TD5 : Intégration de données

UCBL - Base de données pour la bioinformatique - 2023 / 2024

Objectif du TD : implémenter un script d'intégration de données

Les fêtes sont une bonne occasion de déguster du chocolat, souvent sous forme de ballotins. Mais l'inconvénient d'un ballotin, c'est que ses chocolats proviennent d'une seule chocolaterie ! Vous envisagez donc d'offrir un service révolutionnaire qui permettra de créer un ballotin "sur mesure", avec des chocolats de plusieurs fournisseurs. Quatre chocolateries fournissent la liste des lots de chocolat produits :

- La Blanche Toque (source S1, relationnel). Un seul lieu de vente. Prix exprimé en euros pour 100 grammes. Attribut *date* indiquant la date limite de consommation.
- Juge de Breff (source S2, XML). Plusieurs lieux de vente. Prix en euros au kilo. Pas de date limite de consommation, car les chocolats sont consommables pendant 10 mois après la date de mise en ligne.
- Bannot (source S3, JSON). Un seul lieu de vente. Prix en euros au kilo. Date de production et durée de conservation du produit en mois, qui permettent de calculer la date limite de consommation.
- Aba & Lizzy (source S4, CSV). Une chocolaterie artisanale qui fournit peu d'informations.

Concernant les contraintes : comme les sources de données sont mises à jour régulièrement, vous décidez d'intégrer les nouveaux chocolats et les modifications chaque jour. À terme, il est envisagé d'inclure de nouvelles sources de données (d'autres chocolateries).

id	desc	prix	date	lieu
1	PRALINE MIEL	5	01/12/2023	Blanche Toque, Lyon
2	PRALINE PIMENT	4.6	15/10/2023	Blanche Toque, Lyon
3	MARRON GLACÉ	8	01/12/2023	Blanche Toque, Lyon

Source S1, table *Produits* (La Blanche Toque)

```xml
<chocolats>
 <chocolat code="G14">
  <nom>Ganache praliné</nom>
  <prix>32</prix>
  <magasin>Lyon, France</magasin>
 </chocolat>
 <chocolat code="G07">
  <nom>Gianduja suprême</nom>
  <prix>46</prix>
  <magasin>Lyon, France</magasin>
 </chocolat>
 <chocolat code="T23">
  <nom>Truffe caramel</nom>
  <prix>47</prix>
  <magasin>Lille, France</magasin>
 </chocolat>
 <chocolat code="M03">
  <nom>Manon noir</nom>
  <prix>35</prix>
  <magasin>Lyon, France</magasin>
  <magasin>Lille, France</magasin>
 </chocolat>
</chocolats>
```

Source S2 (Juge de Breff)

```json
{
 "_id" : 1,
 "nom" : "Bannot",
 "villePays" : "Voiron, France",
 "produits" : [{
  "_id" : 2,
  "nomP" : "Grand cru Cuzco",
  "dureeCons" : 12,
  "dateProd" : "06-2023",
  "prixKilo" : 64
  }, {
  "_id" : 3,
  "nomP" : "Grand cru Chuao",
  "dureeCons" : 12,
  "dateProd" : "10-2023",
  "prixKilo" : 70
 }]
}
```

Source S3 (Bannot)

```csv
id_choco, nom_choco
1, noir poivre de Timut
2, lait miel châtaigne
```

Source S4 (Aba & Lizzy)

Exercice 1 Choix d'une architecture

1. D'après les contraintes, quelle architecture semble la plus adaptée au projet ? Justifiez.
2. Quel type de mappings (GAV ou LAV) semble le plus adapté au projet ? Justifiez.

Exercice 2 Définition du schéma global

1. Définir un schéma global (modèle relationnel) qui inclut le maximum d'informations contenues dans les sources et qui respecte si possible les trois premières formes normales.

Exercice 3 Écriture des mappings

1. Définir les mappings en pseudo-langage entre le schéma global et chacune des sources.

Normalement, des algorithmes spécifiques (e.g., Bucket, Minicon, Inverse Rules) sont utilisés pour répondre aux requêtes dans ce contexte. Dans ce TD, nous n'implémentons pas l'un de ces algorithmes (complexe), mais nous allons simplement dérouler une requête dans un contexte d'interrogation distribuée. On considère la requête suivante, posée sur le schéma global : *"identifiant et nom des chocolats à plus de 60 euros le kilo"*.

2. En utilisant les fragments de mappings utiles, identifier les sources qui peuvent répondre à cette requête.
3. Complétez le schéma ci-dessous, qui illustre l'exécution de cette requête (i.e., distribution vers les sources, propagation des résultats).
4. Proposez une traduction ou implémentation (e.g., pseudo-code, Python) au niveau des adaptateurs pour répondre à la requête.