

Brief “Biens immobiliers”

0. Consignes

Le projet est individuel

Pour mettre à jour le projet, on fait push de la branche en Github

Livrables :

Dépôt Github : il devra contenir tous les fichiers de l'application, le READ-ME, le fonctionnement de l'application dans le READ-ME, réponses aux exercices sous forme de wiki (nom du fichier : “Titre de l'exercice_nom_prenom”)

Deadline : Mardi 07 décembre 2021 minuit

Compétences visées :

C2 : Concevoir une base de données analytique

C3 : Programmer l'import de données

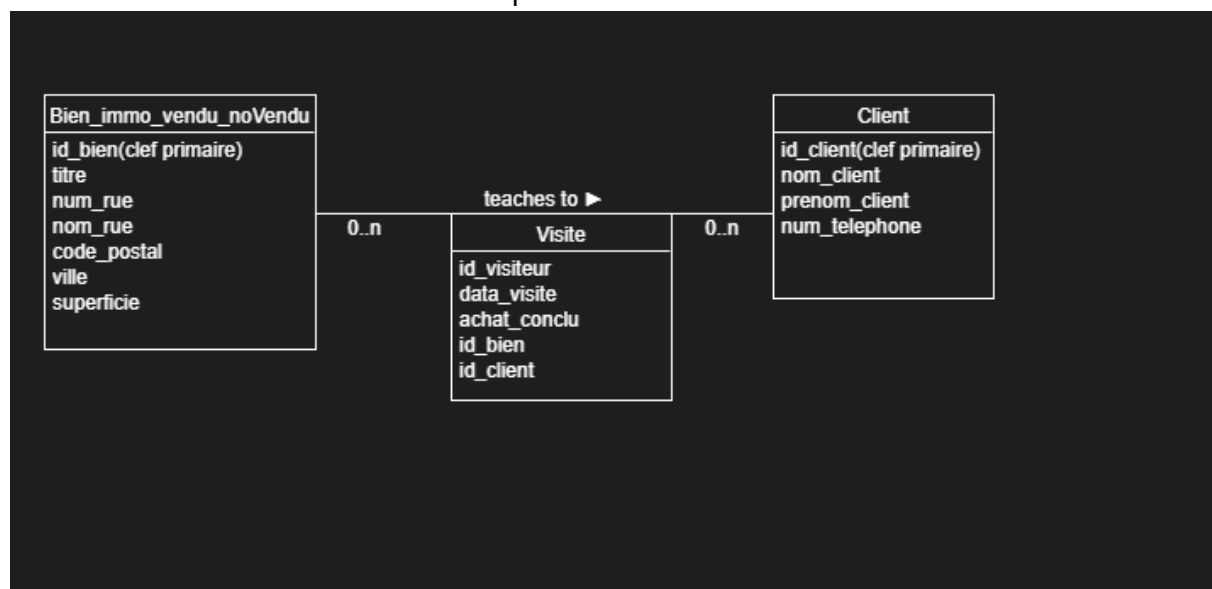
C4 : Préparer les données disponibles depuis la base de données analytique

1. Modélisation

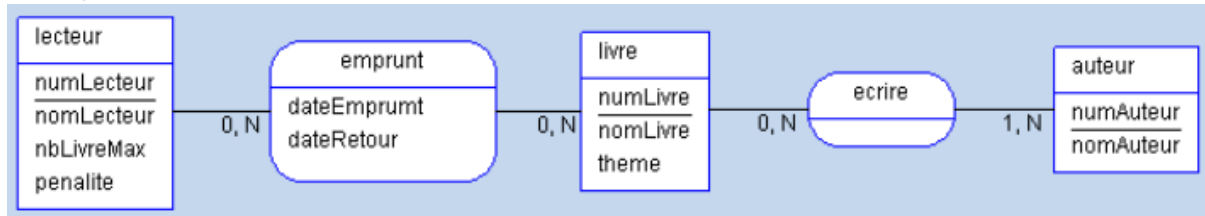
Définir un modèle conceptuel (MCD) pour obtenir des diagrammes Entité / associations. Ils seront ensuite traduits en modèle logique : des tables relationnelles.

