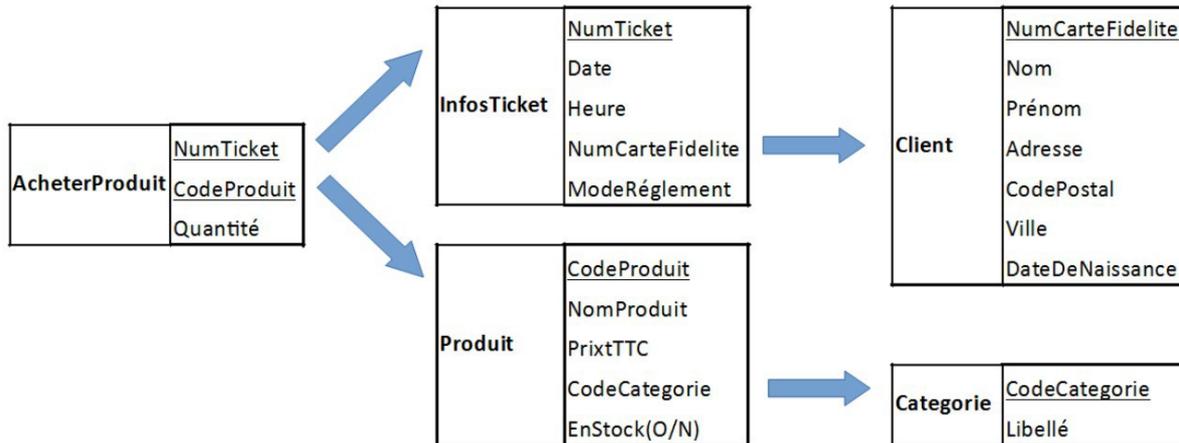


Filtre, projection et fusions

Nous allons, durant ce TP, travailler sur des tables extraites de la base de données d'un magasin (fictif). Le schéma suivant présente la structure des tables utilisées et les relations entre elles.



. RÉVISIONS SUR LES FILTRES ET LES PROJECTIONS

. EXERCICE 1

- Charger le contenu du fichier TableProduit.csv dans une variable que vous appellerez `produits`
Aide : Inspirez-vous de l'exercice 2 du TP1 sur le traitement de données en table.
- Filtrer les articles en rupture de stock (reconnaisables car la valeur associée du descripteur 'EnStock(O/N)' vaut 'N') et stocker le résultat dans une variable que vous appellerez `p_rupture`
Aide : Inspirez-vous de la question 3 de l'exercice 3 du TP1 sur le traitement de données en table.
- Afficher, ligne par ligne, les éléments contenus dans la liste `p_rupture`

. EXERCICE 2

- Charger le contenu du fichier TableInfosTicket.csv dans la variable `tickets`
Aide : Inspirez-vous de l'exercice 1 du TP1 sur le traitement de données en table.
- Filtrer les tickets enregistrés avec sans carte de fidélité (reconnaisables car la valeur associée du descripteur 'NumCarteFidelite' est une chaîne de caractère vide) et stocker le résultat dans une variable que vous appellerez `t_ss_carte`
Aide : Inspirez-vous de la question 3 de l'exercice 3 du TP1 sur le traitement de données en table.
- Afficher, ligne par ligne, les éléments contenus dans la liste `t_ss_carte`

. FUSION DE TABLES

Parfois, la recherche d'informations que l'on souhaite faire nécessite de croiser des tables ayant des descripteurs. On dit que l'on fait une **fusion**.

. EXERCICE 3

1. Charger le contenu du fichier TableInfosTicket.csv dans la variable `tickets`
2. Charger le contenu du fichier TableAcheterProduit.csv dans la variable `ac_prod`
3. Fusionner les deux tables précédentes pour récupérer les numéros de tickets (ainsi que la carte de fidélité utilisée) concernant des passages en caisse où l'article 39 a été acheté. Stocker la liste obtenue dans une variable que vous appellerez `tickets_prod39`
4. Afficher, ligne par ligne, les éléments contenus dans la liste obtenue.

. EXERCICE 4

Supposons que le magasin veuille envoyer des bons de réductions aux clients ayant acheté un article particulier pour l'inviter à le tester un autre article du même fabricant.

1. Utiliser la fonction `input` pour récupérer dans une variable `n_article` un numéro d'article choisi par l'utilisateur
2. Inspirez-vous de l'exercice précédent pour récupérer dans une variable `ticket_p_choisi` les numéros de tickets (ainsi que la carte de fidélité utilisée) concernant des passages en caisse où l'article `n_article` a été acheté.
3. Charger le contenu du fichier TableClient.csv dans la variable `clients`
4. Fusionner le contenu de la liste `ticket_p_choisi` et celui de la variable `clients` pour récupérer les informations sur les clients ayant acheté l'article `n_article`
5. Afficher, ligne par ligne, ces clients.

. A VOUS DE PRENDRE DES INITIATIVES

. EXERCICE 5

Supposons qu'une personne rapporte un article (dans son emballage) avec le ticket de caisse et que le magasin veuille vérifier le ticket de caisse avant de faire le remboursement.

Faire un programme qui demande le numéro de ticket recherché puis affiche le nom des produits achetés ainsi que leurs quantités.

. EXERCICE 6

Proposer un programme qui utilise au moins deux tables différentes.

Commenter le code pour expliquer le rôle de ce programme.