

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

Nombre de pages : 12

### Exercice 1 :

#### 1.1.

```

exercice1.sql x  exercice2.sql
TP4 > exercice1.sql > ...
1  --Exercice 1
2  SET SERVEROUTPUT ON;
3
4  DECLARE
5      -- 1.1 : Somme de deux entiers
6      PROCEDURE calcul_somme IS
7          a NUMBER := &entier_a;
8          b NUMBER := &entier_b;
9      BEGIN
10         DBMS_OUTPUT.PUT_LINE('La somme de ' || a || ' et ' || b || ' est : ' || (a + b));
11     END calcul_somme;
12
13 BEGIN
14     calcul_somme;
15 END;
16 /

```

```

SQL> @C:\Users\cyryn\OneDrive\Desktop\BDA\TP4\exercice1.sql
Entrez une valeur pour entier_a : 5
ancien 4 :          a NUMBER := &entier_a;
nouveau 4 :          a NUMBER := 5;
Entrez une valeur pour entier_b : 3
ancien 5 :          b NUMBER := &entier_b;
nouveau 5 :          b NUMBER := 3;
La somme de 5 et 3 est : 8

Procédure PL/SQL terminée avec succès.

SQL>

```

#### 1.2.

```

TP4 > exercice1.sql > ...
1  --Exercice 1
2  SET SERVEROUTPUT ON;
3
4  DECLARE
5      -- 1.2 : Table de multiplication
6      PROCEDURE table_multiplication IS
7          nombre NUMBER := &saisir_nombre;
8      BEGIN
9          DBMS_OUTPUT.PUT_LINE('Table de multiplication de ' || nombre || ' :');
10         FOR i IN 1..10 LOOP
11             DBMS_OUTPUT.PUT_LINE(nombre || ' x ' || i || ' = ' || (nombre * i));
12         END LOOP;
13     END table_multiplication;
14 BEGIN
15     table_multiplication;
16 END;
17 /

```

```

SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice1.sql
Entrez une valeur pour saisir_nombre : 7
ancien 4 :      nombre NUMBER := &saisir_nombre;
nouveau 4 :      nombre NUMBER := 7;
Table de multiplication de 7 :
7 x 1 = 7
7 x 2 = 14
7 x 3 = 21
7 x 4 = 28
7 x 5 = 35
7 x 6 = 42
7 x 7 = 49
7 x 8 = 56
7 x 9 = 63
7 x 10 = 70

Procédure PL/SQL terminée avec succès.

```

### 1.3.

```

TP4 > exercice1.sql > ...
1  --Exercice 1
2  SET SERVEROUTPUT ON;
3
4  DECLARE
5      v_base NUMBER;
6      v_exposant NUMBER;
7      v_resultat NUMBER;
8
9      FUNCTION puissance(x NUMBER, n NUMBER) RETURN NUMBER IS
10     BEGIN
11         IF n = 0 THEN
12             RETURN 1;
13         ELSE
14             RETURN x * puissance(x, n-1);
15         END IF;
16     END puissance;
17
18
19     BEGIN
20         v_base := &entrez_base;
21         v_exposant := &entrez_exposant;
22
23         v_resultat := puissance(v_base, v_exposant);
24
25         DBMS_OUTPUT.PUT_LINE(v_base || '^' || v_exposant || ' = ' || v_resultat);
26     END;
27 /

```

```
SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice1.sql
Entrez une valeur pour entrez_base : 5
ancien 17 :      v_base := &entrez_base;
nouveau 17 :      v_base := 5;
Entrez une valeur pour entrez_exposant : 3
ancien 18 :      v_exposant := &entrez_exposant;
nouveau 18 :      v_exposant := 3;
5^3 = 125

Procédure PL/SQL terminée avec succès.

SQL>
```

## 1.4.

```
TP4 > exercice1.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_nombre NUMBER := &saisir_nombre;
5      v_result NUMBER := 1;
6      v_table_existe NUMBER;
7
8      e_nombre_negatif EXCEPTION;
9      PRAGMA EXCEPTION_INIT(e_nombre_negatif, -20001);
10
11 BEGIN
12     IF v_nombre < 0 THEN
13         RAISE e_nombre_negatif;
14     END IF;
15
16     BEGIN
17         SELECT 1 INTO v_table_existe
18         FROM user_tables
19         WHERE table_name = 'RESULTATFACTORIEL';
20
21         DBMS_OUTPUT.PUT_LINE('Table existante détectée.');

