

Aharouite Ahmed

PABD / TP08 Oracle

Table des matières

1. Introduction.....	3
2. Manipulation de la base de données.....	3
2.1. Création d'un bloc PL /SQL d'ajout d'un pays.....	3
2.2. Création d'un bloc PL/SQL d'ajout d'un joueur.....	5
3. Conclusion.....	6

1. Introduction

Dans ce TP nous continuons à développer des blocs PL/SQL utilisant des instructions d'interrogation mono-ligne : SELECT ... INTO ...

Dans ce TP nous n'utiliserons pas encore les exceptions. Chaque erreur est testée en début de bloc, et un simple affichage (DBMS_OUTPUT.PUT_LINE) est réalisé pour expliquer l'erreur.

2. Manipulation de la base de données

2.1. Création d'un bloc PL/SQL d'ajout d'un pays

(1) Reprendre le bloc PL/SQL permettant d'ajouter un pays, et compléter ce dernier afin de prendre en compte les spécifications suivantes.

```
SET SERVEROUTPUT ON;

ACCEPT pCio PROMPT 'Saisir le cio du pays : ';
ACCEPT pNom PROMPT 'Saisir le nom de pays : ';

DECLARE
vCio Pays.cio%TYPE := '&pCio';
vNom Pays.nom%TYPE := '&pNom';
n NUMBER := 0;

BEGIN
SELECT COUNT(*) INTO n FROM Pays WHERE vCio = cio;
IF (n = 1) THEN
    DBMS_OUTPUT.PUT_LINE('Le pays existe déjà');
ELSE
    INSERT INTO Pays (cio, nom) VALUES (vCio, vNom);
    -- Affichage d'un message
    DBMS_OUTPUT.PUT_LINE('Pays '||vCio||' '||vNom||' '||' ajouté. ');
    DBMS_OUTPUT.PUT_LINE('Pays enregistré');
END IF;
COMMIT;

END;
/
```

(2) Testez votre bloc PL/SQL avec les différents cas possibles prévus.

vCio Pays.cio%TYPE := 'FRA';	Le pays existe déjà
vNom Pays.nom%TYPE := 'France';	
	Procédure PL/SQL terminée.
vCio Pays.cio%TYPE := 'NEW';	Pays NEW NewPays ajouté.
vNom Pays.nom%TYPE := 'NewPays';	Pays enregistré
	Procédure PL/SQL terminée.

(3) Reprendre le bloc PL/SQL en utilisant RAISE_APPLICATION_ERROR (<code>, <message>).

```
ACCEPT pCio PROMPT 'Saisir le cio du pays : ';
ACCEPT pNom PROMPT 'Saisir le nom de pays : ';

DECLARE
vCio Pays.cio%TYPE := '&pCio';
vNom Pays.nom%TYPE := '&pNom';
n NUMBER := 0;

BEGIN
SELECT COUNT(*) INTO n FROM Pays WHERE vCio = cio;
IF (n = 1) THEN
    RAISE_APPLICATION_ERROR(-20000,'Erreur!');
ELSE
    INSERT INTO Pays (cio, nom) VALUES (vCio, vNom);
    -- Affichage d'un message
    DBMS_OUTPUT.PUT_LINE('Pays '||vCio||' '||vNom||' '||' ajouté. ');
    DBMS_OUTPUT.PUT_LINE('Pays enregistré');
END IF;
COMMIT;

END;
/
```

(4) Testez votre bloc PL/SQL avec cette nouvelle version. Quelle différence constatez vous ?

vCio Pays.cio%TYPE := 'FRA';	Rapport d'erreur -
vNom Pays.nom%TYPE := 'France';	ORA-20000: Erreur!
vCio Pays.cio%TYPE := 'AH';	Pays AH Ahmed ajouté.
vNom Pays.nom%TYPE := 'Ahmed';	Pays enregistré

On remarque que lorsqu'une erreur est levée, elle sera affichée de manière plus précise et spécifique par rapport au langage des bases de données (<code>, <message>) .

