

# Projet 5 - Utiliser les données publiques d'Open Food Facts

## Généralités

Programmes écrits en Python3. Dépôt Github [https://github.com/horlas/OC\\_Project5](https://github.com/horlas/OC_Project5)

**Librairies utilisées:**

- **requests** (interrogation et rappatriement des données d'Open Food Facts).
- **mysql-connector-python**: communication avec la base de données.
- **pandas**: affichage en tableau.
- **sqlalchemy**: ORM, communication avec la base de données.

**Base de données** : Mysql administrée avec PHPAdmin.

## Contenu du projet

Le projet comporte les programmes et les fichiers suivants :

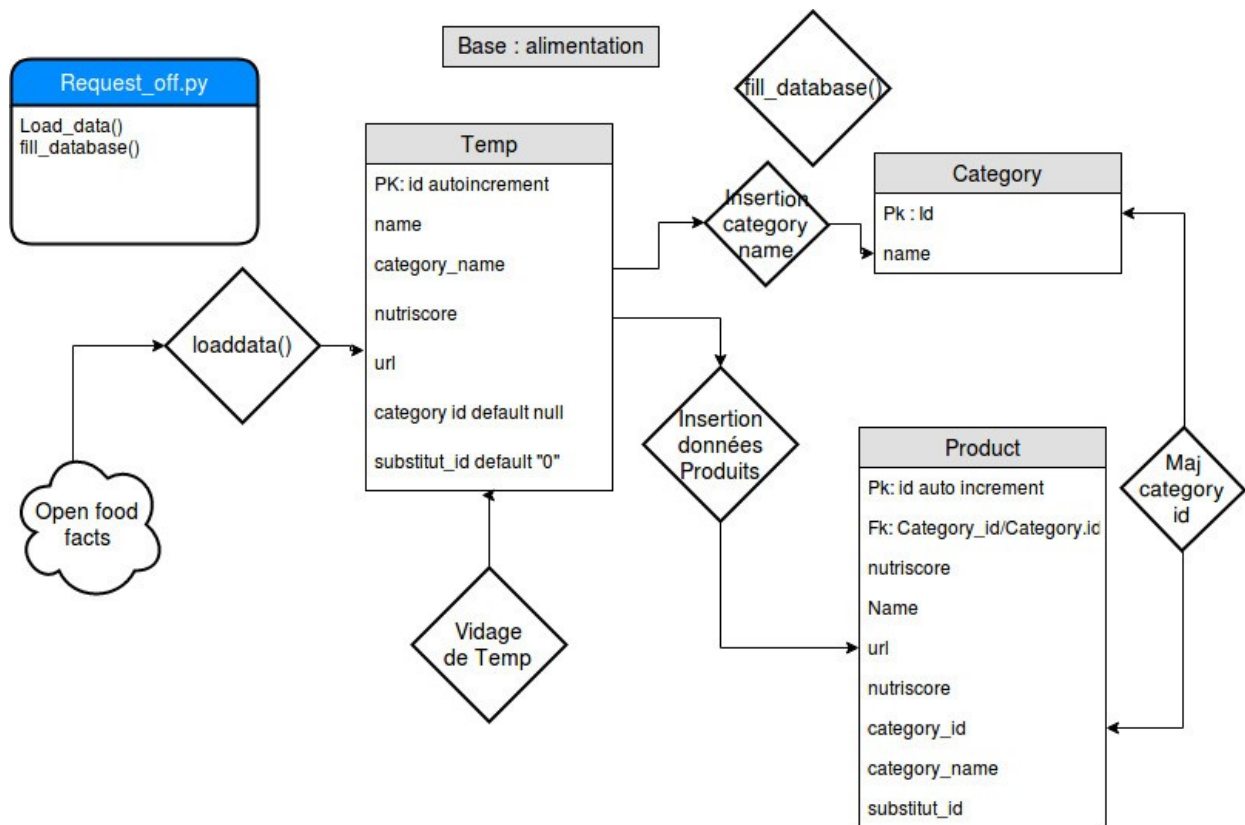
- **create\_database\_alimentation.sql** : script sql de créations des tables
- **request\_off.py** : programme python de remplissage de la base de données à partir des données d'OpenFood Facts
- **substitute\_food.py** : programme python de l'application demandée par la projet : l'utilisateur peut saisir un produit , un substitut lui est proposé, il peut l'enregistrer et par la suite. Il peut retrouver tous les produits substitués sauvegardés.
- **Constants.py** : comportent les constantes et les requêtes sql
- **classes.py** : définit la classe Product et ses fonctionnalités détaillées plus avant dans ce document.  
Ces deux derniers fichiers sont partagés par les deux programmes .
- **data\_analysis** : programme « bonus » pour traiter le jeu de données de la base après l'import. Il est écrit en python3 et utilise l'ORM Sql Alchemy.

