TD-TP : les Procédures et les Fonctions

On s'intéresse au système d'information d'une société de vente de voitures. Les tables suivantes font partie de cette base de données :

Voiture (code_voiture, marque, type, couleur, prixvente)
Client (code_client, nom, sexe, ville)
Vente (num_vente, code_client#, code_voiture#, datevente)

Table Voiture

code_voiture	marque	type	couleur	prixvente
PE106	Peugeot	206	Rouge	10000
OP107	Opel	Astra	Noire	70000
CI277	Citroen	C5	Rouge	45000

Table Client

code_client	nom	sexe	ville
C600	Irma	F	Lille
C900	Irma	F	Barcelone
C800	Julio	M	Barcelone

Table Vente

num_vente	code_client	code_voiture	datevente
VE801	C600	CI277	03-09-2005
VE271	C900	PE106	11-07-2001
VE402	C800	OP107	13-06-2008

Exercice 1:

1) Ecrire un bloc PL/SQL permettant d'afficher le nombre total de voitures, le nombre de voitures de couleur rouge ainsi que le pourcentage des voitures de couleur rouge.

```
SQL> DECLARE
  2    v_total_voitures NUMBER;
  3    v_voitures_rouges NUMBER;
  4    v_pourcentage NUMBER;
  5  BEGIN
  6
  7    SELECT COUNT(*) INTO v_total_voitures FROM Voiture;
  8
  9    SELECT COUNT(*) INTO v_voitures_rouges FROM Voiture WHERE couleur = 'Rouge';
 10
 11    v_pourcentage := (v_voitures_rouges / v_total_voitures) * 100;
 12
 13
 14    DBMS_OUTPUT.PUT_LINE('Nombre total de voitures: ' || v_total_voitures);
 15    DBMS_OUTPUT.PUT_LINE('Nombre de voitures rouges: ' || v_voitures_rouges);
 16    DBMS_OUTPUT.PUT_LINE('Pourcentage de voitures rouges: ' || ROUND(v_pourcentage, 2) ||
'%');
 17  END;
 18  /
Nombre total de voitures: 3
Nombre de voitures rouges: 2
Pourcentage de voitures rouges: 66,67%

ProcÚdure PL/SQL terminÚe avec succÞs.
```

2) Ecrire une procédure qui prend en entrée une couleur et qui permet d'afficher le pourcentage de voitures ayant cette couleur.

```
SQL> SET SERVEROUTPUT ON;
SQL> CREATE OR REPLACE PROCEDURE pourcentage_par_couleur(p_couleur IN VARCHAR2) IS
  2    v_total_voitures NUMBER;
  3    v_voitures_couleur NUMBER;
  4    v_pourcentage NUMBER;
  5  BEGIN
  6
  7    SELECT COUNT(*) INTO v_total_voitures FROM Voiture;
  8
  9    SELECT COUNT(*) INTO v_voitures_couleur FROM Voiture WHERE couleur = p_couleur;
 10
 11    IF v_total_voitures > 0 THEN
 12      v_pourcentage := (v_voitures_couleur / v_total_voitures) * 100;
 13    ELSE
 14      v_pourcentage := 0;
 15    END IF;
 16
 17    DBMS_OUTPUT.PUT_LINE('Pourcentage de voitures de couleur ' || p_couleur || ': ' ||
 18                         ROUND(v_pourcentage, 2) || '%');
 19  EXCEPTION
 20    WHEN NO_DATA_FOUND THEN
 21      DBMS_OUTPUT.PUT_LINE('Aucune voiture trouvée.');
 22    WHEN OTHERS THEN
 23      DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
 24  END pourcentage_par_couleur ;/
ProcÚdure crÚÚe.
SQL> BEGIN
  2    pourcentage_par_couleur('Rouge');
  3  END ;/
Pourcentage de voitures de couleur Rouge: 66,67%
ProcÚdure PL/SQL terminÚe avec succÞs.
```

3) Ecrire un bloc PL/SQL permettant d'afficher le pourcentage des voitures de chaque couleur. Utiliser la procédure crée ci-dessus.

```
SQL> DECLARE
  2    CURSOR c_couleurs IS
  3      SELECT DISTINCT couleur FROM Voiture;
  4  BEGIN
  5    DBMS_OUTPUT.PUT_LINE('Pourcentage des voitures par couleur:');
  6    DBMS_OUTPUT.PUT_LINE('-----------------------------------');
  7
  8    FOR r_couleur IN c_couleurs LOOP
  9      pourcentage_par_couleur(r_couleur.couleur);
 10    END LOOP;
 11  END;
 12  /
Pourcentage des voitures par couleur:
-----------------------------------
Pourcentage de voitures de couleur Rouge: 66,67%
Pourcentage de voitures de couleur Noire: 33,33%

ProcÚdure PL/SQL terminÚe avec succÞs.
```

