[1. Connection 1](#_Toc158058170)

[2. Utilisateur 2](#_Toc705197886)

[3. UtilisateurParticulier 2](#_Toc922040234)

[4. UtilisateurEntreprise 2](#_Toc868601670)

[5. UtilisateurEntrepreneur 2](#_Toc1680180632)

[6. UtilisateurAssociation 2](#_Toc868226269)

[7. Avis 2](#_Toc1486735073)

[8. Administrateur 3](#_Toc600076471)

[9. Vendeur 3](#_Toc447803957)

[10. Client 3](#_Toc287375178)

[11. Planning 4](#_Toc1853756387)

[12. Reservation 4](#_Toc453380622)

[13. Categorie 4](#_Toc1480625914)

[14. Localisation 5](#_Toc1844875753)

[15. Bien 6](#_Toc1614098290)

[16. Services. 7](#_Toc702682457)

[17. Publicite 7](#_Toc2040095375)

[18. Favoris 7](#_Toc901198522)

[19. Demande 8](#_Toc916551743)

[20. Codes 9](#_Toc1276754161)

# Connection

Pour la génération des données de connection nous avons utilisé un code python permettant de générer les clefs primaires

Code :

def generate\_values(self, name, size):  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size+1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ");\n")  
 file.close()

# Utilisateur

Pour la génération de Utilisateur nous avons utilisé un générateur en ligne <https://www.mockaroo.com/>

Et nous avons utilisé un code pour scrapper depuis le site les descriptions des personnes pour la aussi avoir des données réelles

def scrap\_biographie(self, link=""):  
 tab\_desc = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 for i in soup.find\_all("span", {"class": "break-word"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_desc))  
 for i in range(len(tab\_desc)):  
 self.i += 1  
 self.insert\_data\_into\_biographie("biographie", tab\_desc)  
 return False  
  
def insert\_data\_into\_biographie(self, name, desc):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into connection (" + name + ") values (" + str(desc) + ");\n")  
 file.close()

# UtilisateurParticulier

Pour la génération de UtilisateurParticulier nous avons utilisé un générateur en ligne <https://www.mockaroo.com/>

# UtilisateurEntreprise

Pour la génération de UtilisateurEntreprise nous avons utilisé un générateur en ligne <https://www.mockaroo.com/>

# UtilisateurEntrepreneur

Pour la génération de UtilisateurEntrepreneur nous avons utilisé un générateur en ligne <https://www.mockaroo.com/>

# UtilisateurAssociation

Pour la génération de UtilisateurAssociation nous avons utilisé un générateur en ligne <https://www.mockaroo.com/>

# Avis

Pour la génération des données de avis nous avons utilisé un code python permettant de scrapper le site permettant de prendre de véritables données tels que l’avis et la note

Avec le code python :

def scrap\_avis(self, link="https://www.allovoisins.com/p/homeservices-5/avis"):  
 tab\_desc = []  
 tab\_note = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 # print(soup)  
 for i in soup.find\_all(  
 "p", {"class": "normal-text normal-text-bold text-l flex flex-vertical-center"}):  
 tab\_note.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all("div", {"class": "review"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_note))  
 print(len(tab\_desc))  
 for i in range(len(tab\_note)):  
 self.i += 1  
 self.insert\_into\_data\_avis(self.i, tab\_desc[i], tab\_note[i])  
 return False  
  
def insert\_into\_data\_avis(self, i, item1, item2):  
 print("insert into avis values (" + str(i) + ", '" +  
 str(item1) + "'," +  
 str(item2[0]) + "," + str(i) + ");\n")

# Administrateur

Pour la génération des données d'administrateur nous avons utilisé un code python permettant de générer les données

Code :

def generate\_values(self, name, size):  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size+1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ");\n")  
 file.close()

1. Vendeur

Pour la génération des données de vendeur nous avons utilisé un code python permettant de générer les données

Code :

def generate\_values(self, name, size):  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size+1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ");\n")  
 file.close()

