TRAVAIL PRATIQUE 2

Base de données 1
420-2B4-VI
Hiver 2022

Modalité : en équipe de 2
Pondération : 15 %
Date de remise : 6 mai 2022

Mise en contexte

Vous êtes chargé d'implémenter une base de données pour « Donjon Inc. », une entreprise se spécialisant dans la gestion de ressources monstrueuses. Elle fournit une plateforme qui permet aux gestionnaires de donjon d'embaucher des monstres afin de garder les salles et trésors de leur donjon. Le système permet aussi d'enregistrer des groupes d'aventuriers et les informations sur le déroulement de leur quête.

Consignes

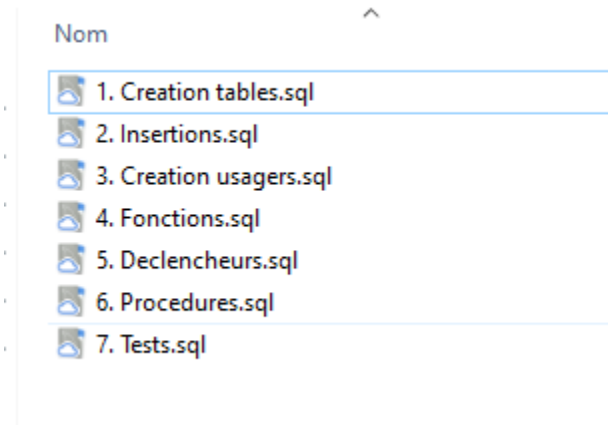
1. Créer la base de données et ajouter les tables. Les colonnes de type entier avec la contrainte de clé primaire dont le nom vérifie la REGEX /^id_[a-z_]+$/ doivent être auto-incrémentées. Le modèle de base de données est joint à l'énoncé. Les données à inclure dans les énumérations
2. Effectuez les requêtes de sélection présentées aux sections B, C, D, E
3. Préparer un script de test selon les consignes de la section F.

Vous trouverez la grille de correction à la section G.

Livrable

Vous devez remettre 7 scripts SQL :

- Un script de création des tables (mis à jour)
- Un script d'insertion des données (modification du script fourni)
- Un script de création des usagers
- Un script de création des fonctions
- Un script de création des déclencheurs
- Un script de création des procédures
- Un script de test



Le dépôt des scripts SQL se fera sur la plateforme GitHub Classroom. Créez un dossier pour le travail 2 (pour ne pas mélanger vos scripts avec le travail 1)

Groupe 02 (mardi 8 h 15 – vendredi 13 h 15) : https://classroom.github.com/a/Wljrw9fQ

Groupe 01 (mardi 10 h 15 – jeudi 13 h 15) : https://classroom.github.com/a/wwENU9JL

SECTION A – CONTRAINTES D'INTÉGRITÉ ET SÉCURITÉ DES DONNÉES

Ajoutez les contraintes suivantes

Humanoïde : Intelligence doit être supérieure ou égale à 0

Mort vivant : vulnerable_soleil vaut 0 ou 1

Mort vivant : infectieux vaut 0 ou 1

Famille_monstre : point_vie_maximal doit être strictement supérieur à 0

Famille_monstre : degat_base doit être strictement supérieur à 0

Responsabilite : niveau_responsabilite doit être supérieur ou égal à 0

Salle : la fonction doit avoir un minimum de 5 caractères

Aventurier : le niveau doit être strictement supérieur à 0

Ligne_coffre : la quantité doit être strictement supérieure à 0

Inventaire_expedition : la quantité doit être strictement supérieure à 0

Objet : la valeur doit être strictement supérieure à 0

Objet : la masse doit être strictement supérieure à 0

SECTION B – BESOINS EN USAGÉS

Créez les rôles suivants dans la BD (CRUD = CREATE, READ, UPDATE, DELETE)

Rôles	Privilèges
administrateur_système	Tout sur toutes les tables
responsable_visites	CRUD pour Visite_salle, Expedition, Expedition_aventurier, Inventaire_expedition et Aventurier
responsable_entretient	CRUD pour Coffre_tresor, Ligne_coffre et Objet
service_ressources_monstrueuses	CRUD pour Humanoide, Mort_vivant, Elementaire, Famille_monstre, Monstre, Responsabilite et Affectation_salle

Créer les utilisateurs suivants dans la BD en leur assignant le bon mot de passe et rôle.

Utilisateurs	Mots de passe	Rôles
daenerys	dragons3	administrateur_systeme
jon	Jenesaisrien	responsable_visites, responsable_entretien
baelish	lord	service_ressources_monstrueuses

SECTION C – CRÉER DES FONCTIONS

Cryptage et décryptage

Créer une fonction qui crypte et décrypte en utilisant le mot de passe. La fonction sera utilisée de la façon suivante :

INSERT INTO Table VALUES (crypter(donnee), donnee);

SELECT decrypter(donnee), donnee FROM Table;

Tous les numéros d'assurance maladie doivent être cryptés. Modifiez le script d'insertion en conséquence. La clé utilisée est « mortauxheros ».

1. FONCTION CRYPTER (XX POINTS)

Cette fonction crypte un texte clair en utilisant la clé de cryptage définie ci-dessus. Elle retourne le texte crypté.

Paramètre : chaîne de caractère à crypter

Valeur de retour : un blob contenant les données cryptées

2. FONCTION DÉCRYPTER (XX POINTS)

Cette fonction décrypte un contenu crypté en utilisant la clé de cryptage définie ci-dessus. Elle retourne le texte clair prêt à être lu par un être humain.

