

PABD / TP14 Oracle

Table des matières

1. Introduction.....	2
2. Manipulation de la base de données.....	2
2.1. Fonctions sans paramètre.....	2
2.2. Fonctions avec paramètres.....	3
2.3. Appel de fonctions.....	5
3. Conclusion.....	5

1. Introduction

Dans ce TP nous revenons à la programmation PL/SQL au sein du SGBD Oracle. Nous utilisons les mécanismes des procédures stockées pour définir des fonctions.

2. Manipulation de la base de données

2.1. Fonctions sans paramètre

(1) Développer une fonction getNBJoueurs qui calcule le nombre de joueurs.

```
CREATE OR REPLACE FUNCTION getNBJoueurs
RETURN number IS
    nbj number :=0;
BEGIN
    SELECT COUNT (*) INTO nbj FROM Joueur;
    RETURN nbj;
END;
/
```

(2) Testez la fonction à partir d'un bloc PL/SQL

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Nombre de joueurs : ' || getNBJoueurs) ;
END ;
```

Nombre de joueurs : 286

Procédure PL/SQL terminée.

(3) Testez la fonction à partir de SQL

```
VARIABLE n NUMBER;
EXEC :n := getNBJoueurs;
PRINT : n;
```

	N

	286

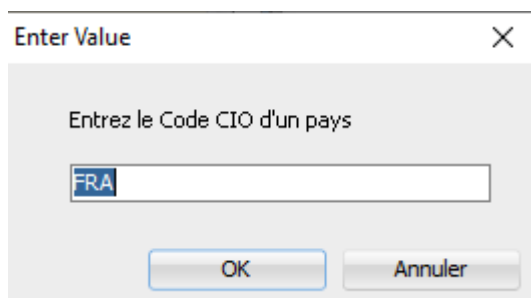
2.2. Fonctions avec paramètres

(1) Créer une fonction getNBJoueursPays conforme aux spécifications suivantes

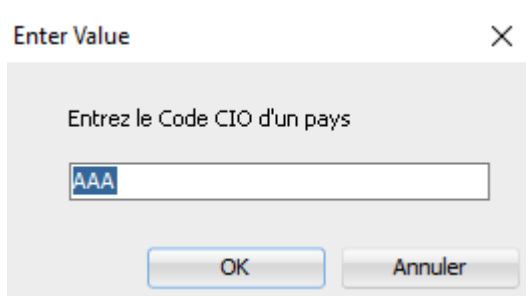
```
CREATE OR REPLACE FUNCTION getNBJoueursPays
( pCio Pays.cio%TYPE)
RETURN number IS
nbp number :=0;
Pays_Existant EXCEPTION;
PRAGMA EXCEPTION_INIT(Pays_Existant, -20010);
nbp number :=0;
BEGIN
    SELECT COUNT (CIO) INTO nbr FROM Pays WHERE cio = pCio;
    SELECT COUNT (nj) INTO nbp FROM Joueur WHERE nat = pCio;
    IF nbr = 1 THEN
        RETURN(nbp);
    ELSE
        RAISE Pays_Existant;
        RETURN(0);
    END IF;
EXCEPTION
    WHEN Pays_Existant THEN
        DBMS_OUTPUT.PUT_LINE('Pays inconnu!');
END;
/
```

(2) Testez la fonction en demandant la saisie du paramètre par une variable issue de l'instruction ACCEPT.

```
ACCEPT pCio PROMPT "Entrez le Code CIO d'un pays" ;
Variable n Number
EXEC :n := getNBJoueursPays(&pCio) ;
print :n;
```



```
      N
-----
      33
```



Pays inconnu !

(3) Créer une fonction getNBMatches conforme aux spécifications suivantes.

```
CREATE OR REPLACE FUNCTION getNBMatches
( pNom Joueur.nom%TYPE, pPre Joueur.pre%TYPE)
RETURN number IS
    nbm number :=0;
BEGIN
    SELECT COUNT (nm) INTO nbm FROM Match WHERE nj1 IN (SELECT nj FROM Joueur WHERE nom = pNom AND pre =pPre)
    OR nj2 IN (SELECT nj FROM Joueur WHERE nom = pNom AND pre =pPre)
    OR nj1b IN (SELECT nj FROM Joueur WHERE nom = pNom AND pre =pPre)
    OR nj2b IN (SELECT nj FROM Joueur WHERE nom = pNom AND pre =pPre);
    RETURN nbm;
END;
/

BEGIN
    DBMS_OUTPUT.PUT_LINE('Nombre de matchs : ' || getNBMatches('NADAL','RAFAEL')) ;
END;
```

Nombre de matchs : 7

(4) Testez la fonction dans une requête SQL (SELECT) qui affiche les matchs joués par les joueurs français classés dans l'ordre des joueurs ayant le plus joué.

```
SELECT pre, nom, getNBMatches(nom, pre)AS "NBM" FROM Joueur
WHERE nat = 'FRA' ORDER BY nbm DESC;
```

	PRE	NOM	NBM
1	KRISTINA	MLADENOVIC	5
2	GAEL	MONFILS	4
3	JEREMY	CHARDY	4
4	BENOIT	PAIRE	4
5	CORENTIN	MOUTET	3

30	HARMONY	TAN	1
31	JESSIKA	PONCHET	1
32	ALIZÉ	CORNET	1
33	Ahmed	Aharouite	0

2.3. Appel de fonctions

(1) Modifier la fonction getNBJoueurs en appelant la fonction getNBJoueursPays de la section 2 pour chaque pays de la BD.

```
CREATE OR REPLACE FUNCTION getNBJoueurs
RETURN number IS
    nbj number :=0;
BEGIN
    SELECT SUM (getNBJoueursPays(cio)) INTO nbj FROM Pays;
    RETURN nbj;
END;
/
```

(2) Testez la nouvelle fonction.

```
BEGIN
    DBMS_OUTPUT.PUT_LINE('Nombre de joueurs : ' || getNBJoueurs) ;
END ;
```

```
Nombre de joueurs : 286
```

3. Conclusion

Ce TP14 a pour but de mettre en application ce que nous avons déjà vu au cours des TD et TP précédents, c'est à dire la manipulation de base de données grâce au langage SQL (Structured Query Langage) et le fonctionnement de blocs PL/SQL.

Durant ce TP, on a pu développer des fonctions avec ou sans paramètres et nous avons fais des appel de fonctions.

Ce TP14 nous a permis d'approfondir nos connaissances sur la gestion des bases de donnés en PL-SQL , plus précisément sur l'utilisation de fonctions.