Documentation

jeudi 21 juillet 2022 09:06

Le but des scripts va être de mettre à jour les données des quatre tables en base. Ainsi, nous avons trois scripts qui mettent à jour respectivement les tables LOC_FICH, LOC_FICH_COLN, et LOC_DATA. La mise à jour de la table LOC_INDC est à faire manuellement sur Azure data studio.

Cependant, avant de lancer les scripts, une partie manuelle est à effectuer au préalable. Notamment pour LOC_FICH, la table des fichiers, qui constitue le point de départ des mises à jour.

Paramètres du projet

Etape 1: téléchargement/déplacement des fichiers de données:

Fichiers INSEE:

- Si le fichier que l'on compte télécharger est de source INSEE, on veillera à bien prendre le format csv
- Quand on téléchargera un nouveau fichier de données, on veillera à ce que le séparateur des données soit bien un point-virgule (c'est essentielle). Normalement, tous les fichier INSEE sont standardisés avec un point-virgule.
- Quand on télécharge le fichier INSEE, on a en réalité téléchargé un fichier zip que l'on déposera dans le répertoire "base initiales" (Simple sauvegarde des données originales)
- Ne pas le décompresser, ce n'est pas nécessaire. Simplement l'ouvrir et récupérer le fichier de données communales (une abréviation dans le nom, "COM", permet de l'identifier) et de le copier dans le répertoire des données "base de donnée". Normalement, un fichier de métadonnée dans le zip lui ait associé, celui-ci est à copier dans le répertoire "metadonnée".

Fichiers AUTRE:

- Simplement veillez à télécharger le fichier au format csv avec comme séparateur un point-virgule.
- Copier le fichier dans le répertoire "base de donnée"

Etape 2: règle de nommage:

Fichier INSEE:

- La règle de nommage des fichiers de données sera la suivante :

  base-(source)-(thème)-(année)-geo(année).csv

  Source correspond au fait qu'il s'agisse d'un fichier INSEE ou autre, donc on mettra "insee" ou "autre". Thème correspond au sujet du fichier, c'est-à-dire logement, tourisme,.... L'année est tout simplement l'année des données et celle suivant le mot geo est l'année du découpage géographique.

- La règle de nommage des fichiers de métadonnées sera la suivante :

  meta-base-(source)-(thème)-(année)-geo(année).csv

Fichier AUTRE:

- Même règle de nommage que pour les fichiers de données INSEE. L'année des données et du découpage géographique est à renseigner si elle est connu. Par exemple, pour le fichier artificialisation des sols, cela donnerai :

  base-autre-artificialisation_sol.csv

- Ouvrir le fichier de données, repérer la colonne correspondant aux codes géographiques et renommer la colonne "CODGEO" pour standardisé avec les fichiers INSEE pour la lecture du fichier par python. (à valider)

Etape 3: cas de suppression/déplacement de fichier du répertoire

- Si vous souhaitez ne plus disposez d'un fichier du répertoire pour une raison quelconque, vous pouvez le supprimer. Cette situation sera prise en charge par le script de mise à jour.

Attention !!! Si vous changer l'emplacement du répertoire, vous devrez adapter le chemin d'accès au répertoire dans le script python. Il y a quatre chemin différent nécessaire aux mises à jour.

Path_donnee : Le chemin d'accès au répertoire des fichiers .csv de données
Path_meta : Chemin d'accès au métadonnée des fichiers INSEE
Path : Chemin d'accès au dossier complet
Path_standard: Chemin d'accès où se situeront dans un premier temps les fichiers Excel des tables générer

Etape 4: Modification des métadonnées

Quand vous aurez déplacer puis renommer un nouveau fichier de métadonnée, il va falloir le "nettoyer":

- Ouvrez le fichier de métadonnée
- Supprimer toute les métadonnées pour qu'il ne reste que la colonne des indicateurs et leur libellé
- Supprimer également toutes les lignes "CODGEO" inutile
- Supprimer aussi les indicateurs qui sont en double, triple,… Se sont en fait des variables qualitatives dont chaque modalité ont un libellé.

Le mieux est de copier directement les données qui nous intéressent dans une autre feuille, renommer la feuille au même nom que le fichier, supprimer la feuille originale.

Une fois toute ces étapes effectuées, on peut passer au lancement du 1er script, le script "maj_fichier.py".

Lancement des scripts

maj_fichier.py

Le script de mise à jour de la table LOC_FICH a été lancé.

Ce que fais le script:

Le script va dans un premier temps effectuer une sauvegarde des données présentes actuellement en base pour pouvoir ainsi détecter, avec le scan, les nouveaux fichiers arrivant et ceux supprimés.
Ensuite, un dataframe pandas est construit, il contient toutes les nouvelles données à insérer en base.
Le cas des suppression est également vérifié et traité.
Et enfin, on procède à l'insertion de ce dataframe en base, en attendant un peu, la nouvelle table s'affichera. On peut maintenant passer à la partie manuelle à effectuer avant le lancement du script de LOC_FICH_COLN.
Une sauvegarde des 3 tables avant mise à jour est effectué à la fin de ce script. Elles sont là pour nous permettre de revenir à une version des tables avant mise à jour. Comme ça, si jamais on fait une erreur, on peut lancer le script de la table où l'on s'est trompé pour la réinitialiser à la version ultérieure.

