

TP n° 4 : Introduction au Procedural Language (PL/SQL) d'Oracle (Partie I)

Remarque : pour activer l'affichage sous SqlDeveloper, aller sur **Affichage** et cliquer sur **sortie SGBD**, sans oublier l'instruction **SET SERVEROUTPUT ON**, au début de votre programme.

Déposer votre travail sur l'ENT.

Exercice n° 1

1. Écrire une procédure anonyme PL/SQL qui permet de demander à un utilisateur de saisir deux entiers et d'afficher leur somme.
2. Écrire une procédure anonyme PL/SQL qui permet de demander à un utilisateur de saisir un nombre et d'afficher sa table de multiplication.
3. Écrire une fonction récursive qui permet de retourner x^n , x et n sont deux entiers positifs.
4. Écrire une procédure anonyme PL/SQL qui calcule la factorielle d'un nombre strictement positif saisi par l'utilisateur. Le résultat sera stocké dans une table *resultatFactoriel*.
5. Modifier le programme précédent pour qu'il calcule et stocke dans une table *resultatsFactoriels* les factorielles des 20 premiers nombres entiers.

Exercice n° 2

Soit le schéma de la base de données de gestion des employés, constituée d'une seule table *employe*. Créer la table et y insérer quelques données. Tester vos méthodes au fur et à mesure.

```
1 CREATE TABLE emp
2 ( matr NUMBER(10) NOT NULL,
3   nom VARCHAR2(50) NOT NULL,
4   sal number(7,2),
5   adresse VARCHAR2(96),
6   dep NUMBER(10) NOT NULL,
7   CONSTRAINT emp_pk PRIMARY KEY (matr)
8 );
```

1. Écrire un bloc anonyme qui permet d'insérer un nouveau employé, dont les valeurs des attributs matr, nom, sal, adresse et dep sont respectivement 4, Youcef, 2500, avenue de la République, 92002.

```
1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_employe emp%ROWTYPE;
4
5 BEGIN
6   v_employe.matr := 4;
7   v_employe.nom := 'Youcef';
8   v_employe.sal := 2500;
9   v_employe.adresse := 'avenue Anatole France';
10  v_employe.dep := 92000;
11  -- Insertion d'un tuple dans la base
12  INSERT into emp VALUES v_employe;
13 END;
```

2. Écrire un bloc anonyme qui permet de supprimer tous les clients dont le *dep* est connu, et utiliser la fonction *ROWCOUNT* pour afficher le nombre de n-uplets supprimés.

```
1 SET SERVEROUTPUT ON;
2 DECLARE
3   v_nb_lignes NUMBER;
4 BEGIN
5   DELETE FROM emp WHERE dep = 10;
6   v_nb_lignes := SQL%ROWCOUNT;
7   dbms_output.put_line('v_nb_lignes : ' || v_nb_lignes);
8
9 END;
```

3. Écrire un bloc anonyme qui permet d'afficher la sommes des salaires de tous les employés, en utilisant un curseur explicite et la boucle *LOOP AND LOOP*. Cette procédure doit vous afficher le même résultat que la requête suivante :

```
1 SELECT SUM(salaire) FROM emp;

1 DECLARE
2   v_salaire EMP.sal%TYPE;
3   v_total   EMP.sal%TYPE := 0;
4   CURSOR c_salaires IS
5     SELECT sal
6     FROM emp;
7 BEGIN
8   OPEN c_salaires;
9
10  LOOP
11    FETCH c_salaires INTO v_salaire;
12    EXIT WHEN c_salaires%NOTFOUND;
13    IF v_salaire IS NOT NULL THEN
14      v_total := v_total + v_salaire;
15    END IF;
16  END LOOP;
17
18  CLOSE c_salaires;
19  dbms_output.put_line('total : ' || v_total);
20 END;
```

4. Modifier la procédure précédente pour calculer le salaire moyen. Cette procédure doit vous afficher le même résultat que la requête suivante :

```
1 SELECT AVG(sal) FROM emp;
```

5. Modifier les deux procédures précédentes, en utilisant la boucle *FOR IN*.
6. Écrire une procédure anonyme qui permet d'afficher les noms des employés du département 92000 et 75000, en utilisant un curseur paramétré.

```
1
2 DECLARE
3   CURSOR c(p_dep EMP.dep%TYPE) IS
4     SELECT dep, nom
5     FROM emp
6     WHERE dep = p_dep;
7 BEGIN
8   FOR v_employe IN c(92000) LOOP
9     dbms_output.put_line('Dep 1 : ' || v_employe.nom);
10  END LOOP;
11  FOR v_employe IN c(75000) LOOP
12    dbms_output.put_line('Dep 2 : ' || v_employe.nom);
13  END LOOP;
14 END;
```

Exercice n° 3

Nous avons défini, en cours, la spécification du package de gestion des clients : deux procédures qui permettent d'ajouter un client. Ces deux procédures portent le même nom, car la surcharge en PL/SQL est possible. Implémenter cette spécification et lever toutes les exceptions.