Morgado-Samagaio Jonathan

PABD / TP8 Oracle

Table des matières

1. Introduction

Dans ce TP, nous allons continuer l'utilisation du PL/SQL avec l'utilisation des SELECT pour faire des vérifications.

2. Bloc PL /SQL d'ajout d'une montagne

Nous allons commencer par créer un bloc nous permettant d'ajouter une montagne en vérifiant si elle existe déjà. Voici le bloc :

```
--1.1
SET SERVEROUTPUT ON;
ACCEPT pNom PROMPT 'Saisissez le nom de la montagne';
ACCEPT pAlt PROMPT 'Saisissez l altitude de la montagn
ACCEPT pCha Prompt 'Saisissez la chaine de la montagne

DECLARE
    vNom Montagne.nom%TYPE := '&pNom';
    vAlt Montagne.altitude%TYPE := &pAlt;
    vCha Montagne.chaine%TYPE := '&pCha';
    vTest NUMBER;

BEGIN
    SELECT COUNT(*) INTO vTest FROM Montagne WHERE nom
    DBMS_OUTPUT.PUT_LINE(vTest);
    IF vTest >= 1 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Montagne déjà
    END IF;
    IF vAlt <= 0 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Altitude néga
    END IF;
```

Nous allons maintenant essayer d'ajouter deux fois la même montagne :

- Nom : Test
- Altitude : 500
- Chaine : TestChaine

Lors de la première exécution, la montagne est ajoutée sans problèmes. Lors de la seconde, nous obtenons l'erreur suivante :

```
Rapport d'erreur -
ORA-20001: Montagne déjà exis
ORA-06512: à ligne 11
```

Nous pouvons voir que la montagne est déjà existante. L'altitude à déjà été testé au dernier TP.

3. Bloc PL /SQL d'ajout d'une montagne à un pays

Nous reprenons le bloc du dernier TP avec l'ajout d'une montagne dans un pays. On le modifie pour ajouter des vérifications d'existence de montagne et de pays. Voici le code :

```
--2.1
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';
    vTestM NUMBER;
    vTestP NUMBER;

BEGIN
    IF vAlt < 0 THEN
        RAISE_APPLICATION_ERROR(-20001, 'Altitude négative'
    END IF;
    SELECT COUNT(*) INTO vTestM FROM Montagne WHERE nom = v
    SELECT COUNT(*) INTO vTestP FROM Pays WHERE nom = vNomP
    IF vTestM >= 1 THEN
        RAISE_APPLICATION_ERROR(-20002, 'Montagne déjà exis
    END IF;
    IF vTestP < 1 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Pays déjà existent
    END IF;
    INSERT INTO Montagne(nm, nom, altitude, chaine)
```

L'altitude et l'existence de la montagne on déjà été tester précédemment, nous allons donc seulement tester l'existence du pays. On saisit :

-Nom : Test3

-Altitude : 400

-Chaine : TestChaine

-Pays : France

Ces valeurs fonctionnent correctement. Si on change Le nom du pays pour un pays qui n'existe pas, on obtient :

```
Rapport d'erreur -
ORA-20003: Pays inexis
ORA-06512: à ligne 19
```

4. Bloc PL/SQL d'ajout d'un pays

Nous allons maintenant reprendre un bloc d'un ancien TP permettant d'ajouter un pays. On va rajouter des vérifications dessus. Voici le bloc :

```
--3.1
ACCEPT pNom PROMPT 'Saisir le nom du pays: ';
ACCEPT pCap PROMPT 'Saisir la capitale du pays: ';
ACCEPT pSup PROMPT 'Saisir la superficie du pays: ';
ACCEPT pPop PROMPT 'Saisir la population du pays: ';
ACCEPT pIs3 PROMPT 'Saisir le code ISO3 du pays: ';
ACCEPT pIs2 PROMPT 'Saisir le code ISO2 du pays: ';
ACCEPT pCon PROMPT 'Saisir le continent du pays: ';

DECLARE
    --Affectations des paramètres aux variables locales
    vNom Pays.Nom%TYPE := '&pNom';
    vCap Pays.Capitale%TYPE := '&pCap';
    vSup Pays.Superficie%TYPE := &pSup;
    vPop Pays.Population%TYPE := &pPop;
    vIs3 Pays.Iso3%TYPE := '&pIs3';
    vIs2 Pays.Iso2%TYPE := '&pIs2';
    vCon Pays.NomC%TYPE := '&pCon';
    vTestP NUMBER;
    vTestI3 NUMBER;
    vTestI2 NUMBER;
    vTestC NUMBER;

BEGIN
    --Vérifications dela superficie et de la population
    IF (vSup <= 0) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Superficie négative');
    END IF;
    IF (vPop <= 0) THEN
        RAISE_APPLICATION_ERROR(-20002, 'Population négative')
    END IF;
    SELECT COUNT(*) INTO vTestP FROM Pays WHERE nom = vNom;
    SELECT COUNT(*) INTO vTestI3 FROM Pays WHERE iso3 = vIs3;
    SELECT COUNT(*) INTO vTestI2 FROM Pays WHERE iso2 = vIs2;
    SELECT COUNT(*) INTO vTestC FROM Pays WHERE nomc = vCon;
    IF vTestP >= 1 THEN
        RAISE_APPLICATION_ERROR(-20003, 'Pays déjà existent');
    END IF;
    IF vTestI3 >= 1 THEN
        RAISE_APPLICATION_ERROR(-20004, 'Iso3 déjà existent');
    END IF;
    IF vTestI2 >= 1 THEN
    RAISE_APPLICATION_ERROR(-20005, 'Iso2 déjà existent');
    END IF;
```

La population, la superficie et le nom du pays ont déjà été testé précédemment.

Si l'on test avec des valeurs d'ISO2, 3 et un nom de continent inexistant, nous obtenons bien les erreurs attendues.

5. Conclusion

Dans ce TP, nous avons pu voir comment utiliser les SELECT INTO dans les blocs PL/SQL pour faire des vérifications de nos valeurs en entrée.