Agence immobilière :

Une agence immobilière veut conserver les informations sur les biens immobiliers (maisons, appartements) qu'elle vend et sur ces clients. Les biens immobiliers vendus et à vendre sont numérotés et décrits par un titre, un numéro de rue, une rue, un code postal, une ville, une superficie. Les clients sont aussi numérotés et sont décrits par un nom, un prénom, et un numéro de téléphone. Les clients visitent les biens. A chaque visite, il faut la décrire par une date et retenir si la visite s'est conclue par un achat.



Exemple de MCD :



Exemple de MLD Table relationnelle :

Lecteur (numlecteur, nonLecteur, nbLivreMax)

Livre (Numlivre, nomLivre)

Auteur(NumAuteur, NomAuteur)

Emprunt(numLecteur, numlivre, dateEmprunt, dateRetour)

Ecrire(numLivre, numAuteur)

2. Choix d'un SGBD

Une BDD : Contient un ensemble d'informations qui sont :

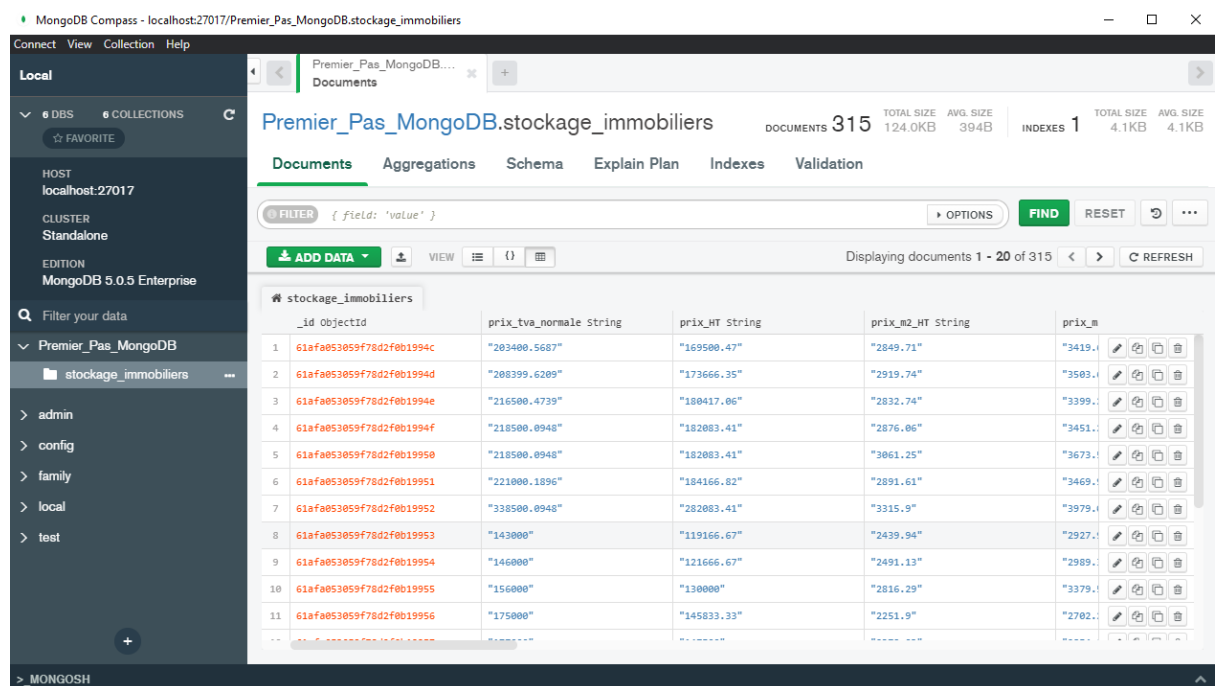
- Stockées,
- Accessibles et
- Gérées

À l'aide d'un Système de Gestion de Base de Données.

