1. 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()
```

2. Utilisateur

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

3. UtilisateurParticulier

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

4. UtilisateurEntreprise

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

5. UtilisateurEntrepreneur

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

6. UtilisateurAssociation

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

7. 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")
```

8. 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()
```

9. 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()
```

10. 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()
```

11. Planning

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

https://www.mockaroo.com/

12. 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()
```

13. 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()
```

14. 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()
```

15. 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()
```

16. 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()
```

17. Publicite

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

18. 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()
```

19. 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()
```

20. 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 main():
    scrap = Scrapping()
    cont = True
    # scrap.insert_pr_fr_into_bien()
    # while cont:
    #    # link = str(input("lien : "))
    #    cont = scrap.scrap_announce()
    # print("Sortie")
    # scrap.insert_into_favoris()
    # scrap.scrap_avis()
    scrap.generate_connection()


if __name__ == '__main__':
    main()

//////////////////////////////////////////////////////////////////////////
```

```python
# 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()
```