420-KEH-LG, Systèmes pour la gestion de bases de données

Travail d'intégration, H2017

Pondération 15%

Objectifs

- Concevoir un modèle de bases de données normalisé, et validé.
- Écrire des procédures/des fonctions et triggers
- Concevoir et réaliser une application JDBC

Mise en contexte : L'or du dragon :

Dans le cadre du projet l'OR du dragon, vous devez concevoir une base de données pour la gestion, le déroulement et l'administration du jeu.

Dans ce contexte, le rôle de la base de données sera décrit comme suit :

1. Contenir les questions et les réponses pour le jeu.
2. Contenir les joueurs avec leurs capitaux et leurs avoirs →immeubles (auberges, manoirs et châteaux)
3. Contenir les objets que les joueurs auraient ramassés durant le jeu. (Mountain Dew, des Dorritos etc ..)

Dans le même contexte, les règles qui régissent la base de données sont :

1. Chaque équipe a une banque de questions à sa disposition. Les questions sont à choix de réponses et ont un degré de difficulté différent **(facile, moyen difficile**)

Préparé par Saliha Yacoub. Idée originale pour le jeu : François Boileau.

2. Les questions sont catégorisées comme suit Histoire, ArtetCulture, et Sciences (d'autres catégories peuvent être rajoutées plus tard). Vous pouvez choisir vos propres catégories si vous voulez. **Chaque catégorie a environ une vingtaine de questions**

3. Les questions sont à choix de réponses. **Chaque question a 4 choix de réponses dont une bonne réponse.**

4. La question et le droit de passage sur un nœud sont en fonction du type d'immeuble sur le nœud. Selon le type d'immeuble :
    a. La question est difficile (**peu importe la catégorie**)
    b. Les droits de passage sont élevés.

5. Une question pigée est soit retirée soit mise à jour par un flag (une question ne peut pas être pigée deux fois durant le jeu.)

6. Le capital du joueur augmente au fur et à mesure qu'il ramasse des pièces d'or

7. Le capital diminue dans les cas suivants :

    a. Il a acheté un immeuble

    b. Il payé les droits de passage.

    c. Sortir de prison

8. Chaque type d'immeuble a un prix fixé au départ.

Ce que la base de données doit permettre :

Dans le cadre du projet l'OR du dragon, votre une application JDBC pour la gestion du jeu doit répondre minimalement aux point suivants:

1. Chercher une question de manière aléatoire en fonction du type de l'immeuble

2. Valider la bonne réponse.

3. Une question posée, ne doit pas être posée plus qu'une fois à un même joueur. (soit la supprimer ou la mettre à jour par un flag). Idéalement la mise à jour du flag est souhaitée.

4. Mettre à jour le capital de son équipe.

5. Mettre à jour les avoirs de son équipe.

6. Mettre à jour la liste d'objets de son équipe.
7. Alimenter la base de données en questions.
8. Afficher les avoirs des joueurs. Exemple l'équipe Primogen a une auberge et un manoir.
9. Afficher les objets des joueurs avec les quantités. Exemple l'équipe Primogen a 3 Dorritos et 2 Mountain Dew.

Contraintes de conception :

1. Toutes les requêtes doivent être écrites en procédures stockées, sauf indications contraire, dans ce cas utilisez de requêtes paramétrées.
2. La base de données est une base de données Oracle.
3. Pour le jeu, les accès à la BD se font par JDBC.

Les équipes

La définition des équipes est laissée à votre discrétion. Le nombre d'étudiants dans une équipe est de 3. Vous ne pouvez pas être tout seul pour ce travail.

Implémentation

L'implémentation débutera : le 07 mars 2017.

La base de données doit-être peuplée en questions et réponses durant votre semaine de mise à niveau. Ce qui veut dire :

1. Insérer trois catégories
2. Insérer 20 questions par catégories. Il doit y avoir autant de questions Facile, moyen et difficiles(ou presque). Vous ne pouvez pas avoir que des questions faciles par exemple. C'est important pour le déroulement du jeu
3. Insérer les choix de réponses pour chaque question

De plus, (à compter du 07 mars)

- Vous devez créer les autres tables nécessaires au jeu.

- Vous devez avoir une interface graphique qui permet :
    - Voir la question et ses réponses.
    - Valider la bonne réponse.
    - Afficher le capital de l'équipe.
    - Afficher les avoirs et les objets de l'équipe
    - Ajouter une question et ses réponses dans la base de données

Correction

La correction de toute l'application se fera : jeudi le 23 mars 2017 durant les cours –420-KEH, 420-KEK. Aucun retard ne sera accepté.

Barème :

Éléments évalués	Pointage
Le modèle de la base de données normalisé, **validé et documenté :**	20%
Fonctionnement du jeu (jdbc : chercher les questions, valider réponses, mise à jour de la bd etc …)	60
Efficacité de programmation	10
Validation	10

Autres informations :

Pour comprendre le déroulement du jeu allez sur le site de François Boileau

D'autres détails seront fournis au besoin.

En tout temps, vous devez sauvegarder le script sql

- De la création des tables
- Des insertions des questions et réponses (et catégorie)
- Les packages et les body package
- Les triggers s'il y en a.

Annexe : Structure de la BD pour les questions (à titre d'indication)

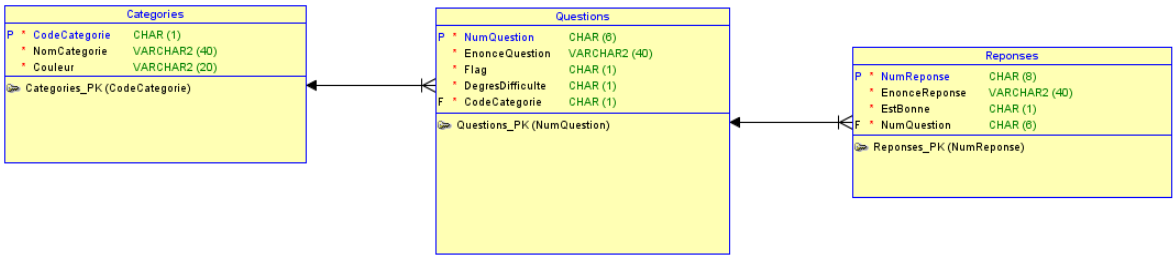

Table Catégories	CodeCategorie doit être un code couleur V pour vert, B pour bleu etc....
Table Questions	NumQuestion doit comprendre le code catégorie lors de la saisie des données Exemple V01 correspond à la question 01 de la catégorie V (vert) B01 : correspond à la question 01 de la catégorie B (bleu) Flag : un caractère qui indique que la question est déjà posée ou non DegresDifficulte : caractère qui indique si la question est facile (F), moyen(M) ou difficile(D) : pensez à une contrainte CHECK
Table Reponses	NumReponse : doit comprendre leNumQuestion lors de la saisie des données exemple : V01A : réponse A de la question V01 V01B : réponse B de la question V01 V01C : réponse C de la question V01 V01D : réponse D de la question V01 EstBonne : prend la valeur Y ou N (Yes ou No) et indique que la réponse pour la question est bonnes. Un caractère.