# 

BUT Informatique Semestre 2 2023-2024

IUT de Bayonne et du Pays Basque, Département Informatique

S2.04 « *Exploitation d'une base de données* »

**Partie 2**

**Présentation de l’application Python**

**TP4 - Groupe n°3**

# Thématique

Étude des logements AirBnB et des hôtes par rapport aux communes.

LATXAGUE Thibault, TD 2 - TP 4

MASSON Rafael, TD 2 - TP 4

Enseignant référent : M. NODENOT Thierry

Livrable remis avant le 06/05/2024 à 14h00

# Table des matières

# 

[**Thématique 1**](#_2lntagu2gp9j)

[**Table des matières 2**](#_2nnoxoamrmut)

[**Présentation de l’application 3**](#_hjkgor3rsqqj)

[**Description de l'enchaînement des menus 3**](#_pde8cthbdq2m)

[**Requête 1 4**](#_j2dxxsd01fko)

[**Entrée utilisateur 4**](#_ssk5d2xy8hj1)

[**Traces d’exécution 4**](#_ui4ptdqcnc4s)

[**Résultats affichés 4**](#_tixpexmxol0o)

[**Requête 2 5**](#_kevj716oh7nt)

[**Entrée utilisateur 5**](#_tnwn53n72jgp)

[**Traces d’exécution 5**](#_pr8pbtbewibz)

[**Résultats affichés 6**](#_g8kmea71r3s5)

[**Requête 3 7**](#_aj6txr7sfw33)

[**Entrée utilisateur 7**](#_ej5icjb52gc3)

[**Traces d’exécution 7**](#_payz0gr8uzzp)

[**Résultats affichés 7**](#_o0stqc7lsbos)

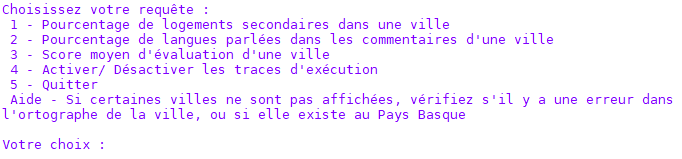
[**Bilan 8**](#_f7fhsopmdle2)

[**Code source 8**](#_3s5m6dqpd5z3)

# Présentation de l’application

## Description de l'enchaînement des menus

Au démarrage de l’application, nous avons le choix entre 3 requêtes, activer les traces d’exécution ou quitter l’application.



Si on saisit 1, l’application va exécuter la requête “[Pourcentage de logements secondaires dans une ville](#_j2dxxsd01fko)”.

Si on saisit 2, l’application va exécuter la requête “[Pourcentage de langues parlées dans les commentaires d’une ville](#_f40dcvhkfzos)”.

Si on saisit 3, l’application va exécuter la requête “[Score moyen d’évaluation d’une ville](#_aj6txr7sfw33)”.

Si on saisit 4, les traces d’exécution seront affichées lors de chaque requête exécutée postérieurement. Ces traces sont désactivées par défaut.

Si on saisit 5, l’application s’arrête.  
  
Si on saisit autre chose, l’application signale une erreur et demande une nouvelle saisie.

Après l’exécution d’une requête ou l’activation / désactivation des traces d’exécution, l’application demande un nouveau choix à l’utilisateur.

Pour toutes les requêtes suivantes, les traces d’exécution ont été activées.



## Requête 1

Autre point important portant sur cette requête. Cette requête créée au total 5 tableaux. Nous en utilisons seulement 2 pour cette requête. Il est donc possible de retravailler la requête afin de l’adapter, chercher de nouveaux résultats avec ces 3 tableaux inutilisés.

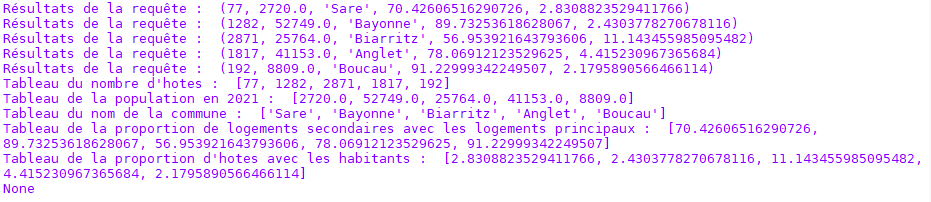
### Entrée utilisateur

### 

### Traces d’exécution

Les traces d’exécution des requêtes sont visibles en ayant entré 4 précédemment.