1. Client

Pour la génération des données de Client nous avons utilisé un code python permettant de générer les données

Code :

def generate\_values(self, name, size):  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size+1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ");\n")  
 file.close()

1. Planning

Pour la génération des données de planning nous avons utilisé un générateur en ligne

<https://www.mockaroo.com/>

1. Reservation

Pour la génération des données de Reservation nous avons utilisé un code python permettant de générer les données

def generate\_reservation(self, name, size):  
 tab = [2, 3, 4, 5, 7, 8, 12, 15, 18, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 36, 39, 41, 44, 46, 49]  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size + 1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ", " + str(random.randint(0, 1))  
 + ", " + str(i) + ", " + str(tab[random.randint(0, len(tab) - 1)]) + ");\n")  
 file.close()

# Categorie

Pour la génération des données de Categorie nous avons utilisé un code python permettant de scrapper le site permettant de prendre de véritables données tels que les noms des categories

def scrap\_categorie\_name(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_data(item.get\_text(strip=True))  
 self.i += 1  
  
def insert\_into\_data\_categorie(self, item1=""):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ", '" +  
 str(item1) + "');\n")  
 file.close()

1. Localisation

Pour la génération des données de localisation nous avons utilisé un générateur pour les adresses et un code python pour les codes postaux et les noms des villes

<https://www.mockaroo.com/>

# Author: Lacroix Baptiste -> https://github.com/BaptisteLacroix  
  
import csv  
import random  
import re  
from typing import List  
  
  
def listenoms(table):  
 *"""*  
***:param*** *table:*  
***:return****:*  
 *"""*  
return [value[0] for value in table[1:]]  
  
  
def choix\_mot(mots: List[str]) -> str:  
 *"""*  
***:param*** *mots:*  
***:return****:*  
 *"""*  
return random.choice(mots)  
  
  
def replace(noms, search\_text):  
 with open(r'./localisation.sql', 'r') as file:  
 data = file.read()  
 data = data.replace(search\_text, "'" + choix\_mot(noms) + "'")  
 with open(r'./localisation.sql', 'w') as file:  
 file.write(data)  
  
  
def main():  
 f = open("laposte\_hexasmal.csv", "r")  
 table = list(csv.reader(f, delimiter=';'))  
 noms = listenoms(table)  
 for i in range(0, 10):  
 search\_text = '1000' + str(i)  
 m = re.search(" ".join(f), search\_text)  
 if m:  
 replace(noms, search\_text)  
 for i in range(9, 28):  
 search\_text = '100' + str(i)  
 m = re.search(" ".join(f), search\_text)  
 if m:  
 replace(noms, search\_text)  
  
 # Printing Text replaced  
 print("Text replaced")  
 f.close()  
 print("Text successfully replaced")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

1. Bien

Pour la génération des données de Bien nous avons utilisé un code python permettant de scrapper le site permettant de prendre de véritables données tels que les noms des Biens étants affiliés a leur catégories, Les prix, desc et clef étrangères ont été rentré a la main.

def scrap\_sous\_categorie(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_bien(item.get\_text(strip=True))  
 self.i += 1  
  
def insert\_into\_bien(self, item1):  
 tab = ["LOCATION", "EMPRUNT"]  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ", '" +  
 random.choice(tab) + "','" +  
 str(item1) + "');\n")  
 file.close()

1. Services.

Pour la génération des données de Service nous avons utilisé un code python permettant de scrapper le site permettant de prendre de véritables données tels que les noms des Services étants affiliés a leur catégories, Les prix, desc et clef étrangères ont été rentré a la main.

def scrap\_sous\_categorie(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_services (item.get\_text(strip=True))  
 self.i += 1  
  
def insert\_into\_services(self, item1):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ",'" +  
 str(item1) + "');\n")  
 file.close()

