B42 – PL/SQL

Bloc anonyme : exécuté 1 fois

Pour exécuter du PL/SQL, utiliser **F5** (Le script)

```
DECLARE                                                    -- La section déclarative
        v_dept departement.nom%TYPE;
        v_ville departement.ville%TYPE;
BEGIN                                                      -- La section exécutable
        SELECT
                nom, ville  INTO v_dept, v_ville
        FROM
                departement
          WHERE
                id= 10;
END;
```

Pour faire des commentaires:
-- sur une ligne
/* sur plusieurs lignes physiques */

Affichage à l'écran

```
set serveroutput on;

-- Dans une fonction/procédure PL/SQL
DBMS_OUTPUT.PUT_LINE ('Résultat : Bonjour');
```

IF/ELSE

```
        …
        IF    LENGTH(v_dept) > 8 THEN
                DBMS_OUTPUT.PUT_LINE (Grand');
        ELSIF LENGTH(v_dept) > 4 THEN
                DBMS_OUTPUT.PUT_LINE ('Moyen');
        ELSE
                DBMS_OUTPUT.PUT_LINE ('Petit');
        END IF;
        …
```

Assigner des valeurs

```
DECLARE
        resultat number(2);
BEGIN
…
        IF    LENGTH(v_dept) > 8 THEN
              DBMS_OUTPUT.PUT_LINE (Grand');
                 resultat = 1;
        ELSIF LENGTH(v_dept) > 4 THEN
              DBMS_OUTPUT.PUT_LINE ('Moyen');
                 resultat = 2;
        ELSE
              DBMS_OUTPUT.PUT_LINE ('Petit');
                 resultat = 3;
        END IF;

        DBMS_OUTPUT.PUT_LINE (resultat);
…
```

Les exceptions

Ce PL/SQL lance une erreur, car aucune donnée n'est retournée au variable (aucun département #100).

```
DECLARE                                                 -- La section déclarative
        v_dept departement.nom%TYPE;
        v_ville departement.ville%TYPE;
BEGIN                                                   -- La section exécutable
        SELECT
                nom, ville  INTO v_dept, v_ville
        FROM
               departement
          WHERE
                id= 100;
END;
```

Pour faire la gestion d'erreurs :

```
…
EXCEPTION
        WHEN NO_DATA_FOUND THEN
              DBMS_OUTPUT.PUT_LINE (‘Aucun département trouvé’);
        WHEN TOO_MANY_ROW THEN
              DBMS_OUTPUT.PUT_LINE (‘Il y a plus de 1 département de trouvé’);
END;
```

Déclarer ses propres gestions

```
DECLARE
        monTypeException  EXCEPTION;
…
        IF LENGTH (v_dept) > 2 THEN
                RAISE monTypeException;
…
EXCEPTION
        WHEN monTypeException THEN
                DBMS_OUTPUT.PUT_LINE ('je gère ma propre exception');
END;
```

Les procédures et fonctions

```
CREATE OR REPLACE PROCEDURE afficher
IS
        v_dept departement.nom%TYPE;
        v_ville departement.ville%TYPE;
BEGIN                                                   -- La section exécutable
        SELECT
                nom, ville  INTO v_dept, v_ville
        FROM
               departement
          WHERE
                id= 10;

        DBMS_OUTPUT.PUT_LINE ('Résultat : '  || v_dept || ' ' || v_ville);
END;
```

Note :

- On ne met pas de () à la déclaration de la procédure s'il n'y a pas de paramètre.
- Pour faire afficher les erreurs : CTRL+Shift+L (Affichage → journal) ou SHOW errors dans SQLPLUS.
- Pour l'exécuter : **execute afficher();**

```
CREATE OR REPLACE FUNCTION analyser (p_id number) RETURN number
IS
 resultat NUMBER;
BEGIN
      SELECT
         avg(salaire) INTO resultat
      FROM
        employe
 WHERE
       id_departement = p_id;

 RETURN resultat;
END;
```

Exécution d'une fonction (à partir d'une procédure/fonction/SQL/programme)

```
set serveroutput on;

VARIABLE v_montant NUMBER;
EXECUTE :v_montant := analyser (10);
select :v_montant from dual;

ou

SELECT
        nom,
        analyser(id)                    -- fonction déclarée dans un autre exemple
FROM
        departement;
```

Les boucles

```
DECLARE                                         -- La section déclarative
        v_dept departement.nom%TYPE;
        v_ville departement.ville%TYPE;
        v_id NUMBER(2);
BEGIN                                           -- La section exécutable
        v_id := 0;

        LOOP
           v_id := v_id + 10;

           SELECT
                nom, ville  INTO v_dept, v_ville
           FROM
               departement
             WHERE
                id= v_id;

           DBMS_OUTPUT.PUT_LINE ('Résultat : '  || v_dept || ' ' || v_ville);
           EXIT WHEN LENGTH(v_ville) > 8;
        END LOOP;
END;
```

Boucle avec condition (FOR)

```
set serveroutput on;

DECLARE
        V_Limite NUMBER(2);
BEGIN
        V_Limite := 50;
        FOR V_Compteur IN 1 .. V_Limite LOOP
           DBMS_OUTPUT.PUT_LINE ('Résultat : ' || V_compteur);
        END LOOP;
END;
```

Les curseurs

L'utilisation d'un curseur nous permet de manipuler ligne par ligne le résultat d'une requête qui retourne plusieurs lignes.

Le curseur pointe sur la ligne courante de l'ensemble des lignes.

Mot clef CURSOR dans la section DECLARE d'un bloc d'instruction PL/SQL

```
set serveroutput on;

DECLARE                                          -- La section déclarative
        v_dept departement.nom%TYPE;
        v_ville departement.ville%TYPE;
        CURSOR cur_ligne IS
        SELECT
                nom, ville
        FROM
               departement;
BEGIN                                            -- La section exécutable
        OPEN cur_ligne;                          -- Ouverture du curseur

        LOOP
            FETCH cur_ligne INTO v_dept, v_ville; -- Assigner la ligne courante dans les variables
            EXIT WHEN  cur_ligne%NOTFOUND;

            DBMS_OUTPUT.PUT_LINE ('Résultat : ' || v_ville);
        END LOOP;

        CLOSE cur_ligne;
END;
```