Compte Rendu TP 4

Exercice 1 :

1.

```
DECLARE
  v_nombre1 INTEGER;
  v_nombre2 INTEGER;
  v_somme   INTEGER;
BEGIN
  v_nombre1 := &Entrez_le_premier_entier;

  v_nombre2 := &Entrez_le_deuxieme_entier;

  v_somme := v_nombre1 + v_nombre2;

  DBMS_OUTPUT.PUT_LINE('La somme de ' || v_nombre1 || ' et ' || v_nombre2 || ' est : ' || v_somme);
END;
```

L'exécution :

```
Output:

Enter value for entrez_le_premier_entier: old   6:   v_nombre1 := &Entrez_le_premier_entier;
new   6:   v_nombre1 := 1;
Enter value for entrez_le_deuxieme_entier: old   8:   v_nombre2 := &Entrez_le_deuxieme_entier;
new   8:   v_nombre2 := 2;
La somme de 1 et 2 est : 3
```

2.

```
DECLARE
  v_nombre INTEGER;
BEGIN
  v_nombre := &Entrez_un_nombre;

  DBMS_OUTPUT.PUT_LINE('Table de multiplication de ' || v_nombre || ':');

  FOR i IN 1..10 LOOP
    DBMS_OUTPUT.PUT_LINE(v_nombre || ' x ' || i || ' = ' || (v_nombre * i));
  END LOOP;
END;
/
```

L'exécution :

```
5

Output:

Enter value for entrez_un_nombre: old   5:   v_nombre := &Entrez_un_nombre;
new   5:   v_nombre := 5;
Table de multiplication de 5:
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
```

3.

La fonction :

```
CREATE OR REPLACE FUNCTION puissance_recursive (x IN INTEGER, n IN INTEGER)
RETURN INTEGER
IS
BEGIN
  IF n = 0 THEN
    RETURN 1;
  ELSIF n > 0 THEN
    RETURN x * puissance_recursive(x, n - 1);
  ELSE
    RAISE_APPLICATION_ERROR(-20001, 'L''exposant n doit être un entier positif.');
  END IF;
END;
/
```

L'appel à la fonction :

```
DECLARE
  v_base INTEGER;
  v_exposant INTEGER;
  v_resultat INTEGER;
BEGIN
  v_base := &Entrez_la_base;

  v_exposant := &Entrez_l_exposant;

  v_resultat := puissance_recursive(v_base, v_exposant);

  DBMS_OUTPUT.PUT_LINE(v_base || '^' || v_exposant || ' = ' || v_resultat);
END;
/

SET SERVEROUTPUT ON;
```

Le résultat :

```
Output:

Enter value for entrez_la_base: old   7:   v_base := &Entrez_la_base;
new   7:   v_base := 5;
Enter value for entrez_l_exposant: old  10:   v_exposant := &Entrez_l_exposant;
new  10:   v_exposant := 2;
5^2 = 25
```

4.

Création de la table des résultats :

```
CREATE TABLE IF NOT EXISTS resultatFactoriel (
    nombre INTEGER PRIMARY KEY,
    factorielle NUMBER
);
```

Procédure anonyme :

```
DECLARE
    v_nombre INTEGER;
    v_factorielle NUMBER := 1; -- Initialisation à 1 pour la multiplication
    v_compteur INTEGER;
BEGIN

    v_nombre := &Entrez_un_nombre_strictement_positif;


    IF v_nombre <= 0 THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : Veuillez saisir un nombre strictement positif.');
    ELSE

        FOR v_compteur IN 1..v_nombre LOOP
            v_factorielle := v_factorielle * v_compteur;
        END LOOP;


        INSERT INTO resultatFactoriel (nombre, factorielle)
        VALUES (v_nombre, v_factorielle);

        DBMS_OUTPUT.PUT_LINE('La factorielle de ' || v_nombre || ' est : ' || v_factorielle);
        DBMS_OUTPUT.PUT_LINE('Le résultat a été stocké dans la table resultatFactoriel.');
    END IF;
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Une erreur est survenue : ' || SQLERRM);
END;
/
```

Le résultat :

```
  5

Output:

Enter value for entrez_un_nombre_strictement_positif: old   7:
new   7:     v_nombre := 5;
La factorielle de 5 est : 120
```

5.

Dans cette question, on va insérer un ensemble de résultats :

```
CREATE TABLE IF NOT EXISTS resultatsFactoriels (
    nombre INTEGER PRIMARY KEY,
    factorielle NUMBER
);

DECLARE
    v_nombre INTEGER;
    v_factorielle NUMBER;
BEGIN
    FOR v_nombre IN 1..20 LOOP
        v_factorielle := 1;
        FOR i IN 1..v_nombre LOOP
            v_factorielle := v_factorielle * i;
        END LOOP;
        INSERT INTO resultatsFactoriels (nombre, factorielle)
        VALUES (v_nombre, v_factorielle);
        DBMS_OUTPUT.PUT_LINE('Factorielle de ' || v_nombre || ' : ' || v_factorielle);
    END LOOP;
    DBMS_OUTPUT.PUT_LINE('Factorielles stockées.');
EXCEPTION
    WHEN OTHERS THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : ' || SQLERRM);
END;
/

SET SERVEROUTPUT ON;
```

Résultat de l'exécution :

```
Output:

Factorielle de 1 : 1
Factorielle de 2 : 2
Factorielle de 3 : 6
Factorielle de 4 : 24
Factorielle de 5 : 120
Factorielle de 6 : 720
Factorielle de 7 : 5040
Factorielle de 8 : 40320
Factorielle de 9 : 362880
Factorielle de 10 : 3628800
Factorielle de 11 : 39916800
Factorielle de 12 : 479001600
Factorielle de 13 : 6227020800
Factorielle de 14 : 87178291200
Factorielle de 15 : 1307674368000
Factorielle de 16 : 20922789888000
Factorielle de 17 : 355687428096000
Factorielle de 18 : 6402373705728000
Factorielle de 19 : 121645100408832000
Factorielle de 20 : 2432902008176640000
```

Exercice 2 :

On commence par créer une table employe :

```sql
CREATE TABLE emp (
    matr INT(10) NOT NULL PRIMARY KEY,
    nom VARCHAR(50) NOT NULL,
    sal DECIMAL(7, 2),
    adresse VARCHAR(96),
    dep INT(10) NOT NULL
);
```

1.

