

Mini Projet Folium de Nassim Manseur

Rajouts des Questions: 4, 5, 6 des cartes et de la structure de la base de données + explications des programmes python dans python et rajout de la base de données en SQL dans le fichier.

Question 1:

On va dans phpmyadmin/localhost puis on crée une nouvelle base de données avec les 3 tables :

- nom de la base : France_regions_depts

Ajouter la table "villes_200", en important le fichier

Ajouter la table "departements", en important le fichier

On va dans l'onglet importer, choisir un fichier et on choisit un par un les 3 fichiers.csv

Question 2:

-On doit d'abord importer les fonctions folium et mysql.connector:

```
import folium
import mysql.connector #Importation des 2 fonctions
```

-Puis on lie le programme python à la base de données franc_region_depts et on écrit la requête SQL dans mycursor.execute:

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="France_regions_depts")
mycursor = mydb.cursor()
#Permet de lier le programme python avec la base de données

mycursor.execute("SELECT latitude, longitude, nom_ville FROM villes_200 ORDER BY villes_200.population DESC LIMIT 20")
#Permet d'exécuter la requête dans la base de données
myresult = mycursor.fetchall() #Transforme le résultat en liste
```

-Ensuite on situe la localisation du point de commencement d'affichage de la carte:

```
c=folium.Map(location=[49.26667, 2.46667], zoom_start=15) #point de départ de la carte
```

-Puis le programme qui permet de rajouter les points sur la cartes :

```
NumeroVille=0 #création d'une variable pour connaître le numéro de chaque ville de la requête
for i in range(20): #pour i qui prend la valeur de 0 a 20

    longitude=float(myresult[NumeroVille][0])#Permet de convertir chaque chiffre en flotant
    latitude=float(myresult[NumeroVille][1])#Permet de convertir chaque chiffre en flotant
    nomville=str(myresult[NumeroVille][2])#Permet de convertir chaque chiffre en integer
    print(longitude,latitude,nomville)#Affiche la longitude , la latitude , le nom de la ville correspondante
    folium.Marker([longitude,latitude],popup=nomville).add_to(c)#Création du point sur la carte avec les bonne coordonnées et le nom mde la ville afficher
    NumeroVille=NumeroVille+1 # la valeur de numéro ville augmente de 1 comme un compteur pour faire le marqueur pour chaque ville

c.save("Question2.html")#crée la carte en htmlS
```

Question 3:

La requête dans php my admin est :

```
INSERT INTO villes_200
(population,nom_ville,longitude,latitude,code_commune,code_postal,dpt_ville)
VALUES (20518,'nogent sur oise',2.46667,49.26667,60463,60180,60),
(37400,'creil',2.48333,49.26667,60175,60100,60),
(13856,'montataire',2.43333,49.26667,60414,60160,60)
```

-On rajoute toutes les informations concernant les villes par rapport aux attributs de la tables .

Question 4:

-On doit d'abord importer les fonctions folium et mysql,connector:

```
import folium
import mysql.connector#Importation des 2 fonctions
```

-Puis on lie le programme python a la base de donnée franc_region_depts :

```
mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="France_regions_depts")
mycursor = mydb.cursor()
#Permet de Lié Le programme python avec La basse de donnée
```

-Ensuite on situe la localisation du point de commencement d'affichage de la carte:

```
c=folium.Map(location=[ 49.26667, 2.46667],zoom_start=15)  
#point de départ de la carte
```

-Enfin le programme qui va permettre de créer les marqueurs dans la carte des villes avec leurs noms:

```
def curseur(NombreVille,requete):  
    cpt=0  
    mycursor.execute(requete)  
    myresult = mycursor.fetchall()  
    for i in range(NombreVille):  
        longitude=float(myresult[cpt][0])  
        latitude=float(myresult[cpt][1])  
        nomville=str(myresult[cpt][2])  
        folium.Marker([longitude,latitude],popup=nomville).add_to(c)  
        cpt=cpt+1  
        print(i,longitude,latitude,nomville)  
    return
```

Ici j'ai décidé de faire une fonction curseur pour simplifier le programme visuellement.

En appelant juste les requêtes les unes à la suite des autres:

```
print(curseur(20,"SELECT latitude,longitude,nom_ville FROM villes_200 ORDER BY villes_200.population DESC LIMIT 20"))  
print(curseur(3,"SELECT latitude,longitude,nom_ville FROM villes_200 WHERE nom_ville='creil' OR nom_ville='nogent sur oise' OR nom_ville='montataire'))
```

-Ainsi je rentre uniquement le nombre de ville à ajouter et la requête pour que les marqueurs se fassent tout seuls.

-Et j'appelle la fonction avec un print pour être sûr que le programme fait ce que je lui demande.

