurs dans la table « Client »
 INSERT INTO Client(idClient, nom, prenom, ville, telephone, email)
 VALUES
    (1, 'NDIOGOYE', 'Jean Marcel Waly', 'Mboro', '772452809', 'marcel01@gmail.com'),
    (2, 'DIAO', 'Kalidou Alassane', 'Tamba', '783051784', 'diaokalidou@gmail.com'),
    (3, 'DIALLO', 'Gora', 'Matam', '760186543', 'gora123@gmail.com'),
    (4, 'FALL', 'Khona', 'Dakar', '709865432', 'fallkhona89@gmail.com'),
    (5, 'NDIAYE', 'Sokhna Diarra', 'Bignona', '770274680', 'boussodiarra@gmal.com');
Création de la table « Commande »
idCommande INT NOT NULL,
          idClient INT NOT NULL,
          idproduit INT NOT NULL,
          dateCommande DATE,
          PRIMARY KEY(idCommande),
          FOREIGN KEY (idClient) REFERENCES Client(idClient),
          FOREIGN KEY (idproduit) REFERENCES Produit(idproduit));
Insertion des valeurs dans la table « Commande »
 INSERT INTO Commande(idCommande, idClient, idproduit, dateCommande)
 VALUES
        (1, 5, 40, '2024-07-15'),
        (2, 2, 20, '2024-07-20'),
        (3, 1, 20, '2024-07-25'),
        (4, 4, 10, '2024-07-30'),
       (5, 1, 50, '2024-08-01'),
        (6, 1, 30, '2024-08-05'),
        (7, 3, 30, '2024-08-10'),
       (8, 3, 10, '2024-09-15'),
        (9, 5, 10, '2024-09-20'),
        (10, 4, 40, '2024-09-30');
```

Création de la table « CommandeProduit »

```
idCommande INT NOT NULL,
       idproduit INT NOT NULL,
       quantite INT NOT NULL,
       PRIMARY KEY (idCommande, idproduit),
       FOREIGN KEY (idCommande) REFERENCES Commande(idCommande),
       FOREIGN KEY (idproduit) REFERENCES Produit(idproduit));
Insertion des valeurs dans la table « CommandeProduit »
  Insert into CommandeProduit(idCommande, idproduit, quantite)
VALUES
   (1, 40, 4),
   (2, 20, 5),
   (3, 20, 8),
   (4, 10, 3),
   (5, 50, 4),
   (6, 30, 2),
   (7, 30, 5),
   (8, 10, 3),
   (9, 10, 3),
   (10, 40, 2);
```

### **EXPLICATION DES LIGNES DE Codes python:**

```
import mysql.connector
from mysql.connector import Error, errorcode
```

Dans ces deux lignes de codes, nous avons import le module « mysql.connector » qui nous permettra de nous connect07/11/2024er à notre base de données et « Error » et « errorcode » pour gérer les erreurs liés à cette connexion

```
config = {
    'user': 'root',
    'password': 'Kalash@2023',
    'host': '127.0.0.1',
    'database': 'Electromenager',
    'raise_on_warnings':True
}
```

Ici, nous enregistrons les informations liées à notre base de données créée dans MySQL dans une variable appelée config

```
try:
    cnx = mysql.connector.connect(**config)
```

Ce bout de code nous permet de gérer les erreurs potentielles rencontrées lors de la connexion en retournant certaines phrases nous permettant de savoir exactement les erreurs commises

Nous passons maintenant aux requêtes effectuées et les résultats retournés sont mis sous forme de liste et sont affichés après chaque requête.

Dans cette capture d'écran se trouve notre première requete après que la connexion à la base de données ait réussi. Cette requête nous renvoie les produits dont le prix unitaire est supérieur à 25000 FCFA.