1. Qu'est-ce qu'un SGBD : C'est un Système de gestion de base de données relationnel.
2. Donner 3 exemples de SGBD relationnel et non relationnel : MySQL, Maria DB, Oracle
3. Quelle est la différence entre un SGBD relationnel et non relationnel : La différence se situe au niveau du stockage des données. La SGBD relationnel stocke les données dans les tables ; tandis que les SGBD non relationnelles stockent les données au format clé-valeur c'est-à-dire stocker d'avantages des grosses quantités des données
4. Dans quel cas choisir un SGBD relationnel ou non relationnel :
Le choix d'un SGBD relationnel : C'est lorsque on a besoin de l'intégrité des données avec un stockages des données dans les tables. Tandis que la NoSQL est plutôt avantageuse en termes de stockage des données volumineuses quand on a une grande quantité de données en termes de vitesse d'accessibilité. D'où la différence entre les BDD SQL et NoSQL est la façon de stocker les données qui diffèrent : L'une stocke les données dans des tables tandis que l'autre les stockées au format clé-valeur de manière à stocker plusieurs en termes de quantité .

3. Mise en place d'une base de données non relationnelle

1. Enumérer les différents types de bases de données NoSQL (clé valeur, document, etc, ...)
 Base de documents : Les données dynamiques y sont stockées, au format JavaScript, Object Notation (JSON). Exemple : CouchDB
 Base de données de colonnes : Utile pour l'analyser des données. Exemple : Apache Cassandra
 Bases de données en cache : Les données sont stockées sur le disque ou le cache. Exemple : Memcache
 Base de données graphiques : Elle est composée de nœuds et crée des relations en utilisant des arêtes. Exemple Oracle NoSQL
2. Mettre en place un POC sur l'utilisation de l'une de ces catégories au choix (démonstration de 5min le mardi 7 décembre après-midi chacun) ; vous êtes libre d'utiliser la bdd de votre choix

