Morgado-Samagaio Jonathan

PABD / TP14 Oracle

Table des matières

1. Introduction

Dans ce TP, nous allons voir comment utiliser les fonctions stockées en PL/SQL

2. Fonction sans paramètre

Nous allons commencer par créer une fonction sans paramètres qui retourne le nombre de pays.

Voici le code :

```
--1.1
CREATE OR REPLACE FUNCTION getNBPays RETURN

n NUMBER;

BEGIN
    SELECT COUNT(p.nom) INTO n FROM Pays p;
    return n;
```

Nous pouvons voir que la fonction retourne un NUMBER n.

Pour pouvoir la tester, on utilise deux méthodes. La première :

```
--1.2
BEGIN
    DBMS_OUTPUT.PUT_LINE('Nombre de pays :'||
END;
```

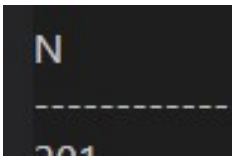
On obtient :

```
SQL> BEGIN
 2    DBMS_OUTPUT.PUT_LINE('Nombre de pays :'|
 3 END;
 4 /
```

La seconde :

```
--1.3
VARIABLE n NUMBER
EXEC :n := getNBP
PRINT :n;
```

Et on obtient :



3. Fonction avec paramètres

1) Nous continuons maintenant avec une fonction paramétrée getNBPaysContinent qui retourne le nombre de pays par continent. Voici le code :

```
--2.1
CREATE OR REPLACE FUNCTION getNBPaysContinent(nomCont IN Cont
RETURN NUMBER AS

n NUMBER;
nbPays NUMBER;

BEGIN
    --Vérification de l'existence du continent
    SELECT COUNT(nomCont) INTO n FROM Continent;
    IF (n <= 0) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Continent invalide')
    END IF;

    --Vérification de la présence de pays
    SELECT COUNT(p.nom) INTO nbPays FROM Pays p WHERE p.nomc
    IF (nbPays <= 0) THEN
        RAISE_APPLICATION_ERROR(-20002, 'Aucun pays dans le c
    END IF;
```

2) On test maintenant de la même manière qu'avant :

```
--2.2
ACCEPT pNomC PROMPT 'Saisissez le nom du continent';

DECLARE
    vNomC Continent.NOM%TYPE := '&pNomC';

BEGIN
    DBMS_OUTPUT.PUT_LINE('Nombre de pays dans le continent '||vNomC||' : '||getNBPa
```

On obtient :

3) On va maintenant faire une procédure paramétrée getNBPaysTraverses qui retourne le nombre de pays traversé par un fleuve en entrée. Voici le code :

```
--2.3
CREATE OR REPLACE FUNCTION getNBPaysTraverses(nomFl IN VARCHAR)
RETURN NUMBER AS

n NUMBER;
nbPays NUMBER;

BEGIN
    --Vérification de l'existance du fleuve
    SELECT COUNT(f.nom) INTO n FROM Fleuve f WHERE f.nom = nomFl;
    IF (n <= 0) THEN
        RAISE_APPLICATION_ERROR(-20001, 'Fleuve saisi invalide');
    END IF;

    --Requête
    SELECT COUNT(t.nomp) INTO nbPays FROM Traverser t, Fleuve f WHERE t.nf = f.n
```

4) On va tester notre fonction en faisant une requête SELECT pour tous les pays d'Océanie. Voici le code :

```
--2.4
SELECT f.nf, f.nom, getNBPaysTraverses(f.
FROM Fleuve f, Traverser t, Pays p
WHERE f.nf = t.nf AND t.nomp = p.nom
AND p.nomc = 'Océanie'
```

On obtient :

Save as: CSV

NF	NOM	NB
227	le Sepik	2
220	la Burdekin River	1
221	la Hastings River	1
222	le Mokau	1
223	le Murray	1
230	la Whanganui	1
225	le Pomahaka	1

(Des colonnes ne sont pas affichées).

4. Appel de fonction

Nous allons maintenant modifier notre première méthode pour obtenir le nombre de pays total en appelant notre deuxième fonction. Voici le code :

```
--3.1
CREATE OR REPLACE FUNCTION getNBPays
RETURN NUMBER AS

nbPays NUMBER := 0;

CURSOR C1 IS SELECT nom FROM Continent WHERE nom != '

BEGIN
    FOR ligne in C1 LOOP
        nbPays := nbPays + getNBPaysContinent(ligne.n
    END LOOP;
```

En testant de la même manière que la première fois, on obtient :

```
Nombre de pays :
```

5. Conclusion

Dans ce TP, nous avons pu voir comment utiliser les procédures stockées avec et sans paramètres et comment les appelés dans un autre bloc PL/SQL ou juste en SQL.