```

```

SQL> @C:\Users\cyryn\OneDrive\Desktop\BDA\TP4\exercice1.sql
Entrez une valeur pour saisir_nombre : 5
ancien 3 :      v_nombre NUMBER := &saisir_nombre;
nouveau 3 :      v_nombre NUMBER := 5;
Table existante détectée.
5! = 120
Résultat stocké avec succès.

Procédure PL/SQL terminée avec succès.

SQL>

```

```
SQL> select * from resultatFactoriel;
```

```

      NOMBRE FACTORIELLE
-----
DATE_CALCUL
-----
          5          120
13/04/25 20:52:43,156000

```

```
SQL> |
```

## 1.5.

```

1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_result NUMBER;
5      v_table_existe NUMBER;
6
7      FUNCTION calcul_factorielle(n NUMBER) RETURN NUMBER IS
8          v_res NUMBER := 1;
9      BEGIN
10         FOR i IN 1..n LOOP
11             v_res := v_res * i;
12         END LOOP;
13         RETURN v_res;
14     END calcul_factorielle;
15
16 BEGIN
17     BEGIN
18         SELECT 1 INTO v_table_existe
19         FROM user_tables
20         WHERE table_name = 'RESULTATSFACIORELS';
21
22         DBMS_OUTPUT.PUT_LINE('Table existante détectée.');

```

```

35     FOR n IN 1..20 LOOP
36         v_result := calcul_factorielle(n);
37
38         BEGIN
39             INSERT INTO resultatsFactoriels(nombre, factorielle)
40             VALUES (n, v_result);
41
42             DBMS_OUTPUT.PUT_LINE(n || '!' = ' || v_result);
43         EXCEPTION
44             WHEN DUP_VAL_ON_INDEX THEN
45                 UPDATE resultatsFactoriels
46                 SET factorielle = v_result,
47                     date_calcul = SYSTIMESTAMP
48                 WHERE nombre = n;
49
50                 DBMS_OUTPUT.PUT_LINE(n || '!' = ' || v_result || ' (mis à jour)');
51         END;
52     END LOOP;
53
54     COMMIT;
55     DBMS_OUTPUT.PUT_LINE('Toutes les factorielles ont été calculées et stockées.');
```

SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice1.sql

Table existante détectée.

Calcul des factorielles de 1 à 20:

1! = 1

2! = 2

3! = 6

4! = 24

5! = 120

6! = 720

7! = 5040

8! = 40320

9! = 362880

10! = 3628800

11! = 39916800

12! = 479001600

13! = 6227020800

14! = 87178291200

15! = 1307674368000

16! = 20922789888000

17! = 355687428096000

18! = 6402373705728000

19! = 121645100408832000

20! = 2432902008176640000

Toutes les factorielles ont été calculées et stockées.

Procédure PL/SQL terminée avec succès.

## Exercice 2 :

1.

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_employe emp%ROWTYPE;
5
6  BEGIN
7      v_employe.matr := 4;
8      v_employe.nom := 'Youcef';
9      v_employe.sal := 2500;
10     v_employe.adresse := 'avenue Anatole France';
11     v_employe.dep := 92000;
12
13     INSERT INTO emp VALUES v_employe;
14     COMMIT;
15
16     DBMS_OUTPUT.PUT_LINE('Employé ajouté avec succès : ' || v_employe.nom);
17
18  EXCEPTION
19      WHEN DUP_VAL_ON_INDEX THEN
20          DBMS_OUTPUT.PUT_LINE('Erreur : Matricule ' || v_employe.matr || ' existe déjà');
21          ROLLBACK;
22      WHEN OTHERS THEN
23          DBMS_OUTPUT.PUT_LINE('Erreur : ' || SQLERRM);
24          ROLLBACK;
25  END;
26  /

```

```

SQL> desc emp;
  Nom                                     NULL ?   Type
-----
MATR                                     NOT NULL NUMBER(10)
NOM                                     NOT NULL VARCHAR2(50)
SAL                                     NOT NULL NUMBER(7,2)
ADRESSE                                NOT NULL VARCHAR2(96)
DEP                                     NOT NULL NUMBER(10)

SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
Employé ajouté avec succès : Youcef

Procédure PL/SQL terminée avec succès.