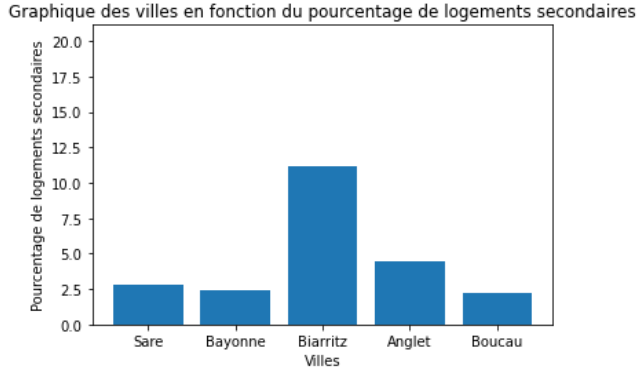
Affichage de l’exécution de la requête n°1 en prenant pour ville Sare, Bayonne, Biarritz, Anglet et Boucau.



### Résultats affichés

Voici le graphique affiché pour cette requête.

Villes choisies : Sare, Bayonne, Biarritz, Anglet et Boucau.



## Requête 2

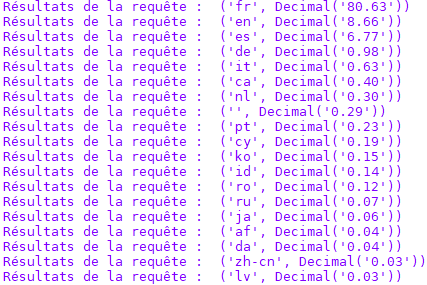
### Entrée utilisateur

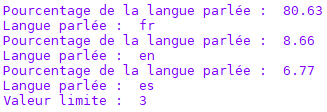
### 

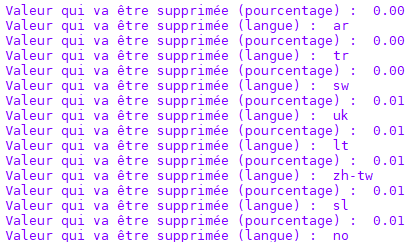
### Traces d’exécution

Les traces d’exécution des requêtes sont visibles en ayant entré 4 précédemment.

Un extrait de l’exécution de la requête n°2 en prenant pour ville Bayonne.