Paramètre : un blob contenant des données cryptées

Valeur de retour : une chaîne de caractère affichant le texte clair

3. FONCTION RESPONSABLE (XX POINTS)

Cette fonction accepte une fonction d'une salle et un objet de type DATETIME et retourne l'identifiant du monstre qui en est responsable (plus haut niveau de responsabilité).

Si au moment de la vérification, la salle n'a pas de responsable alors un avertissement est levé. L'erreur n'est pas gérée par la fonction.

Paramètres :

- Une chaîne de caractère indiquant la fonction de la salle.
- Un DATETIME indiquant à quel moment on souhaite connaître le responsable.

Valeur de retour : l'identifiant du monstre responsable.

4. FONCTION LEADER (XX POINTS)

La fonction retourne l'aventurier qui possède le plus haut niveau dans l'expédition.

Paramètres :

- Une chaîne de caractère indiquant le nom de l'expédition

Valeur de retour : l'identifiant de l'aventurier de plus haut niveau

5. FONCTION DE VÉRIFICATION DE LA VITALITÉ DES MONSTRES (XX POINTS)

La fonction vérifie si un monstre est en vie dans une salle donnée. Un monstre vivant possède un nombre de points de vie strictement plus grand que 0. Si la vérification s'effectue sur une salle qui n'est pas dans la BD ou qu'aucun monstre n'est affecté à la date demandée, une erreur est lancée.

Paramètres :

- L'identifiant de la salle
- La date à laquelle faire la vérification

Valeur de retour : la valeur 0 si tous les monstres sont morts, la valeur 1 si un monstre est encore vivant.

6. FONCTION DE VÉRIFICATION DE LA VITALITÉ DES AVENTURIERS (XX POINTS)

La fonction vérifie si un aventurier est en vie dans une expédition donnée. Un aventurier vivant possède un nombre de points de vie strictement plus grand que 0. Si la vérification s'effectue sur une expédition qui n'est pas dans la BD, une erreur est lancée.

Paramètres :

- L'identifiant de l'expédition pour laquelle faire la vérification

Valeur de retour : la valeur 0 si tous les aventuriers sont morts, la valeur 1 si un aventurier est encore vivant.

SECTION D – CRÉER DES DÉCLENCHEURS

Dans certains déclencheurs vous devrez déclarer des fonctions. Créez ces fonctions directement dans le script des déclencheurs.

1. VALIDATION DE LA CAPACITÉ DES COFFRES (XX POINTS)

Si un coffre contient plus de 15 objets ou que leur masse excède 300 kg, alors une exception doit être lancée. L'exception doit indiquer laquelle des deux règles n'est pas respectée.

Il vous est fortement suggéré d'ajouter une fonction pour vous aider à gérer ce déclencheur.

2. ÉLÉMENTS OPPOSÉS (XX POINT)

On ne peut pas affecter à une même salle un élémentaire du feu et un élémentaire de l'eau. Si cela se produit, un avertissement doit être lancé.

Une fonction vous est fournie pour vous aider avec ce déclencheur.

3. DÉCLENCHEUR DE MORTALITÉ (XX POINTS)

Lorsqu'un monstre meurt, la fin de son affectation est devancée à l'instant actuel.

4. DÉCLENCHEUR DE HACHAGE (XX POINTS)

Lorsqu'on insère une nouvelle valeur dans la table Coffre_tresor, le code secret (entré en clair dans la requête) doit être haché sur une chaîne de 256 bits.

SECTION E – CRÉER DES PROCÉDURES

PROCÉDURE INTIMIDATION (XX POINTS)

PROCÉDURE COMBAT (XX POINTS)

PROCÉDURE PILLAGE DE SALLE (XX POINTS)

PROCÉDURE VISITE_SALLE (XX POINTS)

PROCÉDURE EMBAUCHE (XX POINTS)

PROCÉDURE DE LA MALÉDICTION DE DÉBILITATION (XX POINTS)

SECTION F – TESTER LE SYSTÈME

Écrivez une histoire qui se déroule dans l'univers du donjon. N'hésitez pas à ajouter des données pour rendre votre histoire plus dynamique.

Votre histoire doit au minimum :

- Appeler 2 fonctions
- Appeler 3 déclencheurs
- Appeler 4 procédures stockées

Il est possible d'appeler plus d'éléments, éviter seulement d'allonger inutilement les scripts.

Les auteurs des meilleures histoires recevront la reconnaissance éternelle de leurs pairs et un montage photo (de « très haute qualité ») illustrant un moment marquant de leur histoire.

SECTION G – GRILLE DE CORRECTION

Chaque code SQL doit comporter un en-tête du format suivant :

```
/*
* Description brève du script
*
* Fichier : tp_1.sql
* Auteur : Nom #1
* Langage : SQL
* Date : Février 2022
*/
```

Critères d'évaluation

1. Détermination judicieuse des types de requêtes à formuler
2. Utilisation appropriée des clauses, des opérateurs, des commandes ou des paramètres.
3. Fonctionnement correct des requêtes
4. Gestion correcte des autorisations
5. Cryptage approprié des données
6. Utilisation appropriée des contraintes d'intégrité référentielle, des déclencheurs ou des transactions
7. Création appropriée de procédures stockées ou de scripts
8. Notation claire de la documentation d'aide à la programmation

Élément d'évaluation	**Pondération**
Création des données	
Création des utilisateurs	
Contraintes d'intégrité	
Cryptage et hachage	
Fonctions	
Déclencheurs	
Procédures	
Tests	