```
c.save("Q4_Nassim_Manseur.html")#on enregistre la carte dans le html
```

-Et j'enregistre dans la carte.

Question 5:

-On commence avec l'importation des fonctions et le programme qui lie le python a la base de donnée:

```
import folium
import mysql.connector
#Importation des 2 fonctions

c= folium.Map(location=[50,3],zoom_start=15)
#point de départ de la carte

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="france_regions_depts")
mycursor = mydb.cursor()
```

-Ensuite on va continuer de la même manière pour chaque côté Nord , Sud,Ouest et Est pour trouver les villes les plus éloignées dans la métropole correspondant a ces critères:

```
mycursor.execute("SELECT latitude,longitude,nom_ville FROM villes_200 WHERE latitude BETWEEN 40 AND 52 ORDER BY latitude asc limit 1")
myresult=mycursor.fetchall()

for i in range(len(myresult)):
    latitude = float(myresult[i][0])
    longitude = float(myresult[i][1])
    nom = str(myresult[i][2])
    folium.Marker([latitude, longitude],popup=nom).add_to(c)
c.save('Q5_Nassim_Manseur.html')
```

-On va faire la même chose a chaque fois(donc 4 fois) il y a juste la requête qui change

```
mycursor.execute("SELECT latitude,longitude,nom_ville FROM villes_200 WHERE latitude BETWEEN 40 AND 52 ORDER BY latitude asc limit 1")
mycursor.execute("SELECT latitude,longitude,nom_ville FROM villes_200 WHERE longitude BETWEEN -5 AND 10 ORDER BY longitude desc limit 1")
mycursor.execute("SELECT latitude,longitude,nom_ville FROM villes_200 WHERE longitude BETWEEN -5 AND 10 ORDER BY longitude asc limit 1")
```

-Les requêtes sont tous de la même forme elles prennent la latitude , la longitude et le nom de la ville de chaque villes correspondant aux conditions qui , elles vont changer pour chaque requêtes.

-On cherche a chaque fois entre -5 et 10 pour la longitude et 40 et 52 pour la latitude.

-En faite c'est pour ne pas dépasser les frontières de la France métropolitaine(La Corse est officiellement inclus dedans).

-Et on trie ensuite soit par ordre croissant ou décroissant en limitant le nombre de résultat a 1 pour bien avoir seulement les 4 villes, celles qui seront les plus aux extrémités de la France métropolitaine.

Question 6:

-Donc on commence comme toutes les autres questions c'est adire par l'importation des fonction le lien avec python et la base de donnée :

```
import folium
import mysql.connector
#Importation des 2 fonctions

mydb = mysql.connector.connect(
    host="localhost",
    user="root",
    password="",
    database="france_regions_depts")
mycursor = mydb.cursor()
```

-Puis ensuite on va faire une série de requête pour répondre a la demande de la question:

```
mycursor.execute("ALTER TABLE regions ADD IF NOT EXISTS capt_region VARCHAR(255)")
```

-La requête crée la colonne capt_region dans la relation région si elle n'existe pas déjà.

-Si on ne met pas la condition ça nous affichera une erreur comme quoi il y a déjà la colonne si on refait le programme.

-Puis on continue en faisant:

