Université Constantine 2
جامعة قسنطينة 2

Développement Avancé d'Applications Web

– Cours / TP–

Langage PL/SQL

Dr Bouanaka Chafia

NTIC

chafia.bouanaka@univ-constantine2.dz

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
Nouvelles technologies	TLSI	Licence 3	Génie Logiciel (GL)

Langage PL/SQL

Définition

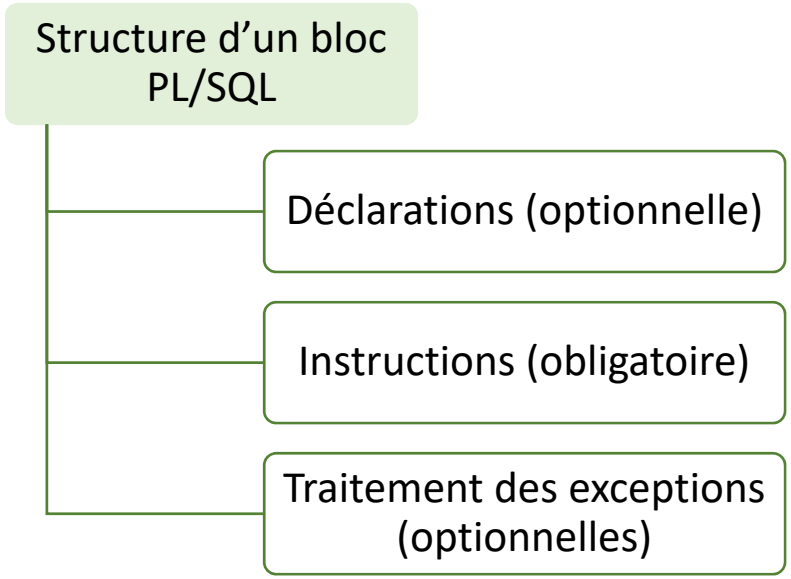
- PL/SQL est un langage procédural (**Procedural Language/SQL**) qui permet en plus de SQL d'avoir les mécanismes des langages algorithmiques.
- L'unité de programmation PL/SQL est **le bloc**.
- Un bloc contient des déclarations de variables ou de constantes et des instructions (qui peuvent être des commandes SQL).
- Des variables peuvent être utilisées pour mémoriser des résultats de requêtes.

Parmi les instructions d'un bloc PL/SQL on a:

- commandes SQL ,
- instructions de boucle (LOOP),
- instructions conditionnelles (IF-THEN- ELSE),
- traitement des exceptions

Langage PL/SQL

Structure d'un bloc PL/SQL



Syntaxe d'un bloc PL/SQL

```
[DECLARE
        <Constantes>
        <Variables>
        <Cursors>
        <Exceptions utilisateurs>]
BEGIN
        <Instruction PL/SQL>
        [EXCEPTION <Traitement d'exception>]
END;
```

Structure d'un bloc PL/SQL : Déclarations de Constantes

Déclarations : Constantes

```
DECLARE
    dateRecrutement date;   /* initialisation implicite avec null */
    nom varchar2(80) := 'Benali';
    trouve boolean;     /* initialisation implicite avec null aussi*/
    incrSalaire constant number(3,2) := 1.5;    /* constante */
```

Langage PL/SQL

Structure d'un bloc PL/SQL : Déclarations de variables

Variable de type colonne

```
DECLARE
   nomEtud etudiant.nom%TYPE;
```

- **nomEtud** est une variable PLSQL de même type que la colonne nom de la table **etudiant**.
- Utile pour garantir la compatibilité des affectations

Variable de type ligne

```
DECLARE
            etudiantRec etudiant%ROWTYPE;
```

etudiantRec est une structure de tuple de même type que le schéma de la relation **etudiant**.
Usage avec SELECT *

Structure d'un bloc PL/SQL : Déclarations de zone curseur

- Le mot **CURSOR** sert à déclarer une zone mémoire qui recevra le résultat d'un SELECT pour un
- parcours tuple par tuple.

Variable de curseur

```
DECLARE
   CURSOR etudCursor IS
   SELECT * FROM etudiant  WHERE ID = 123;
```

Si un CURSOR est utilisé pour mettre à jour un tuple de relation, on le signale avec FOR update en fin de déclaration

Variable de curseur

```
DECLARE
   CURSOR etudCursor IS
   SELECT * FROM etudiant  WHERE ID = 123;
   FOR update of etudiant
```

Langage PL/SQL

Structure d'un bloc PL/SQL : Instructions

- PL/SQL offre la plupart des constructions des langages de programmation:
  - affectation de variables,
  - structures de contrôle de boucle (LOOP)
  - et de teste (if-then- ELSE),
  - appels de procédure et de fonction, etc.
- Les instructions sont écrites entre BEGIN et END
- PL/SQL ne permet pas les commandes SQL de langage de définition comme la création de tab, modification d'attribut etc.
- PL/SQL permet tous les autres types de commandes SQL (insert, delete, update, commit ...)

Langage PL/SQL

Structure d'un bloc PL/SQL : Instruction

Instruction Select monotuple

```
DECLARE
  etudiantRec       etudiant%ROWTYPE;
  nomEtud           etudiant.nom%TYPE;
BEGIN
  SELECT * INTO     etudiantRec
  FROM etudiant     WHERE  ID = '6';
  dbms_output.putline(etudiantRec.nom||' '||etudiantRec.prenom);
END;
```

Langage PL/SQL

Structure d'un bloc PL/SQL : Structure IF

Structure de contrôle IF

IF <condition> THEN <séquence d'instructions>
[ELSIF <condition> THEN <séquence d'instructions>]
. . .
[ELSE <séquence d'instructions>]
END IF ;

Langage PL/SQL

Structure d'un bloc PL/SQL : Boucles

Boucle While

```
WHILE <condition> LOOP
  <séquence  d'instructions>;
END LOOP
```

```
DECLARE
 CURSOR    empCursor IS
           SELECT * FROM EMPLOYEE;
    employeeRec employee%ROWTYPE;
    maxSal employee.SALary%TYPE := 0;
BEGIN
   OPEN empCursor;
   LOOP
            /* Accès à chacun des tuples */
        FETCH empCursor INTO   employeeRec;
        EXIT WHEN empCursor%NOTFOUND;
             /* traitement du tuple */
 IF    maxSal < employeeRec.salary THEN
                  maxSal := employeeRec.salary;
        END IF;
             /* fin traitement tuple */

 END LOOP ;
      dbms_output.putline('Salaire Maximum: '|| maxsal);
      CLOSE empCursor;
END;
```

Calcul du salaire maximum en comparant les salaires de tous les employés

Structure d'un bloc PL/SQL : Boucles

Syntaxe de la boucle FOR :

FOR variableTuple **IN CURSOR**
LOOP
...
END LOOP

Boucle For

```
BEGIN
    FOR employeeRec IN  empCursor LOOP
        /* traitement du tuple */
          IF   maxSal < employeeRec.salary THEN
                    maxSal := employeeRec.salary;
          END IF;
          /* fin traitement tuple */
     END LOOP ;
    dbms_output.putline('Salaire Maximum: '|| maxsal);
END;
```