

Administration des SGBD  
TP 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 mit à jour l'attribut **total\_commande**.
  - a. Ajouter la table.
  - b. Vérifier cette contrainte lors d'une insertion, d'une modification et une suppression.