

## **Méthodologie atelier SIGAT**

---

### **Sommaire :**

#### **Etape 1 :**

1.        **Choix        des données**
2.        **Automatisation**

#### **Etape 2 :**

1.        **Choix        des échelles d'agrégations**
2.        **Choix        des processus de géotraitements**

### **Annexe :**

**Compte rendu des réunions**

**Temps de travail des étudiants**

**Table de GANTT**

**Dictionnaire de variables**

# **ETAPE 1**

## **Intro + présentation du master + présentation de GENERALI + présentation de l'objectif de la commande**

Comme chaque année le MASTER SIGAT, permet aux étudiants de s'immerger dans le monde professionnel par le biais d'ateliers enrichissants, permettant la mise en pratique des connaissances obtenues. La collaboration avec GENERALI (entreprise mondiale d'assurance ) va permettre la mise en pratique de scripts (SQL, PYTHON, R) mais aussi de l'univers de la gestion de projet (appréhension du domaine, connaissance des éléments techniques, enjeu actuel pour les sociétés d'assurances).

L'objectif de cet atelier est ainsi d'offrir à GENERALI des scripts permettant un enrichissement de leur base de données centrée sur les bâtiments. Nous nous concentrerons sur les données « d'urbanisme », GENERALI ayant déjà une équipe de spécialistes centrés sur les questions de l'environnement. Il s'agit dans le cadre de cet atelier de proposer une chaîne de traitements automatisée de récupération, de constitution et d'enrichissement d'une base de données caractérisant le bâti.

## 1. **Choix des données**

Les premières réunions entre GENERALI et les étudiants en Master 2 SIGAT ont permis la mise en place d'une liste effective de données à conserver/ chercher et d'autres à supprimer.

Les données répondent à des thématiques précises émises par le commanditaire. Celles-ci se doivent d'être disponibles en open data, présentes de façon homogène sur tout le territoire français et tirées de ressources sérieuses amenant des précisions à l'échelle du bâtiment.

Ainsi, nous avons pu nous concentrer sur les données suivantes :

- BD adresse / BAN
- DVF
- BD TOPO
- Le cadastre (échelle du bâtiment et de la parcelle)

Chacune de ces données permettra un enrichissement de la base de données des bâtiments. Cela prendra forme au travers de deux échelles distinctes (parcelle + bâti).

Au sein de cette étape, la deuxième démarche à suivre est de sélectionner les champs les plus intéressants et ayant un intérêt pour GENERALI (ci-dessous) :

### **- DVF**

ID PARCELLE, valeur foncière, code\_type\_local, Nombres\_pièces\_principales, Lat, Long

### **- BD ADRESSE**

Numero, nom\_voie, nom\_commune, lat, lon

## **- BD TOPO**

USAGE 1 , USAGE 2, Hauteur, Z\_Min\_SOL, Z\_max\_Sol, Z\_MIN\_Toit, Z\_max\_Toit, NB\_Lgts, Leger, MAT\_Toit, Mat\_MUR, NB\_ETAGES

## **- CADASTRE**

ID, NUMERO, Contenance

### **Dictionnaire des variables**

Un dictionnaire de variable, reprenant les différentes caractéristiques des variables est transmis en annexe (taux de complétion, information sur l'objectif de la donnée). Toutes ces informations ont été établies par le biais de requêtes SQL sur PostgreSQL (interface PGADMIN). En voici un exemple :

#### **Avoir le nombre d'entités présentes dans la base qui ne sont pas nulles :**

```
Select count(*) as nb  
  
From Base_de_donnée  
  
Where nom_de_variable IS NOT NULL
```

#### **Calcul nombre de valeurs uniques des champs texte**

```
select type_local, count(*)  
  
from dvf  
  
group by type_local
```

#### **Calcul minimum et maximum des champs numériques**

```
select min(hauteur), max(hauteur)  
  
from bd_topo
```

**Pour la mise en place d'un histogramme (mettre la méthode avec DATABASE)**

**Réalisation d'histogrammes en Python pour identifier les valeurs extrêmes :**

# Automatisation du téléchargement des données et de l'import des tables dans PostgreSQL

La première étape d'élaboration de scripts consiste en l'automatisation de la récupération des bases de données sur les sites et à leur stockage dans un système de base de données. Le SGBD que nous avons choisi est PostgreSQL.

Nous avons automatisé le téléchargement des données qui sont stockées dans un premier temps dans un dossier intermédiaire.

Dans un second temps, les tables géographiques et non géographiques sont chargées en GeoDataFrame et DataFrame (cf. librairies Geopandas et Pandas) en ne conservant que les champs souhaités, puis importées dans PostgreSQL.

Ce processus représente un gain de temps mais aussi de coût. En effet, une fois le script établi, la récupération des données se fait avec davantage d'aisance.

Voici les différents **scripts de téléchargement de la donnée et son import dans la base de données stockée dans PostgreSQL** réalisés pour chaque donnée, et disponibles sur GitHub :

## DVF

cf. `dvf_telechargement_import.py`

## BD TOPO

cf. `bdtopo_telechargement.py`

cf. `bdtopo_import.py`

## BD ADRESSE

cf. **PAS ENCORE FAIT**

## **BAN**

cf. ban\_telechargement\_import.py

## **CADASTRE BATI**

cf. cadastre\_batiments\_telechargement\_import.py

## **CADASTRE PARCELLE**

cf. cadastre\_batiments\_telechargement\_import.py