

**PROJET DE BASE DE DONNEES:**

**GESTION IMMOBILIERE :**

**Participants :**

Mbombé Gueye

Daouda Dieng

Mamadou Diagne

Cheikh Ibrahima Fall

**Professeur :** Youssouf Ly

1. **Introduction**

Le secteur de la gestion immobilière est en pleine croissance, et l’utilisation d’outils numériques devient indispensable pour faciliter la gestion des biens et des locataires. Ce projet vise à développer une application de gestion immobilière qui permet de simplifier les opérations quotidiennes telles que la gestion des propriétés, des locataires, des loyers, et des paiements.

L'application fera l'interface entre Python et SQL, permettant de stocker, récupérer et manipuler les informations relatives aux biens immobiliers dans une base de données SQL. Python, avec ses nombreuses bibliothèques, sera utilisé pour automatiser et rationaliser ces processus, tandis que SQL assurera un stockage et une récupération efficaces des données. Ce projet constitue un outil puissant pour toute entreprise ou particulier cherchant à optimiser la gestion de son parc immobilier.

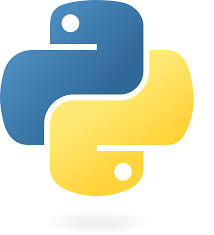
* **Objectifs du projet :**

Notre projet a pour but de créer une application pour une base de données en gestion immobilière en utilisant des technologies comme MySQL et Python pour mieux centraliser et gérer efficacement les informations liées aux biens immobiliers, aux clients, aux transactions et aux agents immobiliers. Mais aussi de permettre de faciliter la recherche, la mise à jour et l'analyse des données.

* **Technologies utilisées :** MySQL pour la base de données et Python pour l’interfaçage
* **MySQL :** est l'un des systèmes de gestion de bases de données relationnelles les plus populaires au monde. Développé à l'origine par MySQL AB, puis acquis par Oracle Corporation, il est apprécié pour sa robustesse, sa flexibilité et sa capacité à gérer de grandes quantités de données. MySQL permet aux développeurs de créer, modifier et interroger des bases de données à l'aide du langage SQL (Structured Query Language), facilitant ainsi la gestion des données pour des applications variées, allant des sites web aux systèmes d'information complexes. Grâce à sa communauté active et à son écosystème riche en outils et bibliothèques, MySQL s'impose comme un choix privilégié pour de nombreux projets, tant pour les petites entreprises que pour les grandes organisations.

****

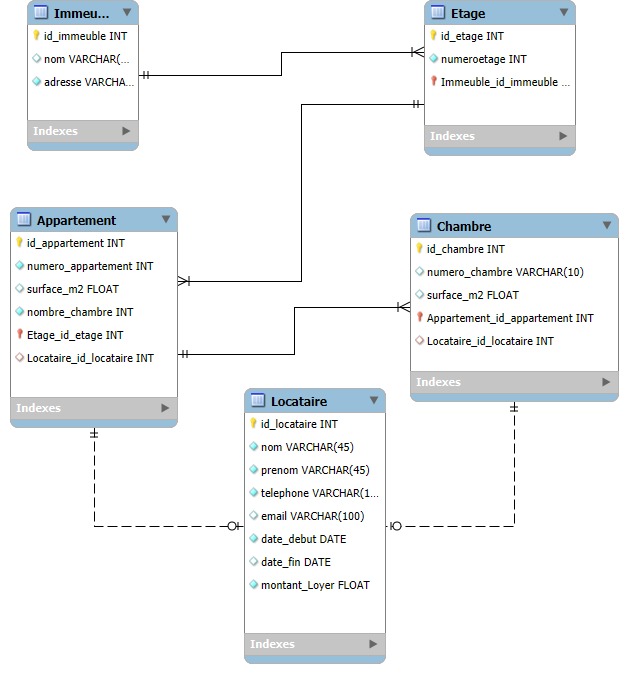
* **Python :** est un langage de programmation polyvalent et facile à apprendre, largement utilisé dans le développement d'applications, l'analyse de données et le machine Learning. L'une de ses applications les plus puissantes est la gestion des bases de données. Que vous travailliez avec des bases de données relationnelles comme MySQL, PostgreSQL ou SQLite, ou avec des bases de données NoSQL comme MongoDB, Python offre une multitude de bibliothèques et d'outils pour interagir avec vos données de manière efficace.

