

TP 3 PLSQL

Pour ce TP, utiliser SQL Developer pour vous connecter à la base de données ORCLIUT.

Quelques précisions :

- 1- Du fait d'une incompatibilité entre SQL Developer et la version d'Oracle installée, la commande DBMS_OUTPUT.PUT_LINE utilisée pour l'affichage des messages dans la console de sortie n'est pas bien gérée. Pour profiter de son ergonomie, utiliser SQL Developer pour vos développements et le terminal (SQL*Plus) pour tester vos scripts et afficher les résultats (notamment avec DBMS_OUTPUT).

Pour vous connecter, ouvrez un terminal et lancer la commande ci-dessous (remplacer username par votre login utilisé dans sql developer)

```
sqlplus  
"username@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=lorien.arda.lan)(Port=1521))(CONNECT_DATA=(SID=ORCLIUT))"
```

- 2- Une fois connecté, exécuter la commande ci-dessous pour activer DBMS_OUTPUT.

```
SQL> SET SERVEROUTPUT ON
```

Soit le schéma de base de données suivant :

Client : (**id**, prenom, nom, email, ville,);

Commande: (**id**, client_id, date_achat, reference);

ligne_commande: (**id**, commande_id, produit_id, quantite, prix_total);

Produit : (**id**, nom_produit, prix_unitaire, quantite_en_stock)

1. Créer une procédure sp_DateDernierAchat(Nom_Produit) qui prend en paramètre un Nom de produit et affiche la dernière date d'achat du produit sous la forme « le produit X a pour dernière date d'achat »
2. Ajouter la colonne quantite_en_stock dans la table Produit. Vous pouvez initialiser la colonne avec une valeur > 0.
3. Créer un trigger trg_GestionStock qui va permettre de gérer les stocks : avant qu'une nouvelle ligne de commande ne soit ajoutée, vérifiez que la quantité commandée <= quantite_stock du produit. Si c'est le cas, la ligne de commande est ajoutée et la quantité est mise à jour dans la table produit dans le cas contraire, une exception est levée.
4. Ecrire le trigger trg_DeleteClient qui gère la suppression d'un client; lorsqu'un client est supprimé de la table client, toutes les commandes de ce client sont supprimées et toutes les lignes de commande supprimées.

5. On souhaite conserver les informations des anciens clients. Créer une table Client_His qui a la même structure que la table client.

Modifier le trigger de la question 4 : les informations du client en cours de suppression doivent être stockées dans la table d'historisation.

6. On souhaite faire l'audit de la table commande.

Créer la table CommandeAuditLog (Utilisateur, ActionSQL, DateMAJ, ActCol) qui contiendra :

- Utilisateur : le nom de l'utilisateur qui effectue l'action. Utiliser la variable sqlplus USER (*SELECT USER FROM DUAL*) qui contient le nom de l'utilisateur SQL connecté.
- ActionSQL contiendra 'INS' pour une insertion, 'DEL' pour une suppression de ligne, 'UPD' pour une mise à jour.
- DateAct qui contiendra la date de l'action. Utiliser la variable SYSDATE.
- ActCol contiendra le nom des colonnes qui ont été modifiées par l'ordre UPDATE.

UPDATING('nom_colonne') retourne TRUE si c'est la colonne nom_colonne qui est mise à jour et false si non.