

PABD / TP15 Oracle

Table des matières

1. Introduction.....	2
2. Manipulation de la base de données.....	2
2.1. Création d'un paquetage.....	2
3. Conclusion.....	7

1. Introduction

Dans ce TP nous utilisons les mécanismes des paquetages.

2. Manipulation de la base de données

2.1. Création d'un paquetage

(1) Reprenez les sous-programmes développés dans le TP12, et les placer dans un paquetage nommé gestion_tennis les sous-programmes ci-dessous.

```
CREATE OR REPLACE PACKAGE gestion_tennis IS
    PROCEDURE AjouterPays (vCIO Pays.cio%TYPE, vNom Pays.nom%TYPE);
    PROCEDURE AjouterJoueur (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE, vGen Joueur.Gen%TYPE,
        vNat Joueur.Nat%TYPE, vVNs Joueur.VNs%TYPE, vPNs Joueur.PNs%TYPE, vDNs Joueur.DNs%TYPE);
    PROCEDURE AjouterMatch (vNT Match.NT%TYPE, vTour Match.Tour%TYPE, vListNJ VARCHAR);
    PROCEDURE AfficherMatches (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE);
END gestion_tennis;
/
```

a) ajouter un pays

```
CREATE OR REPLACE PACKAGE BODY gestion_tennis AS

    PROCEDURE AjouterPays (vCIO Pays.cio%TYPE, vNom Pays.nom%TYPE) IS
BEGIN
    -- Traitement
    INSERT INTO Pays (cio, nom) VALUES (vCIO, vNom);
    DBMS_OUTPUT.PUT_LINE('Pays '||vCIO||' '||vNom||' ajouté.');
```

```
    COMMIT;
EXCEPTION
    WHEN DUP_VAL_ON_INDEX THEN
        DBMS_OUTPUT.PUT_LINE('Pays existant !');
```

```
END AjouterPays;
```

b) ajouter un joueur

```
PROCEDURE AjouterJoueur (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE, vGen Joueur.Gen%TYPE,
vNat Joueur.Nat%TYPE, vVNs Joueur.VNs%TYPE, vPNs Joueur.PNs%TYPE, vDNs Joueur.DNs%TYPE) IS
    erreur_ck EXCEPTION;
    PRAGMA EXCEPTION_INIT(erreur_ck, -2290);
    erreur_fk EXCEPTION;
    PRAGMA EXCEPTION_INIT(erreur_fk, -2291);
BEGIN
    -- Traitement
    INSERT INTO Joueur (nj, pre, nom, gen, nat, vns, pns, dns)
    VALUES (seq_joueur.NEXTVAL, vPre, vNom, vGen, vNat, vVNs, vPNs, vDNs);
    DBMS_OUTPUT.PUT_LINE('Joueur ajouté.');
```

COMMIT;

EXCEPTION

WHEN erreur_ck THEN

DBMS_OUTPUT.PUT_LINE('Genre invalide !');

WHEN erreur_fk THEN

IF (SQLERRM LIKE '%FK_JOUEUR_NAT%') THEN

DBMS_OUTPUT.PUT_LINE ('Nationalité inconnue !');

ELSIF (SQLERRM LIKE '%FK_JOUEUR_PNS%') THEN

DBMS_OUTPUT.PUT_LINE ('Pays de naissance inconnu !');

END IF;

END AjouterJoueur;

c) ajouter un match

```
PROCEDURE AjouterMatch (vNT Match.NT%TYPE, vTour Match.Tour%TYPE, vListNJ VARCHAR) IS
    TYPE vecteur_joueur IS TABLE OF NUMBER INDEX BY BINARY_INTEGER;
    NJ vecteur_joueur;
    i NUMBER:=0;
    erreur_nb_joueur EXCEPTION;
    erreur_fk EXCEPTION;
    PRAGMA EXCEPTION_INIT(erreur_fk, -2291);
BEGIN
    -- Traitement liste joueur & Vérification joueur
    FOR ligne IN (SELECT regexp_substr(vListNJ, '[^~]+', 1, LEVEL) AS colonne
    FROM DUAL
    CONNECT BY regexp_substr(vListNJ, '[^~]+', 1, LEVEL) IS NOT NULL) LOOP
        i:=i+1;
        NJ(i):=ligne.colonne;
    END LOOP;
    -- Vérification nombre de joueurs
    IF (vNT IN ('SM','SD') AND i<>2) OR
    (vNT IN ('DM','DD','DX') AND i<>4) THEN
        RAISE erreur_nb_joueur;
    END IF;
    -- Traitement
    IF (i=2) THEN
        INSERT INTO Match (NM,NT,TOUR,NJ1,NJ2)
        VALUES (Seq_Match.NEXTVAL, vNT, vTour, NJ(1), NJ(2));
    ELSE
        INSERT INTO Match (NM,NT,TOUR,NJ1,NJ1b,NJ2,NJ2b)
        VALUES (Seq_Match.NEXTVAL, vNT, vTour, NJ(1), NJ(2), NJ(3), NJ(4));
    END IF;
    DBMS_OUTPUT.PUT_LINE('Match enregistré');
    COMMIT;
EXCEPTION
    WHEN erreur_nb_joueur THEN
        DBMS_OUTPUT.PUT_LINE('Nombre de joueurs invalide !');

WHEN erreur_fk THEN



IF (SQLERRM LIKE '%FK_MATCH_NT%') THEN



DBMS_OUTPUT.PUT_LINE('Tableau invalide !');



ELSIF (SQLERRM LIKE '%FK_MATCH_NJ%') THEN



DBMS_OUTPUT.PUT_LINE('Joueur inconnu !');



END IF;



END AjouterMatch;


