TP1 : modélisation relationnelle

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

Objectif du TP : modéliser avec un outil et produire le code de création des tables

1 Diagramme E/A (niveau conceptuel)

Dans ce TP, nous allons d'abord utiliser un outil de modélisation basique, afin de modéliser les concepts principaux de bioinformatique. De nombreux outils de modélisation existent[1], mais soit ils ne permettent pas de modéliser au niveau conceptuel, soit leur version gratuite est limitée (JMerise) soit ils sont trop complexes (Navicat, Open Modelsphere, PowerAMC) ou mono-plateformes (Looping-mcd pour Windows). Deux solutions libres et multi-plateformes :

- Mocodo[2] (`http://www.mocodo.net/`) : développé en Python, version en ligne (pas d'installation) ou exécutable (plus d'options), réflexion "sur papier", projet actif. Pour l'utiliser, il faut écrire votre diagramme E/A sous forme textuelle, puis Mocodo produit un joli diagramme, le schéma relationnel, ainsi que le script SQL (pour différents SGBD) ;
- AnalyseSI[3] (`https://launchpad.net/analysesi/+download`) : développé en Java, interface graphique qui permet de travailler directement, projet moyennement maintenu. Pour l'utiliser, il faut télécharger le `.jar`, et soit double-cliquer dessus, soit taper dans un terminal `java -jar AnalyseSI.jar`.

Questions :

1. Avec l'outil de votre choix, modélisez par un diagramme entité/association (formalisme Merise) les besoins suivants. Les laboratoires de recherche observent des transcripts sous une condition d'observation. Un transcript est décrit par un identifiant et un nom, et peut être observé par différents laboratoires et/ou sous différentes conditions. Un laboratoire souhaite pouvoir reproduire une même observation pour vérifier un résultat obtenu. Un laboratoire est représenté par un identifiant, un nom et un acronyme optionnel. Différentes informations sont stockées pour un gène : un identifiant, un nom, un *tag locus*, des coordonnées, et des références du gène sur d'autres bases (Uniprot, ...). Un transcript appartient à un gène, et ce dernier est décrit par un identifiant et un nom. Une maladie, dont on spécifie un identifiant et un nom, peut concerner un ou plusieurs gènes. Un même gène peut évidemment être affecté par plusieurs maladies. Un transcript peut coder pour une protéine. Une protéine, en plus d'un identifiant, est décrite par un nom, un poids, une longueur, une référence vers la base Uniprot et un ensemble de séquences biologiques. La bioinformatique cherche à détecter les interactions entre deux protéines. Une interaction comporte un identifiant, un type et sa référence vers la taxonomie PSI-MI. Les méthodes de détection (des interactions) sont décrites par un identifiant et un label, et sont organisées sous forme hiérarchique (e.g., une méthode est générale, et d'autres en sont des versions spécialisées). Différentes équipes de recherche peuvent détecter une interaction avec une méthode déjà testée ou de avec de nouvelles méthodes. Des chercheuses (identifiant, nom, pays) co-signent des articles pour lesquels l'ordre de signature importe. Chaque article possède un identifiant et des données relatives à son titre, le journal de publication, une année de publication, et une référence PubMed. Enfin un article est une citation soit pour une ou plusieurs interactions, soit une ou plusieurs protéines ou soit une ou plusieurs maladies.

[1] `https://en.wikipedia.org/wiki/Comparison_of_data_modeling_tools`

[2] Sources sur `https://github.com/laowantong/mocodo`, installation avec `pip install mocodo`, voir `https://pypi.org/project/mocodo/`

[3] Site officiel sur `http://www.analysesi.com/`, et code source sur sur `https://github.com/AnalyseSI/AnalyseSI`

2. Dans la théorie, les laboratoires observent des transcripts. En pratique, on observe surtout des gènes. Que faudrait-il modifier dans votre diagramme pour prendre en compte cette contrainte ?

3. Dans la réalité, une protéine possède une séquence biologique primaire (souvent la plus longue). Des modifications appliquées à cette séquence primaire génèrent des séquences alternatives. Enfin, une maladie ne concerne pas simplement un gène, mais concerne plus précisément une protéine et une séquence biologique alternative. Comment feriez-vous pour modéliser ces nouvelles contraintes, et notamment quelle extension du diagramme E/A serait utile ?

4. Enfin, on souhaite distinguer les laboratoires publics, qui ont une dotation financière annuelle, des laboratoires privés qui eux ont une entité de rattachement (e.g., une entreprise). Un laboratoire ne peut pas être à la fois public et privé, et l'on ne connait pas forcément le type de laboratoire. Que faudrait-il modifier dans votre diagramme pour prendre en compte cette contrainte ?

2 Modèle relationnel et script SQL (niveaux logique et physique)

Un outil de modélisation permet souvent de générer automatiquement les modèles suivants (logique et physique). En plus de gagner du temps, cela permet aussi de n'appliquer des modifications qu'au niveau conceptuel, les niveaux suivants étant simplement générés à nouveau si besoin. Attention toutefois, les outils de modélisation ne sont pas parfaits et le schéma relationnel et le script SQL peuvent comporter des erreurs ou nécessiter des modifications.

Questions :

1. Transformez le diagramme de l'exercice précédent (sans les contraintes supplémentaires) en schéma relationnel et en script SQL. Si nécessaire, choisissez le SGBD de sortie (PostgreSQL) dans les paramètres de configuration.
2. Vérifiez les modèles obtenus, et notamment les choix de transformation par rapport aux cardinalités.
3. Exécutez le code SQL dans PosgreSQL (voir TP2 pour la connexion au SGBD) pour créer les tables.