Auteur : *Pellegrini Emanuele*
Projet 5 – OFF Advisor

PROJET 5: FICHE EXPLICATIVE

Vous pouvez trouver mes codes source **ici**.

DÉMARCHE ADOPTÉE

1. Récupération des données depuis le fichier .csv de l'Open Food Facts: j'ai téléchargé la base de données depuis le site web (1,25Go).

2. Triage des données: j'ai crée un script en Python afin de trier seulement les informations utiles à la création de ma base de données 'myoff_db'. Vu que dans le cahier des charges était marqué que l'applicatif concernait uniquement des utilisateurs français, seulement les produits se trouvant en France et ses départements d'outre mer ont été pris en compte.
La taille du fichier .csv avec les données triées fait 3,68Mo.

3. Création de la base de données: j'ai crée ma base de données avec la syntaxe SQL 'DROP DATABASE IF EXISTS' afin d'en créer une nouvelle à chaque fois qu'on initialise ce script. Ensuite j'ai réglé les paramètres comme le codage, les droits d'accès pour moi-même etc.

4. Création des tables: j'ai créé les tables pour ma base de données en choisissant leur type par rapport à leur contenu (ex. *tinyint* s'il y avait un nombre demandant 1 octet d'espace, ou *mediumint* s'il fallait en avoir 3). Ensuite j'ai défini les clés primaires et étrangères pour chaque table. Je vais détailler leur signification et structure:

- Category est la table où se trouvent les 10 catégories entre lesquelles sont divisés les produits du fichier 'off_selected.csv'. La clé primaire est *id*, c'est à dire l'identification d'une catégorie (ex. 1= gâteaux).

- Dish est la table qui héberge tous les aliments (7897 produits regroupés en 316 pages) avec leurs informations. La clé primaire est *id*, le nombre d'identification d'un produit spécifique parmi les autres. Dans cette table on utilise une clé étrangère '*category_id*' qui se réfère à '*Category(id)*' afin de structurer les 10 différents catégories d'aliments.

- Store est la table où il y a tous les magasins dans lesquels l'utilisateur peut trouver les aliments (245 en total). En ce cas aussi, la clé primaire est *id*, qui définit un magasin en particulier.
- Dish_Store est la table où l'utilisateur peut trouver un aliment spécifique grâce à ses attributs 'dish_id' et 'store_id' utilisés en tant que clés étrangères vers les tables Store (en particulier *Store_id*) et Dish (en particulier *Dish_id*).

- Healthier_Substitute est la table contenant les aliments sauvegardés par l'utilisateur; soit les aliments choisis au début, soit l'alternative proposée par le programme. Cela est possible grâce aux attributs '*origin_id*' et '*alternative_id*' se référant à la table Dish et en particulier à *Dish(id)* et utilisés en tant que clés étrangères.

5. Entrée des données: j'ai donc inséré les données dans la base table par table (sauf pour la table Healthier_Substitute, bien entendu, qui contient seulement les éventuelles données sauvegardées par l'utilisateur). Pour faire cela j'ai utilisé des boucles qui parcourent le fichier '*off_selected.csv*' et récupèrent les données souhaitées.

6. Création des classes: je me suis concentré ensuite sur les classes nécessaires au fonctionnement du programme principal.

- Classe Connection pour gérer les connexions et les paramètres concernant la base de données.
- Classe List pour collecter les id des différentes tables de la base de données.
- Classe Mysql_queries afin de structurer l'affichage des données, la recherche d'un aliment substitutif à partir d'un aliment donné et encore la possibilité de sauvegarde de l'aliment choisi et de son substitut.
- Classe Outcome pour montrer les résultats des queries SQL au niveau de l'utilisateur.

7. Écriture du programme principal: au final j'ai écrit en Python le programme de l'OFF Advisor avec lequel l'utilisateur interagira pour avoir des conseils nutritifs et manger des repas plus sains.

DIFFICULTÉS RENCONTRÉES

J'ai eu beaucoup de difficultés dans ce projet au niveau de la programmation SQL. Ma totale inexpérience a rendu l'apprentissage assez difficile et je suis resté bloqué pendant plusieurs semaines sur l'insertion et la structuration des données (clés, contraintes, index…). Pour résoudre le problème j'ai temporairement basculé sur le projet 6 qui m'a apporté plus de notions et plus de temps avec ces nouveaux concepts. Grâce a cela j'ai réussi à terminer mon travail. Les ressources externes comme Stack Overflow et les différents forums de discussion ont été très utiles.

MÉTHODOLOGIE DE PROJET

Pour réaliser ce projet j'ai choisi une méthodologie agile prévoyant la réalisation d'un tableau Trello et la définition des *sprints* à travers des *user stories*. Ces *stories* m'ont permis d'avoir une vue panoramique sur les différentes phases du projet, à partir de sa conception, en passant par la réalisation des fonctionnalités, jusqu'à la phase des tests.
Vu que j'ai travaillé seul et pas en équipe, je n'ai pas utilisé des pratiques communes lorsqu'on parle de méthodologie agile, telles que le *pair programming* ou le *daily standup meeting*, même si une fois par semaine j'avais la vidéoconférence avec mon mentor (bien assis sur ma chaise).
Toutefois j'ai essayé de faire du *sprint planning* et estimer la durée de chaque *user story* car je trouve important de se fixer des deadlines pour se motiver et respecter des temps.
En détail on pourrait définir ma méthodologie Scrum parce qu'au début j'ai listé toutes les fonctionnalités marquées dans le cahier des charges (*backlog*); ensuite j'ai crée un programme minimal avec les fonctionnalités principales qui peu à peu est devenu plus complet grâce à la finalisation des différents *sprints*.