Pur Beurre

Github : https://github.com/hjung06/P05_pur_beurre

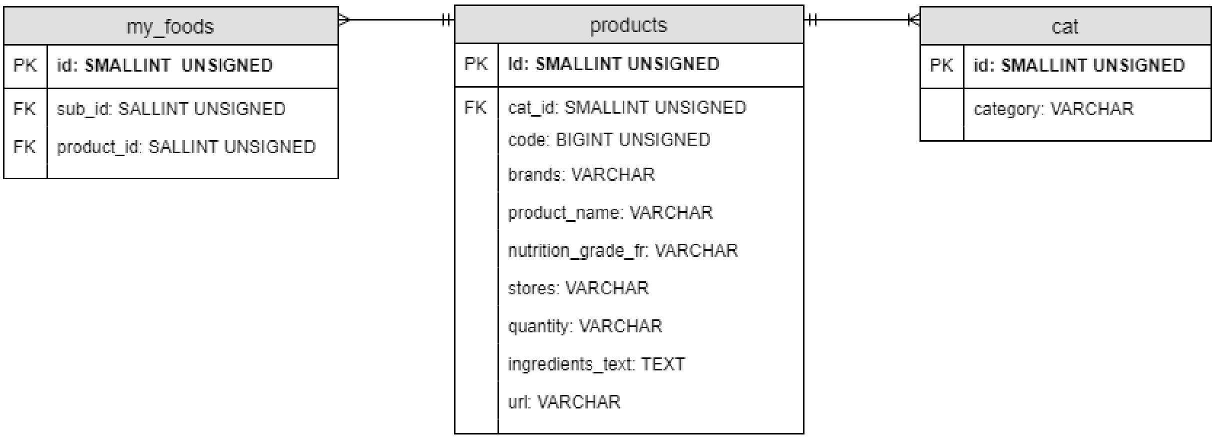
Trello : https://trello.com/b/FOSpG6tf/pur-beurre

Introduction

The goal of this project was to use API and RDBMS to retrieve and extract data in order to provide necessary information for users to help them make better choices by comparing foods they desired.

Openfoodfacts.org API and **MySQL** version 5.7 were used to retrieve and store data needed for this project.

Physical database design



Files

api_requests.py: contains a class object for retrieving json from openfoodfacts.org using requests python module. It also dumps the json into python list object.

Convert.py: contains a class object appends the products from each category to one big list object which will be used to store its data into MySQL.

Mysqlclass.py: contains a class object which creates database and tables on MySQL server. PyMysql module is used to connect to MySQL server. It also inserts data retrieved into tables created.

Menu.py: contains a class object which is responsible for displaying instructions and interface of the application. Textwrap method was used to format some strings from the data to increase the readability.

pur_beurre.py: creates instances then passes user's input to instance methods. Input() function was used to store and pass user's answer to different instances necessaire.

Main : A text file which contains welcome interface of the application.

Development

The challenging part of this project was learning SQL because I did not have any experience with SQL, I had to start from scratch.

The key function of this application is to find a healthier substitute product. In order to achieve this, I had to search for the highest nutrition grade from each category. Since each entity in the table ‘product’ had an attribute ‘nutrition_grade_fr’ which contains the nutritions score of each product in alphabetic order from A to E (A being highest and E being the lowest nutrition score). At first, I used **MIN()** function from MySQL to sort out the minimum values from ‘nutrition_grade_fr’ column and then added **GROUP BY** clause to find the highest score from each category. But when I executed the query I ended up selecting only the first row of items from each group with the minimum nutrition grade values. In order to let MySQL to select all the rows which has the minimum nutrition grade value, I had to include *nutritioin_grade_fr=(SELECT MIN(nutrition_grade_fr FROM products)* in the **WHERE** clause instead of selecting *MIN(nutrition_grade_fr)* as a column. By placing another select query inside of a query, I was able to select all the minimum values of ‘nutrition_grade_fr’ column from each category of products.