**Représentation E-A de la base relationnelle**

[image]

[explication]

Notre base de données représente une sorte d’hôtel des ventes d’un monde fantastique. L’idée est donc que des utilisateurs du site peuvent mettre en ligne une offre de vente pour leurs objets. Nous nous inspirons d’un hôtel de vente dans un jeu de rôle par exemple *World of Warcraft*.

Entités

Une offre est faite de sorte qu’un utilisateur peut vendre le même objet un nombre arbitraire de fois, c’est pourquoi chaque offre a un attribut quantité. Nous avons donc dans offre l’attribut « disponible » qui indique si la quantité restante est supérieure à 0. Donc une offre qui n’a plus de stock ne sera pas effacée de la base, mais aura plutôt son attribut disponible mis à false. Chaque offre aura aussi le prix unitaire en attribut ainsi que son id qui est sa clé primaire. Finalement, l’offre a aussi un attribut qui indique la date à laquelle l’offre a été mise en ligne.

Un utilisateur a aussi son id qui est unique et aussi la clé primaire. Il a aussi son nom d’utilisateur.

Un village a un nom, une position géographique x et une position géographique y qui font toutes parties de la clé primaire.

Un objet a un id unique qui est la clé primaire et aussi un nom. Nous voyons objet comme une classe abstraite, il sera donc impossible de créer un objet tout court, il faudra prendre une des classes qui hérite objet, c’est-à-dire une des suivantes :

* Armure qui a pour attribut son type (casque, armure de jambe, etc.) et son matériau.
* Arme qui a pour attribut son matériau
* Animal qui a pour attribut son type (monture ou bétail) et son sexe.
* Potion qui a pour attribut son type (l’effet de la potion)
* Babiole qui a pour attribut sa description. Tout objet qui n’entre pas dans les autres catégories sera mis ici.

Relations

La relation habite fait le lien entre un utilisateur et le village où il habite.

La relation poste fait le lien entre un utilisateur et l’offre qu’il poste sur le site.

La relation affiche fait le lien entre l’offre et l’objet qui est en vente dans l’offre.

La relation vente fait le lien entre l’offre et un utilisateur qui a fait un achat.

**Le schéma relationnel de la base**

[image]

**Explications du code DDL**

Pour les clés primaires, nous avons décidé de mettre un ID unique pour chaque entité. Donc par exemple, chaque objet aura un « itemId » unique. Ceci fait en sorte que le choix de clé primaire est très simple.

Le seul cas de format standard à gérer dans notre base de données était la date de la vente. Nous avons décidé de [comment on écrit la date].

Nous avons décidé de dupliquer l’attribut « matériau » qui se trouve dans l’entité armure et arme puisque, bien que les deux aient un matériau, la liste des matériaux possibles pour les armes n’est pas la même que celle pour les armures. On voulait aussi garder « Item » comme une classe abstraite, il serait donc impossible de créer un Item qui ne sera pas référencé dans un des sous-types de Item (armure, arme, animal, potion ou babiole). Une particularité de babiole est que c’est le seul objet qui a une description. La raison est que tout objet qui n’entre pas dans les autres catégories et qu’un utilisateur souhaite quand même vendre entrera dans babiole. La description est donc inutile pour les autres types d’objets vu qu’on sait déjà ce qu’ils sont.

La façon que nos id sont codés, nous ne pourrions pas avoir plus que 5000 objets, mais pour les besoins du projet nous pensions que c’était suffisant.

**Requêtes en SQL et explication du résultat attendu**

Tout d’abord, voici nos requêtes que nous jugeons plus complexe et qui font intervenir au moins 4 relations.

1.



Cette requête donne les objets les plus vendus, avec le nom des vendeurs, leur village et les coordonnées des villages.

2.



Cette requête donne, pour chaque type d’armure, l’objet le moins cher.

3.

select title,item\_name,material,unitprice,village from (select \* from armors, items,offers,clients where armors.id=item\_id and item\_id=offers.itemid and offers.clientid=clients.id) as r1

Voici un exemple des requêtes qui sont utilisées dans la barre de navigation principale. Par exemple, celle-ci retourne le titre, le nom, le materiel, le prix et le village de ventes de toutes les armures. Nous utilisons ce gabarit pour les armes, potions, animaux.

Autres requêtes

1.****

Cette requête sort le nom, le type, le matériau, le prix unitaire et la quantité de toutes les armures fait de «Plate» qui sont en vente à Stormwind.

2.



Cette requête sort tous les objets en vente à Edoras.

3.



Cette requête donne toutes les potions de soins en vente à Theramore en sortant leur nom, leur effet, leur prix et la quantité en vente.

4.



Cette requête donne toutes les armures de corps en vente avec la ville où elles sont vendues.

5.



Cette requête retourne la distance entre 2 villages.

6.



Cette requête retourne l’arme de Mace la moins chère ainsi que le client de vente.

**Captures d’écran de l’application**