

TP 4 – Triggers

Syntaxe de création d'un trigger

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[FOR EACH ROW]
WHEN (condition)
BEGIN
--- Instruction PLSQL
END;
/
```

- **CREATE [OR REPLACE] TRIGGER trigger_name** : pour créer ou remplacer un trigger existant.
- **{BEFORE | AFTER}** : le moment du déclenchement du trigger (avant ou après l'opération de mise à jour).
- **{INSERT [OR] | UPDATE [OR] | DELETE}** : l'événement de mise à jour qui provoquera le déclenchement du trigger. Plusieurs événements séparés par OR sont possibles.
- **[OF col_name]** : utilisé dans le cas de l'opération Update appliquée sur une colonne particulière.
- **[ON table_name]** : le nom de la table sur laquelle le trigger est défini.
- **[FOR EACH ROW]** : spécifie si le trigger est lancé pour chaque ligne affecté ou une seule fois.
- **WHEN (condition)** : le trigger est lancé seulement lorsque la ligne affectée vérifie la condition.

Remarque : pour générer une exception et empêcher le programme de continuer, l'utilisateur peut lancer la procédure **raise_application_error (-Num_Message, 'Message à Afficher')** ; Num_Message est compris entre 20000 et 20999.

Questions

Supposons que les tables des TP précédents sont créées et remplies.

1. Créez un trigger qui affiche « **un nouveau employé est ajouté** » après chaque insertion d'un employé. Répétez la même chose pour la modification ou la suppression.
2. Créez un trigger qui affiche « **Un produit est ajouté à la catégorie[Nom de la catégorie]** » après chaque insertion dans la table produit.
3. Créer un triggers qui vérifie avant modification du l'attribut code client dans la table commande que la nouvelle valeur existe réellement, sinon, il refuse l'opération.
4. L'administrateur veut, pour un besoin interne, avoir le chiffre d'affaire généré par chaque produit. Pour cela, il ajoute un attribut : **CHIFFRE_AFFAIRE** dans la table produit.
 - a. Ajouter l'attribut.
 - b. Créer un trigger CHIFFRE_AFFAIRE_TRIGGER qui mit à jour l'attribut CHIFFRE_AFFAIRE lors d'une insertion, d'une modification et une suppression dans la table DETAILCOMMANDE.

5. L'administrateur veut sauvegarder pour chaque date le nombre de commandes passées. A chaque fois une commande est ajoutée, on ajoute une ligne dans la table Historique_commande (DateC, total_commande) ou bien on met à jour l'attribut total_commande.
 - a. Ajouter la table.
 - b. Vérifier cette contrainte lors d'une insertion, d'une modification et une suppression.