

## SIG : Géoinformatique

### Exercice 5 : Opérations sur les couches

Vous êtes employé.e par la commune de Lausanne pour faire une étude sur l'urbanisation de la commune. Vous devez calculer la longueur cumulée des routes de la commune ainsi que la surface au sol occupée par tous les bâtiments. La commune aimerait que vous fournissiez toutes ses informations directement dans une couche vectorielle en format geopackage. Vous devez donc créer un code dans la console python qui vous permette de calculer ces différents éléments et qui vous permette aussi de les ajouter à la couche comme nouveaux attributs.

Vous avez à disposition les données suivantes :

- Commune\_lausanne.gpkg : Les limites de la commune de Lausanne avec les attributs suivants :

	fid	Commune	Canton	numero_id
1		5 Lausanne	Vaud	1

- Buildings\_lausanne.gpkg : La couche vectorielle contenant la surface au sol occupée par tous les bâtiments de la commune de Lausanne avec les attributs suivants :

Anlagen, Revisionen - Gebäude, Seite 14/21, Internet V4.0.7, Index															
	fid	id	swiss	baufahrtenstrasse	baufahrtenstrasse	erstellung_jahr	erstellung_monat	grundlaendung	herkunft	herkunft_jahr	herkunft_monat	objektart	revisions_jahr	revisions_monat	revisions_quart
1	14289	14289 (7548427) 422...	25/06/2014	25/04/2014	2008	4	Herbeizent	herbeizent	2011	4	Gebäude	2020	4	2021_Akt	
2	14290	14290 (3963776) 445...	25/06/2014	25/04/2014	2008	4	Herbeizent	herbeizent	2011	4	Offener Flächen...	2020	4	2021_Akt	
3	14291	14291 (34433430) 434...	25/06/2014	25/04/2014	2008	4	Herbeizent	herbeizent	2011	4	Gebäude	2020	4	2021_Akt	
4	14292	14292 (36077817) 430...	08/10/2014	25/04/2014	2008	4	Herbeizent	herbeizent	2011	4	Gebäude	2020	4	2021_Akt	

Comme bien souvent en Suisse, les attributs sont en allemand. Heureusement pour ceux qui n'ont pas appris l'allemand, nous allons utiliser ici uniquement la géométrie des différentes features.

- Roads\_lausanne.gpkg : La couche vectorielles contenant toutes les routes de la commune de Lausanne avec les attributs suivants :

	fid	id	swiss	daten_erstellung	daten_modifizierung	erstellung_jahr	erstellung_monat	grundlaenderung	herkunft	herkunft_jahr	herkunft_monat	objektart	revisions_jahr	revisions_monat	revisions_quart
1	14488	14488 (34781) 3063306 421...		26/11/2021	27/04/2009	2005	4	Herbeizent	herbeizent	2020	4	4m Strasse	2020	4	2021_Akt
2	14489	14489 (34781) 3063306 421...		26/11/2021	27/04/2009	2005	4	Herbeizent	herbeizent	2020	4	3m Strasse	2020	4	2021_Akt
3	14490	14490 (34781) 3063306 421...		26/11/2021	27/04/2009	2005	4	Herbeizent	herbeizent	2020	4	3m Strasse	2020	4	2021_Akt
4	14491	14491 (34781) 3063306 421...		26/11/2021	27/04/2009	2005	4	Herbeizent	herbeizent	2020	4	4m Strasse	2020	4	2021_Akt

Comme pour les bâtiments, biens que les attributs soient en allemand, nous nous intéressons ici uniquement aux géométries.

A l'aide de ces couches vectorielles, ajoutez les attributs suivants à la couche communes\_lausanne.gpkg :

- La longueur cumulée de toutes les routes de la commune avec comme nom d'attribut « longueur\_cumulee » (float).

- La surface totale occupée par les bâtiments de la commune en hectares avec comme nom d'attribut « `surface_cumulee` » (float).
- Le pourcentage de la surface totale que représentent les bâtiments de la commune avec comme nom d'attribut « `pct_surface` » (entier).

De plus, l'attribut « `numero_id` » provient d'une erreur. Faites en sorte de retirer cet attribut de la couche vectorielle.

Pour mener à bien ce petit mandat, vous aurez probablement besoin de faire appel aux fonctionnalités suivantes de PyQGIS :

#### Créer un nouvel attribut :

```
layer.dataProvider().addAttributes([QgsField(attribute_name,QVariant.type)])
```

- `attribut_name` : une chaîne de caractères comme par exemple "pct\_surface"
- `QVariant.type` : le type de variable conservée dans l'attribut comme par exemple : `QVariant.Double`, `QVariant.Int`, etc. Il faut donc changer *type* par *Int* ou *Double* par exemple.

Une fois les différents attributs créés, il est nécessaire de mettre à jour la couche avec la méthode `updateFields()` tel que :

```
layer.updateFields()
```

#### Ajouter / modifier la valeur d'un attribut :

Pour ajouter une valeur ou modifier la valeur d'un attribut pour un feature spécifique, il est nécessaire d'activer le mode **édition** de la couche sur laquelle on fait ces modifications comme suivant :

with edit(layer) :

```
#appliquer les différentes modifications ici (sur une feature avec la couche en mode édition)
```

```
feature[«attribut_name »] = ....
```

```
layer.updateFeature(feature)
```

Il faut mettre à jour la feature dès qu'on a fini de la modifier avec la méthode `updateFeature(feature)` tel que :

```
layer.updateFeature(feature)
```

#### Supprimer un attribut :

Pour supprimer un attribut, il faut récupérer de l'attribut par le biais de son nom avec la méthode suivante :

```
indice = layer.fields().indexOfName(«attribut_name »)
```

Et ensuite supprimer l'attribut avec la méthode `deleteAttribute(indice)` tel que :

```
layer.deleteAttribute(indice)
```

Attention, si vous ajouter des nouveaux attributs et que vous les modifiez, faites bien attention de mettre à jour les attributs et les features avant de supprimer un attribut pour ne pas avoir de confusion au niveau des indices.