```
paragraphe = 1
cursorSel = cnx.cursor()
cnx.commit()
Aff_query = "SELECT nomproduit FROM Produit WHERE prixUnitaire > 25000"
cursorSel.execute(Aff_query)
resultats = cursorSel.fetchall()
print(resultats)
```

nomproduit
Iphone Xr
Refrigerateur
Micro-ondes
Cuisiniere
Machine a laver

Celle-ci retourne les prénoms et les noms des clients dont l'idClient est supérieur ou égal à 3

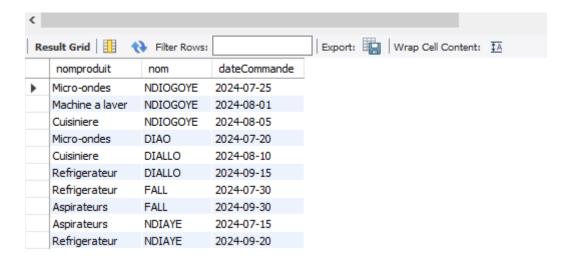
```
kali = "SELECT nom, prenom FROM Client WHERE idClient >= 3"
cursorSel.execute(kali)
result = cursorSel.fetchall()
print(result)
```



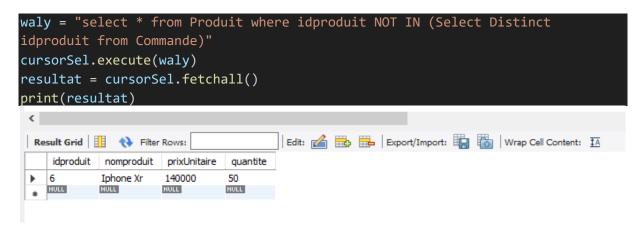
La capture ci-dessus renvoie, s'il y'en a, les clients qui n'ont pas encore effectué de commande, sinon la phrase « aucun résultat trouvé ». Et pour cela, nous avons fait une jointure entre les tables Client et Commande.

Cette requête retourne les produits, les noms des clients ainsi que les dates des commandes en utilisant la jointure entre nos trois tables que sont : Produit, Client et Commande

```
pro_com = "select nomproduit, nom, dateCommande from Produit, Client, Commande
where Produit.idproduit = Commande.idproduit AND Client.idClient =
commande.idClient"
cursorSel.execute(pro_com)
answer = cursorSel.fetchall()
if answer:
    print("les resultats sont:")
    for row in answer:
        print(f"nomproduit: {row[0]}, nom: {row[1]}, dateCommande: {row[2]}")
else:
    print("Pas de resultats")
```



Dans cette requête, nous voulons avoir la liste des produits qui n'ont pas été commandés



Ici, nous effectuons la mise à jour du prix unitaire de l'un de nos produits qu'est la machine à laver.

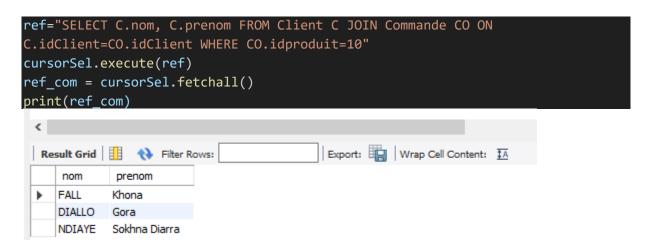
```
gora = "UPDATE Produit SET prixUnitaire = 190000 WHERE idproduit=50"
cursorSel.execute(gora)
diallo = cursorSel.fetchall()
if diallo:
    print("echec")
else:
    print("maj effectue")
```

Pour vérifier que la mise à jour a été effectué, on a affiché le prix unitaire d'une machine à laver après la requête précédente



Là, nous cherchons les produits commandés par le client « NDIOGOYE Jean Marcel Waly »

Nous cherchons à trouver les clients qui ont commandé au moins un Réfrigérateur



Affichage des commandes effectuées en Aout 2024

```
bar="SELECT idproduit, idclient FROM Commande WHERE dateCommande BETWEEN
'2024-08-01' AND '2024-08-31'"
cursorSel.execute(bar)
rab= cursorSel.fetchall()
print(rab)
```