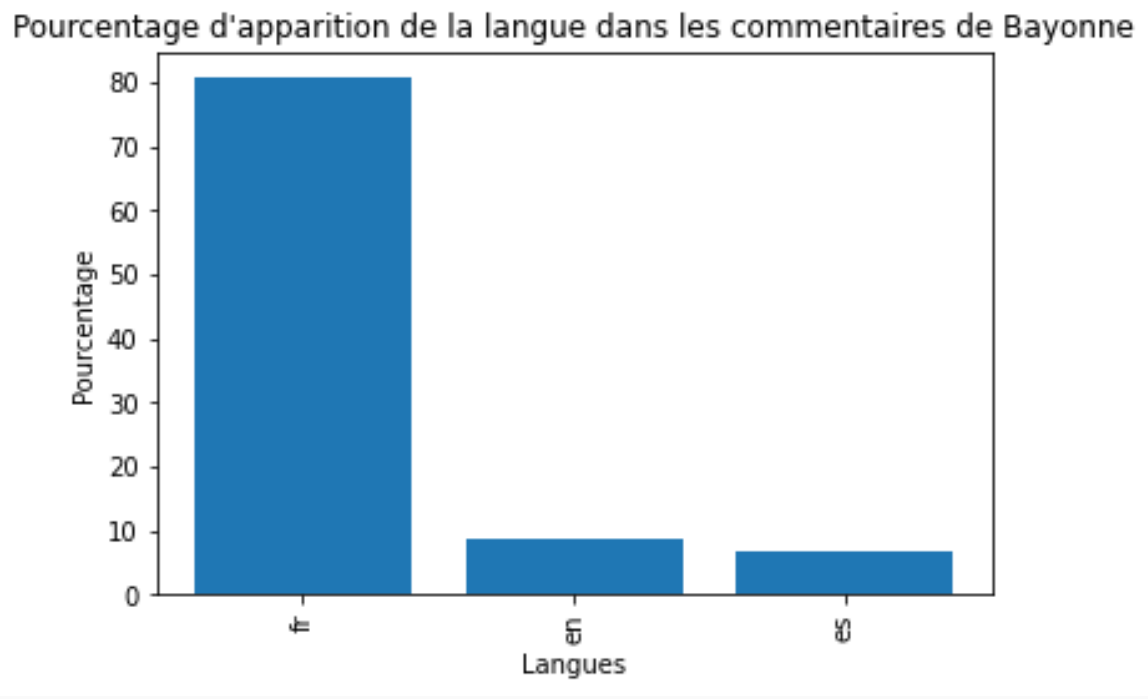




### Résultats affichés

Voici le graphique affiché pour cette requête.

Ville choisie : Bayonne. Pourcentage minimal choisi : 1%



## Requête 3

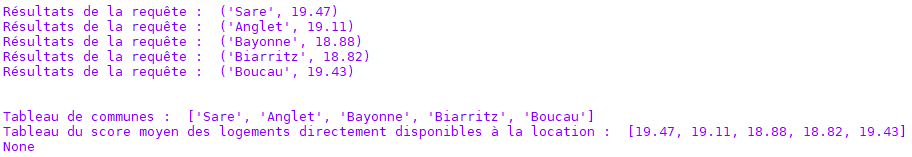
### Entrée utilisateur

### 

### Traces d’exécution

Les traces d’exécution des requêtes sont visibles en ayant entré 4 précédemment.

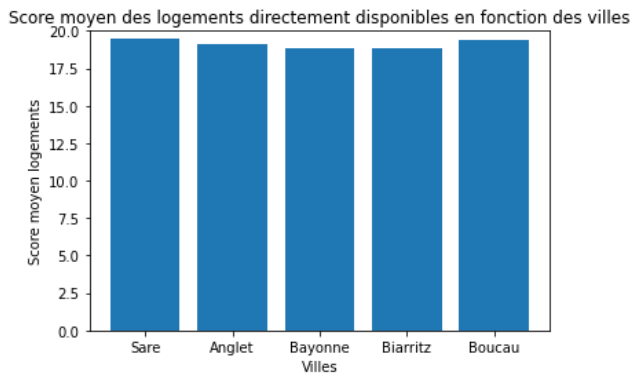
Affichage de l’exécution de la requête n°3 en prenant pour ville Sare, Bayonne, Biarritz, Anglet et Boucau.



### Résultats affichés

Voici le graphique affiché pour cette requête.

Villes choisies : Sare, Bayonne, Biarritz, Anglet et Boucau.



# Bilan

La partie que nous estimons avoir mieux traitée est la requête 2. Elle était également la plus intéressante car elle utilise plusieurs paramètres différents pour son exécution et affiche un résultat trié. L’utilisation des langues parlées est aussi une facette que nous avons aimé utiliser car de nombreux enregistrements étaient différents et avaient un nombre de traitements possibles important.

Chaque membre du groupe à travaillé sur chaque aspect de ce projet (descriptif des requêtes sur le premier livrable, le code SQL exécuté sur les serveurs de Lakartxela, l’application Python sur l’IDE Spyder et enfin le rapport final) et de façon complémentaire et régulière.

Autoévaluation : A; nous estimons avoir rempli les attentes du travail demandé, tout en ayant des requêtes intéressantes, et en ayant subdivisé le travail de façon équitable.

# Code source

"""

Créateurs :

- Latxague Thibault TD2/TP4

- Masson Rafael TD2/TP4

"""

import pyodbc # import de la bibliotheque pyodbc pour faire des requetes sql en python

import matplotlib.pyplot as plt

# on connecte notre base de donnee ODBC

conn = pyodbc.connect('DSN=tlatxague')

cursor = conn.cursor()

#%% S2.04 -- Fonctions requetes

conn = pyodbc.connect('DSN=tlatxague')

cursor = conn.cursor()

tracesExec = False

#Requete 1

def r1():

#Création de listes qui représenteront chacune une colonne de notre BD

nomVille = []

nbrHotes = []

pop2021 = []

comm = []

propLogsPrincSec = []

propHotesHabts = []

nombreParam = input("Combien de villes cherchez-vous ? ") #On demande le nombre de villes pour l'exécution du programme

for i in range(int(nombreParam)):

nomVille.append(input("Quel est le nom de la ville ? : ")) #On ajoute la réponse dans la liste

