

Contrôle terminal de BDBIO

UCBL - Département Informatique de Lyon 1 – janvier 2017

Documents papier autorisés. Anonymisez votre copie. Durée : 1h30.

L'une de vos bonnes résolutions de cette année, c'est le bénévolat pour des associations. Et ça tombe bien, l'association *Chass'o Choc*, en charge d'organiser des "chasses aux oeufs en chocolat" a besoin d'aide en informatique : plusieurs communes souhaitent, pour le bonheur des enfants, mutualiser la gestion de ces chasses aux oeufs.

Exercice 1 Modélisation relationnelle (6 points)

Actuellement, plusieurs communes font appel à l'association *Chass'o Choc*, qui gère notamment une base de données avec des informations sur les communes (*nbH* pour le nombre d'habitant.e.s), les enfants qui y habitent (*numE* est un identifiant numérique), et les types de chocolat. Cette base permet d'obtenir des statistiques annuelles sur le ramassage. Les communes souhaitent désormais stocker de nouvelles informations.

1. En présence d'une base de données existante, une étape de rétro-ingénierie est nécessaire : dessinez un diagramme E/A (formalisme Merise) dont la transformation permettrait d'obtenir le schéma relationnel de *Chass'o Choc* présenté ci-dessous (prévoir de la place pour étendre votre diagramme, voir question suivante).

COMMUNES (nomC, nbH)

ENFANTS (numE, nomE, age, #nomC, anneeArrivee)

TYPESCHOCO (nomT, poids, forme, description)

RAMASSER (#nomC, #numE, #nomT, anneeR, quantite)

2. Complétez votre diagramme E/A avec les nouvelles spécifications suivantes. En plus des statistiques sur le ramassage, les communes souhaitent stocker le nom, l'adresse et le téléphone des entreprises impliquées. Une entreprise peut fabriquer des lots de chocolats en une quantité donnée et à une date donnée. Une entreprise peut aussi se voir confier l'emballage des lots de chocolats. À noter que certaines communes ne veulent pas d'emballage. Chaque lot de chocolat est d'un seul type de chocolat. Enfin, les types de chocolats sont différenciés pour affiner les statistiques : d'une part les fourrés, pour lesquels on stocke le type de ganache, et d'autre part les types de chocolats creux.

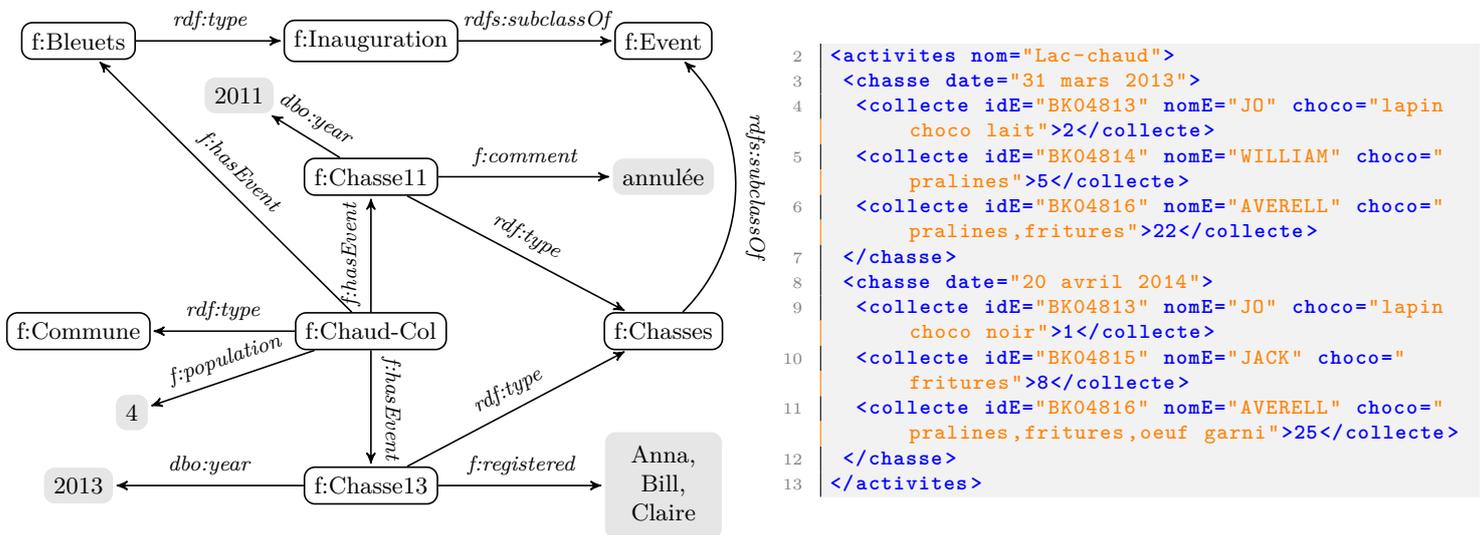
Exercice 2 Interrogation d'une BD relationnelle (4 points)

