

# OPENCLASSROOMS

## PROJET 5 OC

Utilisez les données publiques de l'OpenFoodFacts

*Jimi Bourgeois*

Dépôt repo Github : [https://github.com/Bibjim/P5\\_OpenFoodFacts\\_OC](https://github.com/Bibjim/P5_OpenFoodFacts_OC)

Trello : <https://trello.com/b/FaKmnu5U/projet-5-off>

Bienvenue sur la base de données d'OpenFoodFacts.com (Version FR)

Menu principal de la base de données:

- 1 - Charger la base de données en local\* ?
- 2 - Rechercher un produit de substitution dans la base ?
- 3 - Accéder de vos produits favoris ?
- 4 - Gestion de la base (suppression des données)

\*Pour la première utilisation du programme, charger les données dans la base en local

Votre choix (Entrer Q pour quitter le programme):

- But du projet:  
Utiliser les données du site OpenFoodFacts afin de créer un programme qui propose à l'utilisateur des substitues de produits alimentaires de meilleurs qualités grâce aux nutriscores.
- Le projet a été réalisé en plusieurs étapes:
  - L'analyses des données de l'API REST
  - La construction de la base de données (MPD)
  - La création de la structure du programme
  - L'écriture du script du programme
- La synthèse des données de l'API m'a permis de projeter la base de données. La difficulté était de comprendre la structure du fichier JSON dans son ensemble afin de prendre les informations essentielles pour la base.
- La base de données comporte 3 tables. Une table pour stocker les catégories de produits, la seconde pour les produits et la suivante pour stocker les sauvegardes des produits de substitution.
- Suite à la mise forme de la base, la création de la structure du programme a été une étape essentielle pour pouvoir débiter le script python.

## Utilisez les données publiques de l'OpenFoodFacts

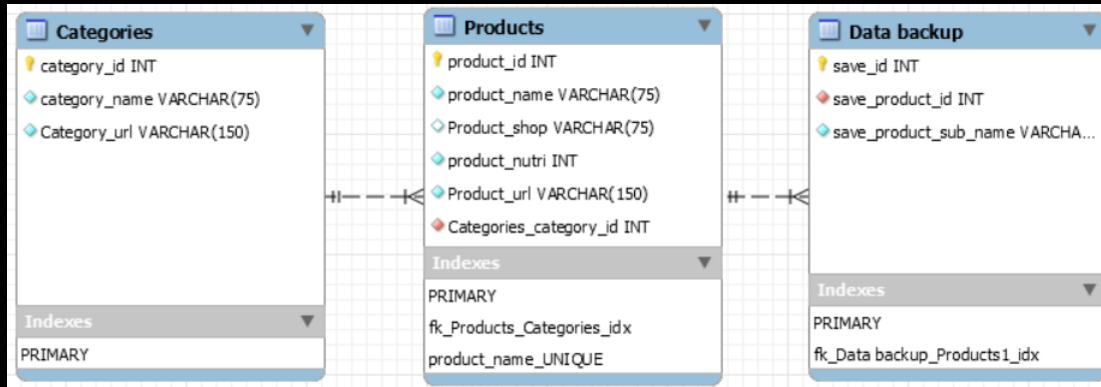
### OPENCLASSROOMS

- Le programme est divisé en 3 modules. Un module principal qui contient le menu principal du programme, un sous-module pour gérer les requêtes de la base de données et un sous-module pour gérer les requêtes de recherche de substitue.
- La principale difficulté était l'importation des données dans la base. Un produit mal référencés sur les données publiques sont l'origine de plusieurs modification du script au fur et à mesure de l'avancement du projet. J'ai dû créer un script afin de nettoyer les données avant de les intégrer dans la base de données.
- La seconde difficulté était de mettre en forme la requête pour rechercher un produit de substitution. J'ai utilisé une requête qui retourne le meilleur produit de la catégorie suivant le produit sélectionné que l'on souhaite substituer. Si le produit sélectionné a déjà le nutriscore le plus élevé de la catégorie, le programme vous demande de choisir un autre produit avec un nutriscore moins élevé.

# Utilisez les données publiques de l'OpenFoodFacts

## OPENCLASSROOMS

### Modèle physique de données (mysql workbench)



### Structure du Programme

**Module principal**  
App.py

**Sous-module**  
mydb\_gestion.py

**Sous-module**  
substitute\_products.py

**Base de données**  
mydb

### Script de création de la base de données

```
mydb_gestion.py mydb_gestion.py db_request

# Connection to the SQL database
self.db_connect = pymysql.connect("localhost","root","","mydb")
db = self.db_connect.cursor()

# Request SQL of the "Categories" table
table_cat = "CREATE TABLE IF NOT EXISTS `mydb`.`Categories` \
(`category_id` INT NOT NULL AUTO_INCREMENT,\
`category_name` VARCHAR(150) NOT NULL,\
`category_url` VARCHAR(200) NOT NULL,\
PRIMARY KEY (`category_id`)) ENGINE = InnoDB"

# Request SQL of the "Products" table
table_prod = "CREATE TABLE IF NOT EXISTS `mydb`.`Products` \
(`product_id` INT NOT NULL AUTO_INCREMENT,\
`product_name` VARCHAR(150) NOT NULL,\
`product_shop` VARCHAR(150) NULL,\
`product_nutri` VARCHAR(75) NULL,\
`product_url` VARCHAR(200) NOT NULL,\
`Categories_category_id` INT NOT NULL,\
PRIMARY KEY (`product_id`),\
INDEX `fk_Products_Categories_idx` (`Categories_category_id` ASC),\
UNIQUE INDEX `product_name_UNIQUE` (`product_name` ASC),\
CONSTRAINT `fk_Products_Categories` \
FOREIGN KEY (`Categories_category_id`) \
REFERENCES `mydb`.`Categories` (`category_id`) \
ON DELETE NO ACTION \
ON UPDATE NO ACTION) \
ENGINE = InnoDB"

# Request SQL of the "Products" table
table_sub = "CREATE TABLE IF NOT EXISTS `mydb`.`Substitute` \
(`save_id` INT NOT NULL AUTO_INCREMENT,\
`save_product_id` INT NOT NULL,\
`save_product_sub_name` VARCHAR(150) NOT NULL,\
PRIMARY KEY (`save_id`),\
INDEX `fk_Substitute_Products1_idx` (`save_product_id` ASC),\
CONSTRAINT `fk_Data_backup_Products1` \
FOREIGN KEY (`save_product_id`) \
REFERENCES `mydb`.`Products` (`product_id`) \
ON DELETE NO ACTION ON UPDATE NO ACTION)"

db.execute(table_cat)
db.execute(table_prod)
db.execute(table_sub)
```