## Structure de la base de données et description de son remplissage

Le programme request\_off.py a deux fonctions :

- **Load\_data()** : importe les données d'OpenFoodFacts. Pour les nécessités du projet, seules les trois premières pages d'un nombre de catégories, choisies arbitrairement, sont importées. Cependant, il est possible d'activer l'importation de l'ensemble des produits contenus dans une catégorie (ligne 23 request\_off.py : `ceil(count/20) # 20 = pagination`)  
Load\_data fait appel à la classe Product pour créer des objets produits avec, comme attributs, les différents champs importés depuis Open Food Facts.
- **Fill\_database** : cette fonction fait appel à la procédure stockée de la base Mysql fill\_database qui fait les actions suivantes :
  1. Insertion des données de la table Temp vers la table Product.
  2. Insertion du nom de la catégorie de la table Temp vers la table Catégorie. L'incrémenter des id Product et Catégorie sont fait par auto-incrémenter ce qui valide une incrémenter correcte.

3. Mise à jour de l'id de la catégorie sur de la table Catégorie vers la table Product.
4. Vidage de la table Temp pour éviter le surpoids de l'ensemble de la base alimentation.



## Application de choix et substitut d'un produit substitut\_food.py

### Fonctionnalité 1

Le programme propose à l'utilisateur de choisir une catégorie puis un produit qu'il consommerai . Lorsque la saisie est faite , un objet de class Product est créé avec tous ses attributs saisis par l'utilisateur : `p_selected` . (méthode `define_product`).

Le programme fait alors appel à la méthode `substitut()` qui sélectionne dans la même catégorie un produit avec un meilleur nutriscore .

**Focus** sur le choix du produit de substitution et le calcul du meilleur nutriscore :

Le nutriscore du produit sélectionné est rentré sous forme de lettre et convertit grâce à la méthode `conversion` de la class Product, qui renvoie le nutriscore en chiffre (`ns_number`).

Une fois ce chiffre obtenu la méthode `substitut()` crée un dictionnaire de tous les produits appartenant à la même catégorie et comportant un nutriscore inférieur en terme de chiffre . Un random choice est opéré sur ce dictionnaire pour en extraire une proposition de produit substituant. Dans le cas où le produit sélectionné est déjà avec un bon nutriscore et ne présente pas de meilleur choix, un message comme quoi il n'y a pas de produit substituant est affiché sur l'écran.

C'est alors qu'un deuxième objet Product est crée sous le nom de `p_substitut`.

Une synthèse est affichée à l'écran : produit sélectionné , produit substituant .

L'utilisateur peut alors , si il le désire, enregistrer son produit.

### Fonctionnalité 2

Grâce à la librairie pandas un affichage de tous les produits enregistrés est fait .

Cet affichage est généré par la fonction du programme `substitut_food` , `recorded_substitut()`. Cette fonction appelle la requête `query_substitut_product`.

**Focus** sur la vue `substitut_product` de la base : Nous avons créé cette vue pour permettre d'afficher le nom du produit substituant et non son id et le nom du produit sélectionné. La double requête devant s'effectuer sur la même base nous avons choisi de créer une vue qui fait une première

correspondance (substitut.id vers name) pour ensuite faire une deuxième requête sur le nom du produit.

Un système de boucles est mis en place permettant à l'utilisateur de saisir plusieurs produits sans sortir du programme ou de passer de la fonctionnalité 1 à la fonctionnalité 2. Le choix 0 permet de sortir du programme proprement.

Les saisies de l'utilisateur sont validées par des blocs « try » except » empêchant ce dernier de saisir autre chose que les choix proposés.