SQL>

```

2.

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_dep NUMBER := &saisir_departement;
5      v_nb_lignes NUMBER;
6  BEGIN
7      DELETE FROM emp WHERE dep = v_dep;
8
9      v_nb_lignes := SQL%ROWCOUNT;
10     COMMIT;
11
12     DBMS_OUTPUT.PUT_LINE('Nombre d employés supprimés du département ' || v_dep || ' : ' || v_nb_lignes);
13
14 END;
15 /

```

```

SQL> @C:\Users\cyrrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
Entrez une valeur pour saisir_departement : 10
ancien      2 :      v_dep NUMBER := &saisir_departement;
nouveau    2 :      v_dep NUMBER := 10;
Nombre d employé@s supprimé@s du d@partement 10 : 0

Procédure PL/SQL terminée avec succès.

```

3.

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_salaire EMP.sal%TYPE;
5      v_total NUMBER := 0;
6      CURSOR c_salaires IS
7      SELECT sal
8      FROM emp;
9  BEGIN
10     OPEN c_salaires;
11
12     LOOP
13         FETCH c_salaires INTO v_salaire;
14         EXIT WHEN c_salaires%NOTFOUND;
15
16         v_total := v_total + NVL(v_salaire, 0);
17     END LOOP;
18
19     CLOSE c_salaires;
20     dbms_output.put_line('total : ' || v_total);
21 EXCEPTION
22     WHEN OTHERS THEN
23         IF c_salaires%ISOPEN THEN
24             CLOSE c_salaires;
25         END IF;
26         dbms_output.put_line('Erreur : ' || SQLERRM);
27 END;
28 /

```

```

SQL> desc emp;
Nom                                NULL ?    Type
-----
MATR                                NOT NULL NUMBER(10)
NOM                                NOT NULL VARCHAR2(50)
SAL                                NOT NULL  NUMBER(7,2)
ADRESSE                             VARCHAR2(96)
DEP                                NOT NULL  NUMBER(10)

SQL> select sum(sal) from emp;

SUM(SAL)
-----
      2500

SQL> @C:\Users\cyrrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
total : 2500

Procédure PL/SQL terminée avec succès.

SQL>

```

## 4.

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2  DECLARE
3      v_salaire EMP.sal%TYPE;
4      v_total  NUMBER := 0;
5      v_count  NUMBER := 0;
6      v_moyenne NUMBER;
7      CURSOR c_salaires IS
8      SELECT sal
9      FROM emp;
10 BEGIN
11     OPEN c_salaires;
12
13     LOOP
14         FETCH c_salaires INTO v_salaire;
15         EXIT WHEN c_salaires%NOTFOUND;
16         IF v_salaire IS NOT NULL THEN
17             v_total := v_total + v_salaire;
18             v_count := v_count + 1;
19         END IF;
20     END LOOP;
21
22     CLOSE c_salaires;
23
24     IF v_count > 0 THEN
25         v_moyenne := v_total / v_count;
26         dbms_output.put_line('Salaire moyen : ' || ROUND(v_moyenne, 2));
27     ELSE
28         dbms_output.put_line('Aucun salaire valide à calculer');
29     END IF;
30
31 EXCEPTION
32     WHEN OTHERS THEN
33         IF c_salaires%ISOPEN THEN
34             CLOSE c_salaires;
35         END IF;
36         dbms_output.put_line('Erreur : ' || SQLERRM);
37 END;

```

```
SQL> select avg(sal) from emp;
```

```

      AVG(SAL)
-----
        2500

```

```
SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
Salaire moyen : 2500
```

```
Procédure PL/SQL terminée avec succès.
```

```
SQL>
```



## 5.

## Première procédure avec la boucle FOR .. IN

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_total NUMBER := 0;
5  BEGIN
6      FOR emp_rec IN (SELECT sal FROM emp WHERE sal IS NOT NULL) LOOP
7          v_total := v_total + emp_rec.sal;
8      END LOOP;
9
10     DBMS_OUTPUT.PUT_LINE('Somme totale des salaires : ' || v_total);
11 EXCEPTION
12     WHEN OTHERS THEN
13         DBMS_OUTPUT.PUT_LINE('Erreur : ' || SQLERRM);
14 END;
15 /

```

```

SQL> @C:\Users\cyryn\OneDrive\Desktop\BDA\TP4\exercice2.sql
Somme totale des salaires : 2500

```

Procédure PL/SQL terminée avec succès.