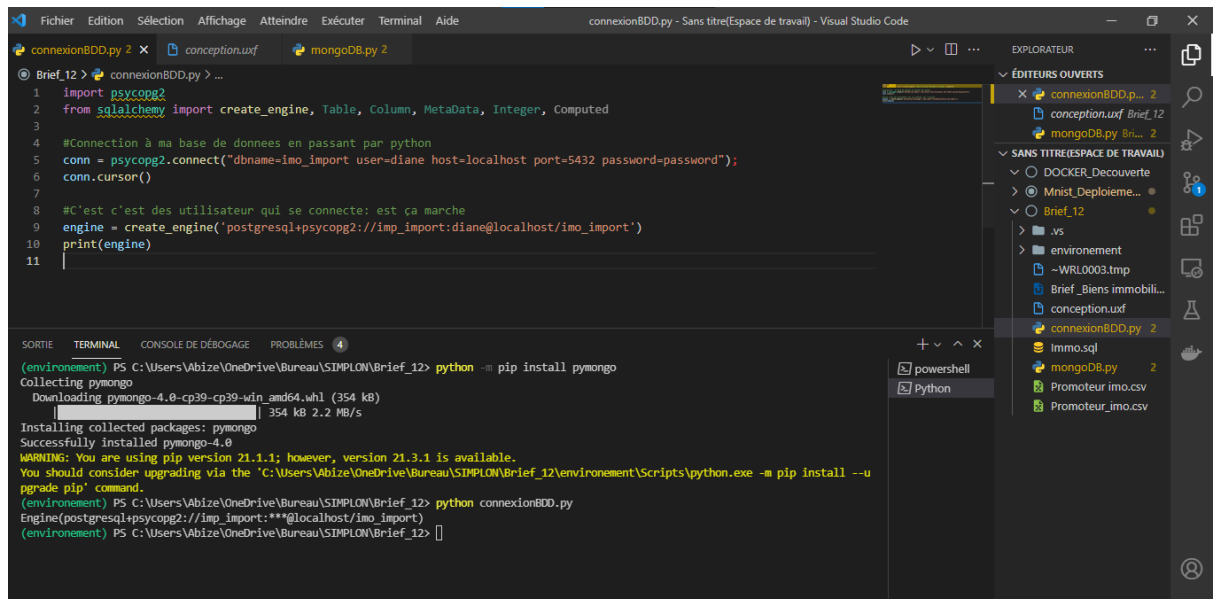
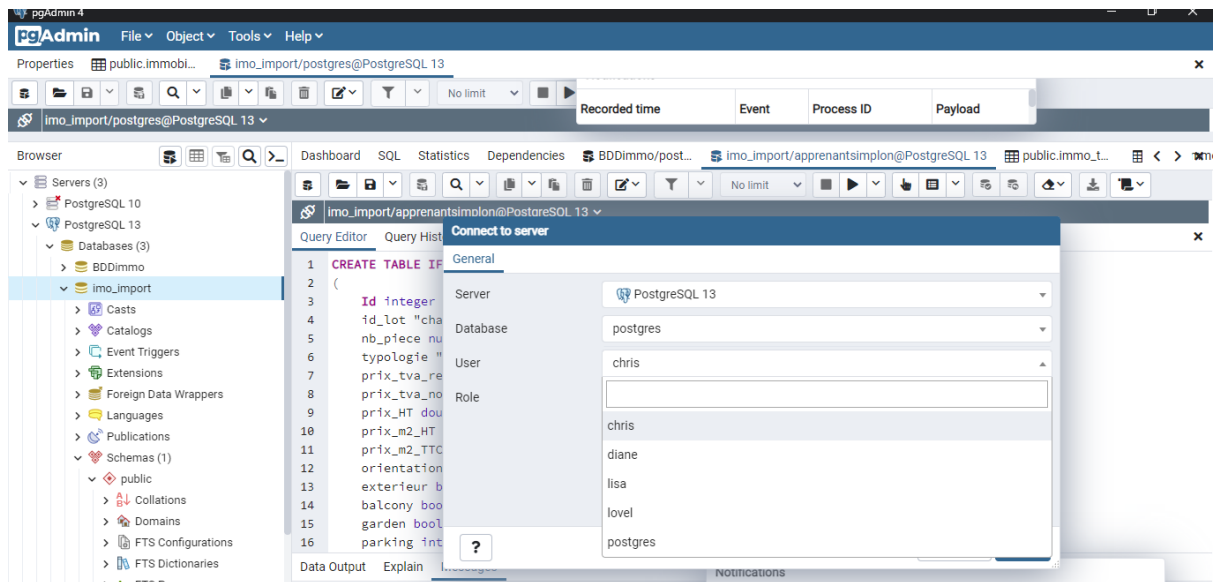