2.2. Création d'un bloc PL/SQL d'ajout d'un joueur

(1) Reprendre le bloc PL/SQL permettant d'ajouter un joueur, et compléter ce dernier afin de prendre en compte les spécifications suivantes.

```
SET SERVEROUTPUT ON;

ACCEPT pPre PROMPT 'Saisir le prénom du joueur : ';
ACCEPT pNom PROMPT 'Saisir le nom du joueur : ';
ACCEPT pGen PROMPT 'Saisir le genre du joueur : ';
ACCEPT pNat PROMPT 'Saisir la nationalité du joueur : ';
ACCEPT pVns PROMPT 'Saisir la ville de naissance du joueur : ';
ACCEPT pPns PROMPT 'Saisir le pays de naissance du joueur : ';
ACCEPT pDns PROMPT 'Saisir la date de naissance du joueur : ';

DECLARE
vNj Joueur.nj%TYPE := SEQ_JOUEUR.nextval;
vPre Joueur.pre%TYPE := 'pPre';
vNom Joueur.nom%TYPE := 'pNom';
vGen Joueur.gen%TYPE := 'pGen';
vNat Joueur.nat%TYPE := 'pNat';
vVns Joueur.vns%TYPE := 'pVns';
vPns Joueur.pns%TYPE := 'pPns';
vDns Joueur.dns%TYPE := 'pDns';
n NUMBER := 0;
p NUMBER := 0;

BEGIN
SELECT COUNT(*) INTO n FROM Pays WHERE vNat = cio;
SELECT COUNT(*) INTO p FROM Pays WHERE vPns = cio;
-- Verification de la validité du genre
IF (vGen IN ('H','F')) AND n=1 AND p=1 THEN
    -- Insertion du nouveau Joueur
    INSERT INTO Joueur (nj, pre, nom, gen, nat, vns, pns, dns) VALUES (vNj, vPre, vNom, vGen, vNat, vVns, vPns, vDns);
    -- Affichage d'un message
    DBMS_OUTPUT.PUT_LINE('Joueur '||vNj||' '||vPre||' '||vNom||' '||vGen||' '||vNat||' '||vVns||' '||vPns||' '||vDns||' ajouté. ');
    DBMS_OUTPUT.PUT_LINE('Joueur enregistré');
ELSE
    RAISE_APPLICATION_ERROR(-20000,'Erreur!');
    IF vGen NOT IN ('H','F') THEN
        DBMS_OUTPUT.PUT_LINE('Genre invalide');
    END IF;
    IF n=0 THEN
        DBMS_OUTPUT.PUT_LINE('Nationalité inconnue');
    END IF;
    IF p=0 THEN
        DBMS_OUTPUT.PUT_LINE('Pays de naissance inconnu');
    END IF;
-- Validation
COMMIT;
END IF;

END;
```

(2) Testez votre bloc PL/SQL avec les différents cas possibles prévus

Joueur 372 AHMED AHAROUITE H FRA TOULOUSE FRA 20/09/01 ajouté.
Joueur enregistré

Procédure PL/SQL terminée.

<pre>nouveau :DECLARE vNj Joueur.nj%TYPE := SEQ_JOUEUR.nextval; vPre Joueur.pre%TYPE := 'AHMED'; vNom Joueur.nom%TYPE := 'AHAROUITE'; vGen Joueur.gen%TYPE := 'M'; vNat Joueur.nat%TYPE := 'M'; vVns Joueur.vns%TYPE := 'TOULOUSE'; vPns Joueur.pns%TYPE := 'M'; vDns Joueur.dns%TYPE := '20/09/01'; n NUMBER := 0; p NUMBER := 0;</pre>	<pre>Rapport d'erreur - ORA-20000: Erreur!</pre>
--	--

3. Conclusion

Ce TP08 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 étudier le fonctionnement de blocs PL/SQL . Nous avons pu déclarer des variables, utiliser des saisies (valeurs entrées au clavier), créer une exception et vérifier les cas d'erreurs.

Ce TP08 nous a permis d'approfondir nos connaissances sur la gestion des bases de données en PL-SQL, plus précisément sur l'utilisation d'insertions de joueur et de pays.