Nom : Benayhya

Prenom : Othman

Filiere : Genie informatique 2

Numero : 11

TP3_PL/SQL

Exercice 1 :

```plsql
CREATE OR REPLACE FUNCTION calculer_stock_restant (
  p_num_article IN Article.num_a%TYPE
) RETURN NUMBER IS
  v_stock NUMBER;
BEGIN

  SELECT STOCK INTO v_stock
  FROM Article
  WHERE num_a = p_num_article;

  RETURN v_stock;
EXCEPTION
  WHEN NO_DATA_FOUND THEN

    RETURN NULL;
END calculer_stock_restant;
```

```sql
SET SERVEROUTPUT ON;
DECLARE
    v_stock_restant number;
    v_num_article ARTICLE.num_A%TYPE := '&num_article';
BEGIN
    v_stock_restant := calculer_stock_restant(v_num_article);
    DBMS_OUTPUT.PUT_LINE('stock restant pour larticle ' || v_num_article || ' est : '||v_stock_restant);
End;
/
```

```
stock restant pour larticle 2est : 240


PL/SQL procedure successfully completed.
```

Exercice 2 :

```sql
CREATE OR REPLACE PROCEDURE inserer_client(
    p_num_c CLIENT.num_c%TYPE,
    p_nom   CLIENT.nom%TYPE,
    p_ville CLIENT.ville%TYPE
)
IS
    v_exists NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_exists
    FROM CLIENT
    WHERE num_c = p_num_c;

    IF v_exists > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : Le client existe déjà.');
        RETURN;
    END IF;

    INSERT INTO CLIENT(num_c, nom, ville)
    VALUES (p_num_c, p_nom, p_ville);

    DBMS_OUTPUT.PUT_LINE('Client inséré avec succès.');
END;
/
```

```
begin
   inserer_client(
      2,
      'Othman',
      'Rabat'
   );
end;
/
```

```
Client inséré avec succès.


PL/SQL procedure successfully completed.

Erreur : Le client existe déjà.


PL/SQL procedure successfully completed.
```

Exercice 3 :

```sql
CREATE OR REPLACE FUNCTION generer_rapport_ventes(
    p_date_debut DATE,
    p_date_fin   DATE
) RETURN VARCHAR2
IS
    v_total_ventes NUMBER := 0;
    v_revenu_total NUMBER := 0;
BEGIN
    SELECT
        NVL(COUNT(*), 0),
        NVL(SUM(quantite * prix_vente), 0)
    INTO
        v_total_ventes,
        v_revenu_total
    FROM vente
    WHERE date_vente BETWEEN p_date_debut AND p_date_fin;

    RETURN 'Total ventes = ' || v_total_ventes ||
           ' | Revenu total = ' || v_revenu_total || ' MAD';
END;
/
```

```sql
DECLARE
    v_result varchar2(500);
BEGIN
    v_result := generer_rapport_ventes(DATE '2025-01-01', DATE '2025-04-01');
    dbms_output.PUT_LINE(v_result);
END;
/
```

```
Total ventes = 4 | Revenu total = 15150


PL/SQL procedure successfully completed.
```

Exercice 4 :

Exercice 1 :

```plsql
CREATE OR REPLACE FUNCTION calculer_stock_restant (
  p_num_article IN Article.num_a%TYPE
) RETURN NUMBER IS
  v_stock NUMBER;
BEGIN

  SELECT STOCK INTO v_stock
  FROM Article
  WHERE num_a = p_num_article;

  RETURN v_stock;
EXCEPTION
  WHEN NO_DATA_FOUND THEN

    RETURN NULL;
END calculer_stock_restant;
```

```plsql
SET SERVEROUTPUT ON;
DECLARE
    v_stock_restant number;
    v_num_article ARTICLE.num_A%TYPE := '&num_article';
BEGIN
    v_stock_restant := calculer_stock_restant(v_num_article);
    DBMS_OUTPUT.PUT_LINE('stock restant pour larticle ' || v_num_article || ' est : '||v_stock_restant);
End;
/
```

```
stock restant pour larticle 2est : 240


PL/SQL procedure successfully completed.
```

Exercice 2 :

```plsql
CREATE OR REPLACE PROCEDURE inserer_client(
    p_num_c CLIENT.num_c%TYPE,
    p_nom   CLIENT.nom%TYPE,
    p_ville CLIENT.ville%TYPE
)
IS
    v_exists NUMBER;
BEGIN
    SELECT COUNT(*)
    INTO v_exists
    FROM CLIENT
    WHERE num_c = p_num_c;

    IF v_exists > 0 THEN
        DBMS_OUTPUT.PUT_LINE('Erreur : Le client existe déjà.');
        RETURN;
    END IF;

    INSERT INTO CLIENT(num_c, nom, ville)
    VALUES (p_num_c, p_nom, p_ville);

    DBMS_OUTPUT.PUT_LINE('Client inséré avec succès.');
END;
/
```

```plsql
begin
   inserer_client(
      2,
      'Othman',
      'Rabat'
   );
end;
/
```

```
Client inséré avec succès.


PL/SQL procedure successfully completed.

Erreur : Le client existe déjà.


PL/SQL procedure successfully completed.
```