****

1. **Partie Base de Données**

**1. Modèle Conceptuel des Données (MCD)**

* Présentation du schéma conceptuel de la base de données



* Définition des entités : Immeuble, Etage, Appartement, Chambre, Locataire

2. Règles de Gestion

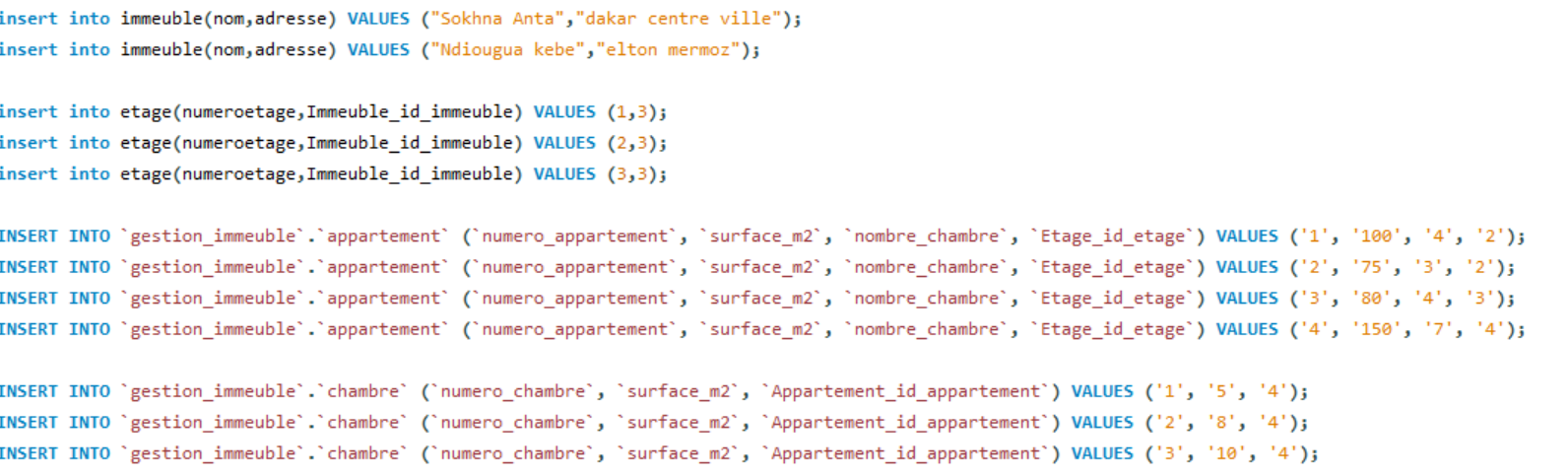
* **Immeuble :**
* Chaque immeuble doit avoir un nom et une adresse unique.
* Un immeuble peut contenir plusieurs étages.
* **Etage :**
* Un étage est lié à un immeuble.
* Un étage peut contenir plusieurs appartements.
* Un numéro d’étage ne peut pas être dupliqué dans le même immeuble.
* Un étage ne peut pas être supprimé s’il contient des appartements.
* **Appartement :** 
* Chaque appartement est lié à un étage.
* Un appartement peut contenir plusieurs chambres.
* Le numéro d’appartement est unique par étage.
* Un appartement ne peut pas être supprimé s’il contient des chambres.
* **Chambre :**
* Chaque chambre est liée à un appartement.
* Le numéro de chambre doit être unique dans l’appartement.
* Une chambre peut être attribuée à un locataire.
* **Locataire : **
* Chaque locataire a des informations obligatoires: nom, prénom, téléphone, email.
* Un locataire peut occuper un appartement ou une chambre.
* Avant de supprimer un locataire, il doit être dés assigné de l'appartement ou de la chambre qu’il occupe.
* Un locataire ne peut occuper qu'un seul appartement ou une seule chambre à la fois.

**3. Création de la base de données :**

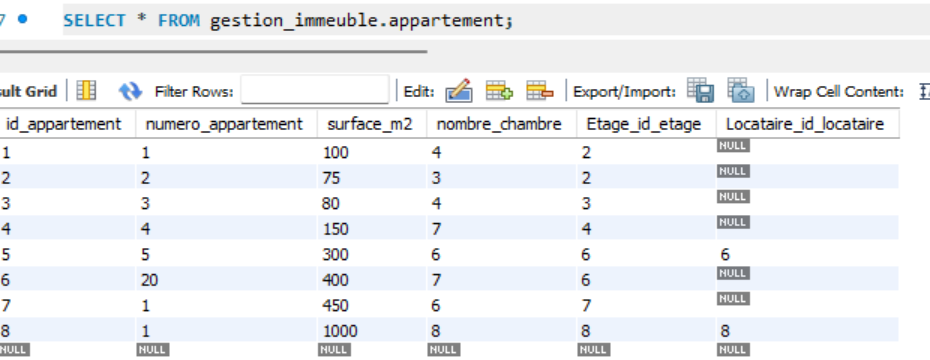
* Script SQL pour créer les tables :
* **Immeuble**
* **Etage**
* **Appartement**
* **Chambre**
* **Locataire**

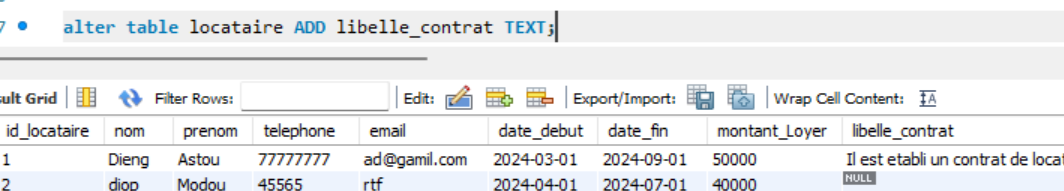
1. **Manipulation des données:**

* Insertion des données (exemples pratiques)



* Requêtes de sélection, mise à jour, suppression





**III. Partie Interfaçage avec Python:**

* 1. **Présentation de l’Interfaçage :**
* **Rôle de Python dans la gestion et manipulation des données via une interface textuelle**

Le code Python joue un rôle clé dans la gestion et la manipulation des données dans un projet de gestion immobilière, en offrant une interface textuelle pour interagir avec la base de données de manière simple et efficace. Voici quelques points essentiels sur ce rôle :

