Université Lumière – Lyon 2, Faculté de Sciences Économiques et de Gestion
Licence IUP ISEA – Année 1999-2000
TD n° 2 – PL/SQL et triggers
J. Darmont (*jerome.darmont@univ-lyon2.fr*), 06/04/00

Exercice 1

Il s'agit d'implémenter le programme de l'Exercice 1 sur PL/SQL vu en cours.

1. Créer la structure de la table PRODUIT (NumProd, Desi, PrixUni). Laisser la table vide.
2. Créer la table PRODUIT2K en recopiant la structure de la table PRODUIT.
3. Saisir le bloc PL/SQL adéquat dans un fichier.
4. Exécuter ce bloc PL/SQL.
5. Peupler la table PRODUIT avec quelques tuples.
6. Ré-exécuter le bloc PL/SQL.
7. Afficher le contenu de la table PRODUIT2K.

Exercice 2

Il s'agit d'implémenter le programme de l'Exercice 2 sur PL/SQL vu en cours.

1. Créer la structure de la table COMMANDE (NumCli, NumProd, Qte). Laisser la table vide.
2. Saisir le bloc PL/SQL adéquat dans un fichier.
3. Exécuter ce bloc PL/SQL.
4. Insérer un tuple avec quantité non NULL dans la table COMMANDE.
5. Ré-exécuter le bloc PL/SQL.
6. Insérer un tuple avec quantité NULL dans la table COMMANDE.
7. Ré-exécuter le bloc PL/SQL.
8. Insérer un tuple avec quantité non NULL dans la table COMMANDE.
9. Ré-exécuter le bloc PL/SQL.
10. Insérer plusieurs tuples avec des quantités NULL et non NULL dans la table COMMANDE.
11. Ré-exécuter le bloc PL/SQL.

Exercice 3

L'entreprise employant les personnes de la table EMP du TD n° 1 est délocalisée des États-Unis en France (si, si). Il est donc nécessaire de convertir leur salaire et leur commission en francs (pour simplifier, on admettra 1 USD = 6,5 FF). Tous les employés voient également augmenter leur salaire de 25 % après conversion.

1) 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.
2) 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.

Exercice 4

Il s'agit d'implémenter le trigger de l'Exercice 1 (sur les triggers) vu en cours.

1. Créer la structure de la table TABL (Clenum).
2. Saisir le code du trigger adéquat dans un fichier.
3. Exécuter le trigger plusieurs fois en insérant des tuples dans la table TABL.
4. Vérifier son bon fonctionnement en affichant le contenu de la table TABL.

Exercice 5

Il s'agit d'implémenter le trigger de l'Exercice 2 (sur les triggers) vu en cours.

1. Créer la structure de la table CLIENT (NumCl, Nom, CP, *NumConjoint*).
2. Saisir le code du trigger adéquat dans un fichier.
5. Exécuter le trigger plusieurs fois en insérant des tuples dans la table CLIENT.
6. Vérifier son bon fonctionnement en affichant le contenu de la table CLIENT.

Correction Exercice 3

```
create table emp_fr as select * from emp;
delete from emp_fr;


-- Remplissage de la table EMP_FR

DECLARE

  n NUMBER(2);
  CURSOR employes IS
    SELECT empno, ename, job, mgr, hiredate, sal, comm, deptno
    FROM emp;
  employe employes%ROWTYPE;
  newsal emp.sal%TYPE;
  newcomm emp.comm%TYPE;
  empvide EXCEPTION;

BEGIN

  -- Test table vide
  SELECT COUNT(*) INTO n FROM emp;
  IF n=0 THEN
    RAISE empvide;
  END IF;

  -- Remplissage emp_fr
  FOR employe IN employes LOOP
    -- Calculs
    newsal:=employe.sal*6;
    newsal:=newsal*1.25;
    IF employe.comm IS NOT NULL THEN
      newcomm:=employe.comm*6;
    ELSE
      newcomm:=NULL;
    END IF;
    -- Insertion donnees
    INSERT INTO emp_fr VALUES(employe.empno, employe.ename, employe.job,
      employe.mgr, employe.hiredate, newsal, newcomm, employe.deptno);
  END LOOP;

EXCEPTION

  WHEN empvide THEN RAISE_APPLICATION_ERROR(-20501,'Pas d employe !');

END;
.
```