Remarque: Pour plus de détails sur le côté programmation, des commentaires ont été écrit tout au long du code. Une version notebook du script est également disponible pour permettre de suivre et de comprendre les différentes étapes des raisonnements suivi.

Partie manuelle:

Cette partie manuelle existe pour la simple raison de nous permettre de décider quelle fichiers nous souhaitons mettre à jour les données (fichiers déjà présent dans LOC_DATA), et lesquelles on souhaite importer les nouvelles données (nouveaux fichiers).
Il a été décidé de procéder ainsi :

- Ouvrir Azure data studio et se connecter à la base Carto.
- Clic droit sur la table LOC_FICH et sélectionner "Edit Data".
- Mettre à 0 la colonne des B_IMPR pour les fichiers non souhaité.
- Mettre à 1 la colonne des B_IMPR pour les fichiers souhaité mettre à jour et importé.
- Mettre à 1 la colonne des B_EXPR pour indiquer le dernier fichier en date pour ce thème.
- Compléter les donnée qui peuvent être renseigné si elles sont connu.

Remarque d'utilisation du logiciel: Quand on complète une cellule de la table, pour que cette donnée soit enregistrer il ne faut pas oublier de faire la touche Entrer pour valider le contenu de la cellule.

Si vous pensez avoir terminé, on clique sur run et les données saisi sont alors prise en compte. On procéder à la mise à jour de la table LOC_FICH_COLN.

maj_correspondance.py

Le script de mise à jour de LOC_FICH_COLN a été lancé.

Ce que fait le script:

Le script va dans un premier temps effectuer une sauvegarde dans un dataframe pandas de la table LOC_FICH_COLN. Ensuite il récupère les fichiers à B_IMPR à 1 de la table LOC_FICH où il va les ouvrir et récupérer les toutes les variables dans deux dictionnaires. Un pour les fichiers INSEE et un deuxième pour les fichiers AUTRE. S'il s'agit de fichier INSEE, le script ne s'embête pas il va ouvrir les métadonnées associé et récupérer l'indicateur et son libellé.
Ainsi, un dataframe est créée avec une correspondance pour chaque indicateur INSEE (rien sinon) et le libellé est ajouté également. Ce dataframe affiche donc toute les lignes à insérer dans la table LOC_FICH_COLN. Le cas d'un ajout ou d'une suppression d'indicateur au sein du fichier est également pris en compte.
Pour plus de détails sur le code lui-même, des commentaires ont été écrit tout au long des lignes de codes. Une version notebook du script est disponible aussi pour visualiser étapes par étapes le raisonnement suivi.

Parti manuelle:

La partie manuelle pour cette étape est à faire pour les tables LOC_FICH_COLN et LOC_INDC. La procédure se fait comme suit:

LOC_INDC:

- Si de nouveaux indicateurs souhaité font leur apparition, on renseigne dans la table LOC_INDC dans une nouvelle ligne l'indicateur ADEME, sont libellé (la signification de l'indicateur, ce qu'il représente), et le thème (logement, artificialisation,…)

Pour saisir une nouvelle ligne sur Azure data studio, il suffit de commencer à saisir la nouvelle ligne sur la ligne où il n'y que des valeurs *NULL*. Ne pas oublier la touche ENTRER après chaque saisi. La clef primaire se générera automatiquement dès que vous commencerez à saisir une nouvelle ligne.
Conseil : Quand vous n'avez plus de donnée à insérer, commencez à saisir une nouvelle ligne pour que la clef primaire puisse se générer. Dès que celle-ci est générer vous pouvez supprimé la ligne que vous étiez en train de saisir.

LOC_FICH_COLN:

- Mettre à 1 les B_IMPR des indicateurs souhaités.
- Si un indicateur souhaité n'est pas un indicateur INSEE, alors il n'aura pas de correspondance ADEME, il faudra donc la renseigner et renseigner la même correspondance que celle indiqué dans LOC_INDC précédemment.
- Si vous souhaité ajouter et/ou supprimer un indicateur dans un fichier à importer, si c'est dans un fichier INSEE, ne pas oublier de le saisir également dans son fichier de métadonnée sinon il n'apparaîtra pas en base ou apparaîtra dans le cas d'une suppression.
- Si vous ajouter un indicateur, ne pas oublier de mettre son B_IMPR à 1 et de renseigner ca correspondance dans LOC_INDC.
- **(Optionnel)** Renseigner les libellés et commentaire pour chaque indicateurs.
- Si deux fichiers souhaités ont le même thème, alors ils auront les mêmes indicateurs ADEME qui sont eux intemporelles. Par conséquent, il faudra choisir quelle correspondance on supprime pour n'en avoir qu'une. La correspondance qui sera retenue importera alors les données de l'indicateur du fichier correspondant. Si l'on souhaite implanté les deux fichiers au date différentes, alors il faudra procéder par étapes (on en importe 1 des 2, ensuite on intervertir, on accorde la correspondance à l'autre indicateur et met le précédent à nulle).

