

Filière : TDI

Niveau : TS

Durée : 2h00

Intitulé du module : Système de gestion de BD II

Barème: 40 / 40

I) : Partie Théorique (1/40 points)

- 1) Citer les trois types de triggers dans l'SQL server. (1pt)

II) : Partie Pratique (39/40 points)

Soit le schéma relationnel suivant :

Service (Num_serv, Nom_serv, Date_creation)

Employe (Matricule, Nom, Prenom, DateNaissance, Adresse, Salaire, Fonction, Num_serv#)

Projet (Num_prj, Nom_prj, Lieu, nbr_limite_taches, Num_serv#)

Tache (Num_tach, Nom_tache, date_debut, date_fin, cout, Num_prj#)

Travaille (Matricule#, Num_tache#, Nombre_heure)

Questions :

A. Ecrire les fonctions suivantes (3pts):

1. Fonction nommée fn_personnel_rendement

6 pts

Permettant de calculer et renvoyer le taux de rendement d'un employé donné pour un projet donné en paramètre.

Rendement = somme (nombre des heures de travailles) x 10 / somme (coût des taches)

B. Gérer les exceptions (3pts):

1. Ajouter les messages suivants au catalogue système.

Avertissement N° 60000 :

- Langue française : opération de « type opération » bien effectuée par l'utilisateur « utilisateur courant de BD » à la date de « date opération ».
- Langue anglaise: the « type of operation » operation has been performed by the user « current user of the database » on the date of « date of operation ».

Le type de l'opération est soit : ajout, modification ou suppression

C. Ecrire les procédures stockées suivantes (8pts):

1. Procédure nommée ps_Projet_supprimer

Permettant de supprimer en cascade un projet dont le numéro est passé en paramètre (supprimer toutes les lignes correspondantes de la table travaille, tache puis projet.

- opération réussie : exception journalisée N°60000 (valeur de retour = 0).
- opération non terminée : exception journalisée avec le texte d'erreur, la date et l'utilisateur courant.

Penser à encapsuler les requêtes dans une transaction.

2. Procédure nommée **ps_Tache_ajouter**

8 pts

Accepte en paramètre le numéro de projet, le nom d'une tache, sa durée et le cout (par défaut = null), puis ajouter une ligne à la table tache.

La procédure doit effectuer le traitement suivant :

- Renvoyer -1 si le numéro de projet n'existe pas.
- Si le numéro du projet existe, ajouter une tache tel que :
 - ✓ numéro tache = max(num_tach)+1
 - ✓ date_debut =
 - ↳ s'il existe déjà une tache pour ce projet alors la date de début est :
= max (date_fin) pour le projet passé en paramètre + 1 jour
 - ↳ si non (c'est la 1^{ère} tache pour ce projet) la date de début est :
= la date d'aujourd'hui
 - ✓ Date_fin = date_debut + Durée (j)
- Si l'ajout est effectué avec succès, lever l'exception N°60000. (la procédure renvoie 0 avec le numéro de tache ajouté).
- Si la base de données renvoie une erreur, la procédure renvoie -2.

3. Procédure nommée **ps_Personnel_augmenter**

8 pts

Qui permet d'augmenter le salaire des trois employés les plus rentables qui ont participé à la réalisation d'un projet passé en paramètre selon la règle suivante :

- l'employé au 1^{er} rang : augmentation de 2%
- l'employé au 2^{ème} rang : augmentation de 1%
- l'employé au 3^{ème} rang : augmentation de 0.5%

Et de renvoyer le montant total d'augmentation.

N.B : l'employé le plus rentable est celui qui a le plus grand taux de rendement.

D. Ecrire les déclencheurs suivants (13pts):

1. trigger nommé **tg_salaire_log**

5 pts

- Donner le script permettant de créer la table suivante :

SalaireLog (Num_auto, matricule, date_modification, ancien_salaire, nouveau_salaire, taux, utilisateur)

- Créer un trigger pour ajouter une ligne d'historisation à chaque modification du salaire.

Taux = (nouveau_salaire – ancien_salaire) / ancien_salaire.

N.B : n'oublier pas de traiter le cas de la mise à jour par lot.

2. trigger nommé **tg_tache_ajouter**

Qui permet de contrôler le nombre des taches ajouté (le nombre des taches d'un projet doit être toujours inférieur à la valeur du champ « nbr_limite_taches » de ce projet.

3. trigger nommé **tg_projet_supprimer**

Permettant de supprimer tous les travaux et toutes les taches à la suppression d'un projet (suppression en cascade d'un projet).

4. trigger nommé **tg_projet_ajouter**

8 pts

- Créer ce trigger pour ajouter N taches à l'insertion d'un nouveau projet comme suite:
 - les noms des taches sont : tache 1, tache 2, tache 3, ...

- la durée est fixe à 40 jours
- le cout est null.

N.B : N est égale à la valeur du champ nbr_limite_taches.

Num_tach	Nom_tach	Date_debut	Date_fin	Cout	Num_prj
Max(num_tach)+1	Tache 1	Date système	Date système +40j	Null	Num_prj inséré
Max(num_tach)+2	Tache 2	Date système +41j	Date système +81j	Null	Num_prj inséré
Max(num_tach)+3	Tache 3	Date système +82j	Date système +102j	Null	Num_prj inséré
...
Max(num_tach)+N	Tache N	Date système + (N-1)x41j	Date système + (N-1)x41j +40	Null	Num_prj inséré