TP 2 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)

Les données à utiliser sont les mêmes que celles du 1ier TP.

Pour connecter à la base via le terminal : *sqlplus "username@(DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(Host=lorien.arda.lan)(Port=1521))(CONNECT_DATA=(SID=ORCLIUT)))"*

Exécuter *SET SERVEROUTPUT ON* afin d'activer la sortie.

1- Créer une séquence seq_tp ayant pour première valeur 121. Afficher sa valeur actuelle et la valeur suivante.

2- Implémenter une procédure stockée **sp_insert_ligne_commande(p_commande_id, p_produit_id, p_quantite)** qui permet d'insérer une ligne de commande dans la table ligne_commande. Utiliser la séquence seq_tp créée précédemment pour l'id de la ligne de commande. Le prix total doit être calculer dans la procédure.

Tester votre procédure avec un id de commande ou un id de produit qui n'existe pas.

Modifier la procédure précédente afin de gérer le cas où p_commande_id ou p_produit_id n'existe pas.

NB : Une ligne de commande ne peut être insérée uniquement si la commande et le produit existent déjà.

3- On souhaite calculer les frais de livraisons des commandes. Implémenter une procédure **sp_frais_livraison (p_commande_id INT, p_frais_livraison OUT NUMBER)** qui calcule les frais de livraison d'une commande.
p_frais_livraison est un paramètre de sortie.

Les frais de livraisons valent :

- 0 si commande>= 100 euros
- 4 si commande>= 50 euros et < 100euros
- 10 si commande < 50 euros.

4- Implémenter la question ci-dessus en utilisant une fonction.

5- Points de fidélité : Implémenter une fonction ***fn_points_fidelite(Id_client)*** qui retourne les points de fidélité d'un client.
1 point de fidélité tous les 10 euros d'achat.
Traiter le cas où le client n'a jamais passé de commande ou le client n'existe pas.

Afficher le résultat sous la forme suivante.

```
Da costa a 0 point de fidélité
Vespasien a 246 points de fidélité
Collin a 244 points de fidélité
Camus a 185 points de fidélité
```

6- Implémenter une fonction ***fn_DateDernierAchat(Nom_Produit)*** qui prend en paramètre un Nom de produit et retourne la phrase suivante : « le produit X a pour dernière date d'achat yyyy/mm/dd »