Projet 5 : Utilisez les données publiques de l'OpenFoodFacts (OFF)

Etudiant : SENGMANY Nicolas
Mentor : THOMAS Gaël
Évaluateur de soutenance : Ranga GONNAGE
Code source : https://github.com/nicoseng/Projet-5-OFF

Ce document résume le projet 5 et s'inscrit dans le cadre de la formation de développeur d'application Python d'Openclassrooms.

Objectif du projet

Créer une application permettant de proposer à l'utilisateur un aliment de substitut de meilleure qualité nutritionnelle que celui choisi au départ.

Choix de l'algorithme

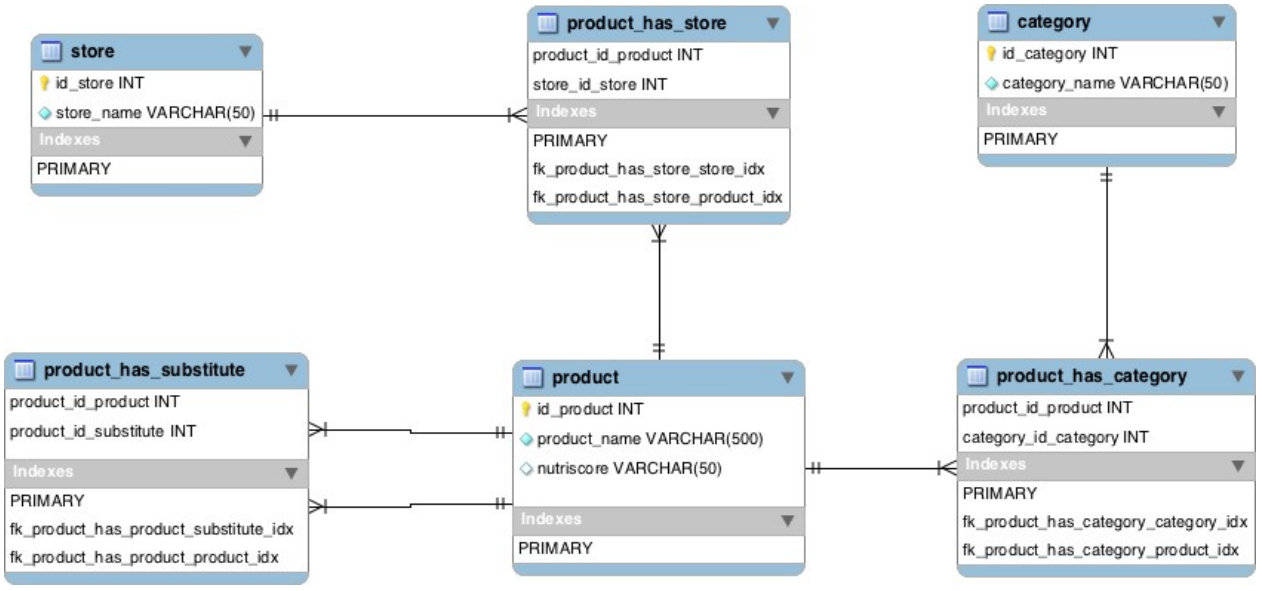
Création et choix de la base de données

Un produit peut appartenir à différentes catégories, tout comme une catégorie peut renfermer plusieurs produits. Il en va de même avec les magasins : Un magasin peut renfermer plusieurs produits et un produit peut-être répertorié dans plusieurs magasins. De ce fait, le modèle relationnel adopté est celle de relier chaque table (product, category et store) via une table dite « intermédiaire » permettant ainsi de matérialiser les différentes relations de type « plusieurs à plusieurs ». Ainsi, un produit est relié à son magasin via une table de relation nommée product_has_store. De la même façon, un produit est relié à une catégorie via une table intermédiaire : product_has_category. Enfin, un produit peut également posséder un substitut, d'où la table relationnelle liant un produit à un substitut potentiel : product_has_susbstitute.

Les différentes tables intermédiaires contiendront ainsi les différentes combinaisons possibles via un jeu de clé primaire/étrangère.

Ainsi par exemple, si deux produits numérotées respectivement 2 et 3 appartiennent à une même catégorie numérotée 5, la table de relation nommée product_has_category contiendra alors les combinaisons 2-5 et 3-5. Ces numéros permettront ensuite de rechercher les noms des catégories possédant ces numéros dans la table nommée category.

