Rapport projet BD

Dans le cadre de l'unité d'enseignement de base de données avancées du premier semestre de Master 1, il nous a été demandé de réaliser un projet consistant à créer une application permettant la gestion de thésaurus. De ce fait, nous aborderons dans un premier temps la phase analyse de la mise en place de cette application, en présentant le cahier des charges ainsi que nos choix quant à la conception de la base de données. Nous aborderons ensuite la phase conception de l'application avec entre-autres la présentation du schéma objet relationnel réalisé. Nous présenterons ensuite la phase de conception de l'application avec la création de la base de données ainsi que la création du site web. Enfin, nous parlerons de l'organisation du projet avec notamment l'organisation effective du travail.

I.Phase d'analyse

I.1 Cahier des charges

I.1.1 Définition du projet

Il nous a été demandé dans le cadre de l'unité d'enseignement de base de données avancées de réaliser une application permettant la gestion de thésaurus. Notre projet consiste à mettre en place une application travaillant avec une base de données générique permettant l'enregistrement de thésaurus de sujets différents. Le but de l'application sera de permettre à différents utilisateurs de créer ses propres thésaurus.
Le thème que nous avons choisi pour la réalisation de notre projet sera le concept des Pokemons.

I.1.2 Besoins des utilisateurs

Comme on peut le voir dans le cas d'utilisation de la figure …,un utilisateur classique doit pouvoir se connecter et se déconnecter de l'application.
Il doit pouvoir créer ses propres concepts et lier un terme vedette à son concept.
Un utilisateur doit aussi avoir la possibilité d'ajouter un synonyme à un terme ainsi que de modifier les descriptions des termes et concepts lui appartenant.

En plus des fonctionnalités accessibles aux utilisateurs classiques, un administrateur doit pouvoir supprimer un concept, un terme vedette ou encore un synonyme. En effet, cela permet un meilleur contrôle des données.

Figure cas d'utilisation

I.1.3 Spécifications techniques

La base de données devra être entièrement réalisée en objet, en utilisant le langage SQL3 et devrai être implémentées en utilisant Oracle.

L'application de gestion de thésaurus sera créée sous la forme d'un site web, en utilisant les langages PHP, HTML, CSS et JavaScript.

I.1.4 Répartition du travail

Il est souvent difficile lors d'un travail de groupe de gérer chaque membre, à savoir, qui fait quoi et quand. Pour cela, nous avons nommé Morgane Vidal en tant que chef de projet. Elle s'occupera donc entre-autres de la répartition du travail ainsi que de la planification des réunions en s'appuyant sur l'application Producteev permettant de fixer les devoirs de chacun ainsi que les dates limites de rendu.

I.2 Base de données

figure schema objet

Comme on peut le voir dans la figure … ci-dessus, nous avons choisi de mettre en place quatre entités : « Utilisateur », « Concept », «TermeVedette» et « Synonyme ».

L'entité Utilisateur a pour but de permettre aux utilisateurs du site web de se connecter ainsi que de conserver un suivi des différents concepts, termes vedettes et synonymes créés. De ce fait, un utilisateur possède un concept, un terme vedette et un synonyme et inversement, ceux-ci sont possédés par un et un seul utilisateur.

L'entité «Concept» quant à elle correspond à un concept. A un « Concept » est lié son « TermeVedette ». Par exemple le concept « ConceptPokemons » a pour terme vedette « Pokemons ». De plus, un concept peut avoir un ou plusieurs parents ainsi que un ou plusieurs fils.

L'entité « TermeVedette », comme son nom le désigne, correspond au terme vedette d'un concept. Par concéquent, elle désigne un concept. De plus, un terme vedette peut avoir plusieurs synonymes. En effet, si l'on prend le terme vedette « Pokemon », il peut avoir comme synonyme « Monstre ». De ce fait, l'entité « TermeVedette » doit être reliée à l'entité « Synonyme ».

Enfin, l'entité « Synonyme » correspond aux différents synonymes d'un terme vedette.

II Phase conception

II.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.

II.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.

III.1 Base de données

III.1.2 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.

III.1.3 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.

III.2 Implémentation du site web

IV. Organisation du projet

IV.1 Choix technologies

//Pour Git j'ai repris ce qu'on avait dans notre ancien rapport, je trouvais ça bien.

Nous avons choisi Git comme gestionnaire de version. C'est un logiciel libre créé par Linus Torvald, simple et performant, qui nous a permis la mise en commun du travail et la gestion de versions tout au long de l'avancée de notre projet.

Notre choix s'est porté sur ce gestionnaire de version car, contrairement à d'autres tels que Subversion, Git est décentralisé. C'est à dire qu'il ne repose pas sur un seul et même dépôt en ligne sur lequel chaque nouvelle version d'un fichier envoyée remplace la précédente et affecte donc l'ensemble du groupe. Git met en place, en plus du dépôt de référence, commun à tous, un dépôt en ligne et un dépôt local pour chaque utilisateur. De plus, l'utilisation de dépôts publics permet d'éviter les conflits, notamment lorsque deux personnes travaillent sur un même fichier et souhaitent l'envoyer simultanément sur le dépôt en ligne puisque ces envois seront faits sur les dépôts personnels de chaque personne puis intégrés sur le dépôt de référence par la personne en charge.

IV.2 Organisations du travail

Nous avions prévu dès le départ de nous réunir chaque semaine afin de discuter ensemble de l'avancée du projet, des éventuelles difficultés rencontrées par les membres du groupes ainsi que pour répartir le travail à effectuer. De plus, étant donné le fait que notre groupe est composé de huit personnes et qu'il était difficile de trouver un créneau réunissant tous les membres du groupes, nous mettions à disposition un compte rendu de chaque réunion relatant les éventuelles décisions prises ainsi que la répartition du travail.

Au départ, tous les membres du groupe ont participé à la mise en place du schéma objet de la base de données.

Par la suite, notre chef de projet s'est chargée de la planifications des différentes réunions, de la répartition du travail ainsi que de la rédaction des comptes rendus de réunion.

Geoffrey Dumas, Quentin Philbert et Arlemi Turpault ont réalisé les différents diagrammes.

Clément Agret c'est chargé de la réalisation du schéma objet relationel.

Jimmy Lopez a réalisé toute l'interface graphique et la maquette du site web.

Manuel Chataigner, Mengyue Xue et Morgane Vidal ont créé la base de données avec Oracle.

Morgane Vidal, Geoffrey Dumas, Jimmy Lopez et Manuel Chataigner ont écrit les différentes requêtes SQL utilisées dans l'application.

Manuel Chataigner, Morgane Vidal, Quentin Philbert et Clément Arget se sont occupés de la rédaction du rapport.

Jimmy Lopez, Mengyue Xue, Arlémi Turpault et Geoffrey Dumas ont implémenté le site web.

IV.3 Difficultés rencontrées

Lors de la mise en place du schéma objet, nous avions, au départ, mal compris le sujet. De ce fait, nous n'avions pas réalisé un schéma objet correspondant aux attentes. Nous avons donc dû reprendre entièrement notre réflexion sur la conception du schéma objet de la base de données, ce qui nous a fait perdre du temps dans la mise en place de l'application.

De plus, lorsque nous avons mis en place la création de la base de données, nous avons fait face à un problème par rapport au fait qu'un concept est parent de plusieurs autres concepts mais aussi fils de concepts. En effet, étant donné le fait que nous avions dans ce cas là de l'interdépendance, il a fallu faire une pré-déclaration du type Concept_t ainsi que gérer correctement les références.

Conclusion A faire