1. Facilité d'accès aux données : Grâce à Python, les utilisateurs peuvent interroger la base de données et afficher des informations importantes (immeubles, locataires, appartements, etc.) de manière lisible et structurée. Par exemple, avec des bibliothèques comme pandas, il est possible de présenter les données sous forme de tableau, facilitant leur lecture et manipulation.
2. Automatisation des tâches répétitives : Python permet d'automatiser des actions comme l'ajout de nouveaux enregistrements (locataires, appartements) ou la mise à jour des données, ce qui réduit considérablement le risque d'erreurs humaines et accélère le processus de gestion.
3. Gestion des erreurs : Le code Python, en gérant les exceptions, assure que l'application ne plante pas en cas d'erreurs. Cela permet d'identifier et de corriger les problèmes tout en maintenant le bon fonctionnement du système. Par exemple, si une insertion échoue à cause de contraintes dans la base de données, l'application peut capturer cette erreur et offrir une solution alternative à l'utilisateur.
4. Interaction intuitive : L'interface textuelle permet une interaction simple où l'utilisateur peut entrer des informations et recevoir des retours en temps réel sur les actions effectuées. Ce type d'interface est moins complexe à implémenter qu'une interface graphique, tout en restant fonctionnel et pratique.
5. Sécurité des données : Les requêtes SQL intégrées dans le code Python sont paramétrées pour prévenir les injections SQL, assurant ainsi un accès sécurisé aux données.
   1. **Connexion à la Base de Données :**

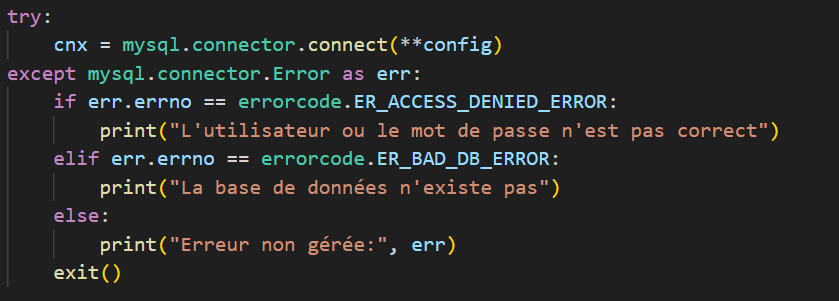
* **Détails sur la configuration de la connexion entre Python et la base MySQL**

Une fois le connecteur installé avec “ **pip install mysql-connector-python** ”, la connexion à la base de données se fait via les informations d'identification comme l'hôte, l'utilisateur, le mot de passe, et le nom de la base de données.

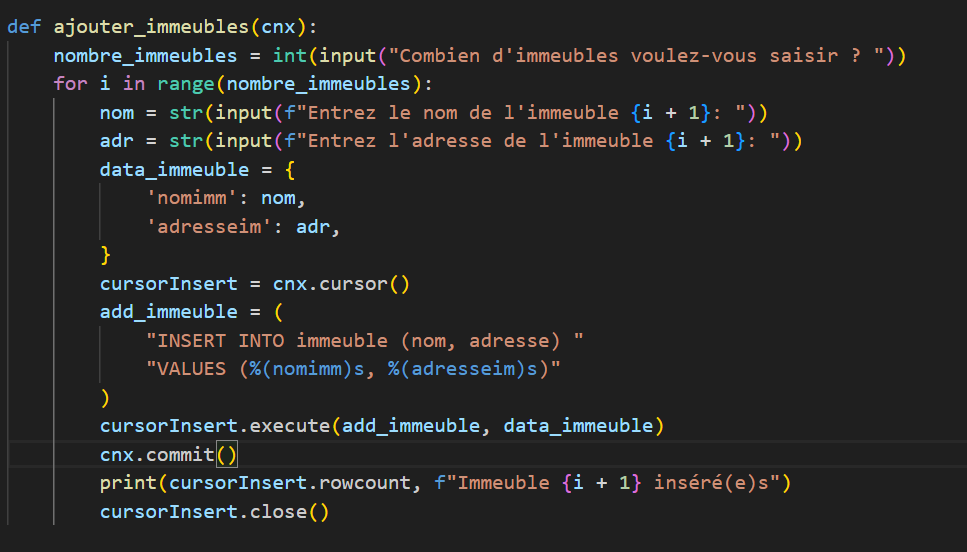


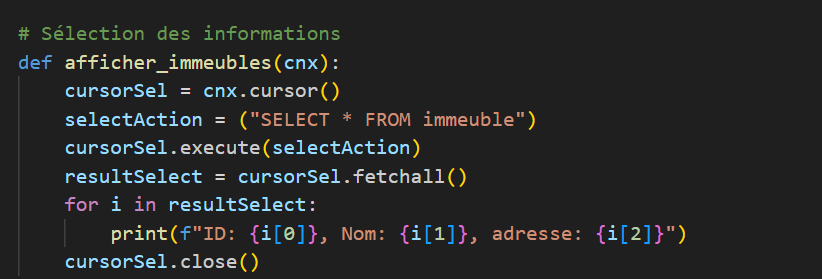
* **Gestion des erreurs de connexion**

La gestion des erreurs de connexion dans une application Python liée à MySQL est essentielle pour éviter que l'application ne plante en cas de problème de communication avec la base de données.