1. Publicite

La génération des données de Publicite ont été générés a la main

1. Favoris

La table favoris a été généré avec le code python ci dessous

def insert\_into\_favoris(self):  
 vendeur = [3, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23, 26, 29, 32, 35, 37, 40, 42, 43, 45, 46, 47, 49, 50]  
 client = [2, 3, 4, 5, 7, 8, 12, 15, 18, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 36, 39, 41, 44, 46, 49]  
 for i in range(35):  
 with open(r'./test.txt', 'a') as file:  
 ve = vendeur[random.randint(0, len(vendeur) - 1)]  
 cl = client[random.randint(0, len(client) - 1)]  
 file.write("insert into favoris values (" +  
 str(i + 1) + "," +  
 str(cl) + "," +  
 str(ve) + ");\n")  
 file.close()

1. Demande

Pour la génération des données de demande nous avons utilisé un code python permettant de scrapper le site permettant de prendre de véritables données

def scrap\_announce(self, link="https://www.allovoisins.com/r/-3/0/0/10/location-vente"):  
 tab\_name = []  
 tab\_prix = []  
 tab\_categories = []  
 tab\_desc = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 # print(soup)  
 for i in soup.find\_all(  
 "p", {"class": "nearYou\_\_searchItemName mainlight bold h4"}):  
 tab\_name.append(i.get\_text(strip=True))  
 for i in soup.find\_all(  
 "span", {"class": "nearYou\_\_searchItemLabel badge badge--green"}):  
 tab\_prix.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all(  
 "h3", {"class": "nearYou\_\_searchItemTitle mg-top-s mg-bottom-s"}):  
 tab\_categories.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all(  
 "p", {"class": "nearYou\_\_searchItemDescription mainlight"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_name))  
 print(len(tab\_prix))  
 print(len(tab\_categories))  
 print(len(tab\_desc))  
 for i in range(len(tab\_prix)):  
 self.i += 1  
 self.insert\_into\_data(self.i, tab\_name[i], tab\_prix[i], tab\_categories[i], tab\_desc[i])  
 return False

def insert\_into\_data(self, i, item1, item2="", item3="", item4=""):  
 with open(r'./test.txt', 'a') as file:  
 print(item1)  
 print(item2)  
 print(item3)  
 print(item4)  
 file.write("insert into demande values (" + str(i) + ", '" +  
 str(item1) + "'," + "'" +  
 str(item2) + "'," + "'" +  
 str(item3) + "'," + "'" +  
 str(item4) + "');\n")  
 file.close()

# Codes

# Author: Lacroix Baptiste -> https://github.com/BaptisteLacroix  
  
import requests  
from bs4 import BeautifulSoup  
import random  
  
  
class Scrapping:  
  
 def \_\_init\_\_(self):  
 self.i = 20  
  
 def scrap\_categorie\_name(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_data\_categorie(item.get\_text(strip=True))  
 self.i += 1  
  
 def insert\_into\_data\_categorie(self, item1=""):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ", '" +  
 str(item1) + "');\n")  
 file.close()  
  
 def scrap\_sous\_categorie\_bien(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_bien(item.get\_text(strip=True))  
 self.i += 1  
  
 def insert\_into\_bien(self, item1):  
 tab = ["LOCATION", "EMPRUNT"]  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ", '" +  
 random.choice(tab) + "','" +  
 str(item1) + "');\n")  
 file.close()  
  
 def scrap\_sous\_categorie\_services(self, link="https://www.allovoisins.com/near\_you", balise1="li", balise2="class",  
 baliseName="near\_you\_col\_right\_links\_tracked\_ga"):  
  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 data1 = soup.find('div', {'class': 'nearYou\_\_leftNav card shadow mg-right'})  
 if data1 is None:  
 return False  
 data2 = data1.find('ul')  
 print(data2)  
 for item in data2.find\_all(balise1, {balise2: baliseName}):  
 self.insert\_into\_services(item.get\_text(strip=True))  
 self.i += 1  
  
 def insert\_into\_services(self, item1):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into demande values (" + str(self.i) + ",'" +  
 str(item1) + "');\n")  
 file.close()  
  
