

**TP-1 : PL/SQL****1. Objectif :**

L'objectif de ce TP est de créer vos propres programmes PL/SQL.

**2. Application Emploi :**

1. Ecrire un programme PL/SQL permettant l'insertion de nombres dans la table des messages :

- créer la table des Messages (Resultats VARCHAR2 (60));
- insérer les nombres 1 à 10 en excluant 6 et 8.
- valider avant la fin du programme.

2. Créer un programme PL/SQL qui calcule le montant de la commission pour un employé donné en se basant sur le salaire de l'employé (vous pouvez utiliser la table EMP fournie par Oracle ou la créer si nécessaire) :

- le numéro de l'employé est donné par l'utilisateur.
- si l'employé a un salaire inférieur à 1000€, la commission vaut 10% du salaire
- si l'employé a un salaire entre 1000 et 1500€, le montant de la commission vaut 15% du salaire.
- si le salaire de l'employé dépasse 1500€, le montant de sa commission vaut 20% de son salaire.
- si l'employé a un salaire NULL, le montant de la commission vaut 0.
- valider par Commit.

3. L'entreprise employant les personnes de la table EMP est délocalisée des États-Unis vers la France. Il est donc nécessaire de convertir leur salaire et leur commission en Euro (pour simplifier, on admettra 1 USD = 0.85 Euro). Tous les employés voient également augmenter leur salaire de 25 % après conversion.

- créer une nouvelle table vide EMP\_FR de même structure que EMP. On pourra, par soucis de rapidité, recopier la table EMP dans EMP\_FR pour en obtenir la structure, puis effacer tous les tuples de EMP\_FR.
- écrire un programme PL/SQL permettant de recopier tous les tuples de la table EMP dans la table EMP\_FR en effectuant au passage les opérations nécessaires sur le salaire et la commission. Traiter le cas où la table EMP est vide comme une exception.

4. Créer une procédure pour interroger la table EMP, en recherchant le salaire et la fonction de l'employé en utilisant le numéro de l'employé. Tester avec le numéro employé 7839.

5. Créer la fonction SAL\_ANNUEL qui retourne le salaire annuel d'un employé et ayant pour paramètres d'entrée le salaire mensuel de l'employé et sa commission.

Utiliser cette fonction dans une requête Select sur la table Employé. S'assurer que la fonction traite les valeurs NULL.

6. Créer une procédure qui permet d'augmenter le salaire des employés travaillant dans un département donné comme paramètre de la procédure. Le pourcentage de l'augmentation du salaire est aussi donné comme paramètre de la procédure

7. Créer pour chaque ligne, un trigger sur la table Personne qui pourra afficher toute commande du DML invoquée (Insert, Update ou Delete) ainsi que le nom de la personne concernée.

8. Créer une séquence qui pourra être utilisée comme clé primaire de la table Personne puis créer un trigger qui pourra automatiquement remplir l'ID.

9. Créer un trigger qui doit permettre qu'un salaire soit augmenté mais jamais diminué.