masquer=1

TD 10 : PL/SQL

PROGRAMMES PL/SQL

Nous travaillerons dans ce TD sur les trois tables suivantes qui permettent de décrire des employés, des projets, l'affectation des employés aux projets et les grilles de salaire pour différents profils de postes :

EMPLOYE (NumSS, NomE, PrenomE, VilleE, DateNaiss)
PROJET (NumProj, NomProj, RespProj, VilleP, Budget)
EMBAUCHE (NumSS, NumProj, DateEmb, Profil)
GRILLE_SAL (Profil, Salaire)

La clé primaire de chaque table est soulignée.

Exercices

1. Expliquez le fonctionnement du programme PL/SQL suivant:

```
DECLARE
      trouve BOOLEAN;
BEGIN
     FOR  r  IN (Select numproj, nomproj, villep from Projet ORDER BY nomproj) LOOP
               dbms_output.put_line('Projet: '||r.nomproj||' dans la ville  '||r.villep);
               trouve := FALSE;
               for r2 IN (Select nome, prenome from employe e, embauche b
                         Where e.numss=b.numss and b.numproj=r.numproj )
               LOOP
                   dbms_output.put_line('Employé: '|| r2.nome||', '|| r2.prenome);
                   trouve := TRUE;
              END LOOP;
              IF(trouve = FALSE) THEN
                   dbms_output.put_line('Pas d''employé');
             END IF;
        END LOOP;
END;
/
```

2. (*Curseur*). Modifiez le programme précédent afin de remplacer la première boucle FOR par un curseur.

3. (*Curseur avec des paramètres*). Modifiez le programme de la question 2 afin de remplacer la boucle FOR par un curseur avec des paramètres. Utilisez les attributs des curseurs au lieu de la variable trouve pour vérifier l'employé.

4. (*Curseur Implicite*). Écrivez un bloc PL/SQL anonyme qui supprime tous les employés de la table EMPLOYE qui ont 50 ans ou plus et qui affiche le nombre de lignes qui ont été supprimées ou le message 'Aucun employé supprimé' si aucun employé n'a plus de 50 ans. *Utilisez les attributs*

d'un curseur implicite.

5. Optionnel - (*Curseur et CASE*). Écrire un bloc anonyme qui augmente tous les salaires dans la table Grille_Sal. Les salaires inférieurs à 40000 sont augmentés de 30%, les salaires compris entre 40000 et 60000 de 20% et les salaires supérieurs à 60000 sont augmentés de 10%. Utilisez un CASE pour tester les différentes valeurs possibles du salaire.

6. (*Exception prédéfinie*). Écrire une procédure PL/SQL qui a 3 paramètres: le *numSS*, le *nomE* et *prenomE* d'un nouvel employé à insérer dans la table Employé. S'il existe déjà un autre employé avec le même numSS dans la table Employé elle doit afficher un message d'erreur sans insérer l'utilisateur. Dans le cas contraire la procédure doit insérer le nouvel employé et afficher un message de confirmation. Utilisez l'exception prédéfinie **NO_DATA_FOUND**.

7. (*Exception utilisateur*). Écrire une procédure qui prend comme paramètre le numéro et le budget d'un projet et qui l'ajoute à la table projet seulement si la somme tous les budgets après l'ajout ne dépasse pas 400000. Dans le cas contraire le projet n'est pas ajouté et un message d'erreur est affiché à l'utilisateur. Utilisez une exception utilisateur.