```
SQL>
```

## Deuxième procédure avec la boucle FOR .. IN

```

TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      v_total NUMBER := 0;
5      v_count NUMBER := 0;
6      v_moyenne NUMBER;
7  BEGIN
8      FOR emp_rec IN (SELECT sal FROM emp WHERE sal IS NOT NULL) LOOP
9          v_total := v_total + emp_rec.sal;
10         v_count := v_count + 1;
11     END LOOP;
12
13     IF v_count > 0 THEN
14         v_moyenne := v_total / v_count;
15         DBMS_OUTPUT.PUT_LINE('Salaire moyen : ' || ROUND(v_moyenne, 2));
16     ELSE
17         DBMS_OUTPUT.PUT_LINE('Aucun salaire valide à calculer');
18     END IF;
19 EXCEPTION
20     WHEN OTHERS THEN
21         DBMS_OUTPUT.PUT_LINE('Erreur : ' || SQLERRM);
22 END;
23 /

```

```
SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
Salaire moyen : 2500

Procédure PL/SQL terminée avec succès.

SQL>
```

## 6.

```
TP4 > exercice2.sql > ...
1  SET SERVEROUTPUT ON;
2
3  DECLARE
4      -- Curseur paramétré
5      CURSOR c_emp(p_deptno NUMBER) IS
6          SELECT nom
7          FROM emp
8          WHERE dep = p_deptno;
9  BEGIN
10     -- Affiche les employés du département 92000
11     DBMS_OUTPUT.PUT_LINE('Employés du département 92000:');
12     FOR emp_rec IN c_emp(92000) LOOP
13         DBMS_OUTPUT.PUT_LINE('- ' || emp_rec.nom);
14     END LOOP;
15
16     -- Affiche les employés du département 75000
17     DBMS_OUTPUT.PUT_LINE('Employés du département 75000:');
18     FOR emp_rec IN c_emp(75000) LOOP
19         DBMS_OUTPUT.PUT_LINE('- ' || emp_rec.nom);
20     END LOOP;
21 END;
22 /
```

```
SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice2.sql
Employés du département 92000:
- Youcef
Employés du département 75000:

Procédure PL/SQL terminée avec succès.

SQL>
```

## Exercise 3 :

```

TP4 > exercice3.sql > {} gestion_clients
1  CREATE OR REPLACE PACKAGE gestion_clients AS
2      PROCEDURE ajouter_client(
3          p_id IN clients.id%TYPE,
4          p_nom IN clients.nom%TYPE,
5          p_email IN clients.email%TYPE
6      );
7
8      PROCEDURE ajouter_client(
9          p_client_rec IN clients%ROWTYPE
10     );
11 END gestion_clients;
12 /
13
14 CREATE OR REPLACE PACKAGE BODY gestion_clients AS
15     PROCEDURE ajouter_client(
16         p_id IN clients.id%TYPE,
17         p_nom IN clients.nom%TYPE,
18         p_email IN clients.email%TYPE
19     ) IS
20     BEGIN
21         INSERT INTO clients(id, nom, email)
22         VALUES(p_id, p_nom, p_email);
23
24         COMMIT;
25         DBMS_OUTPUT.PUT_LINE('Client ajouté (version 1)');
26     EXCEPTION
27         WHEN DUP_VAL_ON_INDEX THEN
28             RAISE_APPLICATION_ERROR(-20001, 'ID client déjà existant');
29         WHEN OTHERS THEN
30             RAISE_APPLICATION_ERROR(-20002, 'Erreur ajout client: ' || SQLERRM);
31     END ajouter_client;
32
33     PROCEDURE ajouter_client(
34         p_client_rec IN clients%ROWTYPE
35     ) IS
36     BEGIN
37         INSERT INTO clients VALUES p_client_rec;
38
39         COMMIT;
40         DBMS_OUTPUT.PUT_LINE('Client ajouté (version 2)');
41     EXCEPTION
42         WHEN DUP_VAL_ON_INDEX THEN
43             RAISE_APPLICATION_ERROR(-20001, 'ID client déjà existant');
44         WHEN OTHERS THEN
45             RAISE_APPLICATION_ERROR(-20003, 'Erreur ajout client (record): ' || SQLERRM);
46     END ajouter_client;
47 END gestion_clients;
48 /

```

```
SQL> @C:\Users\cyrin\OneDrive\Desktop\BDA\TP4\exercice3.sql  
Procédure PL/SQL terminée avec succès.  
  
Package créé.  
  
Corps de package créé.  
  
Pas d'erreur.  
SQL>
```