

TP N°5

Rappel (declencheurs/triggers)

- Un **déclencheur (trigger)** est une procédure stockée, dans une BD, qui s'appelle automatiquement à chaque fois qu'un événement spécial se produit sur la table à laquelle le déclencheur est attaché.
 - Un **déclencheur** peut être utilisé pour valider les données, enregistrer les anciennes et les nouvelles valeurs dans une table d'audit (log) ou s'assurer que les règles métier sont respectées.
 - Par exemple, un **déclencheur** peut être invoqué lorsqu'une ligne est insérée dans une table spécifique ou lorsque certaines colonnes de la table sont mises à jour.
 - On distingue **deux types** de déclencheurs : **(1)** Le **déclencheur BEFORE** qui exécute son action **avant** l'exécution de l'instruction de déclenchement (**ÉVÉNEMENT**). **(2)** Le **déclencheur AFTER** qui exécute son action **après** l'exécution de l'instruction de déclenchement.
 - Quant aux **événements** auxquels un **déclencheur** peut être assigné, il y en a **trois** : **INSERT**, **UPDATE** et **DELETE**.
 - Parmi les avantages des déclencheurs on cite : **(1)** renforcer l'intégrité des données, **(2)** détecter les erreurs **(3)** auditer les changements de données, **(4)** enregistrer les événements **(5)** aider à prévenir les transactions invalides, ...etc.
 - le mot-clé **OLD** est utilisé pour accéder aux données de ligne qui sont remplacées par la mise à jour ;
 - Le mot-clé **NEW** permet d'accéder aux données de ligne entrantes qui remplaceront l'ancienne ligne, en cas de succès ;
 - Selon le type de **déclencheur** créé, les lignes **OLD** et **NEW** peuvent ne pas être disponibles :
 - (1) INSERT TRIGGER** : Accès possible à **New uniquement**.
 - (2) UPDATE TRIGGER** : Accès aux deux pseudo-lignes **NEW** et **OLD**
 - (3) DELETE TRIGGER** : Accès **uniquement** à **OLD** pseudo-lignes,
- Bref, il n'y a pas de ligne **OLD** sur un déclencheur **INSERT** et pas de ligne **NEW** sur un déclencheur **DELETE**.

Syntaxe

```
CREATE TRIGGER [nom_declencheur]
[before | after]
{insert | update | delete}
ON [nom_table]
[for each row]
[corps_declencheur]
```

- **[before | after]** ([avant | apres]) : Ceci spécifie quand le déclencheur sera exécuté.
- **{insert | update | delete}** ({insérer | mise à jour | supprimer}) : ceci spécifie l'opération DML.
- **ON [nom_table]** : Ceci spécifie le nom de la table associée au déclencheur.
- **[for each row]** : Ceci spécifie un déclencheur au niveau de la ligne, c'est-à-dire que le déclencheur sera exécuté pour chaque ligne affectée.
- **[corps_declencheur]** : Ceci fournit les opérations à effectuer lorsque le déclencheur est déclenché.

Applications

1. Créer une BD nommée **Supermarket** et contenant les tables :
client (numClt,nom, prenom, dateNaissance, adresse)
employe (numEmp,nomEmp, prenomEmp, salaire)
commande (numComd, dateComd, dateLivr)
archiveEmp (numArch, nomEmp, prenomEmp, salaire)
2. Écrire puis tester un trigger qui permet de refuser l'insertion d'un client suite à son âge inférieur à 18 ans et affiche le message explicatif : "Le client doit être âgé plus de 18 ans".
3.
 - a) Créer et initialiser une table **budgetEmpl** contenant la somme des salaires des employés.
 - b) Assigner cette table un trigger permettant de mettre à jour la table **budgetEmpl** après chaque insertion d'un nouvel employé.
4. Écrire puis tester un trigger qui permet, **avant** la mise à jour de la date de naissance d'un client, de contrôler l'âge de celui-ci : s'il est âgé de plus 18 la mise à jour aura lieu sinon le message explicatif : "Le client doit être âgé plus de 18 ans" sera affiché.
5. Écrire puis tester un déclencheur qui permet de mettre le salaire à 3500 dhs pour un employé n'ayant pas de salaire.
6. Écrire puis tester un trigger qui permet de contrôler les insertions et les modifications dans la table commande selon cette contrainte : *"La date de livraison ne doit pas être inférieure à la date de commande"*.
7. Écrire puis tester un déclencheur permettant **avant** de supprimer un employé de le sauvegarder dans la table d'archivage **archiveEmp**.
8. Écrire puis tester un déclencheur permettant **après** la suppression d'un salaire de mettre à jour la somme des salaires stockée dans la table **budgetEmpl**.
9.
 - a) Lister tous les déclencheurs de la BD **Supermarket**.
 - b) Supprimer la table **archiveEmp**.
 - c) Réafficher déclencheurs de la BD **Supermarket**.
 - d) Qu'en est-il au trigger de la table **archiveEmp** ? Conclure.
 - e) Supprimer le déclencheur créé en **3.b)** et vérifier sa suppression.