

Administration des SGBD
TP Triggers

RELARQUE IMPORTANTE

Pour l'ensemble des TP, on demande de réaliser un compte-rendu du travail effectué. Ce compte-rendu comprendra un rapport (manuscrit ou réalisé sur traitement de texte) détaillant les points intéressants abordés à chaque question :

- requêtes SQL correspondant aux questions posées
- résultat de l'exécution de la requête
- réponses aux questions soulevées dans le sujet,
- explication des parties non triviales des requêtes complexes,
- analyse et commentaire des jeux d'essais.

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 écraser un trigger existant.
- **{BEFORE | AFTER | INSTEAD OF }** : 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 lance 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 créés et remplies.

1. Créez un trigger qui affiche « **un nouveau client est ajouté** » après chaque insertion d'un client. Répétez la même chose pour la modification ou la suppression.
2. Créez un trigger qui affiche « **un nouveau modèle est ajouté à la marque [Nom de la marque]** » après chaque insertion d'un modèle.
3. Créer un trigger qui vérifie que lors de la modification du **salaire d'un employé**, la nouvelle valeur ne peut jamais être inférieure à la précédente.
4. Chaque intervention est prise en charge par un ou plusieurs employés dans des périodes différentes. Créer un trigger qui vérifie que la période d'intervention d'un employé est comprise dans la période d'intervention.
5. L'administrateur veut, pour un besoin interne, avoir le nombre total des interventions pour chaque **employé**. Pour cela, il ajoute un attribut : **TOTAL_INTERVENTIONS** dans la table employé.
 - a. Ajoutez l'attribut.
 - b. Créez un trigger **TOTAL_INTERVENTIONS_TRIGGER** qui met à jour l'attribut **TOTAL_INTERVENTIONS**.
6. L'administrateur veut sauvegarder pour chaque mois de l'année le total de gains de toutes ses interventions. A chaque fois une intervention est ajoutée dans un mois soit c'est la première intervention dans ce mois donc on ajoute une ligne dans la table **CHIFFRE_AFFAIRE (MOIS, ANNEE, TOTAL_GAINS)** ou bien on met à jour l'attribut **TOTAL_GAINS**.