Exercice 2:

1) Ecrire une fonction SUM_prixvente qui permet de retourner la somme des prix de vente des voitures d'une marque donnée.

```
SQL> CREATE OR REPLACE FUNCTION SUM_prixvente(p_marque IN VARCHAR2)
  2  RETURN NUMBER IS
  3    v_somme NUMBER := 0;
  4  BEGIN
  5    SELECT SUM(prixvente) INTO v_somme
  6    FROM Voiture
  7    WHERE marque = p_marque;
  8
  9    RETURN v_somme;
 10  EXCEPTION
 11    WHEN NO_DATA_FOUND THEN
 12      RETURN 0;
 13    WHEN OTHERS THEN
 14      DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
 15      RETURN NULL;
 16  END SUM_prixvente;
 17  /

Fonction crÚÚe.

SQL> SELECT SUM_prixvente('Peugeot') FROM dual;

SUM_PRIXVENTE('PEUGEOT')
------------------------
                   10000
```

2) Ecrire une fonction MOY_prixvente qui permet de retourner la moyenne des prix de vente des voitures d'une marque donnée.

```
SQL> CREATE OR REPLACE FUNCTION MOY_prixvente(p_marque IN VARCHAR2)
 2  RETURN NUMBER IS
 3    v_moyenne NUMBER := 0;
 4  BEGIN
 5    SELECT AVG(prixvente) INTO v_moyenne
 6    FROM Voiture
 7    WHERE marque = p_marque;
 8
 9
10    RETURN v_moyenne;
11  EXCEPTION
12    WHEN NO_DATA_FOUND THEN
13      RETURN 0;
14    WHEN OTHERS THEN
15      DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
16      RETURN NULL;
17  END MOY_prixvente;
18  /

Fonction crÚÚe.

SQL> SELECT MOY_prixvente('Peugeot') FROM dual;

MOY_PRIXVENTE('PEUGEOT')
------------------------
                   10000

SQL>
```

3) Ecrire un bloc PL/SQL permettant d'augmenter de 20% les prix de vente des voitures d'une marque donnée si leur prix de vente est inférieur à la moyenne des prix de vente des voitures ayant cette marque.

```
SQL> DECLARE
  2    v_marque VARCHAR2(50) := '&marque';
  3    v_moyenne NUMBER;
  4    v_compteur NUMBER := 0;
  5  BEGIN
  6    v_moyenne := MOY_prixvente(v_marque);
  7
  8    DBMS_OUTPUT.PUT_LINE('Moyenne des prix pour ' || v_marque || ': ' || v_moyenne);
  9
 10    UPDATE Voiture
 11    SET prixvente = prixvente * 1.2
 12    WHERE marque = v_marque
 13    AND prixvente < v_moyenne
 14    RETURNING COUNT(*) INTO v_compteur;
 16    DBMS_OUTPUT.PUT_LINE('Nombre de voitures mises à jour: ' || v_compteur);
 17
 18          DBMS_OUTPUT.PUT_LINE('Nouvelle moyenne des prix pour ' || v_marque || ': ' ||
MOY_prixvente(v_marque));
 20    COMMIT;
 21  EXCEPTION
 22    WHEN OTHERS THEN
 23      DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
 24      ROLLBACK;
 25  END;/
Entrez une valeur pour marque : bmw
ancien   2 :   v_marque VARCHAR2(50) := '&marque';
nouveau   2 :   v_marque VARCHAR2(50) := 'bmw';
ProcÚdure PL/SQL terminÚe avec succÞs.
```

Exercice 3:

1) Ecrire une procédure permettant d'afficher toutes les dates d'achat avec les marques de voitures correspondantes relatives à un client de code donné.

```
SQL> CREATE OR REPLACE PROCEDURE afficher_ach_client(p_code_client IN VARCHAR2) IS
 2  CURSOR c_achats IS
 3   SELECT v.datevente, vt.marque
 4   FROM Vente v
 5   JOIN Voiture vt ON v.code_voiture = vt.code_voiture
 6   WHERE v.code_client = p_code_client
 7   ORDER BY v.datevente;
 8
 9  v_nom_client VARCHAR2(50);
10   v_count NUMBER := 0;
11 BEGIN
12  BEGIN
13   SELECT nom INTO v_nom_client
14   FROM Client
15   WHERE code_client = p_code_client;
16  EXCEPTION
17   WHEN NO_DATA_FOUND THEN
18    DBMS_OUTPUT.PUT_LINE('Aucun client trouve adans le code: ' || p_code_client);
19    RETURN;
20  END;
21
22  DBMS_OUTPUT.PUT_LINE('Achats du client: ' || v_nom_client || ' (' || p_code_client || ')');
23  DBMS_OUTPUT.PUT_LINE('----------------------------------------');
24
25  -- Parcourir et afficher les achats
26  FOR r_achat IN c_achats LOOP
27   DBMS_OUTPUT.PUT_LINE('Date: ' || TO_CHAR(r_achat.datevente, 'DD-MM-YYYY') ||
28                ' | Marque: ' || r_achat.marque);
29   v_count := v_count + 1;
```

```
30   END LOOP;
31
32   IF v_count = 0 THEN
33     DBMS_OUTPUT.PUT_LINE('Ce client ne pas effectu aucun achat.');
34   ELSE
35     DBMS_OUTPUT.PUT_LINE('----------------------------------------');
36     DBMS_OUTPUT.PUT_LINE('Total des achats: ' || v_count);
37   END IF;
38 EXCEPTION
39   WHEN OTHERS THEN
40     DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
41 END afficher_achats_client;
42 /
```

Avertissement : ProcÚdure crÚÚe avec erreurs de compilation.

2) Ecrire une procédure permettant d'afficher toutes les informations relatives aux clients de nom donné en entrée en affichant pour chacun les marques des voitures achetées avec les dates d'achats correspondantes. Utiliser la procédure de la question1.

SQL> CREATE OR REPLACE PROCEDURE afficher_clients_par_nom(p_nom IN VARCHAR2) IS

```
 2   CURSOR c_clients IS
 3     SELECT code_client, nom, sexe, ville
 4     FROM Client
 5     WHERE nom = p_nom;
 6
 7   v_count NUMBER := 0;
 8 BEGIN
 9   DBMS_OUTPUT.PUT_LINE('Informations des clients nommés ''' || p_nom || ''':');
10   DBMS_OUTPUT.PUT_LINE('----------------------------------------');
11
12   FOR r_client IN c_clients LOOP
13     DBMS_OUTPUT.PUT_LINE('Code client: ' || r_client.code_client);
14     DBMS_OUTPUT.PUT_LINE('Nom: ' || r_client.nom);
```

```
15      DBMS_OUTPUT.PUT_LINE('Sexe: ' || r_client.sexe);
16      DBMS_OUTPUT.PUT_LINE('Ville: ' || r_client.ville);
17      DBMS_OUTPUT.PUT_LINE('----------------------------------------');
18
19      afficher_ach_client(r_client.code_client);
20
21
DBMS_OUTPUT.PUT_LINE('========================================');
22      v_count := v_count + 1;
23    END LOOP;
24
25    IF v_count = 0 THEN
26      DBMS_OUTPUT.PUT_LINE('Aucun client trouvé avec le nom: ' || p_nom);
27    ELSE
28      DBMS_OUTPUT.PUT_LINE('Nombre total de clients: ' || v_count);
29    END IF;
30  EXCEPTION
31    WHEN OTHERS THEN
32      DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
33  END afficher_clients_par_nom;
34  /
```

Avertissement : ProcÚdure crÚÚe avec erreurs de compilation.

3) Ecrire un bloc PL/SQL permettant d'afficher pour tous les noms de clients, toutes les informations relatives à ce client, ainsi que la liste des dates d'achats avec les marques des voitures correspondantes. Utiliser la procédure de la question2. DECLARE

```
4)    2   CURSOR c_noms_distincts IS
5)    3     SELECT DISTINCT nom
6)    4     FROM Client
7)    5     ORDER BY nom;
8)    6  BEGIN
9)    7   DBMS_OUTPUT.PUT_LINE('RAPPORT COMPLET DES CLIENTS ET LEURS ACHATS');
10)   8   DBMS_OUTPUT.PUT_LINE('========================================');
```

```
11)
12)   FOR r_nom IN c_noms_distincts LOOP
13)     afficher_clients_par_nom(r_nom.nom);
14)     DBMS_OUTPUT.PUT_LINE('========================================');
15)   END LOOP;
16) EXCEPTION
17)   WHEN OTHERS THEN
18)     DBMS_OUTPUT.PUT_LINE('Erreur: ' || SQLERRM);
19) END;
20) /
```