

Mémoire du projet Openfoodfact - Python

1. Mise en place la base de données.

J'ai lu l'énoncé du projet et j'ai schématisé la base de données sous le logiciel Mysql WorkBench, j'ai ensuite fait un export de du schéma en requêtes SQL dans un fichier. Comme toute requête SQL, elles sont séparées par des points-virgules.

Dans mon script d'initialisation de la base de données, si la base n'existe pas, le programme l'a créée. Ensuite créée, j'update mon singleton avec comme paramètres de connexion le nom de la base de donnée en supplément. Ce me permettra d'exécuter l'ensemble des requêtes du fichier exporté de MysqlWorkbench, après l'avoir parser et identifier les requêtes séparées par les points virgules. Cette opération s'effectue uniquement à la premier initialisation du programme. Comment je sais si le programme est en première ou deuxième utilisateur? Grâce à l'existence de la base de données.

2. Importation des catégories dans la base depuis l'API.

Grâce au module "requests", mon programme est capable d'effectuer vers une source extérieur. Je suis allé sur le site de l'API et j'ai cherché la liste des catégories. J'en ai choisit 2, le programme envoie donc 2 requêtes et l'API renvoie les données correspondants à ces 2 catégories. J'ai d'abord afficher la réponse pour lire la structure de l'API, puis j'ai utilisé les fonctions pythons pour parser la réponse. Pour chaque catégorie, je vérifie d'abord que la catégorie existe en base et si elle n'existe pas, alors je fais l'insert en base. Puis j'ajoute les produits correspondant à la catégorie, les marques, et les relations entre catégorie et produit si requête mysql renvoie un id de catégorie non nulle. A chaque instance de programme des requêtes sont lancées pour savoir si les catégories existent.

3. Programme avec les questions réponses.

L'explication de cette partie sera en partie expliquée à l'oral lors de la soutenance. Je vais néanmoins expliquer l'algorithme de substitution du programme.

L'utilisateur réponds au programme en choisissant un produit. Il a donc un score appelé nutriscore. Ce nutriscore va de la lettre A à Z. J'ai donc listé toute les lettres de l'alphabet. Si le nutriscore est égale à A alors je renvoie A comme nutriscore meilleur car c'est le maxium. Si le nutriscore du produit est C, pour pouvoir trouver son substitut, le programme renverra une liste de lettre, A et B. C'est donc la recherche du substitut que je passerais en paramètre la liste des lettres, que je parserais et que je passerais dans where_in. Ensuite la fin de la requête trie le nutriscore du substitut par ordre ASC et limit de 1, ce qui permet au programme de renvoyer le meilleur substitut.

En jaune les données, en rose les questions et en vert les réponses du programme.