Cartouche du document

Année : ING 2
Activité : Travail dirigé

Objectifs

L'objectif de cette série d'exercices est :

- de maîtriser la conception de fonctions et procédures stockées
- d'apprendre à utiliser des curseurs dans un bloc PL/SQL

Sommaire des exercices

Corps des exercices

1 - Producteurs de Vins

Énoncé :

On reprend les tables Producteur, Vin et Recolte.

Question 1)

Énoncé de la question

Ecrire la procédure qui permet de créer un nouveau producteur. Ce nouveau producteur devra avoir un couple (nom, prenom) différent de tous les couples existants. Cette procédure devra afficher un message pour préciser si l'opération s'est bien déroulée.

Solution de la question

```
CREATE OR REPLACE PROCEDURE creer_producteur(pnom varchar2, pprenom
varchar2, pregion varchar2) is
cpt number;
BEGIN
    if pnom is null then
        dbms_output.put_line('Veuillez donner le nom du producteur');
    elsif pprenom is null then
        dbms_output.put_line('Veuillez donner le prénom du producteur');
    elsif pregion is null then
        dbms_output.put_line('Veuillez donner la région du producteur');
    else
        SELECT count(*) INTO cpt FROM producteur
        WHERE nom = pnom and prenom = pprenom;
        IF cpt > 0 THEN
            dbms_output.put_line('Le couple (' || pnom || ',' || pprenom || ')  existe déjà');
        ELSE
            INSERT INTO Producteur(nom,prenom,region)
VALUES(pnom,pprenom,pregion);
```

```
                dbms_output.put_line('Insertion réussie');
            END IF;
        END IF;
    END creer_producteur;
    /
    select * from user_errors WHERE name = 'CREER_PRODUCTEUR';
```

.

Question 2)

Énoncé de la question

Reprendre la question précédente en traitant les erreurs avec la fonction raise_application_error.

Rappel : le code d'erreur doit être un entier inférieur à -20000. Tout nombre supérieur ou égal est code réservé par Oracle.

Solution de la question

```
CREATE OR REPLACE PROCEDURE creer_producteur(pnom varchar2, pprenom
varchar2, pregion varchar2) is
cpt number;
BEGIN
    if pnom is null then
        raise_application_error(-20001,'Veuillez donner le nom du producteur');
    elsif pprenom is null then
        raise_application_error(-20001,'Veuillez donner le prénom du producteur');
    elsif pregion is null then
        raise_application_error(-20001,'Veuillez donner la région du producteur');
    else
        SELECT count(*) INTO cpt FROM producteur
        WHERE nom = pnom and prenom = pprenom;
        IF cpt > 0 THEN
            raise_application_error(-20001,'Le couple (' || pnom || ',' || pprenom || ')  existe
déjà');
        ELSE
            INSERT INTO Producteur(nom,prenom,region)
VALUES(pnom,pprenom,pregion);
            dbms_output.put_line('Insertion réussie');
        END IF;
    END IF;
END creer_producteur;
/
select * from user_errors WHERE name = 'CREER_PRODUCTEUR';
```

.

Question 3)

Énoncé de la question

On suppose qu'on rajoute dans l'association Recolte un attribut **indépendant** noté no_recolte. Il permet de préciser qu'un producteur peut faire plusieurs récoltes d'un même vin.

Que faut-il changer dans le MLD et le MPD ?

Pour toutes les questions qui suivent, on travaillera avec ce nouveau schéma de la base.

Solution de la question

Dans le MLD on transforme la relation Recolte : (#numvi,#numprod, no_recolte,quantite).

Dans la table recolte on ajoute la colonne no_recolte de type number. Cette colonne est ajoutée à la clé primaire.

Question 4)

Énoncé de la question

Ecrire une procédure nommée reporting_recoltes qui reçoit le nom et le prénom d'un producteur et qui

- Sa région de production et la quantité totale produite
- Un tableau des récoltes classées par type de vin

Solution de la question

```
CREATE OR REPLACE PROCEDURE reporting_recoltes(pnom varchar2, pprenom
varchar2) is
vregion producteur.region%type;
cursor cur_recoltes is
    select * from recolte
    WHERE numprod in (select numprod from producteur
        WHERE nom = pnom and prenom = pprenom)
    order by numvin;
vnumvin vin.numvin%type;
cpt number;
BEGIN
    if pnom is null then
        dbms_output.put_line('Veuillez donner le nom du producteur');
    elsif pprenom is null then
        dbms_output.put_line('Veuillez donner le prénom du producteur');
    else
        SELECT region INTO vregion FROM producteur
        WHERE nom = pnom and prenom = pprenom;
        dbms_output.put_line('La région du producteur (' || pnom || ',' || pprenom || ')  est ' ||
vregion);
        vnumvin := 'bad';

        for enr_recoltes in cur_recoltes
        loop
```

```
                if vnumvin <> enr_recoltes.numvin then
                    dbms_output.put_line('Recoltes de ' || enr_recoltes.numvin);
                    cpt := 1;
                end if;
                vnumvin := enr_recoltes.numvin;
                dbms_output.put_line('Recolte ' || cpt || ' : ' || enr_recoltes.quantite || '
bouteille(s)');
                cpt := cpt + 1;
            end loop;
        END IF;
    END reporting_recoltes;
    /
    select * from user_errors WHERE name = 'REPORTING_RECOLTES';
```

.