#La requete SQL

SQLCommand = ("""

SELECT COUNT(L.host\_id) AS nombreHosts,

APB.Population\_2021, APB.Commune,

(APB.Res\_principales\_2020/APB.Logements\_2020)\*100

AS proportionLgtsPrincSec,

(COUNT(L.host\_id)/APB.Population\_2021)\*100

AS proportionHotesHabitants FROM logements L

JOIN nodenot\_bd.agglo\_paysbasque APB ON APB.code\_insee = L.logt\_codeINSEE

WHERE APB.Commune = ?

GROUP BY APB.Commune, APB.Population\_2021, proportionLgtsPrincSec

""")

print("\n")

#On exécute la requête autant de fois qu'il y a de paramètres donnés par l'utilisateur

for i in range(int(nombreParam)):

param = (nomVille[i])

cursor.execute(SQLCommand, param)

#On ajoute les élements de réponse dans les listes correspondantes

for row in cursor.fetchall():

if tracesExec : print("Résultats de la requête : ", row)

nombreHotes, population2021, commune, proportionLogementsPrincipauxSecondaires, proportionHotesHabitants = row

nbrHotes.append(nombreHotes)

pop2021.append(population2021)

comm.append(commune)

propLogsPrincSec.append(proportionLogementsPrincipauxSecondaires)

propHotesHabts.append(proportionHotesHabitants)

#Affichage des tableaux pour les traces d'exécution

if tracesExec : print("Tableau du nombre d\'hotes : ", nbrHotes)

if tracesExec : print("Tableau de la population en 2021 : ", pop2021)

if tracesExec : print("Tableau du nom de la commune : ", comm)

if tracesExec : print("Tableau de la proportion de logements secondaires avec les logements principaux : ", propLogsPrincSec)

if tracesExec : print("Tableau de la proportion d\'hotes avec les habitants : ", propHotesHabts)

#On génère notre graphique avec les listes comm et propHotesHabts

plt.bar(comm, propHotesHabts)

plt.xlabel("Villes")

plt.ylabel("Pourcentage de logements secondaires")

plt.title("Graphique des villes en fonction du pourcentage de logements secondaires")

plt.ylim(0, max(propHotesHabts)+10)

plt.show()

#○Requete 2

def r2():

#Création de listes qui représenteront chacune une colonne de notre BD

langueParlee = []

pourcentageLangue = []

#On demande à l'utilisateur le nm de l'unique ville

nomVille = (input("Quel est le nom de la ville ? : "))

#On demande à l'utilisateur le pourcentage minimal de pourcentage de la langue parlée

pourcentageMinimal = (input("Quel est le pourcentage minimale de la langue parlée souhaité ? (nombre positif entier ou décimal) : "))

#La requete SQL

SQLCommand = ("""

SELECT R.langue\_detectee, ROUND(COUNT(R.langue\_detectee)/(SELECT COUNT(R.review\_id)

FROM reviews R

JOIN logements L on L.log\_id = R.logement\_id

JOIN nodenot\_bd.agglo\_paysbasque APB ON APB.code\_insee = L.logt\_codeINSEE

WHERE APB.Commune = ?)\*100,2) AS langueSurRevues

FROM reviews R

JOIN logements L ON L.log\_id = R.logement\_id

JOIN nodenot\_bd.agglo\_paysbasque APB ON APB.code\_insee = L.logt\_codeINSEE

WHERE APB.Commune = ?

GROUP BY R.langue\_detectee

ORDER BY langueSurRevues DESC""")

#On exécute la requête

param = (nomVille, nomVille)

cursor.execute(SQLCommand, param)

#On ajoute les élements de réponse dans les listes correspondantes

for row in cursor.fetchall():

if tracesExec : print("Résultats de la requête : ", row)

lang, prct = row

langueParlee.append(lang)

pourcentageLangue.append(prct)

longueurLangues = len(langueParlee)

print("\n")