 def scrap\_announce(self, link="https://www.allovoisins.com/r/-3/0/0/10/location-vente"):  
 tab\_name = []  
 tab\_prix = []  
 tab\_categories = []  
 tab\_desc = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 # print(soup)  
 for i in soup.find\_all(  
 "p", {"class": "nearYou\_\_searchItemName mainlight bold h4"}):  
 tab\_name.append(i.get\_text(strip=True))  
 for i in soup.find\_all(  
 "span", {"class": "nearYou\_\_searchItemLabel badge badge--green"}):  
 tab\_prix.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all(  
 "h3", {"class": "nearYou\_\_searchItemTitle mg-top-s mg-bottom-s"}):  
 tab\_categories.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all(  
 "p", {"class": "nearYou\_\_searchItemDescription mainlight"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_name))  
 print(len(tab\_prix))  
 print(len(tab\_categories))  
 print(len(tab\_desc))  
 for i in range(len(tab\_prix)):  
 self.i += 1  
 self.insert\_into\_data(self.i, tab\_name[i], tab\_prix[i], tab\_categories[i], tab\_desc[i])  
 return False  
  
 def insert\_into\_data(self, i, item1, item2="", item3="", item4=""):  
 with open(r'./test.txt', 'a') as file:  
 print(item1)  
 print(item2)  
 print(item3)  
 print(item4)  
 file.write("insert into demande values (" + str(i) + ", '" +  
 str(item1) + "'," + "'" +  
 str(item2) + "'," + "'" +  
 str(item3) + "'," + "'" +  
 str(item4) + "');\n")  
 file.close()  
  
 @staticmethod  
 def insert\_pr\_fr\_into\_bien():  
 vendeur = [3, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23, 26, 29, 32, 35, 37, 40, 42, 43, 45, 46, 47, 49, 50]  
 planning = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27]  
 categorie = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,  
 28]  
 localisation = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26,  
 27]  
 tabVe = []  
 tabPl = []  
 tabCa = []  
 tabLo = []  
 for i in range(27):  
 with open(r'./test.txt', 'a') as file:  
 ve = vendeur[random.randint(0, len(vendeur) - 1)]  
 pl = planning[random.randint(0, len(planning) - 1)]  
 ca = categorie[random.randint(0, len(categorie) - 1)]  
 lo = categorie[random.randint(0, len(localisation) - 1)]  
  
 while ve in tabVe and len(tabVe) != len(vendeur):  
 ve = vendeur[random.randint(0, len(vendeur) - 1)]  
  
 while pl in tabPl and len(tabPl) != len(planning):  
 pl = planning[random.randint(0, len(planning) - 1)]  
  
 while ca in tabCa and len(tabCa) != len(categorie):  
 ca = categorie[random.randint(0, len(categorie) - 1)]  
  
 while lo in tabLo and len(tabLo) != len(localisation):  
 lo = categorie[random.randint(0, len(localisation) - 1)]  
  
 tabVe.append(ve)  
 tabPl.append(pl)  
 tabVe.append(ca)  
 tabLo.append(lo)  
  
 file.write("insert into bien values (" +  
 str(lo) + "," +  
 str(pl) + "," +  
 str(ve) + "," +  
 str(ca) + ");\n")  
 file.close()  
  
 def insert\_into\_favoris(self):  
 vendeur = [3, 9, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 23, 26, 29, 32, 35, 37, 40, 42, 43, 45, 46, 47, 49, 50]  
 client = [2, 3, 4, 5, 7, 8, 12, 15, 18, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 36, 39, 41, 44, 46, 49]  
 for i in range(35):  
 with open(r'./test.txt', 'a') as file:  
 ve = vendeur[random.randint(0, len(vendeur) - 1)]  
 cl = client[random.randint(0, len(client) - 1)]  
 file.write("insert into favoris values (" +  
 str(i + 1) + "," +  
 str(cl) + "," +  
 str(ve) + ");\n")  
 file.close()  
  
