

Morgado-Samagaio Jonathan

**PABD / TP7 Oracle**

## Table des matières

Table des matières .....	2
1. Introduction.....	3
2. Création d'un bloc PL /SQL d'ajout d'une montagne à un pays.....	3
3. Création d'un bloc PL/SQL d'ajout de plusieurs montagnes à un pays.....	4
4. Conclusion .....	4

## 1. Introduction

Dans ce TP, nous allons continuer l'utilisation du PL/SQL avec l'utilisation des conditions et des boucles.

## 2. Création d'un bloc PL /SQL d'ajout d'une montagne à un pays

- 1) Nous allons commencer par créer un bloc PL/SQL permettant d'ajouter une montagne à un pays. Voici le bloc :

```
--1.1
SET SERVEROUTPUT ON;
ACCEPT pNomM PROMPT 'Saisissez le nom de la montagne';
ACCEPT pAlt PROMPT 'Saisissez l\'altitude de la montagne';
ACCEPT pChaine PROMPT 'Saisissez le nom de la chaine de la :
ACCEPT pNomP PROMPT 'Saisissez le nom du pays';

DECLARE
    vNomM Montagne.nom%TYPE := '&pNomM';
    vAlt Montagne.altitude%TYPE := '&pAlt';
    vChaine Montagne.chaine%TYPE := '&pChaine';
    vNomP Pays.nom%TYPE := '&pNomP';

BEGIN
    IF vAlt < 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Altitude négative'
    END IF;
    INSERT INTO Montagne(nm, nom, altitude, chaine)
    VALUES(seq_montagne.NEXTVAL, vNomM, vAlt, vChaine);
    DBMS_OUTPUT.PUT_LINE('Montagne'||vNomM||' ajoutée.');
```

Nous pouvons voir que le bloc à la même forme que ceux du dernier TP mais cette fois il y a une condition IF présente. Le bloc s'exécute sans problèmes.

- 2) Nous allons maintenant tester deux cas :

- Un cas qui fonctionne
- Un cas avec une altitude trop faible

1<sup>er</sup> cas : Valeur entré : ('Test', 500, 'TestChaine', 'France'). Ces valeurs passent sans soucis.

2eme cas : Valeurs entré : ('Test2', -600, 'TestChaine', 'France) Nous obtenons bien le code d'erreur suivant :

```
Rapport d'erreur -
ORA-20001: Altitude négat
ORA-06512: à ligne 9
```

### 3. Création d'un bloc PL/SQL d'ajout de plusieurs montagnes à un pays

- 1) Nous allons maintenant gérer une chaîne de caractère pour l'insertion de multiples valeurs. Nous devons faire un bloc prenant en entrée plusieurs numéros de montagne à ajouter dans un pays. Voici le bloc :

```
--2.1
SET SERVEROUTPUT ON;
ACCEPT pListe PROMPT 'Saisir la liste des montagnes';
ACCEPT pNom PROMPT 'Saissiez le nom du pays';

DECLARE
    vListe VARCHAR(30) := '&pListe';
    vNom Pays.nom%TYPE := '&pNom';
    listeEx BOOLEAN := FALSE;

BEGIN
    FOR i IN (SELECT regexp_substr(vListe, '^[^-]+', 1, level)
              FROM DUAL
              CONNECT BY regexp_substr(vListe, '^[^-]+', 1, level) IS NOT NULL)
    LOOP
        listeEx := TRUE;
        INSERT INTO Localiser(nm, nomp) VALUES(i.colonne, vNom);
        DBMS_OUTPUT.PUT_LINE('Localisation enregistré');
    END LOOP;

    IF (listeEx = FALSE;) THEN
        RAISE APPLICATION_ERROR(-20001, 'Liste vide');
```

- 2) En testant avec des valeurs correctes, les montagnes sont bien localiser dans le bon pays. Avec des valeurs incorrectes, nous obtenons bien le message d'erreur disant que la liste est vide.

### 4. Conclusion

Dans ce TP, nous avons pu voir comment utiliser des conditions et des boucles dans des blocs PL/SQL. Nous avons vu comment utiliser une requête dans un FOR.