* 1. **Gestion des Immeubles**

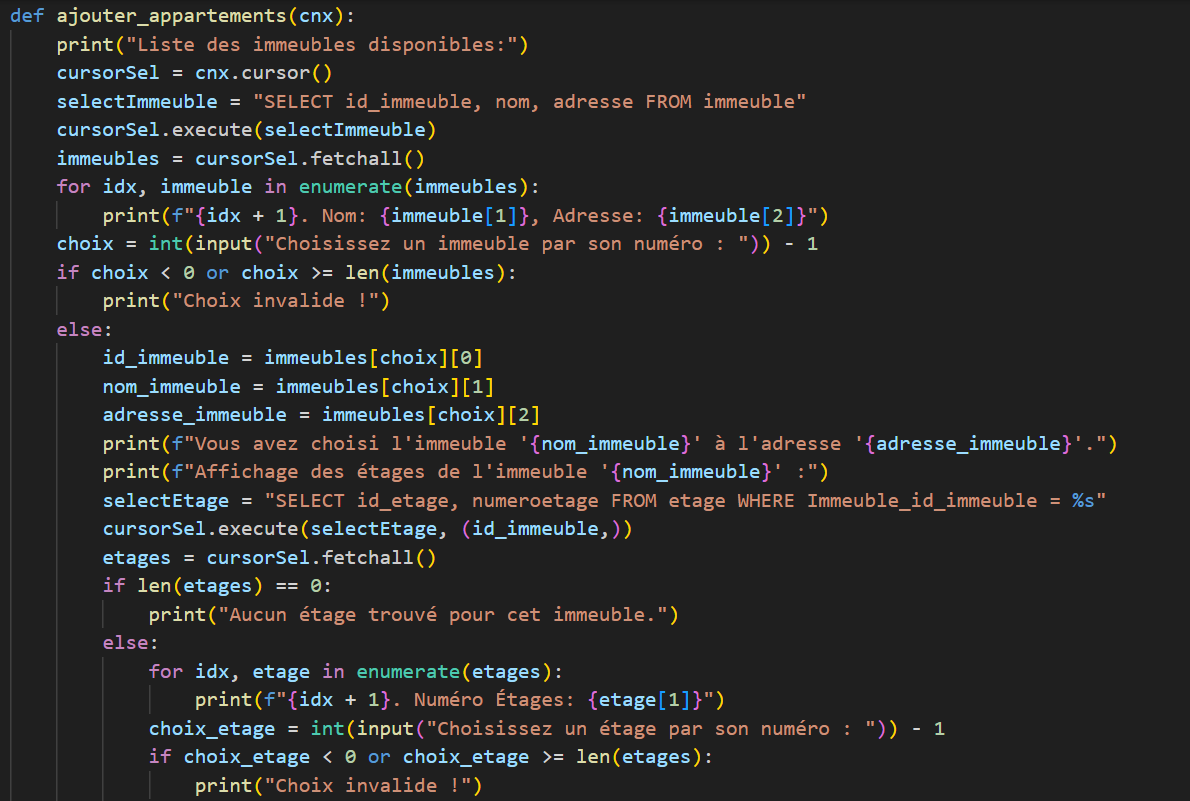
****

****

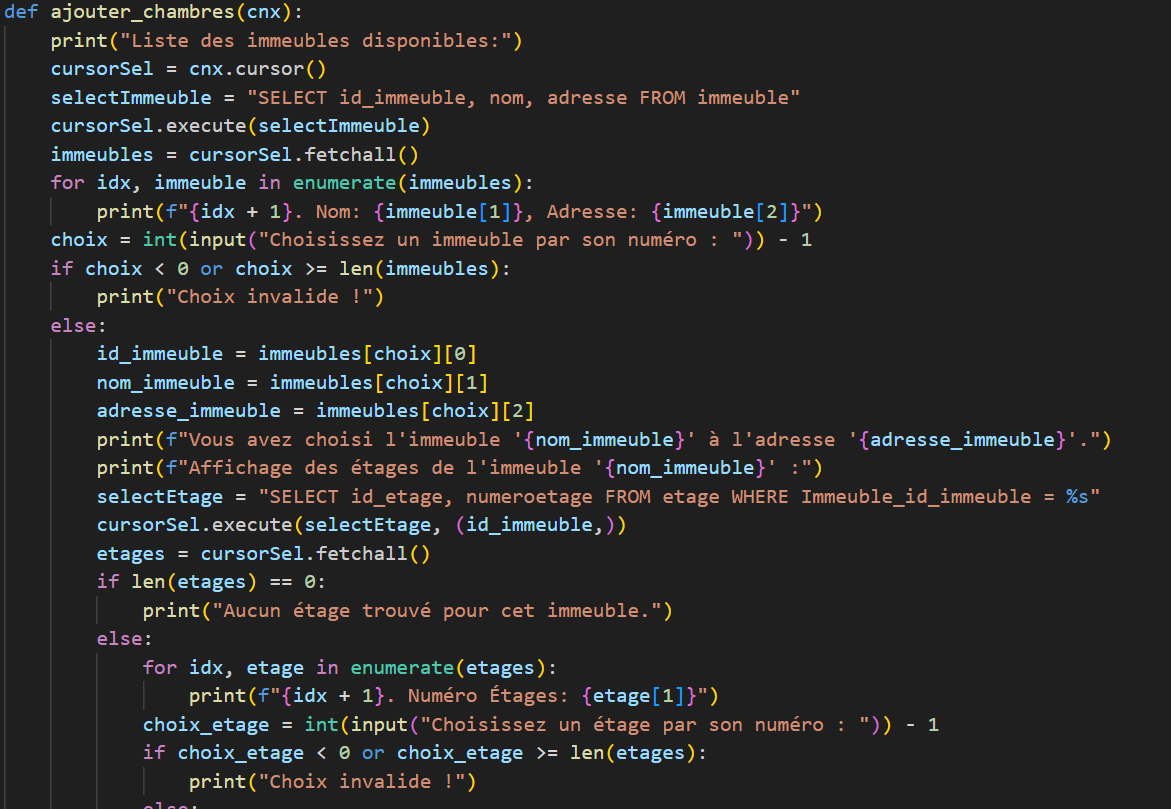
* 1. **Gestion des Étages**

****

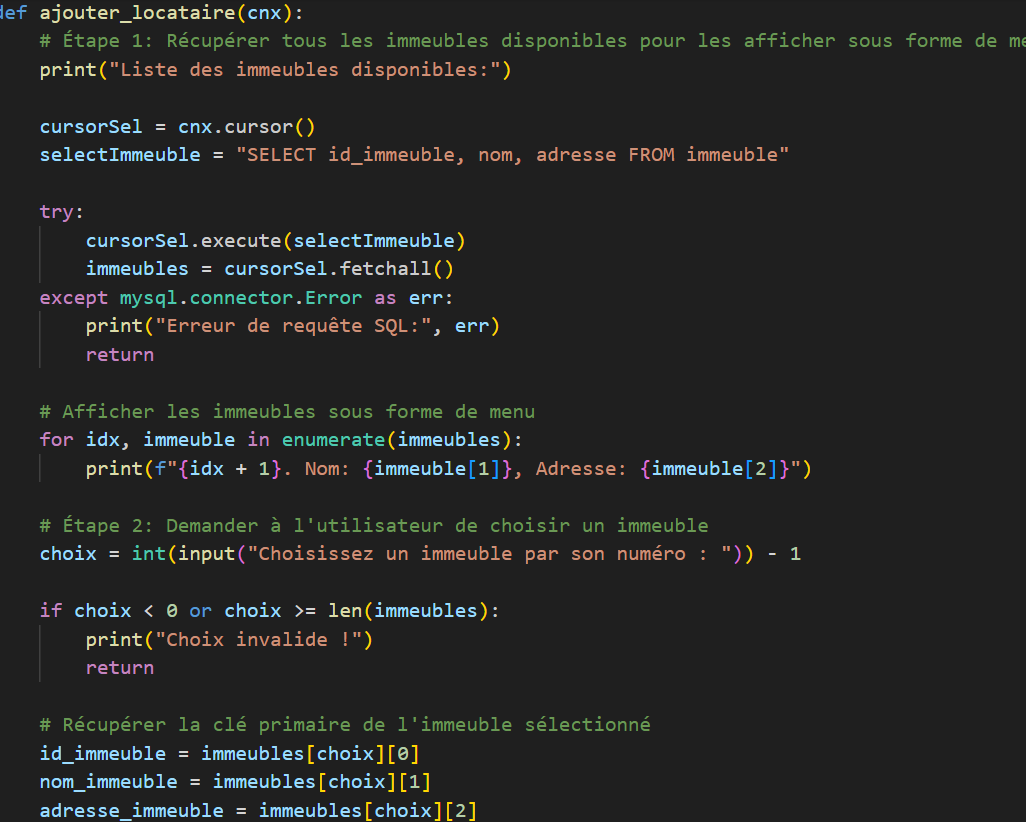
* 1. **Gestion des Appartements**

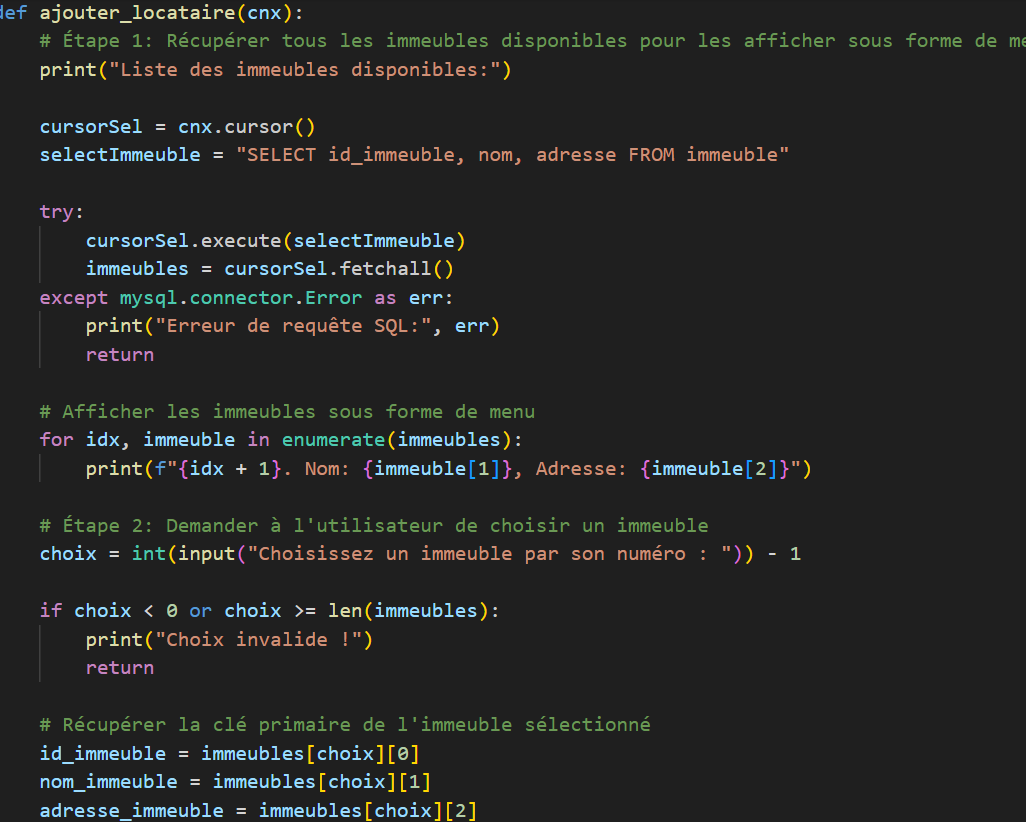