```
mycursor.execute("SELECT nom_ville FROM villes_200 JOIN departements ON
villes_200.dpt_ville=departements.num_dpt JOIN regions ON
regions.code_region=departements.code_region WHERE
code_commune=code_commune_capt")
```

-Pour trouver les villes qui sont les capitales de régions en liant la table departement ,la table ville_200 et la table regions car dans la table departement et la table region il y a les codes de ces capitales et dans la table ville_200 il y a le nom de ces capitales .

-Pour terminer :

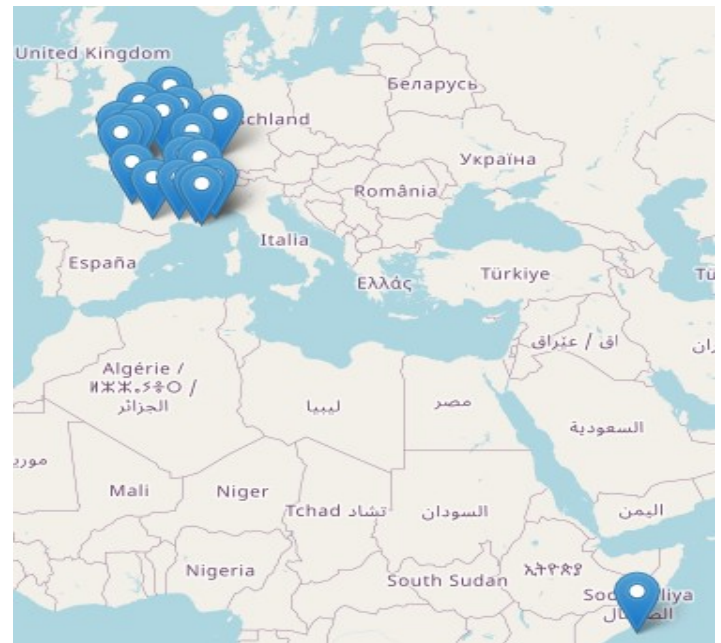
```
myresult=mycursor.fetchall()
#Permet d'enregistrer dans la variable myresult les donnée trouvé sous forme de matrice donc une liste de liste
print (myresult)
#On affiche le résultat pour vérifier si la requête est bonne