Exercice 4 :

```plsql
CREATE OR REPLACE FUNCTION generer_rapport_ventes(
    p_date_debut DATE,
    p_date_fin   DATE
) RETURN VARCHAR2
IS
    v_total_ventes NUMBER := 0;
    v_revenu_total NUMBER := 0;
BEGIN
    SELECT
        NVL(COUNT(*), 0),
        NVL(SUM(quantite * prix_vente), 0)
    INTO
        v_total_ventes,
        v_revenu_total
    FROM vente
    WHERE date_vente BETWEEN p_date_debut AND p_date_fin;

    RETURN 'Total ventes = ' || v_total_ventes ||
           ' | Revenu total = ' || v_revenu_total || ' MAD';
END;
/
```

Code Fonction Version 2, sans requête imbriquée

```sql
CREATE OR REPLACE FUNCTION rapport_articles_fournisseur(
    p_num_f FRS.num_f%TYPE
) RETURN VARCHAR2
IS
    v_result VARCHAR2(4000) := '';
BEGIN
    FOR rec IN (
        SELECT *
        FROM (
            SELECT a.num_a,
                   a.des,
                   NVL((SELECT SUM(v.qte)
                        FROM VENTE v
                        WHERE v.num_a = a.num_a), 0) AS total_vendu
            FROM ARTICLE a
            WHERE a.num_f = p_num_f
        )
        ORDER BY total_vendu DESC
    )
    WHERE ROWNUM <= 3
    LOOP
        v_result := v_result ||
                    'Article ' || rec.num_a || ' - ' ||
                    rec.des || ' : ' || rec.total_vendu || ' vendus' || CHR(10);
    END LOOP;

    IF v_result IS NULL THEN
        RETURN 'Aucun article trouvé pour ce fournisseur.';
    END IF;
    RETURN v_result;
END;
/
```

Exercice 5 :

Code fonction :

```sql
CREATE OR REPLACE FUNCTION chiffre_affaires_fournisseur(
    p_num_f FRS.num_f%TYPE
) RETURN NUMBER
IS
    CURSOR c_art IS
        SELECT a.num_a
        FROM ARTICLE a
        WHERE a.num_f = p_num_f;

    v_ca NUMBER := 0;
    v_qte NUMBER;
    v_prix NUMBER;
    v_num_a ARTICLE.num_a%TYPE;
    v_sold NUMBER := 0;
BEGIN
    OPEN c_art;
    LOOP
        FETCH c_art INTO v_num_a;
        EXIT WHEN c_art%NOTFOUND;
        SELECT NVL(SUM(qte), 0),
               NVL(SUM(qte * prix_vente), 0)
        INTO v_qte, v_prix
        FROM VENTE
        WHERE num_a = v_num_a;
        v_sold := v_sold + v_qte;
        v_ca := v_ca + v_prix;
    END LOOP;
    CLOSE c_art;
    IF v_sold = 0 THEN
        RAISE_APPLICATION_ERROR(-20050,
            'Aucune vente pour ce fournisseur.');
    END IF;
    RETURN v_ca;
END;
/
```

```
DECLARE
    v_result number;

BEGIN
    v_result := chiffre_affaires_fournisseur(1);
    dbms_output.PUT_LINE(v_result);

END;
/
```

Exercice 6 :

```sql
CREATE OR REPLACE FUNCTION marge_beneficiaire_article(
    p_num_f FRS.num_f%TYPE
) RETURN NUMBER
IS
    CURSOR c_art IS
        SELECT a.num_a, a.prix_achat
        FROM ARTICLE a
        WHERE a.num_f = p_num_f;

    v_num_a   ARTICLE.num_a%TYPE;
    v_prix_a  ARTICLE.prix_achat%TYPE;
    v_prix_v  NUMBER;
    v_marge_tot NUMBER := 0;
    v_count NUMBER := 0;

-- fonction principale

BEGIN
    OPEN c_art;
    LOOP
        FETCH c_art INTO v_num_a, v_prix_a;
        EXIT WHEN c_art%NOTFOUND;
        SELECT NVL(AVG(prix_vente), 0)
        INTO v_prix_v
        FROM VENTE
        WHERE num_a = v_num_a;
        IF v_prix_v > 0 THEN
            v_marge_tot := v_marge_tot + (v_prix_v - v_prix_a);
            v_count := v_count + 1;
        END IF;
    END LOOP;
    CLOSE c_art;
    IF v_count = 0 THEN
        RAISE_APPLICATION_ERROR(-20051,
            'Aucun article vendu pour ce fournisseur.');
    END IF;
    RETURN v_marge_tot / v_count;
END;
/

DECLARE
    v_result number;

BEGIN
    v_result := marge_beneficiaire_article(1);
    dbms_output.PUT_LINE(v_result);
```

```
46        dbms_output.put_line(v_result);
47
48    END;
49    /
50
```