****

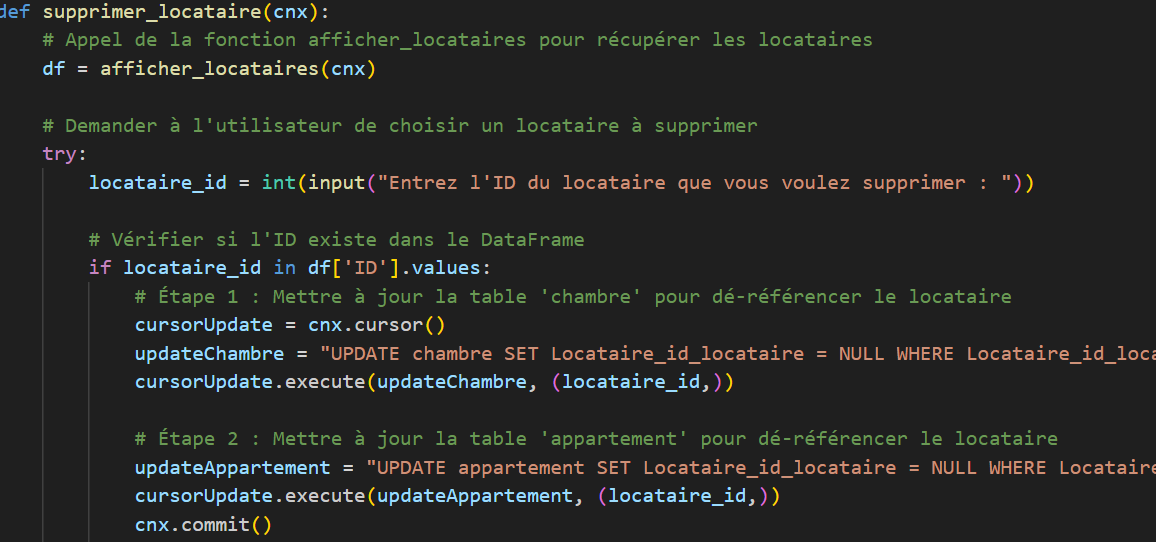
* 1. **Gestion des chambres**

****

* 1. **Gestion des Locataires**

****

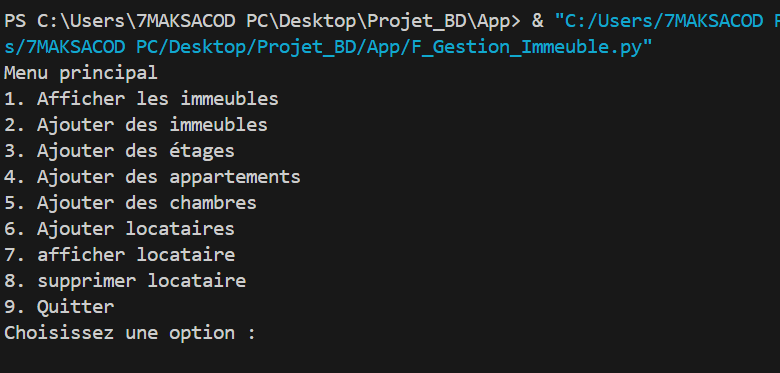
****

****

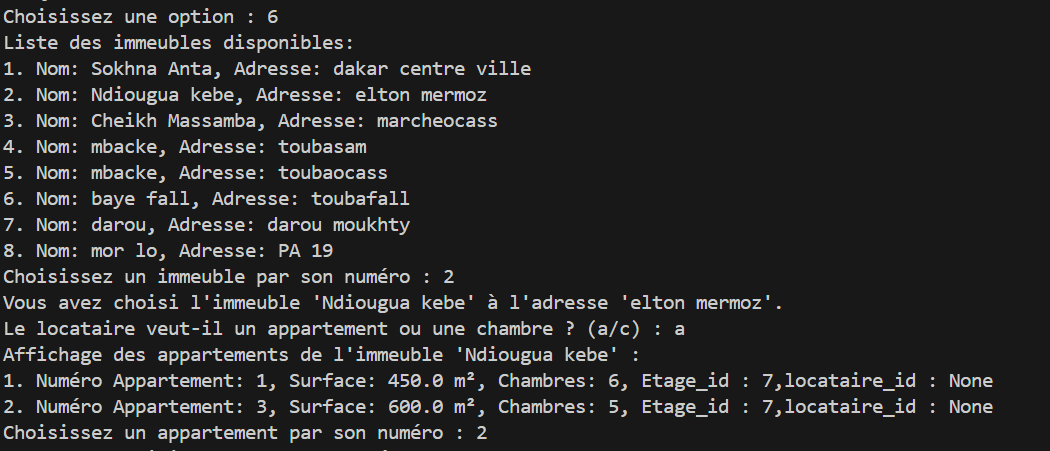
Ces fonctions illustrent comment le code Python interagit avec la base de données MySQL pour réaliser des opérations CRUD (Créer, Lire, Mettre à jour, Supprimer). Le code complet, qui contient l'ensemble des fonctionnalités, est disponible sur GitHub, permettant une meilleure compréhension et extension du projet.