```

d) afficher matchs

```
PROCEDURE AfficherMatches (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE) IS
-- Variables
vJoueur Joueur%ROWTYPE;
estVide BOOLEAN:=TRUE;
erreur_match EXCEPTION;
BEGIN
-- Cas d'erreur Joueur inconnu (NO_DATA_FOUND)
SELECT * INTO vJoueur
FROM Joueur
WHERE pre = vPre AND nom = vNom;
-- Traitement
FOR ligne IN (SELECT M.Tour, J1.pre, J1.nom, j1sset1, j1sset2, j1sset3, j1sset4, j1sset5, j2sset1, j2sset2, j2sset3, j2sset4, j2sset5
FROM Match M, Joueur J1, Joueur J2
WHERE J1.nj = nj1
AND J2.nj = nj2 AND J2.pre = vPre AND J2.nom = vNom
AND nt LIKE 'S_'
UNION
SELECT M.Tour, J2.pre, J2.nom, j2sset1, j2sset2, j2sset3, j2sset4, j2sset5, j1sset1, j1sset2, j1sset3, j1sset4, j1sset5
FROM Match M, Joueur J1, Joueur J2
WHERE J1.nj = nj1 AND J1.pre = vPre AND J1.nom = vNom
AND J2.nj = nj2
AND nt LIKE 'S_') LOOP
IF estVide THEN
estVide:=FALSE;
DBMS_OUTPUT.PUT_LINE('Match de '||vPre||' '||vNom);
END IF;
DBMS_OUTPUT.PUT(' '||ligne.Tour||' : '||ligne.pre||' '||ligne.nom);
IF ligne.j1sset1+ligne.j2sset1 > 0 THEN DBMS_OUTPUT.PUT(' ('||ligne.j2sset1||'-'||ligne.j1sset1||')'); END IF;
IF ligne.j1sset2+ligne.j2sset2 > 0 THEN DBMS_OUTPUT.PUT(' ('||ligne.j2sset2||'-'||ligne.j1sset2||')'); END IF;
IF ligne.j1sset3+ligne.j2sset3 > 0 THEN DBMS_OUTPUT.PUT(' ('||ligne.j2sset3||'-'||ligne.j1sset3||')'); END IF;
IF ligne.j1sset4+ligne.j2sset4 > 0 THEN DBMS_OUTPUT.PUT(' ('||ligne.j2sset4||'-'||ligne.j1sset4||')'); END IF;
IF ligne.j1sset5+ligne.j2sset5 > 0 THEN DBMS_OUTPUT.PUT(' ('||ligne.j2sset5||'-'||ligne.j1sset5||')'); END IF;
DBMS_OUTPUT.PUT_LINE(' ');
END LOOP;
IF estVide THEN
RAISE erreur_match;
END IF;
EXCEPTION
WHEN NO_DATA_FOUND THEN
DBMS_OUTPUT.PUT_LINE('Joueur inconnu !');
WHEN erreur_match THEN
DBMS_OUTPUT.PUT_LINE('Pas de match de simple !');
END AfficherMatches;
```

END gestion_tennis;

/

(2) Testez le paquetage.

a) ajouter un pays

```
ACCEPT vcIO PROMPT 'Entrer le cio : '
ACCEPT vNom PROMPT 'Entrer le nom : '
EXECUTE gestion_tennis.AjouterPays('&vcIO', '&vNom');
```

vcio = 'FRA' vnom = 'France'

vcio = 'AAA' vnom = 'AAA'

Pays existant !

