JUNIOR DATA ENGINEER

This test is divided into two parts:
- Exercise #1: A data analysis
- Exercise #2: A SQL Challenge

Exercise #1: Data analysis

Context, and problem to solve

ExpiryApp is a CodaBene application that helps teams in stores to manage expiry dates of products. It relies on the following principle: at the init of the application (its first day of use), the user creates an aisle (e.g. "Rayon des Yaourts"), sets an alerting parameter for this aisle (e.g. I want to be warned 3 days before expiry), and for each product present in the shelves he.she saves in the app all the shortest expiry dates. Then, the following days, the application asks the user to control the expiry dates of products which are nearing their expiry dates (following the alerting parameter). At each control, the new shortest date is set.

For example: If today is 27th of August, and if at the init of the store 1 month ago the user set that the expiry date of Danette was the 30th of August, then the application will ask him/her to control that product today (if the alerting parameter is 3))

The problem to solve is the following: when new products are physically added in the shelves, the teams in store do not always recall to init these products in the app ; and to set expiry dates for them. Therefore, *ExpiryApp* does not know them, and products can be missed during the daily control routine.

Task

You will be tasked with creating a python script to : import and analyse a provided set of data, extract meaningful information and display results in the terminal.

The analysis aims to determine if *ExpiryApp* successfully tracks all the relevant products inside a specific shop, and if not, suggest products to track.

To perform this task, you will be provided with the following data:

- A fake set of products currently tracked by the app in a specific shop (references initialized in shop.csv)
- A fake extraction of the shop internal system which contains all the products which can be present in the shop. (retailer extract.csv)

From the attached csv files:

Create a Python script which parses both files and prints out:

1. The total number of references not tracked in the app but present in the shop assortment
2. The list (EAN and Reference Name) of products which are not tracked by the app, but are relevant (should be tracked). We consider that a product is relevant if:
    1. It is not "dereferenced" ("data deref" in the retailer extract)

    2. The sub family ("Libellé Sous-Famille" in the retailer extract) where it belongs is tracked in at least one aisle in the shop
3. The total size of the list of relevant but not tracked products
4. Bonus: For each reference considered in the previous question, suggest an aisle where the reference could be found by the user.

    Expected result

Create a git repo and add your python script. Include in the readme, any instructions to run it and where we should put csv files. We should be able to run the script with the files that we provided. Therefore, the script should perform all the necessary transformations if needed.

Exercise #2: SQL Challenges

Using the same csv files as in the first exercise, import them in a PostgreSQL database and perform an SQL query for each question:

1. The total number of references not tracked in the app but present in the shop assortment (the result should match your result in the first exercise)
2. If we consider that today is the 2021-10-20, count in a single query :
    a. How many products are expired (expiry date is passed)
    b. How many products will expire in less than 5 days
    c. How many products are safe (will expire in more than 5 days)
3. Improve your query from 2/, to split the results per aisle. Also sort the results to display first the aisles where there are more at risk products.
4. Improve your query from 3/, to only show products which belong to the "Libellé Groupe de Famille" -> Charcuterie. Also, split the results by "Libellé Sous-Famille" as well as aisles.

    Expected result

Using the same repo as in the first exercise, add a .sql file for each of the 4 challenges, with the query to solve it.

Good luck, and many thanks for applying to this CodaBene position ! 💪