



Chapitre 3 : Introduction au Procedural Language for SQL (PL/SQL)

Tarek HAMROUNI
Maître de Conférences en Informatique
hamrouni.tarek@gmail.com

Présentation du PL/SQL

- **PL/SQL : Procedural Language for SQL**
- PL/SQL est propre à Oracle. D'autres variantes sont offertes pour d'autres SGBDs.
- PL/SQL est un langage procédural 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.

Présentation du PL/SQL

- PL/SQL n'est pas sensible à la casse.
- Parmi les instructions d'un bloc PL/SQL :
 - commandes SQL,
 - instructions conditionnelles (IF-THEN-ELSE, CASE),
 - instructions répétitives (LOOP, FOR, WHILE),
 - traitement des exceptions,
 - appels à d'autres blocs PL/SQL.
- Les instructions se terminent par « ; ».
- Un bloc PL/SQL peut-être un bloc **anonyme** ou bien **nommé** en tant qu'une **procédure**, une **fonction** ou un **déclencheur (trigger)**.

Structure d'un bloc PL/SQL

■ Un Bloc :

- une partie dédiée aux **déclarations** (**optionnelle**) : déclaration et initialisation de types, variables et constantes
- une partie dédiée aux **instructions d'exécution**,
- une partie (**optionnelle**) dédiée au **traitement des erreurs** (communément appelées **exceptions**).

DECLARATION

DECLARE - Optionnelle

/* déclaration variables, constantes, types, curseurs,... */

EXECUTABLE

BEGIN - Obligatoire

/* exécution des traitements */

.....

EXCEPTION - Optionnelle

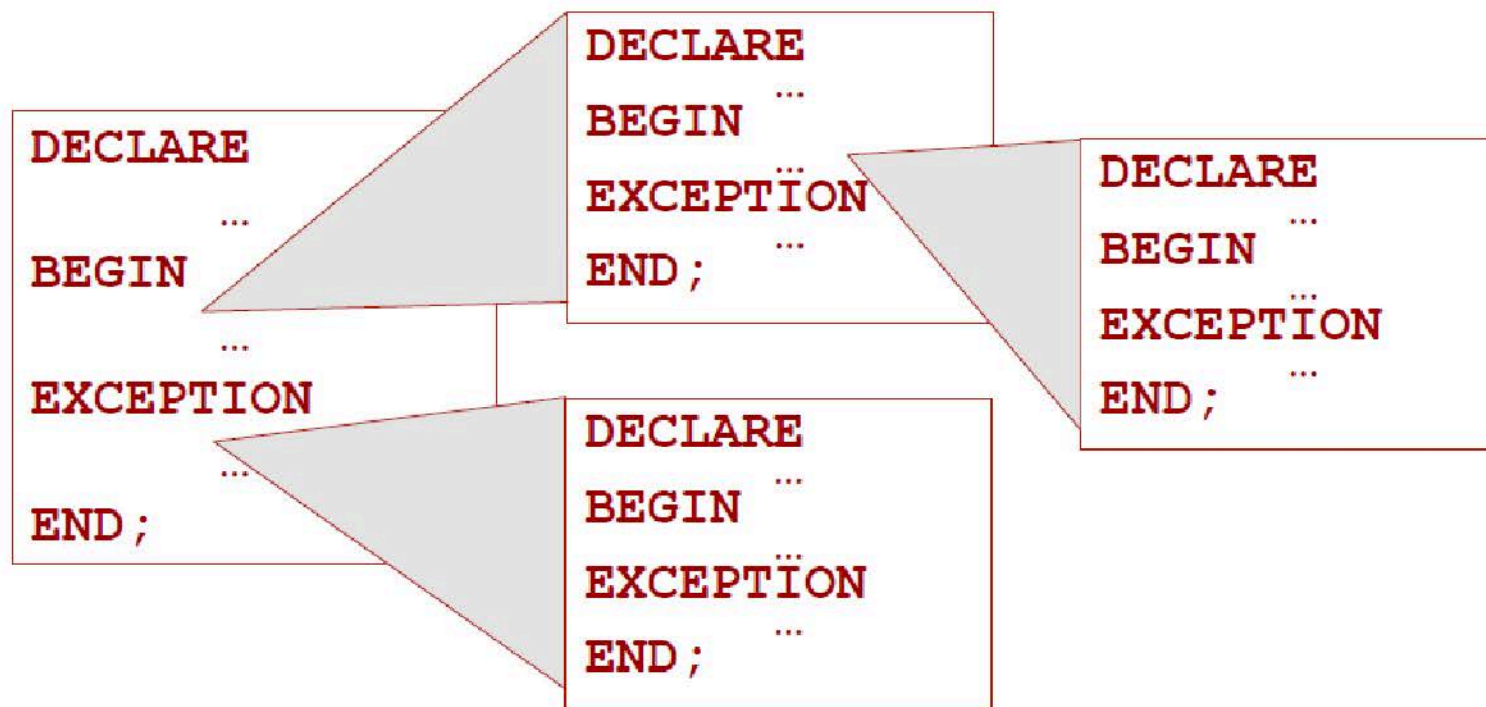
/* traitement des exception */

END - Obligatoire;

/

Structure d'un bloc PL/SQL

- Les blocs PL/SQL peuvent être imbriqués les uns dans les autres :



Structure d'un bloc PL/SQL

- PL/SQL ne permet pas les commandes SQL du langage de définition des données (LDD) comme la création de table (CREATE TABLE), la modification d'attribut (ALTER TABLE), etc.
- PL/SQL permet d'autres types de commandes SQL (SELECT, INSERT, UPDATE, DELETE, COMMIT, ROLLBACK)
- La commande SELECT (sauf si imbriqué dans un curseur (cf. prochain chapitre)) est toujours utilisée avec **INTO**, pour affecter les données retrouvées aux variables PL/SQL.

Le select **doit** retourner
exactement une ligne
(sinon notion de curseur)

```
SELECT colonne INTO v_col  
FROM nom de table  
WHERE condition(s);
```

Affichage

- Pour afficher le contenu d'une variable, d'une expression, etc., la procédure **DBMS_OUTPUT.PUT_LINE()** prend en argument le message à afficher.
- **DBMS_OUTPUT.PUT()** affiche un message sans retour à la ligne.
- Par défaut, les fonctions d'affichage sont désactivées.
 - ➔ Activation avec la commande **SET SERVEROUTPUT ON**

Lecture des données

- La directive ACCEPT permet de faire une lecture d'une variable de substitution au clavier :

ACCEPT nomVariable PROMPT 'messageàAfficher'

- Il est possible de passer en paramètres d'entrée d'un bloc PL/SQL des variables définies sous SQL*PLUS ou SQL Developer (outils à utiliser sous Oracle pour exécuter des blocs PL/SQL).

→ On accède aux valeurs d'une telle variable dans le code PL/SQL en faisant préfixer le nom de la variable par un « & ».

ACCEPT nbVol PROMPT 'Entrer durée du Vol : '

→ Instruction à faire au niveau de SQL*PLUS ou SQL Developer

DBMS_OUTPUT.PUT_LINE('La durée est : ' || &nbVol);

→ Instruction à faire au niveau d'un bloc PL/SQL