mycursor.execute("UPDATE regions JOIN villes_200 ON villes_200.code_commune=regions.code_commune_capt SET capt_region=nom_ville WHERE code_commune=code_commune_capt")
#Permet de remplir la colonne qu'on a crée auparavant et on y met Les capitales de chaque régions.
```

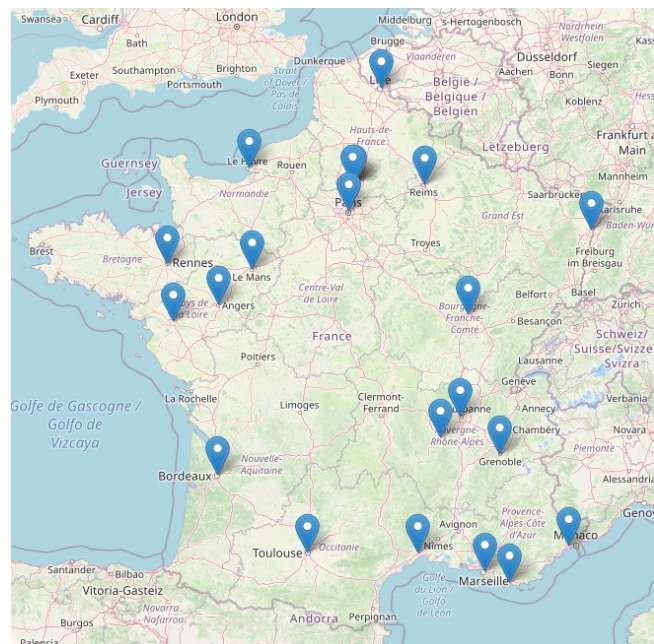
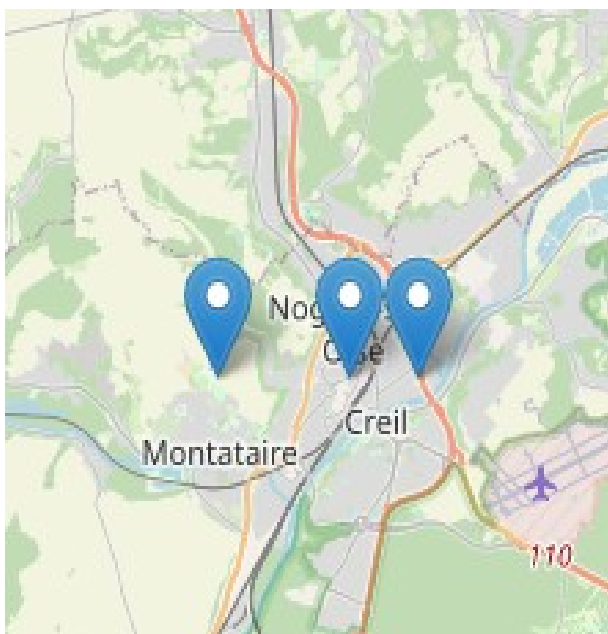

-On enregistre le nom des capitales dans la matrice puis on les affiche pour être sûr qu'il n'y pas d'erreur et on refait une autre requête qui va rajouter dans la nouvelle colonne les nom des capitales correspondant a leurs régions.

RESULTATS DES CARTES

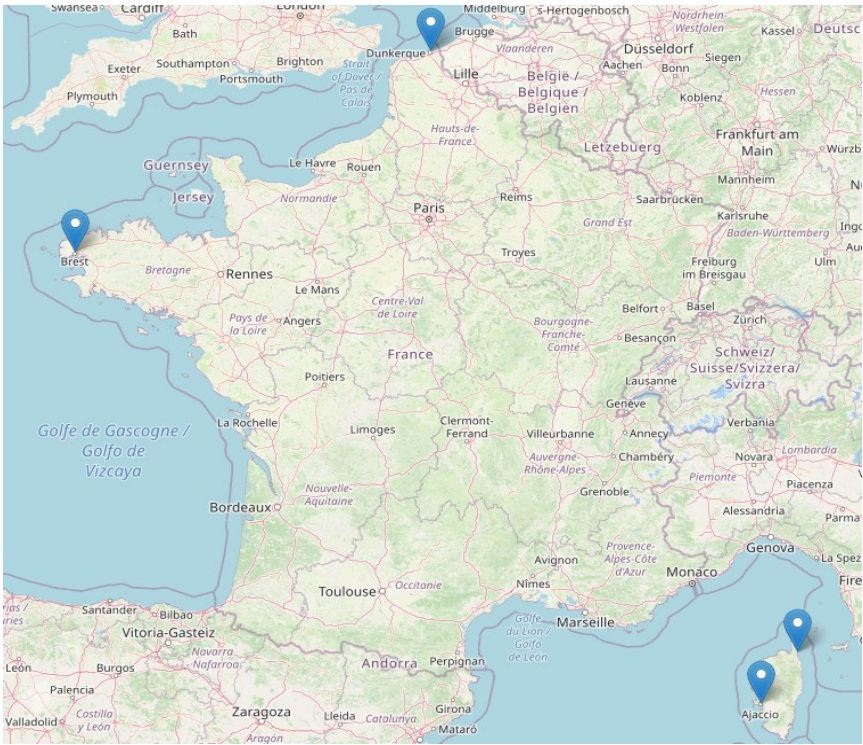
Les 20 plus grandes villes de France:



Les 20 plus grandes villes de France + Creil, Nogent et Montataire:



Les villes les plus au Nord, au Sud, à l'Est et à l'Ouest de la métropole:



Structure des bases de la base de donnée:

départements:

← Serveur : 127.0.0.1 » Base de données : france_regions_depts » Table : departements

Parcourir Structure SQL Rechercher Insérer Exporter Importer Privilèges Opérations Suivi

Structure de table Vue relationnelle






















#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1 num_dpt	varchar(7)	utf8_general_ci		Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	2 nom_dpt	varchar(23)	utf8_general_ci		Oui	NULL			Modifier Supprimer Plus
<input type="checkbox"/>	3 code_region	varchar(11)	utf8_general_ci		Oui	NULL			Modifier Supprimer Plus

↑ ☐ Tout cocher Avec la sélection : Parcourir Modifier Supprimer Primaire Unique Index Spatial

region:

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	code_region	varchar(11)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	2	nom_region	varchar(21)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	3	code_commune_capt	varchar(17)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	4	capt_region	varchar(255)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus

ville_200:

	#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/>	1	code_commune	varchar(12)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	2	nom_ville	varchar(21)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	3	code_postal	varchar(125)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	4	dpt_ville	varchar(9)	utf8_general_ci		Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	5	population	int(10)			Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	6	longitude	float			Oui	NULL			 Modifier  Supprimer  Plus
<input type="checkbox"/>	7	latitude	float			Oui	NULL			 Modifier  Supprimer  Plus