 def scrap\_avis(self, link="https://www.allovoisins.com/p/homeservices-5/avis"):  
 tab\_desc = []  
 tab\_note = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 # print(soup)  
 for i in soup.find\_all(  
 "p", {"class": "normal-text normal-text-bold text-l flex flex-vertical-center"}):  
 tab\_note.append(i.get\_text(strip=True))  
  
 for i in soup.find\_all("div", {"class": "review"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_note))  
 print(len(tab\_desc))  
 for i in range(len(tab\_note)):  
 self.i += 1  
 self.insert\_into\_data\_avis(self.i, tab\_desc[i], tab\_note[i])  
 return False  
  
 def insert\_into\_data\_avis(self, i, item1, item2):  
 print("insert into avis values (" + str(i) + ", '" +  
 str(item1) + "'," +  
 str(item2[0]) + "," + str(i) + ");\n")  
  
 def generate\_values(self, name, size):  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size + 1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ");\n")  
 file.close()  
  
 def generate\_reservation(self, name, size):  
 tab = [2, 3, 4, 5, 7, 8, 12, 15, 18, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 34, 36, 39, 41, 44, 46, 49]  
 with open(r'./test.txt', 'a') as file:  
 for i in range(1, size + 1):  
 file.write("insert into connection (" + name + ") values (" + str(i) + ", " + str(random.randint(0, 1))  
 + ", " + str(i) + ", " + str(tab[random.randint(0, len(tab) - 1)]) + ");\n")  
 file.close()  
  
 def scrap\_biographie(self, link=""):  
 tab\_desc = []  
 print("Scrapping")  
 r = requests.get(link)  
 soup = BeautifulSoup(r.text, 'html.parser')  
 for i in soup.find\_all("span", {"class": "break-word"}):  
 tab\_desc.append(i.get\_text(strip=True))  
  
 print(len(tab\_desc))  
 for i in range(len(tab\_desc)):  
 self.i += 1  
 self.insert\_data\_into\_biographie("biographie", tab\_desc)  
 return False  
  
 def insert\_data\_into\_biographie(self, name, desc):  
 with open(r'./test.txt', 'a') as file:  
 file.write("insert into connection (" + name + ") values (" + str(desc) + ");\n")  
 file.close()  
  
  
def main():  
 scrap = Scrapping()  
 cont = True  
 # scrap.insert\_pr\_fr\_into\_bien()  
 while True:  
 link = str(input("lien : "))  
 # cont = scrap.scrap\_announce()  
 scrap.scrap\_biographie(link)  
 print("Sortie")  
 # scrap.insert\_into\_favoris()  
 # scrap.scrap\_avis()  
 # scrap.generate\_connection()  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

//////////////////////////////////////////////////////////////////////////

# Author: Lacroix Baptiste -> https://github.com/BaptisteLacroix  
  
import csv  
import random  
import re  
from typing import List  
  
  
def listenoms(table):  
 *"""*  
***:param*** *table:*  
***:return****:*  
 *"""*  
return [value[0] for value in table[1:]]  
  
  
def choix\_mot(mots: List[str]) -> str:  
 *"""*  
***:param*** *mots:*  
***:return****:*  
 *"""*  
return random.choice(mots)  
  
  
def replace(noms, search\_text):  
 with open(r'./localisation.sql', 'r') as file:  
 data = file.read()  
 data = data.replace(search\_text, "'" + choix\_mot(noms) + "'")  
 with open(r'./localisation.sql', 'w') as file:  
 file.write(data)  
  
  
def main():  
 f = open("laposte\_hexasmal.csv", "r")  
 table = list(csv.reader(f, delimiter=';'))  
 noms = listenoms(table)  
 for i in range(0, 10):  
 search\_text = '1000' + str(i)  
 m = re.search(" ".join(f), search\_text)  
 if m:  
 replace(noms, search\_text)  
 for i in range(9, 28):  
 search\_text = '100' + str(i)  
 m = re.search(" ".join(f), search\_text)  
 if m:  
 replace(noms, search\_text)  
  
 # Printing Text replaced  
 print("Text replaced")  
 f.close()  
 print("Text successfully replaced")  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()