* 1. **Lancement du programme**

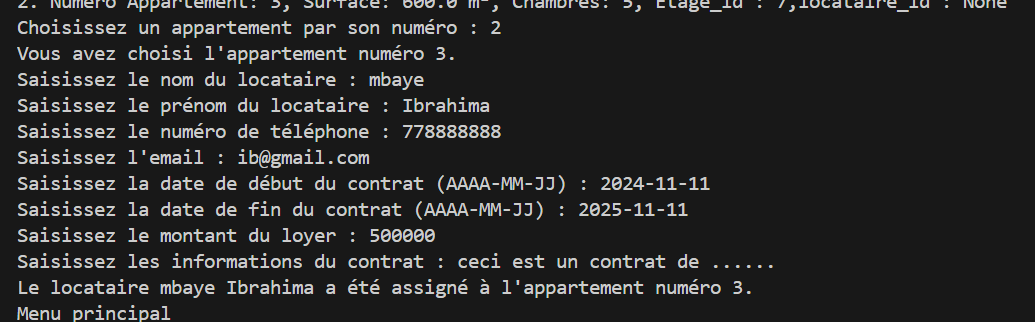
Si on lance l ‘application voici le menu qui s’affiche

****

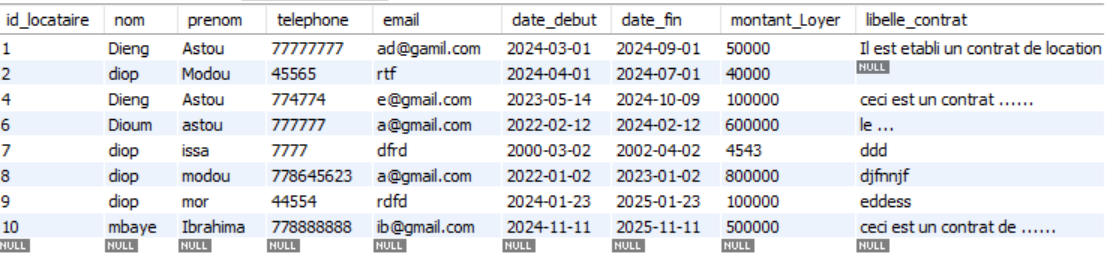
Prenons l’exemple de l’option 6 : ajout d’un locataire



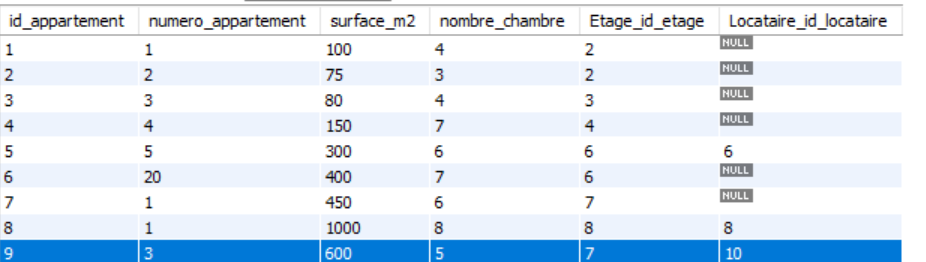
Donc ici si on veut ajouter un locataire on affiche d’abord la liste des immeubles disponible après avoir choisi y’a les options entre un appartement ou une chambre ,dans le cas présent on choisit un appartement ensuite on nous affiche les appartements disponibles et enfin on fait le choix sur le type d’appartement choisi .



Ici on ne fait que remplir les éléments essentiels du locataire et après remplissage on peut visualiser les données directement dans MySQL pour être bien sûr que ces données sont bien enregistrées dans la base donnée.



Une fois que le locataire a été assigné à l’appartement choisi, automatiquement sa clé primaire migre vers la table appartement comme clé étrangère qui est par défaut nulle pour montrer que cet appartement est d’Aure avent occupé.