#On enlève les communes qui ont un pourcentage de langues parlées égal à pourcentageMinimal

for i in range(len(langueParlee)):

if pourcentageLangue[i] < float(pourcentageMinimal):

valeurLimite = i

if tracesExec : print("Valeur limite : ", valeurLimite)

print("\n")

break

if tracesExec : print("Pourcentage de la langue parlée : ", pourcentageLangue[i])

if tracesExec : print("Langue parlée : ", langueParlee[i])

#On supprime tous les éléments après la première langue qui atteint 0.1%

for j in range(longueurLangues - valeurLimite):

if tracesExec : print("Valeur qui va être supprimée (pourcentage) : ", pourcentageLangue[-1])

if tracesExec : print("Valeur qui va être supprimée (langue) : ", langueParlee[-1])

pourcentageLangue.pop()

langueParlee.pop()

print("\n")

#On affiche les tableaux pour les traces d'exécution

if tracesExec : print("Tableau des langues parlées : ", langueParlee)

if tracesExec : print("Tableau des langues : ", pourcentageLangue)

#On génère notre graphique avec les listes langueParlee et pourcentageLangue

plt.bar(langueParlee, pourcentageLangue)

plt.xlabel("Langues")

plt.ylabel("Pourcentage")

plt.title("Pourcentage d'apparition de la langue dans les commentaires de " + nomVille)

plt.xticks(rotation=90) # rotation des étiquettes à 90 degrés

plt.show()

plt.ylim(0, 100)

def r3():

#Création de listes qui représenteront chacune une colonne de notre BD

nomVille = []

comm = []

avgLgtScoreMoyen = []

#On demande le nombre de villes pour l'exécution du programme

nombreParam = input("Combien de villes cherchez-vous ? ")

for i in range(int(nombreParam)):

#On ajoute la réponse dans la liste

nomVille.append(input("Quel est le nom de la ville ? : "))

#La requete SQL

SQLCommand = ("""

SELECT a.Commune, ROUND(AVG(l.logt\_review\_scores\_rating)\*4,2) AS score\_moyen

FROM logements l

INNER JOIN reviews r ON l.log\_id = r.logement\_id

INNER JOIN nodenot\_bd.agglo\_paysbasque a ON l.logt\_codeINSEE = a.code\_insee

WHERE l.logt\_instant\_bookable = 1 AND a.Commune = ?

GROUP BY a.Commune;

""")

print("\n")

#On exécute la requête autant de fois qu'il y a de paramètres donnés par l'utilisateur

for i in range(int(nombreParam)):

param = (nomVille[i])

cursor.execute(SQLCommand, param)

#On ajoute les élements de réponse dans les listes correspondantes

for row in cursor.fetchall():

if tracesExec : print("Résultats de la requête : ", row)

nomCommune, scoreMoyenLgt = row

comm.append(nomCommune)

avgLgtScoreMoyen.append(scoreMoyenLgt)

print("\n")

#On affiche les tableaux pour les traces d'exécution

if tracesExec : print("Tableau de communes : ", comm)

if tracesExec : print("Tableau du score moyen des logements directement disponibles à la location : ", avgLgtScoreMoyen)

#On génère notre graphique avec les listes comm et propHotesHabts

plt.bar(comm, avgLgtScoreMoyen)

plt.xlabel("Villes")

plt.ylabel("Score moyen logements")

plt.title("Score moyen des logements directement disponibles en fonction des villes")

plt.ylim(0, 20)

plt.show()

#%% S2.04 -- Programme principal

while(True):

#On demande à l'utilisateur la requête qu'il souhaite exécuter

choixRequete = input(""" \n \n

Choisissez votre requête :

1 - Pourcentage de logements secondaires dans une ville

2 - Pourcentage de langues parlées dans les commentaires d'une ville

3 - Score moyen d'évaluation d'une ville

4 - Activer/ Désactiver les traces d'exécution

5 - Quitter

Aide - Si certaines villes ne sont pas affichées, vérifiez s'il y a une erreur dans l'ortographe de la ville, ou si elle existe au Pays Basque \n

Votre choix : """)

#Appel des requêtes en fonction du choix de l'utilisateur

match choixRequete:

case "1":

print(r1())

case "2":

print(r2())

case "3":

print(r3())

case "4":

if tracesExec:

tracesExec = False

print("Traces d'exécution désactivées")

else:

tracesExec = True

print("Traces d'exécution activées")

case "5":

break

case \_:

print("\n \n Erreur en entrée, reessayez : ")