

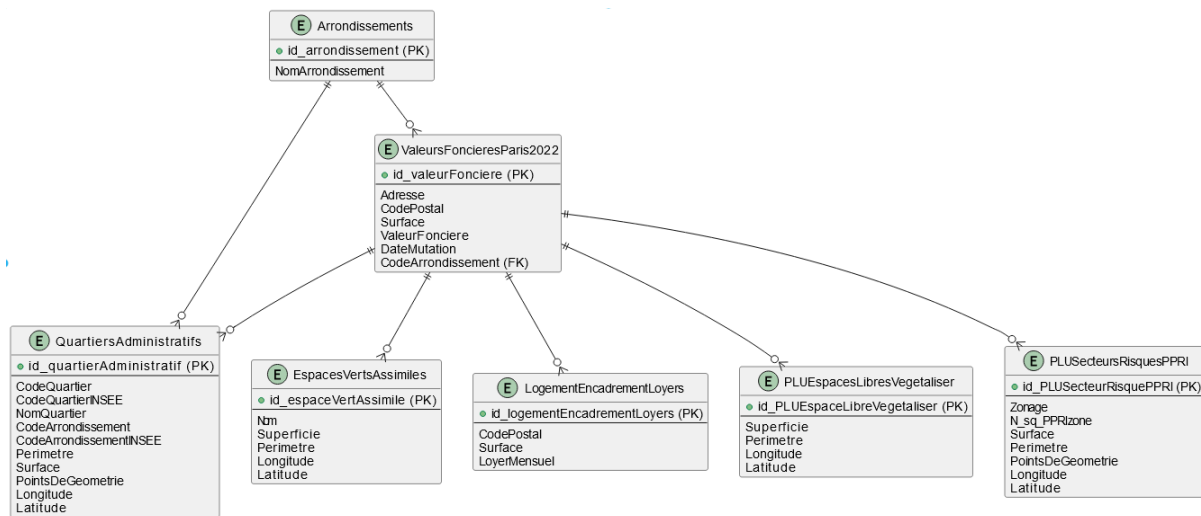
# **Rapport Etape 2**

## **Analyse de l'immobilier à Paris**

## Introduction

Cette suite de projet explore la modélisation et la gestion de données à travers l'utilisation d'un diagramme entité-association (ER) et de scripts SQL. Le diagramme ER permet de visualiser les relations entre les entités, tandis que le premier script SQL définit la structure de la base de données. Par la suite, un script d'alimentation sera utilisé pour importer des données à partir de fichiers JSON et CSV dans la base de données. Cette approche garantit une gestion efficace des données et simplifie le développement de systèmes informatiques fiables.

# 1. Diagramme entité-association



Ce schéma représente un diagramme entité-association (ER) des différentes entités et de leurs relations dans une base de données pour la gestion des données immobilières et urbaines à Paris.

Arrondissements et QuartiersAdministratifs :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De Arrondissements vers QuartiersAdministratifs (Un arrondissement peut contenir plusieurs quartiers administratifs, mais chaque quartier administratif appartient à un seul arrondissement).

Arrondissements et ValeursFoncièresParis2022 :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De Arrondissements vers ValeursFoncièresParis2022 (Un arrondissement peut avoir plusieurs valeurs foncières enregistrées pour différentes adresses, mais chaque valeur foncière est associée à un seul arrondissement).

ValeursFoncièresParis2022 et EspacesVertsAssimiles :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De ValeursFoncièresParis2022 vers EspacesVertsAssimiles (Une valeur foncière peut être associée à plusieurs espaces verts assimilés, mais chaque espace vert assimilé est lié à une seule valeur foncière).

ValeursFoncièresParis2022 et LogementEncadrementLoyers :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De ValeursFoncièresParis2022 vers LogementEncadrementLoyers (Une valeur foncière peut être associée à plusieurs logements soumis à l'encadrement des loyers, mais chaque logement appartient à une seule valeur foncière).

ValeursFoncièresParis2022 et PLUEspacesLibresVegetaliser :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De ValeursFoncièresParis2022 vers PLUEspacesLibresVegetaliser (Une valeur foncière peut être associée à plusieurs espaces libres à végétaliser, mais chaque espace libre appartient à une seule valeur foncière).

ValeursFoncièresParis2022 et PLUSecteursRisquesPPRI :

- Relation : One-To-Many (Un à Plusieurs)
- Direction : De ValeursFoncièresParis2022 vers PLUSecteursRisquesPPRI (Une valeur foncière peut être associée à plusieurs secteurs à risques délimités par le PPRI, mais chaque secteur appartient à une seule valeur foncière).

Dans l'ensemble, ces relations indiquent les liens entre les différentes entités de données, ce qui permet de modéliser efficacement les relations complexes entre les données dans le contexte de l'immobilier à Paris.

