TP 4 : PLSQL

Solution :

Supposons que les tables des TP précédents sont crées et remplies.

1. **Ecrire un bloc PL/SQL qui affiche la culture_principale et la taille_exploitation de l'agriculteur 'Benali Ahmed'.**

   Si taille_exploitation est supérieure à 10 hectares, afficher : "Grande exploitation" sinon afficher "Petite exploitation". Exemple : 'Benali Ahmed a une Petite exploitation (5 hectars) de tomates'
   Gérer l'exception NO_DATA_FOUND et afficher : "Aucun agriculteur trouvé avec ce nom."

```
SET SERVEROUTPUT ON;
--------------------------------------------------------------------------------
-- 1. Bloc PL/SQL : infos sur Benali Ahmed
--------------------------------------------------------------------------------
DECLARE
    v_taille NUMBER;
    v_culture VARCHAR2(100);
    v_nom VARCHAR2(100) := 'Benali Ahmed';
BEGIN
    SELECT taille_exploitation, culture_principale
    INTO v_taille, v_culture
    FROM Agriculteur
    WHERE nom = v_nom;

    IF v_taille > 10 THEN
        dbms_output.put_line(v_nom || ' a une Grande exploitation ('
        || v_taille || ' hectares) de ' || v_culture);
    ELSE
        dbms_output.put_line(v_nom || ' a une Petite exploitation ('
        || v_taille || ' hectares) de ' || v_culture);
    END IF;

EXCEPTION
    WHEN NO_DATA_FOUND THEN
        dbms_output.put_line('Aucun agriculteur trouve avec ce nom.');
END;
/
```

   SQLPlus>

```
Benali Ahmed a une Petite exploitation (5 hectares) de tomates

PL/SQL procedure successfully completed.
```

2. **Refaire la question 1 pour l'ensemble des agriculteurs. Utiliser un curseur qui parcourt la table Agriculteur.**

```sql
-- 2. Bloc PL/SQL pour tous les agriculteurs avec curseur
--------------------------------------------------------------------------------
DECLARE
    CURSOR cr IS
        SELECT nom, taille_exploitation, culture_principale
        FROM Agriculteur;
BEGIN
    FOR i IN cr LOOP
        IF i.taille_exploitation > 10 THEN
            dbms_output.put_line(i.nom || ' : Grande exploitation ('
            ||i.taille_exploitation || ' hectares) de ' ||i.culture_principale);
        ELSE
            dbms_output.put_line(i.nom || ' : Petite exploitation ('
            ||i.taille_exploitation || ' hectares) de ' ||i.culture_principale);
        END IF;
    END LOOP;
END;
/
```

SQLPlus>

```
Benali Ahmed : Petite exploitation (5 hectares) de tomates
Khelifi Samira : Grande exploitation (12 hectares) de ble dur
Touati Mourad : Petite exploitation (8 hectares) de pommes de terre
Zerrouki Fatma : Petite exploitation (3 hectares) de olives
Bensaid Rachid : Grande exploitation (15 hectares) de dattes

PL/SQL procedure successfully completed.
```

3. **Créer un bloc PL/SQL qui :**
   a. **insère une nouvelle ligne dans la table Production (6, 4, 3, 10000, 'hiver 2024')**
   b. **Met à jour la quantité_produite de la production 3 à 15000. .**
   c. **Supprimer de la table Approvisionnement toutes les lignes dont la date_approvisionnement est antérieure à '01-03-2024'**

```sql
-- 3. Bloc PL/SQL : insertion, mise a jour et suppression
--------------------------------------------------------------------------------
BEGIN
    -- a. Inserer
    INSERT INTO Production(production_id, agriculteur_id, produit_id, quantite_produite,
                           saison)
    VALUES (6, 4, 3, 10000, 'hiver 2024');

    -- b. Maj
    UPDATE Production
    SET quantite_produite = 15000
    WHERE production_id = 3;

    -- c. Suppression
    DELETE FROM Approvisionnement
    WHERE date_approvisionnement < TO_DATE('01-03-2024','DD-MM-YYYY');

    COMMIT;
END;
/
```

SQLPlus>

```
PL/SQL procedure successfully completed.
```

- ajoute une nouvelle ligne dans la table **Production** :

```
SQL> SELECT * FROM Production WHERE production_id = 6;

PRODUCTION_ID AGRICULTEUR_ID PRODUIT_ID QUANTITE_PRODUITE
------------- -------------- ---------- -----------------
SAISON
--------------------------------------------------------------------
            6              4          3             10000
hiver 2024
```

- modifie la quantité produite pour la production n°3 : avant 20000 apres 15000

```
SQL> SELECT *FROM Production WHERE production_id = 3;

PRODUCTION_ID AGRICULTEUR_ID PRODUIT_ID QUANTITE_PRODUITE
------------- -------------- ---------- -----------------
SAISON
--------------------------------------------------------------------
            3              3          3             15000
hiver 2024
```