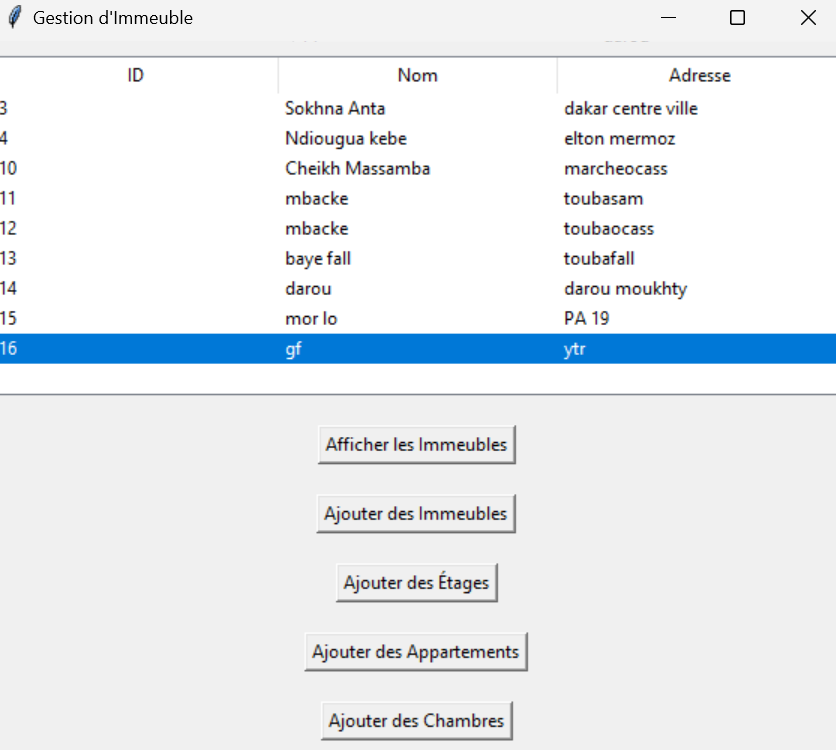


* 1. **Partie graphique**

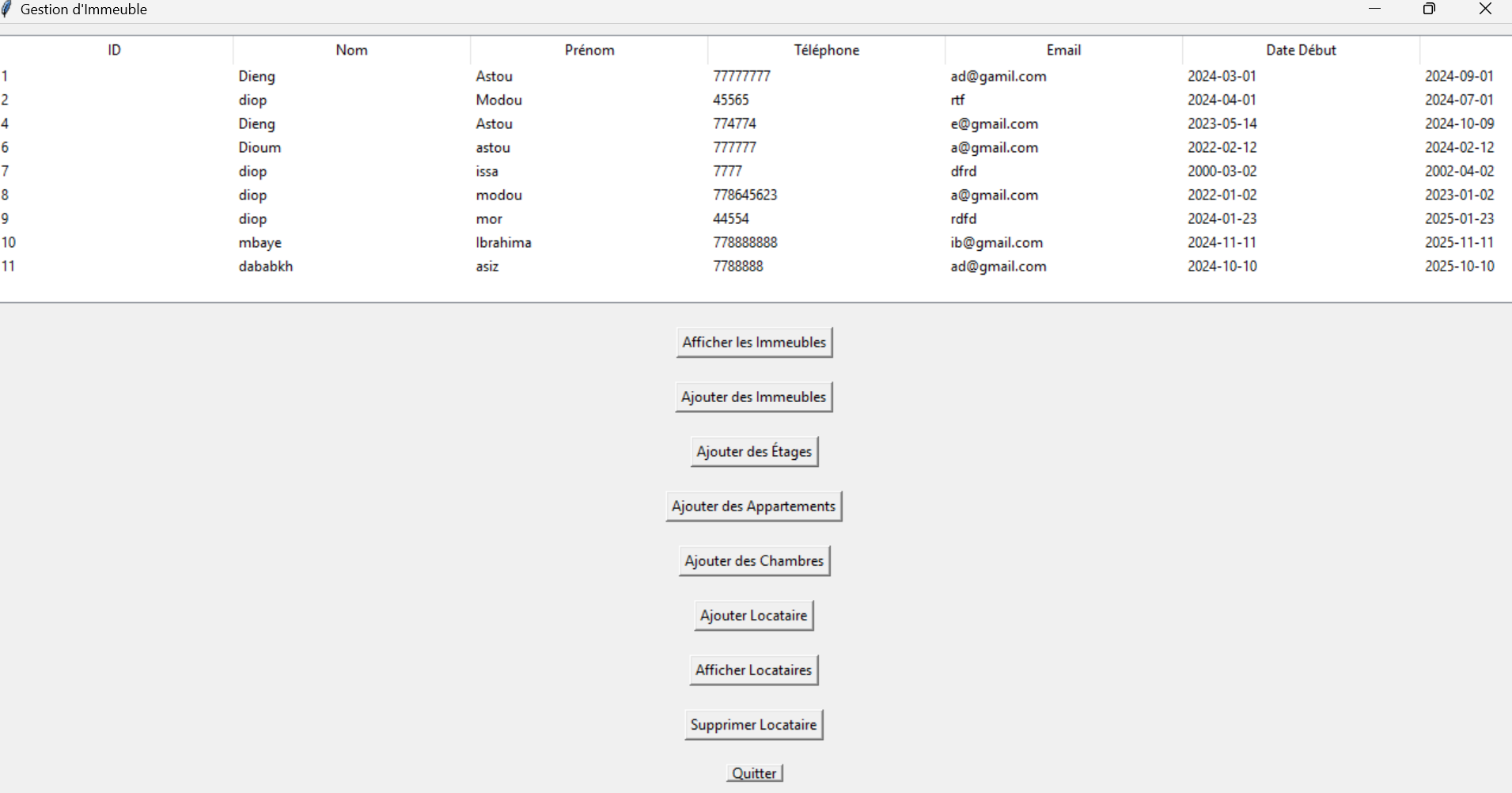
Utilisation de la bibliothèque **tkinter**

**Tkinter** est une bibliothèque standard de Python qui permet de créer des interfaces graphiques simples et interactives pour des applications de bureau.

**Affichage des immeubles**

****

**Affichage des locataires**

****

À partir de l’interface on peut aussi insérer des locataires, des immeubles, ajouter des locataires ….

**IV. Conclusion**

La création de notre base de données en gestion immobilière utilisant MySQL et Python nous offre une solution robuste et flexible pour gérer efficacement les informations liées aux immeubles, aux chambres, aux appartements, aux locataires, aux transactions et aux clients. MySQL, en tant que système de gestion de bases de données relationnelles, nous permet de structurer les données de manière logique et sécurisée, facilitant ainsi les requêtes complexes et l'intégrité des données.

L'intégration de Python dans ce système nous permet de développer des scripts automatisés pour des tâches telles que l'insertion de données, les mises à jour et l'analyse des tendances du marché immobilier.

En résumé, l'utilisation de MySQL combinée avec Python nous on crée une solution dynamique pour notre gestion immobilière, nous permettant non seulement une gestion efficace des données, mais également des analyses avancées pour optimiser les opérations et améliorer la satisfaction client. Cette approche favorise l'adaptabilité aux évolutions du marché et contribue à la prise de décision éclairée.