

TP 1 - Triggers

Le langage PL/pgSQL est une extension du PostgreSQL avec des caractéristiques propres aux langages de programmation. L'objectif du TP n'est pas de recenser l'ensemble des caractéristiques du langage mais d'utiliser les fonctions essentielles afin d'appréhender les principes des triggers (déclencheurs).

Les exercices ci-dessous sont à traiter sur la base de données `magasin.sql`. Je vous conseille de travailler dans des fichiers de script séparés pour chaque question (à lancer avec `\i` dans le terminal Postgres).

Pour chacun des triggers demandés, vous procéderez de la manière suivante :

- Écrire la partie de création du trigger, ce qui vous forcera à déterminer le type de trigger, le(s) événements déclencheurs et le moment d'appel.
- Écrire en français la spécification du comportement attendu du trigger.
- Proposer une série de tests qui valideront le bon comportement du trigger.
- Écrire la fonction de trigger.
- Activer le trigger et le tester.

► Exercice 1 : Échauffement

1. Reprenez le trigger `verif_stock.sql` commencé en cours. Testez les commandes pour créer et supprimer le trigger. Cherchez dans la documentation comment activer ou désactiver le trigger sans le supprimer.
2. Expliquez le comportement du trigger, proposez des tests et vérifiez que le trigger se comporte bien comme attendu.
3. Améliorez le trigger pour que l'ajout d'un produit à une commande mette automatiquement à jour la quantité restante du produit concerné dans les stocks du magasin. Proposez un comportement raisonnable si le produit n'est pas disponible dans la quantité demandée.
4. Améliorez le trigger pour fixer automatiquement le prix du produit sur la facture au prix actuellement pratiqué par le magasin.

► Exercice 2 : Historique des prix

1. Créez une table `historiquePrix(log_id, date, idmag, idpro, ancienPrix, nouveauPrix)` qui servira à stocker l'évolution des prix des produits dans les différents magasins de la base de données.
2. Écrivez un trigger pour remplir automatiquement l'historique des prix chaque fois que le prix d'un produit est modifié.
3. Améliorez votre trigger pour afficher automatiquement une annonce lorsqu'un magasin se met à pratiquer le meilleur prix du marché pour un produit donné. L'annonce devra indiquer le nom et le lieu du magasin, la référence et le libellé du produit, ainsi que le prix et la quantité disponible.

► **Exercice 3 : Cartes de fidélité**

1. Créez un trigger qui met à jour les points de fidélité d'un client lors d'un nouvel achat. Pour simplifier, on supposera que le nombre de points gagnés correspond au montant de l'achat.

Si le client n'a pas de carte de fidélité dans le magasin concerné, un message s'affichera indiquant au client le nombre de points qu'il aurait pu gagner.

Vérifiez bien l'interaction de votre nouveau trigger avec celui de l'exercice 1.4.

2. Créez un trigger pour plafonner le nombre de points de fidélité à 1000. Vérifiez bien comment votre nouveau trigger interagit avec celui de la question précédente.

► **Exercice 4 : Triggers et vues - points de jury**

1. Reprenez le script `universite.sql` vu en cours. Chargez le dans votre base de données, expliquez son comportement et testez le.
2. Améliorez le script pour résoudre les problèmes vus en cours, en particulier :
 - Si le cours *Points de jury* n'octroie pas assez d'ECTS pour permettre à l'étudiant de passer en L3, un message s'affichera, indiquant que l'étudiant a trop de retard pour que le jury puisse forcer le passage.
 - Si les données rentrées pour l'étudiant sont incorrectes (par exemple, le numéro d'étudiant ne correspond pas au nom fourni), la modification devra être annulée.
 - Si le cours *Points de jury* n'existe pas, il sera automatiquement créé et octroiera 10 ECTS.

Prenez bien le temps de tester tous les cas possibles.

► **Exercice 5 : Triggers et vues - cas simple**

La chaîne de magasins *La cabale des câbles* souhaite contrôler et aligner ses prix simultanément dans toutes ses boutiques.

1. Construisez une vue regroupant les références, libellés et prix de tous les produits vendus dans un magasin *La cabale des câbles*.
2. Discutez en petits groupes du sens des opérations **INSERT**, **UPDATE** et **DELETE** sur cette vue. Validez vos interprétations auprès de votre chargé de TP.
3. Écrivez des triggers pour implémenter votre spécification des opérations qui ont du sens sur cette vue.

► **Exercice 6 : Triggers et vues - cas très discutable**

1. Créez une vue qui rassemble pour chaque produit ses informations ainsi que l'information du magasin qui le propose au meilleur prix avec le prix correspondant et le nombre d'exemplaires en stock.
2. Proposez des situations dans lesquelles les opérations **INSERT**, **UPDATE** et **DELETE** ont du sens. Validez vos spécifications auprès de votre chargé de TP.
3. Écrivez des triggers pour implémenter votre spécification des opérations qui ont du sens sur cette vue.