

420-KEH-LG, TP no1

1. Pondération : 10%.
2. Date de de remise : le 28 février avant **minuit**.
3. Modalité de remise : **boite de remise**.
4. Ce travail doit se faire en **individuel** ; cependant vous pouvez vous entraider.
5. Vous pouvez poser des questions cependant je ne ferai pas le travail à votre place.

Objectifs

Ce travail vise l'atteinte des objectifs suivants :

1. Produire un modèle conceptuel de données
2. Produire un modèle relationnel de données.
3. Écrire des packages et body package
4. Écrire des triggers.

Mise en contexte :

Nous sommes dans un environnement du jeu vidéo de type médiéval, et nous souhaitons développer une application qui permet la vente en ligne d'un certains nombres d'items concernant le jeu.

Dans l'environnement du jeu, il y a plusieurs joueurs, chaque joueur peut acheter plusieurs items, des joueurs différents peuvent acheter des items identiques. La quantité d'un item acheté peut-être supérieure à 1. Pour acheter des items nous devons utiliser un panier d'achats.

Voici comment sont décrits les items du jeu.

1. Les items sont catégorisés par type. Il existe plusieurs types.
2. Les items ont un numéro, un nom, et un type, et une quantité disponible

3. Les types sont pour l'instant : des Armes, des armures et des potions.
4. Les armes ont une efficacité (nombre de points de dommage) un genre (une main ou deux mains ou autre.)
5. Les armures ont une matière qui la compose, poids, taille.
6. La potion a un nom, effet attendu, la durée pour l'effet.
7. Tous les joueurs peuvent acheter n'importe quel item. Un joueur a un alias unique, un nom, un prénom et un montant initial en écus
8. Chaque item a un prix unique fixé au départ

Questions :

1. Donner le dictionnaire de données du jeu.
2. Donner le modèle conceptuel et le modèle relationnel du jeu.
3. Écrire **les packages et les bodys packages** pour la gestion du jeu.
 - a. Un package **Jeu** qui va contenir :
 - i. Une procédure qui permet de faire une insertion dans la table Items, puis dans une des tables Armes Ou Armure selon le type de l'Item.
 - ii. Une procédure qui permet de modifier la quantité disponible d'un item.
 - iii. Une procédure suppression, qui permet de supprimer un item.
 - iv. Une fonction qui retourne la liste des items selon le type de l'item
 - b. Le Body Package **Jeu** qui va contenir les détails de chacune des procédures et fonctions.
 - c. Un package **Joueurs** qui va contenir
 - i. Une procédure insertion du joueur du joueur.
 - ii. Une procédure Achats. Cette procédure ou fonction permet à un joueur de faire un achat d'un ou plusieurs items. (Ajouter des items au panier)

- iii. Une procédure qui modifie le panier d'un joueur
 - iv. Une procédure qui supprime un item du panier du joueur.
 - v. Une fonction qui liste le contenu du panier d'un joueur. Afficher également le nom de l'item et son type.
- d. Le Body Package **Joueurs** qui va contenir les détails de chacune des procédures et fonctions.
4. Écrire les triggers suivants :
- a. Le trigger qui fait la suppression en cascade d'un item.
 - b. Le trigger qui permet de garantir qu'un numéro inséré dans la table Armures ne sera pas utilisé (inséré) dans la table Arme. Un autre trigger qui fait la même chose pour la table Armes.
 - c. Un trigger qui contrôle les achats des joueurs. Lorsqu'un joueur achète un item si le montant exigé par l'achat est plus élevé que le montant disponible chez le joueur alors l'achat ne pourra pas se faire.
 - d. Un trigger qui contrôle les achats des joueurs. Lorsqu'un joueur achète un item avec une quantité N, si cette quantité dépasse la quantité disponible alors l'achat ne sera pas effectué.

Ce qu'il faut remettre dans un dossier Zippé portant votre nom :

1. Les modèles de la base de données. La BD doit être normalisée. Ces modèles seront remis au format papier.
2. Le dictionnaire de données.
3. Un script SQL contenant la création des tables
4. Un script SQL contenant le package Jeu et le body Package jeu
5. Un script SQL contenant le package Joueurs et le body Package joueurs
6. Un script SQL contenant TOUS les triggers
7. Un script SQL contenant les requêtes d'exécution des procédures, des fonctions et de la vérification du déclenchement des triggers.