

420-KHG-LG, intégration des BD en II

Travail pratique numéro2

Pondération 20%

Objectifs

Ce devoir vise principalement à vous faire expérimenter les aspects suivants:

- Définir le modèle de données adéquat.
- Réaliser une application C#, ADO.NET.
- Travailler en équipe

Ce travail peut-être fait en équipe de 2.

Date de remise et correction est ; le 25 avril 2016

Mise en contexte :

Nous sommes dans un environnement du jeu vidéo, et nous souhaitons développer une application pour la gestion des items du jeu, et des joueurs.

Dans l'environnement du jeu, il y a plusieurs joueurs. Chaque joueur peut posséder plusieurs items. (Ou zéro) Des joueurs peuvent avoir des items identiques en quantité supérieur ou égale à 1.

Les joueurs peuvent acquérir des items pendant le jeu.

Pour des raisons de simplicité un joueur est confondu avec son personnage.

1. Les items sont catégorisés par genre. Il existe plusieurs genres.
2. Un genre peut-être par exemple : Arme, Armure, potion, etc...
3. Chaque arme est caractérisée par un nom, son efficacité, son poids son prix et une photo.

4. Une armure a un nom, matière qui la compose, poids, taille, efficacité, durabilité et un prix.
5. La potion a un nom, effet attendu, la durée pour l'effet et un prix
6. Un joueur a un alias unique, un nom, un prénom, un montant initial en écus, un nombre de points initial qu'il augmente lors du jeu et un avatar.
7. Chaque item a un prix, un poids, le nombre de points requis pour le joueur pour l'utiliser, une quantité en stock.

Indications concernant votre application C# ADO.NET

Minimalement, l'application doit permettre :

- l'ajout d'un item (nouvelle arme, nouvelle armure, etc), la modification du prix unitaire, et de la quantité en stock d'un item ainsi que la suppression d'un item.
- Recherche d'un item par type d'item (Rechercher la liste des armes ou la liste des armures etc ...). Ici, on pourrait afficher une arme à la fois et utiliser les boutons suivant et précédent pour voir le détail d'un arme. L'affichage de la photo est souhaité
- La rechercher d'un item par son nom ou le début du nom.
- La consultation de l'historique des joueurs (les joueurs avec la liste des items qu'ils possèdent). On doit afficher un joueur à la fois. On doit pouvoir se déplacer avec les méthodes nécessaires entre les joueurs.
- Ajouter supprimer un joueur.
- Modifier le capital d'un jouer.
- Effectuer un achat d'items.

Contraintes sur la base de données :

1. Le modèle de données (Entités/Relations) doit-être validé avant la création des tables.
2. Pour toutes les requêtes de MAJ, vous devez utiliser OracleParameter.
3. Pour les listes, utiliser un OracleDataReader.

Critères de correction	Pondération
Normalisation et modèle relationnel	15%
Démonstration du fonctionnement	60%
Efficacité de programmation et validation	20%
Évaluation secrète de l'équipière ou l'équipier (s'il y a lieu)	5%