Pays AAA AAA ajouté.

b) ajouter un joueur

```
ACCEPT vPre PROMPT 'Entrer le prénom : '
ACCEPT vNom PROMPT 'Entrer le nom : '
ACCEPT vGen PROMPT 'Entrer le genre : '
ACCEPT vNat PROMPT 'Entrer la nationalité : '
ACCEPT vVNs PROMPT 'Entrer la ville de naissance : '
ACCEPT vPNs PROMPT 'Entrer le pays de naissance : '
ACCEPT vDNs PROMPT 'Entrer la date de naissance : '
EXECUTE gestion_tennis.AjouterJoueur('&vPre', '&vNom', '&vGen', '&vNat', '&vVNs', '&vPNs', '&vDNs');

Joueur ajouté.
```

	PRE	NOM	GEN	CLA	NAT	VNS	PNS
1	461 Ahmed	Aha	H	(null)	AAA	aaa	AAA


c) ajouter un match







```
ACCEPT vNT PROMPT 'Entrer le nt : '
ACCEPT vTour PROMPT 'Entrer le tour : '
ACCEPT vListNJ PROMPT 'Entrer les numéros des joueurs : '
EXECUTE gestion.tennis.AjouterMatch('&vNT', '&vTour', '&vListNJ');
```

1) $nt = 'SX'$

Tableau invalide !

2)


 Trier... | Filtre : `tour = 'finale' AND nm = 442`

	 1	NT	TOUR	 NJ1	 NJ1B	 NJ2	 NJ2B	 J1SSET1
1	442 SM	finale		1 (null)		2 (null)		(null)

Match enregistré

d) afficher matchs

```
ACCEPT vPre PROMPT 'Entrer le prénom : '
ACCEPT vNom PROMPT 'Entrer le nom : '
EXECUTE gestion.tennis.AfficherMatches('vPre', 'vNom');
```

Match de RAFAEL NADAL

1/2 finale : ROGER FEDERER (6-3) (6-4) (6-2)
1/4 de finale : KEI NISHIKORI (6-1) (6-1) (6-3)
1/8 de finale : JUAN IGNACIO LONDERO (6-2) (6-3) (6-3)
1er tour : YANNICK HANFMAN (6-2) (6-1) (6-3)
2ème tour : YANNICK MADEN (6-1) (6-2) (6-4)
3ème tour : DAVID GOFFIN (6-1) (6-3) (4-6) (6-3)
finale : DOMINIC THIEM (6-3) (5-7) (6-1) (6-1)

Procédure PL/SQL terminée.

(3) Modifiez le paquetage en ajoutant une nouvelle procédure SupprimerJoueur respectant les spécifications suivantes :

```
CREATE OR REPLACE PACKAGE gestion_tennis IS
    PROCEDURE AjouterPays (vCIO Pays.cio%TYPE, vNom Pays.nom%TYPE);
    PROCEDURE AjouterJoueur (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE, vGe
vNat Joueur.Nat%TYPE, vVNs Joueur.VNs%TYPE, vPNs Joueur.PNs%TYPE, vDNs J
    PROCEDURE AjouterMatch (vNT Match.NT%TYPE, vTour Match.Tour%TYPE, vListN
    PROCEDURE AfficherMatches (vPre Joueur.Pre%TYPE, vNom Joueur.Nom%TYPE);
    PROCEDURE SupprimerJoueur (vNom Joueur.nom%TYPE, vPre Joueur.pre%TYPE);
END gestion_tennis;
/
```

```
END AfficherMatches;
```

```







PROCEDURE SupprimerJoueur (vNom Joueur.nom%TYPE, vPre Joueur.pre%TYPE) IS
    vNj Joueur.nj%TYPE;
BEGIN
    SELECT nj INTO vNj
    FROM Joueur
    WHERE nom = vNom AND pre = vPre;
    -- Traitement
    DELETE FROM Match
    WHERE nj1 = vNJ
           OR nj2 = vNJ
           OR nj1b = vNJ
           OR nj2b = vNJ;
    DELETE FROM Joueur
    WHERE nom = vNom AND pre = vPre;
    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Joueur inconnu !');
END SupprimerJoueur;








END gestion_tennis;
/

```

```
ACCEPT vNom PROMPT 'Entrer le nom : '
ACCEPT vPre PROMPT 'Entrer le prénom : '
EXECUTE gestion.tennis.SupprimerJoueur('&vNom', '&vPre');
```

Colones | Données | Model | Contraintes | Droits | Statistiques | Déclencheurs | Flashback | Dépendances | Détails

      | Trier... | Filtre : nom = 'Aha' AND pre = 'Ahmed'

 ¹ NJ |  PRE |  NOM |  GEN |  CLA |  NAT |  VNS

Procédure PL/SQL terminée. Le joueur précédemment créé vient d'être supprimé.

3. Conclusion

Ce TP15 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 Language) et le fonctionnement de blocs PL/SQL.

Durant ce TP, on a pu développer un paquetage rempli avec diverses procédures.

Ce TP15 nous a permis d'approfondir nos connaissances sur la gestion des bases de données en PL-SQL , plus précisément sur l'utilisation de paquetage.