4. Mise en place d'une base de données relationnelle

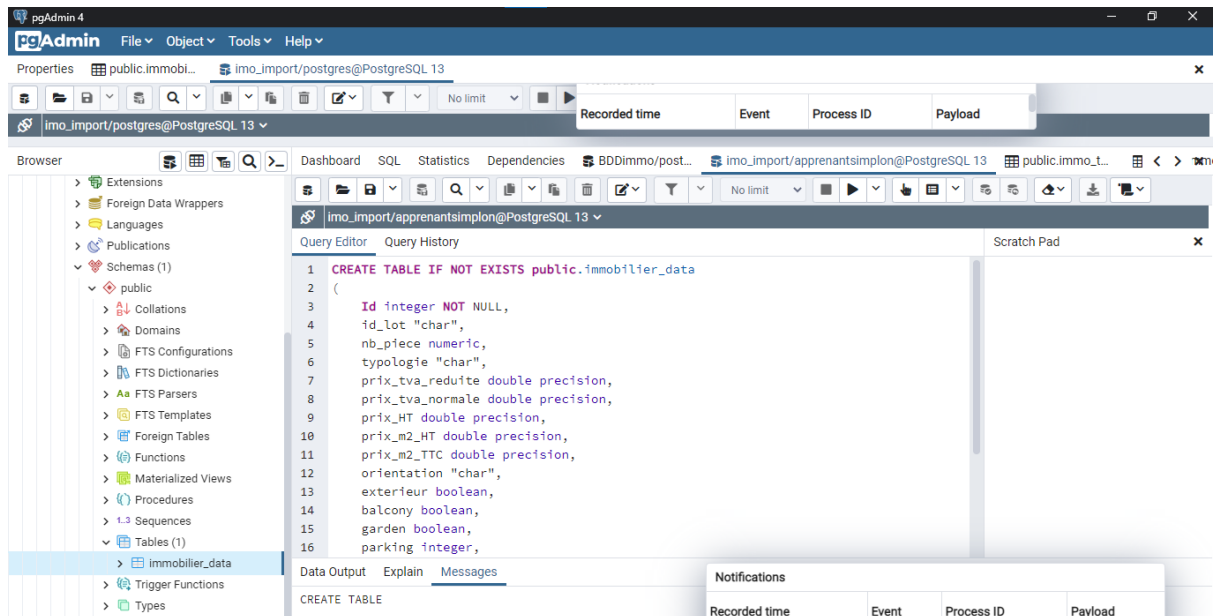
Insertion & acquisition des données de manière programmatique

Vous êtes consultant Python chez un promoteur immobilier. Le chef de projet BI a besoin de la liste des biens immobiliers de la région dans un fichier Excel pour alimenter un dashboard depuis Power BI.

1. Quels sont les avantages du SGBD POSTGRESQL ?
 Rep : PostgreSQL est plus fiable concernant l'intégralité des données
2. Mettre en place et configurer un SGBD (POSTGRESQL)
 Lors de la configuration du SGBD POSTGRESQL réaliser les actions suivantes :
 Création de users, Gestion des droits sur les tables, namespace, etc



3. Créer une table avec les champs définis dans le fichier **Promoteur imo.csv**
Créer et exécuter un script de création de la table et des champs associés
Champs à ajouter après traitement :



- a. Surface, étage, nom du programme, promoteur
Ces 3 champs sont contenues dans le champ
id_lot = surface_etage_nom programme_promoteur
4. Récupérer ces données depuis Python puis les enregistrer dans la base de donnée POSTGRESQL
5. Interroger la base de donnée sur la table précédemment créée en ne gardant que les biens immobiliers possédant un balcon.
6. Enregistrer ces données dans un fichier Excel au format.xlsx

5. Planification du script

1. Quelle solution existe-t-il pour planifier l'exécution de ce script ?
2. Choisir l'une des solutions en expliquant pourquoi,
Puis appliquer une planification du lancement du script toutes les 10 minutes.