  Attention !!! Si vous supprimer des éléments d'une table, ne pas la laisser blanche, il faut mettre "*NULL*" en majuscule puis faire ENTRER.
  Attention !!! Azure data studio n'affiche que les 200 premières lignes, or LOC_FICH_COLN en possède bien plus. Par conséquent, il faut modifié l'option "Max row" à côté de la flèche verte pour run et cliquer sur run pour afficher toute les lignes, ensuite vous pouvez éditer la table.

- Vérifiez que tout est correct puis lancer le fichier maj_data.py.

maj_data.py

Ce que fait le script:

Etape 1 : Récupération des 3 tables et téléchargement des fichiers Excel

Dans un premier temps, le script récupère dans des dataframes les données de LOC_FICH (fichiers aux B_IMPR =1) LOC_FICH_COLN et LOC_INDC. Ensuite, il télécharge leur fichiers Excel dans le dossier FDLD de l'application.

Etape 2 : Les vérifications

On effectue les vérifications indiquer dans la procédure sous forme de 4 fonctions renvoyant True ou False. Ainsi, l'import des données ne pourra se faire si le résultat retourné de chaque fonction n'est pas True. A noter qu'il il y a en réalité 5 fonctions de vérification. En effet, la fonction "verif_correspondance_uniq" vérifie qu'il n'existe qu'une correspondance dans d'un indicateur dans la table LOC_INDC.

Etape 3 : Création du dataframe des données à insérer

- Une liste de tuple est créer associant le fichier, son id et l'année du fichier s'il en a.
- Une autre contenant des tuples associant le fichier, l'indicateur INSEE, la correspondance, et l'année. Si pas d'année et de correspondance alors il n'y a rien (None ou nan)
- On créer la même liste que précédemment mais en associant en plus l'Id de l'indicateur correspondant dans la table LOC_INDC.
- Pour créer ensuite le dataframe, on génère un dictionnaire avec comme clef le tuple (id_fichier, fichier, année) de la première liste de tuple. Et comme valeur une liste contenant tous les tuples des indicateurs lui appartenant.
- Création du dataframe avec les code géographique. Une correction est appliqué pour les codes comme ceci : "1001" au lieu de "01001". Et le choix de l'année à choisir entre celle indiquer pour le fichier et celle de l'indicateur est gérer à ce moment-là de la procédure.

Etape 4 : Insertion/suppression des données

- Téléchargement d'un fichier Excel contenant simplement les Id des fichiers ajouter pour pouvoir supprimer et réimplanter les données supprimer au cas où on voudrai revenir en arrière.
- Suppression des données
- Insertion des données

Pour plus de détails sur le code lui-même, des commentaires ont été écrit tout au long des lignes de codes. Une version notebook du script est disponible aussi pour visualiser étapes par étapes le raisonnement suivi.

Export_open_dat.py

Ce que fait le script

Etape 1 : Récupération de la vue "EXPORT_CSV"

- On récupère dans un premier temps le nom de toute les colonnes pour ne pas taper à la main les colonnes que l'on souhaite par la suite. (Fonction "info_view")

- Récupération de la table "EXPORT_CSV". La récupération est faite dans le désordre (toute les communes délégués sont mises en première), donc on réarrange la table par ordre alphabétique des codes communes.
- Puis on supprime les colonnes qui nous sont inutiles ("C_CODE_LOGEMENT",...).

Etape 2: Agrégation, niveau départementale, régionale et EPCI

- On effectue une copie du dataframe créer à partir de la récupération de "EXPORT_CSV". Ce dataframe est donc au niveau communale.
- On remarque que certaine variable sont des proportion d'autre indicateur (Population municipale et taux de femme). Donc on créer une fonction qui va transformer cette proportion en des valeurs comptés pour pouvoir sommer correctement à la phase de l'agrégation. Cette fonction s'appelle "tx_en_valeur".
- On agrège les données
- On recalcule les taux avec la fonction "recalc_tx".

Etape 3: Export fichier CSV

- Test si le fichier existe déjà dans le répertoire où ces fichiers csv se trouveront.
- Si ils sont présent, on supprime et on génère le fichier csv et on le déplace dans le répertoire.
- Sinon on génère simplement le fichier csv et on le déplace dans le répertoire.

Parti manuelle :

Si vous remarquer dans la vue des données niveau communales qu'il y a la présence d'indicateur qui se trouve être des proportions calculés à partir d'autre indicateur présent dans la table, il va falloir procéder comme suit:

Ligne 131

- Se mettre à la ligne
- Appeler la fonction "tx_en_valeur" avec en argument le dataframe "df", l'indicateur de référence qui est utilisé pour calculé la proportion (à saisir entre guillemets), puis l'indicateur correspondant aux proportions calculées.

Ligne 149

- Se mettre à la ligne
- Appeler la fonction "recalc_tx" puis renseigner en argument la même chose que la fonction précédente dans le même ordre.