

# Fichiers finaux :

---

## Requêtes SQL

Fichier	Description
<b>1_bdd_one.sql</b>	permet la création du schéma de base de données. 'one' fait référence au fait qu'elle ne permet qu'un seul livre par point, ce qui facilitera le traitement sur la base de données (déduplication des lignes, calcul, ...) Pour la version final il faudra juste passer sur la version de base de données qui permet plusieurs livres par points d'acupunctur.
<b>2_meridiens.sql</b>	permet la création des méridiens en bdd. Ce fichier est distinct car il a été modifié à la main pour être nettoyé afin d'éviter des doublons inutiles (espace en plus, ...) en base.
<b>3_inserts.sql</b>	permet l'insertion des données en base.
<b>4_attentions_a_ne_pas_oublier.sql</b>	permet de s'assurer qu'aucune indication n'est orpheline, le cas contraire serait anormal.
<b>all.sql</b>	regroupe toutes les requêtes précédentes.

Ces requêtes sont à exécuter manuellement dans le SGBD PostgreSQL une fois la base de données créée manuellement. Je conseille d'exécuter que **all.sql** car moins d'action nécessaire, ce fichier a été créé manuellement alimenté par le résultat du script **3\_inserts.py**, fichier **meridiens.sql** et du fichier **1\_bdd\_one.sql**.

## Autres

Fichier	Description
<b>la video .mp4</b>	vidéo permettant de retracer la création de la plupart des requêtes SQL (orientées DEV pas forcément client car difficilement compréhensible).
<b>INFOS_-Chuan_Min_Wang-_Introd_2024_05_22.xlsx</b>	copie renommée (avant out_clean.xlsx) du fichier de données final utilisé pour créer les requêtes SQL, principalement par <b>3_inserts.py</b> .

---

## Fichier développement (*dossier script*)

---

### Processus

- Livre Google Drive en .DOCX (microsoft word)
- Transformer en Google Docs (ouvrir et convertir) (car GAS ne peut pas traiter les documents Word)
- Depuis le Google sheets servant de table de stockage, exécuter le google apps script lié (Google Sheets > Extensions > Apps Script)
- Copier coller le code de **0\_googleAppsScript.gs** dans l'ide. Modifier l'id du document source (visible dans l'url du fichier voulu) et exécuter le script
- Télécharger sous forme **.xlsx** le résultat stocké sur Google Sheets
- Disposer le classeur **répertoire\_des\_scripts/data/extract\_script\_clean.xlsx**
- Exécuter le script **1\_script\_v3.py** celui-ci exécutera automatiquement le script suivant **2\_clean.py**
- **out\_clean.xlsx** Corriger les quelques erreurs de données restantes du fichier de sortie
- **out\_clean.xlsx** Récupérer tous les méridiens et les insérer en base (c'est le fichier **2\_meridiens.sql**).

- [3\\_inserts.py](#) exécuter comme dans la [vidéo](#)

Pour d'autres détails [cf.readme\\_image\\_process.png](#) [source](#)

## Fichier

Fichier	entrée	sortie	Description	colonnes
<b>0_googleAppsScript</b> (sous Google Sheets)	id de livre spécifié dans le script	feuille au nom du fichier d'entrée (à exporter)	Premier script exécuté, récupère les noms des [Point] avec leurs contenu [texte] Point est le délimiteur et Texte le contenu de se point.	2 [ Point , Texte ]
<b>1_script_v3.py</b>	data/extract_script_clean.xlsx (export_depuis_google_sheets)	out.xlsx	Décompile la colonne Texte parmi les 2 colonnes en entrée en autant de colonne qu'il peut décompiler tout en se limitant aux clés (Indication, Meridien, ...) répertorié dans le script > variable <a href="#">allowed_keys</a> .	2 + Autant qu'il y en a [Point, Texte, Localisation, ...]
<b>2_clean.py</b>	out.xlsx	out_clean.xlsx	Transforme les données, exemple regrouper des colonnes de mêmes sens mais orthographié différemment, ...	2 + les précédentes - celles fusionnées
<b>3_inserts.py</b>	out_clean.xlsx + ses corrections manuel	inserts.sql	Génère les requêtes contenant les données à insérer en base	
<b>mapping.json</b>			Utilisé par <a href="#">3_inserts.py</a> . Permet de donner la correspondance [ nom_attribut_livre_dans_excel ] et [ nom_attribut_bdd ] et également dans quel [ table_bdd ]	

## Points d'amélioration

- Transférer le process de Google Apps Script dans le script python 1\_script\*.py faire sur un script python
- Récupérer les images également (fonctionne sur Google Apps Script)
- Simplifier le processus pour l'intégration en base, stocker tous les points dans un fichier Excel contenant tous les attributs nécessaire afin de faire l'intégration en BDD qu'une seul fois.

## Lien entre les BDD

Raimanu avait proposé de lier les bases Acupunctur et Plantes sur l'attribut Indication, ce qui à première vue paraît très bien.

# Interface base de données

---

Avec Microsoft Access on peut utiliser un connecteur ODBC PostgreSQL facilement pour synchroniser l'état visible dans Access et la base de données.

Je pense qu'il serait plus facile et rapide d'utiliser cette solution en dépit de codage.

En effet, les besoins ne sont pas très élevés sur cette partie et MS Access dispose déjà de formulaire et de tout un fonctionnement conçu dans le but.

- Il faut mettre en place un formulaire permettant la consultation et l'insertion manuel des données.

## Application

---

L'utilisation de Unity semble une bonne solution.

Je pense que cela nécessitera d'ajouter des attributs en base de données pour cette application afin qu'elle puisse attribuer à une coordonnée du modèle 3D un point d'acupuncture.