- supprime toutes les lignes plus anciennes que le 1er mars 2024.

```
SQL> SELECT * FROM Approvisionnement WHERE date_approvisionnement < DATE '2024-03-01';

no rows selected
```

4. **Créer une fonction total_production_agriculteur qui retourne la somme totale des quantités produites pour un agriculteur donné. Afficher un message d'erreur si l'agriculteur mentionné n'existe pas. Exécuter la fonction pour tous les agriculteurs.**

```sql
-- 4.a Fonction total_production_agriculteur
--------------------------------------------------------------------
CREATE OR REPLACE FUNCTION total_production_agriculteur(id_agri NUMBER)
RETURN NUMBER
IS
    total NUMBER;
    v_count NUMBER;
BEGIN
    SELECT COUNT(*) INTO v_count
    FROM Agriculteur
    WHERE agriculteur_id = id_agri;

    IF v_count = 0 THEN
        raise_application_error(-20001, 'Agriculteur inexistant');
    END IF;

    SELECT NVL(SUM(quantite_produite),0)
    INTO total
    FROM Production
    WHERE agriculteur_id = id_agri;

    RETURN total;

END;
/
```

SQLPlus>

```
Function created.
```

```
-- 4.b Executer la fonction pour tous les agriculteurs
--------------------------------------------------------------------------------
DECLARE
    CURSOR cr IS SELECT agriculteur_id, nom FROM Agriculteur;
    total NUMBER;
BEGIN
    FOR i IN cr LOOP
        total := total_production_agriculteur(i.agriculteur_id);
        dbms_output.put_line('Total production de ' || i.nom || ' = ' || total);
    END LOOP;
END;
/
```

```
SQL> DECLARE
  2      CURSOR cr IS SELECT agriculteur_id, nom FROM Agriculteur;
  3      total NUMBER;
  4  BEGIN
  5      FOR i IN cr LOOP
  6          total := total_production_agriculteur(i.agriculteur_id);
  7          dbms_output.put_line('Total production de ' || i.nom || ' = ' || total);
  8      END LOOP;
  9  END;
 10  /
Total production de Benali Ahmed = 15000
Total production de Khelifi Samira = 30000
Total production de Touati Mourad = 15000
Total production de Zerrouki Fatma = 18000
Total production de Bensaid Rachid = 25000

PL/SQL procedure successfully completed.
```

5. **Créer une procédure qui permet d'insérer une production à partir de tous les attributs nécessaires. N'oublier pas de vérifier l'unicité de la clé et l'existence de clé étrangère vers les tables référencées. Affichez les messages d'erreurs en cas de problèmes.**

```sql
-- 5. Procedure d'insertion securisee pour Production
--------------------------------------------------------------------------------
CREATE OR REPLACE PROCEDURE inserer_production(
    p_production_id NUMBER,
    p_agriculteur_id NUMBER,
    p_produit_id NUMBER,
    p_quantite NUMBER,
    p_saison VARCHAR2
) IS
    v_count NUMBER;
BEGIN
    -- Verifier unicite cle
    SELECT COUNT(*) INTO v_count
    FROM Production
    WHERE production_id = p_production_id;

    IF v_count > 0 THEN
        raise_application_error(-20002, 'Erreur : production_id deja utilise.');
    END IF;

    -- Verifier cle etrangere agriculteur
    SELECT COUNT(*) INTO v_count
    FROM Agriculteur
    WHERE agriculteur_id = p_agriculteur_id;

    IF v_count = 0 THEN
        raise_application_error(-20003, 'Erreur : agriculteur_id inexistant.');
    END IF;

    -- Vérifier cle etrangere produit
    SELECT COUNT(*) INTO v_count
    FROM Produit_Alimentaire
    WHERE produit_id = p_produit_id;

    IF v_count = 0 THEN
        raise_application_error(-20004, 'Erreur : produit_id inexistant.');
    END IF;

    -- Insertion finale
    INSERT INTO Production VALUES(
        p_production_id,
        p_agriculteur_id,
        p_produit_id,
        p_quantite,
        p_saison
    );

    dbms_output.put_line('Insertion reussie.');

EXCEPTION
    WHEN OTHERS THEN
        dbms_output.put_line('Erreur : ' || SQLERRM);
END;
/
```

```
Procedure created.

SQL> EXEC inserer_production(10, 1, 2, 5000, 'printemps 2025');
Insertion reussie.

PL/SQL procedure successfully completed.

SQL> SELECT * FROM Production WHERE production_id = 10;

PRODUCTION_ID AGRICULTEUR_ID PRODUIT_ID QUANTITE_PRODUITE
------------- -------------- ---------- -----------------
SAISON
--------------------------------------------------------------------------------
           10              1          2              5000
printemps 2025
```