

Laboratoire3 : Les triggers.

Ce laboratoire est noté. Date de remise le 20 février.

Objectifs :

- Écrire des triggers
- Vérifier le fonctionnement des triggers
- Activer et désactiver des triggers

Consignes :

1. Chaque trigger doit être écrit séparément. (Il n'y aura pas plus qu'un trigger dans un fichier SQL).
2. Pour chacun des triggers, vous devez écrire le code SQL qui permet de le tester.
3. Lors de la correction, vous devez me montrer : Le code SQL du trigger et la vérification que le trigger fonction.

Utilisez les tables de dernier laboratoire : EMPLOYESBIDON et TYPEEMPLO puis réaliser les exercices suivants :

Exercice1 : Écrire un trigger sur la table EMPLOYESBIDON qui permet des numérotations automatiques (numéro séquentiel et automatique du numéro employés) lors des insertions.

Exercice 2 : Écrire un trigger sur la table EMPLOYESBIDON qui contrôle la modification des salaires. Lorsque le salaire modifié est inférieur au salaire actuel, il y a un message d'erreur : « le salaire ne peut pas être modifié à la baisse »

Exercice 3 : Écrire un trigger qui permet de contrôler les ajouts et les modifications des salaires. Si EMPLOI n'est pas 'DIRECTEUR', alors Lors de **l'ajout, modification** d'un nouvel employé, nous devons vérifier si le salaire de celui-ci se retrouve dans la fourchette des salaires. Les salaires sont en fonction des EMPLOI et sont dans la table TYPEEMPLOI

Tester les triggers avec:

```
INSERT INTO employesbidon VALUES (13, 'PATOCHÉ','ALAIN','FINANCES',25000)
```

```
UPDATE employesbidon SET salaire =66000 WHERE numemp= 11;
```

```
UPDATE employesbidon SET salaire =40000 WHERE numemp =11;
```

Exercice 4 : Écrire un trigger qui permet de faire une mise à jour en cascade lors de la suppression d'un type d'emploi dans la table TYPEEMPLOI ou la mise à jour de la modification d'un emploi dans la même table. ie : lorsque EMPLOI dans la table TYPEEMPLOI est modifié (update) alors il faut que EMPLOI se met à jour dans la table EMPLOYESBIDON et lorsqu'un EMPLOI est supprimé dans la table TYPEEMPLOI, il faut que la suppression se reflète dans la table EMPLOYESBIDON.

Vérifier que votre Trigger fonctionne. Peut-être que vous devez désactiver un des triggers déjà créés.

Modifier votre Trigger pour qu'il fasse aussi les suppressions en cascade

Exercice 5 : Créer un trigger sur la table EMPLOYESBIDON qui interdit les mises à jour de la table les fins de semaines (samedi et dimanche)

Vérifier que votre trigger fonctionne. Tester pour la journée actuelle (mercredi)

Exercice 6 : Créer la table Logs comme suit :

```
CREATE TABLE LOGS (NUM NUMBER, DATEMAJ DATE, NSALAIRE NUMBER (8,2),  
COMMENTAIRE VARCHAR2(40), NOM varchar2(40));
```

Cette table aura comme rôle de consigner les modifications apportées à la table EMPLOYESBIDON.

Écrire un trigger qui permet d'insérer dans la table Logs, lorsque les modifications sont apportées aux salaires de la table EMPLOYESBIDON. Dans la table Logs, on y insère un numéro d'employé affecté par la modification, la date de modification, le nouveau salaire, un commentaire et l'USER ayant fait cette modification.

Vérifier que votre trigger fonctionne.

Exercice 7

Réécrire votre procédure insertion en tenant compte de votre trigger d'auto incrémentation. Tester.

Vous devez remettre un fichier SQL contenant les exercices numérotés et un fichier SQL d'exécution pour vérifier que vos triggers fonctionnent.