6. Programmation orientée objet

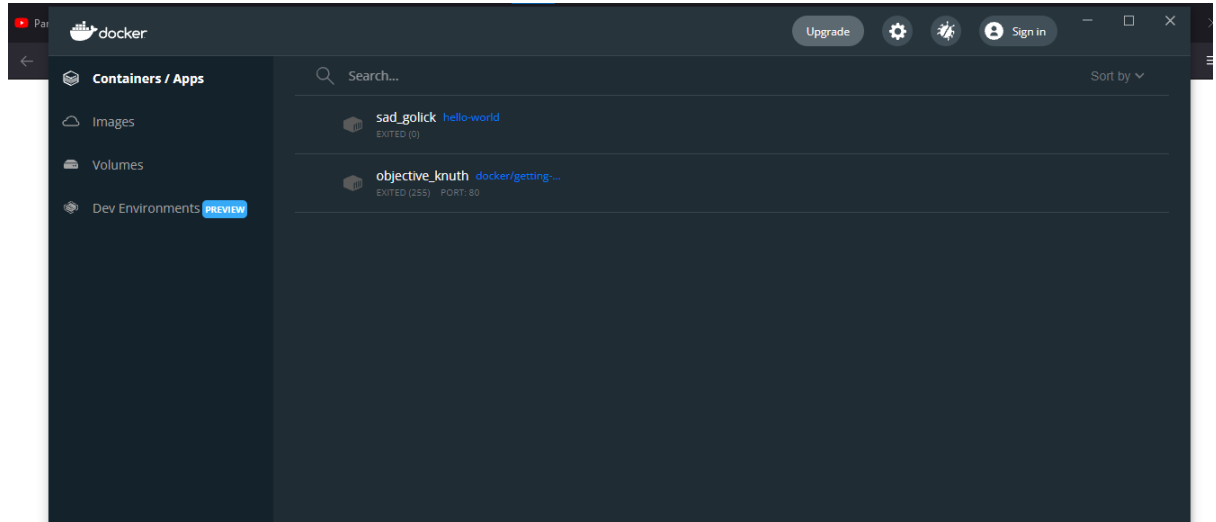
1. depuis l'application Python précédente (§4), créer une classe bien_imo contenant comme attribut les champs contenus dans le fichier **Promoteur imo.csv**
2. Pour chaque ligne du fichier Promoteur imo.csv, instancier un objet bien_imo
3. créer une liste d'objet bien_imo contenant la liste de tous les objets instanciés
4. Ré itérer les étapes §4.2 en travaillant avec les objets créés

7. Containerisation

1. Donner la définition de Docker, Container
Docker: est une plateforme des conteneurs

Containerisation : est le fait de créer un environnement sur une machine physique de tel sorte que tous le contenu du programme ces librairies,... soient exécuter sur un autre machine malgré les différent OS sans qu'il ait des modifications.

2. Installer Docker sur votre machine Windows



3. Créer un container de votre application (§4)

8. Requêtes SQL (Bonus n°1)

Exercice : Gestion des trains

Les relations suivantes permettent la gestion des conducteurs assurant les liaisons ferroviaires de la SNCF.

```

CONDUCTEUR(c_num, c_nom, c_prenom, c_ville)
    c_num          Clé primaire

TRAIN(t_num, t_type, t_localite, wagon_r, wagon_l)
    t_num          Clé primaire
    wagon_r        'oui' si a un wagon restaurant, 'non' sinon
    wagon_l        'oui' si a un wagon lit, 'non' sinon

LIAISON(l_num, ville_d, ville_a, heure_d, heure_a, c_num, t_num)
    l_num          Clé primaire
    c_num          Clé étrangère issue de CONDUCTEUR
    t_num          Clé étrangère issue de TRAIN
  
```

Ecrivez des requêtes SQL pour répondre aux questions suivantes :

- Quels sont les noms et prénoms de tous les conducteurs ?
- Quels sont les noms et prénoms du conducteur numéro 456 ?
- Quels sont les numéros et localités d'origine des trains TGV ou TER ?
- Quel est l'ensemble des villes de départ ?
- Quels sont les numéros des trains qui possèdent un wagon-lit ?
- Quel est le numéro de train assurant la liaison numéro 626 ?
- Quels sont les numéros des conducteurs qui partent de Bordeaux entre 14h et 15h ?

- h) Quels sont les numéros des trains à destination de Loudun ?
- i) D'où part le train qui assure la liaison numéro 323 ?
- j) De quelle ville est originaire le conducteur 46545 ?
- k) Quels sont les trains qui sont soit TGV soit Corail ?
- l) Quels sont les trains qui sont soit TGV avec wagon-restaurant sans wagon lit, soit Corail avec wagon-lit mais sans wagon restaurant

9. Déploiement de l'application (Bonus n°2)

1. Réaliser un déploiement manuel de l'application sur une machine Linux (Debian ou Ubuntu)
2. Déploiement avec GITLAB dans un workflow CI/CD