Question 5)

Énoncé de la question

Ecrire une fonction qui reçoit l'identifiant d'un vin et qui renvoie une chaîne de caractères contenant les caractéristiques de ce vin.

Solution de la question

```
    CREATE or replace function Carac_Vin(pnumvin varchar2) return varchar2 is
    cpt number;
    res varchar2(50);
    begin
        select count(*) into cpt from vin WHERE numvin = pnumvin;
        if cpt = 0 then
            res := 'vin inconnu';
        else
            select 'Cru : ' || cru || ' -- degré : ' || degre || ' -- annee : ' || annee into res from
            vin WHERE numvin = pnumvin;
        end if;
        return res;
    end Carac_Vin;
    /
    select * from user_errors WHERE name = 'CARAC_VIN';
```

.

Question 6)

Énoncé de la question

Réécrire la procédure reporting_recoltes qui appelle la fonction précédente pour une meilleure présentation du reporting des récoltes.

Solution de la question

```
CREATE OR REPLACE PROCEDURE reporting_recoltes(pnom varchar2, pprenom
varchar2) is
vregion producteur.region%type;
cursor cur_recoltes is
    select * from recolte
    WHERE numprod in (select numprod from producteur
        WHERE nom = pnom and prenom = pprenom)
    order by numvin;
vnumvin vin.numvin%type;
cpt number;
BEGIN
    if pnom is null then
        dbms_output.put_line('Veuillez donner le nom du producteur');
    elsif pprenom is null then
        dbms_output.put_line('Veuillez donner le prénom du producteur');
    else
        SELECT region INTO vregion FROM producteur
        WHERE nom = pnom and prenom = pprenom;
        dbms_output.put_line('La région du producteur (' || pnom || ',' || pprenom || ')  est ' ||
vregion);
        vnumvin := 'bad';

        for enr_recoltes in cur_recoltes
        loop
            if vnumvin <> enr_recoltes.numvin then
                dbms_output.put_line('Recoltes du ' || carac_vin(enr_recoltes.numvin));
                cpt := 1;
            end if;
            vnumvin := enr_recoltes.numvin;
            dbms_output.put_line('Recolte ' || cpt || ' : ' || enr_recoltes.quantite || '
bouteille(s)');
            cpt := cpt + 1;
        end loop;
    END IF;
END reporting_recoltes;
/
select * from user_errors WHERE name = 'REPORTING_RECOLTES';
```

.

Question 7)

Énoncé de la question

Dans le cadre d'une utilisation plus globale d'une gestion de l'agriculture, on vous demande de créer un package nommé pck_prod_vin qui regroupe ces procédures et fonctions.

