

LIFBDW2 – BASES DE DONNÉES AVANCÉES

TP4 – PL/SQL : Premiers pas

Licence informatique – Automne 2017–2018

Résumé

Ce TP est une première prise en main de PL/SQL (procédures, fonctions, curseurs, déclencheurs). *Il est donc nécessaire de parcourir les différents supports disponibles sur la page du cours avant de débiter ce TP.*

1 Création de la table

1. Exécuter le script disponible à l'adresse suivante : http://liris.cnrs.fr/~mplantev/ENS/LIFBDW2/TP/creation_SCOTT_DB.sql
2. Etudier les tables nouvellement créées et les contraintes associées.

2 Premiers pas en PL/SQL

1. Exécuter le bloc PL/SQL suivant :

```
SET SERVEROUTPUT ON
/*(commande d'environnement de SQL rendant visible le fonctionnement des procédures
de DBMS_OUTPUT)*/
DECLARE
/*section déclarative des variables et curseurs du bloc PL/SQL, facultative*/
Nb NUMBER;
BEGIN -- section obligatoire des instructions DBMS_OUTPUT.PUT_LINE('Hello World !!!') ;
-- procédure du package DBMS_OUTPUT visualisant une donnée
Nb := 10;
DBMS_OUTPUT.PUT_LINE('Nb est ' || Nb);
-- || opérateur de concaténation
END;
/ /*(Ce caractère slash lance l'exécution du bloc PL/SQL
dans la mémoire tampon SQL. Ne pas mettre d'espace avant).*/
```

2. Ecrire un bloc PL/SQL affichant le nombre de lignes de la table Employee. Il contiendra la commande SQL SELECT assignant une valeur à une variable (ou commande PL/SQL SELECT INTO maVariable FROM;) . On notera que la commande de SQL SHOW ERRORS est utile pour mettre au point un programme PL/SQL.
3. Ecrire une fonction (mémoire) de paramètre un numéro de département et qui retourne le nombre de métiers (différents) de ce département. On pourra typer le numéro de département par

```
{Employee.DeptNo%TYPE}
```

Appeler ensuite cette fonction stockée en utilisant un bloc PL/SQL. Quindique la vue USER_OBJECTS ? La fonction créée et le bloc PL/SQL sont-ils des procédures stockées dans votre compte (stored procedure ou plus précisément stored function) ?

4. De la même façon, écrire une fonction qui étant donné un numéro de département retourne le nombre d'employés dans le département. Lever une exception lorsque le département n'existe pas et/ou le nombre d'employés est inférieur à 1.
5. Ecrire une fonction qui assure la conversion d'un salaire défini en dollars vers un salaire correspondant en euros.

3 Curseurs

1. A l'aide d'un curseur, pour chaque ligne de la table EMPLOYEE, afficher une « phrase résumé » indiquant la situation de l'employé (ex : Employé MILLER(num :7782) travaille dans le département 10 depuis 23/01/82 pour 1300 euros par mois.)
2. Ajouter la colonne NbMétiers , qui contiendra le nombre des métiers de chaque département. On mettra à jour NbMétiers à partir Employee.

4 Triggers

1. Mettre en oeuvre un déclencheur et une exception déclarée pour l'assertion « le salaire dun employé ne peut pas diminuer ». On vérifiera son fonctionnement, puis on commentera ses propriétés à partir de la vue USER_TRIGGERS.
2. De la même façon, créer un déclencheur qui pour chaque mise à jour (insertion ou suppression ou modification) de laffectation dun employé, le déclencheur incrémente ou décrémente le nombre demployés par département. Ajouter d'abord une colonne nbemployé dans la table Departement et la mettre à jour.
3. Imaginer d'autres triggers.