II Phase conception

1. Schéma objet relationnel

Suite à la création du diagramme de classes UML, nous avons conçu le schéma objet relationnel. Chacune des classes du diagramme UML correspond à une table dans le schéma objet relationnel avec l'ensemble de ses attributs, comme le présente la figure suivante.

Schéma objet relationnel

Afin de modéliser l'appartenance d'un concept, d'un terme ou d'un synonyme à un utilisateur, nous avons ajoutés trois collections correspondant à chacun de ces objets. De même, un terme possède un tableau contenant l'ensemble des synonymes qui lui sont liés pour modéliser la relation « a pour synonyme ».

Pour lier un concept et le terme qui le représente, nous avons ajouté dans la table Concept une référence vers le terme associé.

Enfin, il nous a fallu représenter la notion d'arborescence entre les concepts. En effet, chaque concept peut posséder des sous-concepts, et peut être lui-même sous-concept d'un autre concept. Pour modéliser cette relation de parenté, chaque concept contient une liste de parents, dans lesquels il est inclus, et possède également une liste de fils correspondant à l'ensemble de ses sous-concepts.

2. ODL

Par la suite, dans l'optique d'implémenter le code de la base de données en SQL3, nous avons également écrit le code ODL correspondant.

La classe Utilisateur possède les quatre attributs du diagramme de classes. Le login, le mot de passe et le mail sont représentés par des chaines de caractères. L'attribut admin est un booléen qui vaut vrai lorsque l'utilisateur est administrateur. De même que pour le schéma objet relationnel, la classe possède trois collections de référence vers les objets créés par l'utilisateur. Il y a également les méthodes associées de création et suppression des objets Concept, Terme et Synonyme, ainsi que la méthode de suppression de compte.

Les classes Concept, Terme et Synonyme possèdent un attribut id de type entier, un nom de type chaine de caractères, et un terme et un concept ont également une description.

Pour chaque concept, il y a une relation vers le terme associé, l'utilisateur qui le possède, ainsi que des collections des références vers les concepts parents et fils.

La classe Terme contient également les relations vers le concept et l'utilisateur associés, ainsi que l'ensemble des synonymes du terme.

Enfin, chaque syonyme possède les relations vers l'utilisateur qui le possède, et le terme vedette qu'il désigne.

III Phase implémentation

Une fois la conception de la base de données terminée, nous avons divisé le travail entre création de la base de données et réalisation de l'interface Web, permettant l'interaction avec la base de données via les opérations d'affichage et de saisie.

1. Base de données

1.1. Création de la BD

A l'aide du schéma objet relationnel et du code ODL précédemment réalisés, nous avons écrit le code de création de la base de données en SQL3.

Pour chacune des quatre classes, nous avons donc créé un type associé, ainsi qu'une table de chacun de ces types. Nous avons donc un type Synonyme, contenant un entier représentant l'identifiant, et le nom du synonyme, de type varchar. Afin de pouvoir lister l'ensemble des synonymes d'un terme, nous avons créé le type GroupeSynonyme_t, qui représente une table de références vers les synonymes associés.

De ce fait, le type TermeVedette_t contient l'identifiant, le nom, la description ainsi que la liste de ses synonymes, de type GroupeSynonyme_t.

Concernant la création de concepts, il nous a fallu ajouter des références vers l'objet lui-même, pour pouvoir modéliser les relations de parenté entre les différents concepts. Cette représentation incluant une interdépendance, nous avons donc défini un TAD incomplet, en créant un type Concept_t sans propriétés. Cette « pré-déclaration » nous a permis de définir le type GroupeConcept_t comme étant une table de références vers des concepts, et donc d'ajouter à notre type Concept les attributs parents et fils de type GroupeConcept_t. En plus de ces collections, le type Concept contient l'identifiant, le nom, la description ainsi qu'une référence vers un objet de type TermeVedette_t.

Pour terminer, afin de modéliser les possessions d'objets par un utilisateur, nous avons créé le type GroupeTerme_t, qui de même que pour les concepts et

les synonymes, représente une collection de références vers des termes vedettes.

Ceci fait, nous avons pu créer le type Utilisateur, avec un login, un mot de passe, un mail, et le booléen admin, représenté ici par un « number » de taille 1. Nous y avons ajouté trois attributs correspondant aux tables de références vers les objets possédés par cet utilisateur.

Enfin, une fois les types créés, nous avons ajoutés les créations des tables de chacun de ces types, en ajoutant les différentes contraintes de clés primaire et de domaines. Les identifiants des tables Synonyme, TermeVedette et Concept sont donc clés primaires. Pour ces tables, des contraintes sont également ajoutées aux noms et descriptions afin d'empêcher l'affectation de valeur nulle, et donc l'absence de valeur.

Pour la table TermeVedette, nous indiquons la création d'une nested table pour l'ensemble des synonymes associés. La table Concept contient également la contrainte « NOT NULL » sur la référence vers la vedette, et les deux nested table pour les parents et fils.

Pour finir avec la création de la base de données, nous avons ajouté la table Utilisateur, utilisant le login comme clé primaire. En plus des contraintes « NOT NULL », nous avons ajouté une contrainte d'unicité sur le mail qui n'est pas clé primaire, et des contraintes de valeurs sur le mail et la valeur admin. Le mail doit respecter la forme indiquée à l'aide d'une expression régulière, quant à la valeur admin, elle doit être comprise entre 0 et 1 et vaut 0 par défaut. Ici encore les nested table correspondant aux collections sont déclarées.

1.2. Création des requêtes de manipulation

Suite à la création de la base de données, nous avons écrit les requêtes d'insertions, modifications et suppressions des valeurs, afin qu'elles puissent être utilisées dans le code du site web.