## 2. Création de la base de données

Ce script SQL `db_creation_script.sql` crée un schéma de base de données pour stocker des informations sur divers aspects de Paris. Il comporte plusieurs tables qui sont liées les unes aux autres de différentes manières. Voici une explication optimisée du script avec une meilleure mise en évidence des relations entre les tables :

### Table "ValeursFoncièresParis2022"

- Stocke des informations sur les valeurs foncières à Paris en 2022.
- Colonnes : `id_valeurFonciere` (identifiant unique), `Adresse`, `CodePostal`, `Surface`, `ValeurFonciere`, `DateMutation`.
- Clé primaire : `id_valeurFonciere`.
- Relation : La colonne `CodeArrondissement` fait référence à la clé primaire de la table "Arrondissements".

```
CREATE TABLE ValeursFoncièresParis2022 (  
    id_valeurFonciere INT PRIMARY KEY,  
    Adresse VARCHAR(255),  
    CodePostal VARCHAR(10),  
    Surface FLOAT,  
    ValeurFonciere FLOAT,  
    DateMutation DATE,  
    CodeArrondissement INT,  
    FOREIGN KEY (CodeArrondissement) REFERENCES  
Arrondissements(CodeArrondissement)  
);
```

### Table "Arrondissements"

- Stocke des informations sur les arrondissements de Paris.
- Colonnes : `id_arrondissement` (identifiant unique), `NomArrondissement`.
- Clé primaire : `id_arrondissement`.

```
CREATE TABLE Arrondissements (  
    id_arrondissement INT PRIMARY KEY,  
    NomArrondissement VARCHAR(255)  
);
```

### Table "EspacesVertsAssimiles"

- Stocke des informations sur les espaces verts assimilés.
- Colonnes : id\_espaceVertAssimile (identifiant unique), Nom, Superficie, Perimetre, Longitude, Latitude.
- Clé primaire : id\_espaceVertAssimile.

```
CREATE TABLE EspacesVertsAssimiles (  
    id_espaceVertAssimile INT PRIMARY KEY,  
    Nom VARCHAR(255),  
    Superficie FLOAT,  
    Perimetre FLOAT,  
    Longitude FLOAT,  
    Latitude FLOAT  
);
```

### Table "LogementEncadrementLoyers"

- Stocke des informations sur les logements soumis à l'encadrement des loyers.
- Colonnes : id\_logementEncadrementLoyers (identifiant unique), CodePostal, Surface, LoyerMensuel.
- Clé primaire : id\_logementEncadrementLoyers.

```
CREATE TABLE LogementEncadrementLoyers (  
    id_logementEncadrementLoyers INT PRIMARY KEY,  
    CodePostal VARCHAR(10),  
    Surface FLOAT,  
    LoyerMensuel FLOAT  
);
```

### Table "PLUEspacesLibresVegetaliser"

- Stocke des informations sur les espaces libres à végétaliser.
- Colonnes : id\_PLUEspaceLibreVegetaliser (identifiant unique), Superficie, Perimetre, Longitude, Latitude.
- Clé primaire : id\_PLUEspaceLibreVegetaliser.

```
CREATE TABLE PLUEspacesLibresVegetaliser (  
    id_PLUEspaceLibreVegetaliser INT PRIMARY KEY,  
    Superficie FLOAT,  
    Perimetre FLOAT,  
    Longitude FLOAT,  
    Latitude FLOAT  
);
```

Table "PLUSecteursRisquesPPRI"

- Stocke des informations sur les secteurs à risques délimités par le PPRI.
- Colonnes : id\_PLUSecteurRisquePPRI (identifiant unique), Zonage, N\_sq\_PPRIzone, Surface, Perimetre, PointsDeGeometrie, Longitude, Latitude.
- Clé primaire : id\_PLUSecteurRisquePPRI.

```
CREATE TABLE PLUSecteursRisquesPPRI (  
    id_PLUSecteurRisquePPRI INT PRIMARY KEY,  
    Zonage VARCHAR(10),  
    N_sq_PPRIzone INT,  
    Surface FLOAT,  
    Perimetre FLOAT,  
    PointsDeGeometrie GEOMETRY,  
    Longitude FLOAT,  
    Latitude FLOAT  
);
```

Table "QuartiersAdministratifs"

- Stocke des informations sur les quartiers administratifs de Paris.
- Colonnes : id\_quartierAdministratif (identifiant unique), CodeQuartier, CodeQuartierINSEE, NomQuartier, CodeArrondissement, CodeArrondissementINSEE, Perimetre, Surface, PointsDeGeometrie, Longitude, Latitude.
- Clé primaire : id\_quartierAdministratif.
- Relation : La colonne CodeArrondissement fait référence à la clé primaire de la table "Arrondissements".

```
CREATE TABLE QuartiersAdministratifs (  
    id_quartierAdministratif INT PRIMARY KEY,  
    CodeQuartier VARCHAR(10),  
    CodeQuartierINSEE VARCHAR(10),  
    NomQuartier VARCHAR(255),  
    CodeArrondissement INT,  
    CodeArrondissementINSEE VARCHAR(10),  
    Perimetre FLOAT,  
    Surface FLOAT,  
    PointsDeGeometrie GEOMETRY,  
    Longitude FLOAT,  
    Latitude FLOAT,  
    FOREIGN KEY (CodeArrondissement) REFERENCES  
Arrondissements(CodeArrondissement)  
);
```