Les requêtes fréquemment utilisées par les communes sont données ci-dessous. Traduisez-les en SQL en utilisant uniquement la base de données existante (schéma relationnel de l'exercice 1).

1. Le nom des types de chocolat ramassés par des enfants de la commune *Lac au Chat*
2. Les enfants arrivés dans une commune en 2015 et qui n'ont jamais ramassés de chocolats. Le résultat sera trié par âge décroissant
3. Pour chaque commune, le ratio entre nombre d'enfants et nombre d'habitant.e.s
4. Pour chaque type de chocolat ramassé par des enfants d'au moins 10 communes, donner la moyenne des quantités ramassées

Exercice 3 Questions de cours (3 points)

L'engouement suscité par votre projet attire les communes de Chaud-Col et Lac-chaud, dont les sources de données sont fournies ci-après. Lors d'une réunion avec ces nouveaux intéressés, vous êtes interrogé.e sur certains aspects techniques. Répondez aux questions en cochant les affirmations qui vous semblent correctes.



Dans la source 2 (RDF, à gauche), la population est exprimée en milliers d'habitants. Dans la source 3 (XML, à droite), le texte des balises `<collecte>` indique le nombre total de chocolats ramassés.

- "La base de données de *Chass'o Choc* (schéma relationnel de l'exercice 1) est-elle bien modélisée ?"
 - Non, elle n'est pas en première forme normale
 - Non, elle n'est pas en deuxième forme normale
 - Non, elle n'est pas en troisième forme normale
 - Évidemment, c'est moi qui l'ai vérifiée !
- "À Chaud-Col, nos données sont en RDF. Quelles contraintes sont valides sur nos données ?"
 - `f:hasEvent rdfs:range f:Event`
 - `f:comment rdfs:isDefinedBy rdf:Property`
 - `f:registered rdfs:domain f:Chasses`
 - `f:population rdfs:range rdfs:Class`
- "Nos données, à Lac-chaud, sont stockées en XML. Quelles contraintes sont valides sur nos données ?"
 - Une contrainte `use="require"` dans la définition de l'élément `<collecte>` permet d'indiquer que l'attribut `idE` est obligatoire
 - L'élément `<activites>` est défini comme un type complexe utilisant `xsd:all`
 - Une contrainte `maxOccurs="unbounded"` dans la définition de l'élément `<activites>` permet de préciser que le nombre de chasses n'est pas limité
 - L'attribut `idE` est de type `"xsd:IDREF"`

Exercice 4 Intégration de données (7 points)

Pour intégrer les données des nouvelles communes, vous prévoyez d'utiliser une architecture centralisée (entrepôt). Le schéma global sera le schéma relationnel de l'exercice 1, car plusieurs communes l'utilisent déjà. Actuellement, la BD de *Chass'o Choc* stocke des enfants numérotés de 1 à 1000.

- Dessinez le schéma de cette architecture centralisée, en faisant apparaître les notions de schéma global, source 1 (relationnelle), source 2 (RDF), source 3 (XML), chargeur ETL, ainsi que les langages d'interrogation et de définition utilisés.
- Écrire les mappings entre la source 2 (RDF) et le schéma global sous forme d'un programme informatique (exécuté au niveau d'un chargeur ETL). Vous utiliserez les langages de requêtes appropriés pour extraire les données de la source et pour peupler la BD globale. Le reste du programme sera codé en pseudo-langage (syntaxe libre, mais suffisamment explicite pour que le programme soit implémentable). Pour le niveau de détail, utilisez des appels de fonctions pour simplifier le code (e.g., si vous devez trier un tableau, écrivez `tabSorted = sort(tab)` accompagné d'un commentaire mais n'écrivez pas un algorithme complet de tri de tableau !). Utilisez des commentaires, par exemple pour expliquer comment vous résolvez les conflits.
- Même question pour la source 3 (XML) et le schéma global.