

TP n°6 – Implémentation Jpa sur le projet open food facts

Objectifs

Vous connaissez peut-être l'application **Yuka**, disponible sur smartphone. **Yuka** fournit des informations nutritionnelles sur pratiquement tous les produits alimentaires commercialisés en France. En plus d'informations, elle fournit également un score nutritionnel, de A (excellent) à F (mauvais).

Cette application à succès s'est construite sur une **base de données open source** appelée **Open Food Facts**.

La base de données **Open Food Facts** est une base de données **mondiale**, qu'on peut télécharger sous la forme d'un fichier CSV. Le fichier que je vais vous demander de traiter dans le cadre de ce TP est le même que celui sur lequel s'est basé Yuka. Il ne concerne que les **les produits alimentaires fabriqués en France**.

Dans ce TP vous allez mettre en base de données un fichier open food facts contenant près de 13432 produits commercialisés en France.

Description

Développez une nouvelle application permettant d'intégrer en base de données le fichier **open food facts**.

Pour ce faire vous devrez identifier toutes les entités (classes) dont vous aurez besoin ainsi que les relations entre elles (cardinalités) afin de pouvoir créer vos tables.

Le fichier fourni comporte 13 432 références avec 30 informations associées par produit :

- catégorie
- marque
- nom
- score nutritionnel : A (excellent) à F (Faible)
- liste des ingrédients
- énergie pour 100g (en joules)
- quantité de graisse pour 100g
- liste des allergènes
- liste des additifs
- présence d'huile de palme
- toutes les vitamines,
- calcium, magnesium, etc..

Au total chaque référence de produit a 30 attributs.

Type de fichier

Le fichier fourni est un fichier CSV.

Un fichier CSV est un fichier dans lesquelles les informations sont séparées par des séparateurs. Ici le séparateur utilisé est le | :

- la première ligne fait la liste des attributs séparés par des caractères |
- chaque ligne suivante correspond à un article et les valeurs des attributs sont séparées par des |
 - o la liste des ingrédients est séparée par des virgules, parfois des ; ou encore des tirets.
 - o la liste des additifs et des allergènes également
 - o Attention certaines données, notamment chiffrées, ne sont pas toujours renseignées.

Instructions

- Créez un nouveau projet STS nommé **traitement-fichier-jpa-off**
- Ajoutez un « source folder » resources
- Convertissez ce projet en projet MAVEN
- Poussez-le sur GitHub
- Créez une nouvelle base de données appelée **open-food-facts**.
- Créez une classe exécutable **IntegrationOpenFoodFacts**.
- Vous devrez mettre en place des **DAO** afin de gérer vos diverses entités. Pas de requête en dehors des DAO !!
- Le traitement que vous allez écrire doit a minima :
 - o gérer la table des **catégories**
 - o gérer la table des **marques**
 - o gérer la table des **produits**
 - **Un produit a une marque et une catégorie**
 - o gérer la table des **ingrédients** rattachés aux produits.
 - Attention aux cardinalités, un produit a plusieurs ingrédients mais un ingrédient peut être rattaché à plusieurs produits. Par exemple il y a des chances que le sucre soit rattaché à de très nombreux produits.
 - o Gérer la table des **allergènes**
 - o Gérer la table des **additifs**.
- Précautions à prendre :
 - o il n'est pas question d'avoir des doublons dans les tables des **catégories** et des **marques**.
 - o Votre programme doit être rejouable, c'est-à-dire qu'il est capable d'intégrer un produit seulement s'il n'existe pas en base.
 - o La table des ingrédients, des allergènes et des additifs doivent être gérées de la même manière, c'est-à-dire **sans doublon**.
- Réfléchissez aux classes que vous allez mettre en place avant de vous lancer. Attention aux cardinalités entre classes et l'impact sur le modèle de données.

Livrables attendus

- Le diagramme de classes.
- Le modèle physique de données
- L'application
- Le fichier SQL contenant la structure des tables.

COMMITEZ