Solution de la question

Dans ce package, on a déclaré une variable globale au package pour gérer proprement un code d'erreur.

```
--------------------------------
-- La déclaration du PACKAGE --
--------------------------------
CREATE OR REPLACE PACKAGE TP2 AS
    -- Variable Globale
    errnum NUMBER := -20001 ;

    PROCEDURE creer_producteur(pnom varchar2, pprenom varchar2, pregion varchar2);
    PROCEDURE creer_producteur2(pnom varchar2, pprenom varchar2, pregion varchar2);
    PROCEDURE reporting_recoltes(pnom varchar2, pprenom varchar2);
    PROCEDURE reporting_recoltes2(pnom varchar2, pprenom varchar2);
    FUNCTION Carac_Vin(pnumvin varchar2) RETURN varchar2;
END TP2;
/
------------------------
-- Le corps du PACKAGE --
------------------------
CREATE OR REPLACE PACKAGE BODY TP2 AS
    PROCEDURE creer_producteur(pnom varchar2, pprenom varchar2, pregion varchar2) IS
        cpt number;
    BEGIN
        IF pnom is null THEN
            dbms_output.put_line('Veuillez donner le nom du producteur');
        ELSIF pprenom is null THEN
            dbms_output.put_line('Veuillez donner le prénom du producteur');
        ELSIF pregion is null THEN
            dbms_output.put_line('Veuillez donner la région du producteur');
        ELSE
            SELECT count(*) INTO cpt FROM producteur
            WHERE nom = pnom OR prenom = pprenom;
            IF cpt > 0 THEN
                dbms_output.put_line('Le couple ('|| pnom|| ','|| pprenom|| ') existe déjà');
            ELSE
                INSERT INTO Producteur(nom,prenom,region)
VALUES(pnom,pprenom,pregion);
                dbms_output.put_line('Insertion réussie');
            END IF ;
        END IF ;
```

```
    END creer_producteur;
    PROCEDURE creer_producteur2(pnom varchar2, pprenom varchar2, pregion varchar2)
IS
        cpt number;
    BEGIN
        IF pnom is null THEN
            raise_application_error(errnum,'Veuillez donner le nom du producteur');
        ELSIF pprenom is null THEN
            raise_application_error(errnum,'Veuillez donner le prénom du producteur');
        ELSIF pregion is null THEN
            raise_application_error(errnum,'Veuillez donner la région du producteur');
        ELSE
            SELECT count(*) INTO cpt FROM producteur
            WHERE nom = pnom OR prenom = pprenom;
            IF cpt > 0 THEN
                raise_application_error(errnum,'Le couple ('|| pnom|| ','|| pprenom|| ') existe
déjà');
            ELSE
                INSERT INTO Producteur(nom,prenom,region)
VALUES(pnom,pprenom,pregion);
                dbms_output.put_line('Insertion réussie');
            END IF ;
        END IF ;
    END creer_producteur2;
    PROCEDURE reporting_recoltes(pnom varchar2, pprenom varchar2) IS
        vregion producteur.region%type;
        CURSOR cur_recoltes IS
            SELECT * FROM recolte
                WHERE numprod IN (SELECT numprod FROM producteur
                                WHERE nom = pnom AND prenom = pprenom)
                ORDER BY numvin;
        vnumvin vin.numvin%type;
        cpt NUMBER;
    BEGIN
        IF pnom is null THEN
            dbms_output.put_line('Veuillez donner le nom du producteur');
        ELSIF pprenom is null THEN
            dbms_output.put_line('Veuillez donner le prénom du producteur');
        ELSE
            SELECT region INTO vregion FROM producteur
                WHERE nom = pnom OR prenom = pprenom;
```

```
            dbms_output.put_line('La région du producteur ('|| pnom|| ','|| pprenom|| ') est '||
vregion);
            vnumvin := 'bad';
            FOR enr_recoltes IN cur_recoltes
            LOOP
                IF vnumvin <> enr_recoltes.numvin THEN
                    dbms_output.put_line('Recoltes de '|| enr_recoltes.numvin);
                    cpt := 1;
                END IF ;
                vnumvin := enr_recoltes.numvin;
                dbms_output.put_line('Recolte '|| cpt|| ' : '|| enr_recoltes.quantite|| '
bouteille(s)');
                cpt := cpt + 1;
            END LOOP;
        END IF ;
    END reporting_recoltes;
    PROCEDURE reporting_recoltes2(pnom varchar2, pprenom varchar2) is
        vregion producteur.region%type;
        CURSOR cur_recoltes is
            SELECT * FROM recolte
                WHERE numprod IN (SELECT numprod FROM producteur
                WHERE nom = pnom OR prenom = pprenom)
                ORDER BY numvin;
        vnumvin vin.numvin%type;
        cpt number;
    BEGIN
        IF pnom is null THEN
            dbms_output.put_line('Veuillez donner le nom du producteur');
        ELSIF pprenom is null THEN
            dbms_output.put_line('Veuillez donner le prénom du producteur');
        ELSE
            SELECT region INTO vregion FROM producteur
                WHERE nom = pnom OR prenom = pprenom;
            dbms_output.put_line('La région du producteur ('|| pnom|| ','|| pprenom|| ') est ' ||
vregion);
            vnumvin := 'bad';
            FOR enr_recoltes IN cur_recoltes
            LOOP
                IF vnumvin <> enr_recoltes.numvin THEN
                    dbms_output.put_line('Recoltes du '||
carac_vin(enr_recoltes.numvin));
                    cpt := 1;
```

```
                    END IF ;
                    vnumvin := enr_recoltes.numvin;
                    dbms_output.put_line('Recolte '|| cpt|| ' : '|| enr_recoltes.quantite|| '
bouteille(s)');
                    cpt := cpt + 1;
                END LOOP;
            END IF ;
        END reporting_recoltes2;
        FUNCTION Carac_Vin(pnumvin varchar2) RETURN varchar2 IS
            cpt number;
            res varchar2(50);
        BEGIN
            SELECT count(*) INTO cpt FROM vin WHERE numvin = pnumvin;
            IF cpt = 0 THEN
                res := 'vin inconnu';
            ELSE
                SELECT 'Cru : '|| cru|| ' -- degré : '|| degre|| ' -- annee : '|| annee INTO res FROM
vin
                    WHERE numvin = pnumvin;
            END IF ;
            return res;
        END Carac_Vin;
END TP2;
/
```

.