L'ensemble de la démarche nous a permis d'aboutir au modèle de données ci-dessous :



Le choix de l'algorithme adopté lors de la conception du code repose initialement sur la question suivante : Si je devais chercher des informations sur un produit comme dans un catalogue de supermarché, quelles sont les actions dont j'aurai besoin pour trouver et stocker les informations souhaitées ?

A chaque table créée est associée un lot d'actions ordonnées. Le tableau ci-dessous représente les fonctions internes propres à chaque table (hors création de la base de données en elle-même):

Nom de la table	Description brève	Actions correspondantes à la table	Nom des fonctions liées aux actions
product	Recueille les données relatifs aux produits	1. Créer une table product 2. Vérifier le nombre de produits de la table product ; 3. Récupérer les produits dans OFF; 4. Ajouter les produits dans la table product; 5. Sélectionner la table product; 6. Afficher la table product pour l'utilisateur.	1. create_product_table() 2. fill_category_table() 3. select_product_table() 4. display_product_by_category_a nd_store() 5. display_product_selected()
category	Recueille les catégories de produits	1. Créer une table category ; 2. Vérifier le nombre de catégories de la table category; 3. Récupérer les catégories dans OFF; 4. Ajouter les catégories dans la table category; 5. Sélectionner la table catégory; 6. Afficher la table category pour l'utilisateur.	1. create_category_table() 2. fill_product_table() 3. select_category_table()
product_has_category	Recense les combinaisons des numéros liant un produit à une catégorie	1. Créer une table product_has_category 2. Récupérer les numéros de produit et de catégorie 3. Insérer ces numéros dans la table product_has_category.	1. create_product_has_category_t able() 2. fill_product_has_category_tab le ()
store	Recense les magasins des produits	1. Créer une table store ; 2. Récupérer les magasins dans OFF; 3. Vérifier si le magasin en question figure déjà dans la table store; 4. Ajouter les magasin dans la table store; 5. Afficher la table store pour l'utilisateur.	1. create_store_table() 2. add_or_get_store() 3. select_store_table()
product_has_store	Recense les combinaisons des numéros liant un produit à un magasin	1. Créer une table product_has_store 2. Récupérer les numéros de produit et de magasin ; 3. Insérer ces numéros dans la table product_has_store.	1. create_product_has_store_ta bl e() 2. fill_product_has_store_table()
product_has_substitute	Recense les combinaisons des numéros liant un produit à un substitut potentiel	1. Créer une table product_has_substitute 2. Récupérer les numéros de produit et du substitut ; 3. Vérifier si la combinaison des numéros produit/substitut ne figure pas déjà dans la table product_has_substitute; 4. Insérer ces numéros dans la table product_has_substitute. 5. Sélectionner la table product_has_substitu te 6. Afficher la table product_has_substitute pour l'utilisateur	1. create_product_has_substitute _table() 2. check_substitute() 3. add_or_get_substitute() 4. select_substitute_table() 5. display_substitute_basket() 6. delete_susbstitute_basket()

Méthodologie de projet

La méthodologie de projet adoptée est la méthodologie agile (voir tableau Trello) : A chaque fonctionnalité est assignée un ensemble de tâches à réaliser selon un calendrier et des délais définis préalablement.

Difficultés rencontrées

Pour le cinquième projet du parcours Python, les difficultés résidaient principalement dans :

- la compréhension des bases de données (concept nouveau pour un débutant)
- l'apprentissage dans la gestion du projet (système de cartes sur Trello, gestion du temps)
- la gestion des erreurs à anticiper : cette difficulté a été surmontée en choisissant de se mettre à la place de l'utilisateur et en posant les questions suivantes :
    - Quelles sont les erreurs que l'utilisateur est susceptible de produire ?
    - A partir des erreurs potentielles listées, quelles sont les méthodes apprises en Python afin de remédier à ces erreurs ?

Au delà des aspects techniques et des automatismes de code à acquérir, il est à noter que même si les phases de recherches de bugs constituent parfois des moments de découragements, le fait de trouver la solution procure une joie immense contribuant au plaisir de coder et font sans aucun doute partie des aspects du métier du développeur.