Maintenant on va écrire une procédure anonyme qui va insérer un n-uplet:

```sql
SET SERVEROUTPUT ON;
DECLARE
  v_employe emp%ROWTYPE;
BEGIN
  v_employe.matr := 4;
  v_employe.nom := 'Youcef';
  v_employe.sal := 2500;
  v_employe.adresse := 'avenue de la Republique';
  v_employe.dep := 92002;
  INSERT INTO emp VALUES v_employe;
  COMMIT;
  DBMS_OUTPUT.PUT_LINE('Employé inséré avec succès.');
END;
/
```

Lors de l'exécution :

```
Employé inséré avec succès.

PL/SQL procedure successfully completed.
```

2. On a écrit une procédure qui supprime les employés du département = 10 :

```
SET SERVEROUTPUT ON;
DECLARE
  v_nb_lignes NUMBER;
BEGIN
  DELETE FROM emp WHERE dep IS NOT NULL;
  v_nb_lignes := SQL%ROWCOUNT;
  dbms_output.put_line('Nombre de lignes supprimées : ' || v_nb_lignes);
END;
/
```

Le résultat :

```
Nombre de lignes supprimées : 1

PL/SQL procedure successfully completed.
```

3.

```
DECLARE
  v_salaire EMP.sal%TYPE;
  v_total EMP.sal%TYPE := 0;
  CURSOR c_salaires IS
    SELECT sal
    FROM emp;
BEGIN
  OPEN c_salaires;
  LOOP
    FETCH c_salaires INTO v_salaire;
    EXIT WHEN c_salaires%NOTFOUND;
    IF v_salaire IS NOT NULL THEN
      v_total := v_total + v_salaire;
    END IF;
  END LOOP;
  CLOSE c_salaires;
  dbms_output.put_line('Somme des salaires : ' || v_total);
END;
/
```

Le résultat de l'exécution :

```
Somme des salaires : 2500.00

PL/SQL procedure successfully completed.
```

4.La procédure suivante calcule la moyenne des

```
DECLARE
  v_salaire EMP.sal%TYPE;
  v_total EMP.sal%TYPE := 0;
  v_count NUMBER := 0;
  v_moyenne EMP.sal%TYPE;
  CURSOR c_salaires IS
    SELECT sal
    FROM emp;
BEGIN
  OPEN c_salaires;
  LOOP
    FETCH c_salaires INTO v_salaire;
    EXIT WHEN c_salaires%NOTFOUND;
    IF v_salaire IS NOT NULL THEN
      v_total := v_total + v_salaire;
      v_count := v_count + 1;
    END IF;
  END LOOP;
  CLOSE c_salaires;
  IF v_count > 0 THEN
    v_moyenne := v_total / v_count;
    dbms_output.put_line('Moyenne des salaires : ' || v_moyenne);
  ELSE
    dbms_output.put_line('Aucun salaire trouvé.');
  END IF;
END;
/
```

Le résultat : ça nous a donné le même résultat car on n’a qu' une ligne.

```
Moyenne des salaires : 2500.00

PL/SQL procedure successfully completed.
```

5.
Utilisation de FOR IN :
La somme

```
DECLARE
  v_total EMP.sal%TYPE := 0;
  CURSOR c_salaires IS
    SELECT sal
    FROM emp;
BEGIN
  FOR emp_rec IN c_salaires LOOP
    IF emp_rec.sal IS NOT NULL THEN
      v_total := v_total + emp_rec.sal;
    END IF;
  END LOOP;
  dbms_output.put_line('Somme des salaires (FOR IN) : ' || v_total);
END;
/
```

La moyenne

```
DECLARE
  v_total EMP.sal%TYPE := 0;
  v_count NUMBER := 0;
  v_moyenne EMP.sal%TYPE;
  CURSOR c_salaires IS
    SELECT sal
    FROM emp;
BEGIN
  FOR emp_rec IN c_salaires LOOP
    IF emp_rec.sal IS NOT NULL THEN
      v_total := v_total + emp_rec.sal;
      v_count := v_count + 1;
    END IF;
  END LOOP;
  IF v_count > 0 THEN
    v_moyenne := v_total / v_count;
    dbms_output.put_line('Moyenne des salaires (FOR IN) : ' || v_moyenne);
  ELSE
    dbms_output.put_line('Aucun salaire trouvé.');
  END IF;
END;
/
```

6.

Utilisation de curseur paramétré :

```
DECLARE
  CURSOR c_employes_dep (p_dep EMP.dep%TYPE) IS
    SELECT nom
    FROM emp
    WHERE dep = p_dep;
BEGIN
  FOR v_nom IN c_employes_dep(92002) LOOP
    dbms_output.put_line('Département 92002 : ' || v_nom.nom);
  END LOOP;
  FOR v_nom IN c_employes_dep(75000) LOOP
    dbms_output.put_line('Département 75000 : ' || v_nom.nom);
  END LOOP;
END;
/
```