

Nom: Akrouchi Rihab
Matricule: 202031050231
groupe: Master 2 IL groupe 2

TP SDED 5

Introduction:

Dans le cadre de l'apprentissage des bases de données relationnelles et de leur gestion, ce travail a pour objectif de modéliser, créer et manipuler une base de données à l'aide du langage SQL. Le projet consiste en la mise en place de plusieurs tables interconnectées représentant un système de gestion (par exemple, gestion de véhicules, trajets, pannes, etc.), ainsi que le remplissage automatique de ces tables avec des données générées dynamiquement.

L'objectif principal est de mettre en œuvre les concepts suivants :

- Création de schémas relationnels
- Utilisation des contraintes d'intégrité
- Remplissage automatique des tables
- Manipulation des données avec des requêtes SQL

Table Temps :

```
SQL> -- Table Temps
SQL> CREATE TABLE Temps (
  2     CodeT NUMBER PRIMARY KEY,
  3     Jour DATE NOT NULL,
  4     LibJour VARCHAR2(20),
  5     Mois VARCHAR2(20),
  6     LibMois VARCHAR2(20),
  7     Annee NUMBER NOT NULL
  8 );
```

Table created.

Table VilleDepart

```
SQL>
SQL> -- Table VilleDepart
SQL> CREATE TABLE VilleDepart (
  2     CodeVilleD NUMBER PRIMARY KEY,
  3     NomVilleD VARCHAR2(50) NOT NULL
  4 );
```

Table created.

Table VilleArrivee :

```
SQL> -- Table VilleArrivee
SQL> CREATE TABLE VilleArrivee (
  2     CodeVilleA NUMBER PRIMARY KEY,
  3     NomVilleA VARCHAR2(50) NOT NULL
  4 );
```

Table created.

Table Performance

```
SQL> -- Table Performance
SQL> CREATE TABLE Performance (
  2     CodeT NUMBER NOT NULL,
  3     CodeVilleD NUMBER NOT NULL,
  4     CodeVilleA NUMBER NOT NULL,
  5     NumVehicule NUMBER NOT NULL,
  6     NBTrajets NUMBER,
  7     Kilometrage NUMBER,
  8     DureeObligatoire INTERVAL DAY(2) TO SECOND(6),
  9     CarburantConsomme NUMBER,
 10     NBpannes NUMBER,
 11     PRIMARY KEY (CodeT, CodeVilleD, CodeVilleA, NumVehicule),
 12     FOREIGN KEY (CodeT) REFERENCES Temps(CodeT),
 13     FOREIGN KEY (CodeVilleD) REFERENCES VilleDepart(CodeVilleD),
 14     FOREIGN KEY (CodeVilleA) REFERENCES VilleArrivee(CodeVilleA),
 15     FOREIGN KEY (NumVehicule) REFERENCES Vehicule(NumVehicule)
 16 );
```

Table created.

Sequence seqTemps

```
SQL> CREATE SEQUENCE seqTemps
  2     MINVALUE 1
  3     MAXVALUE 100000
  4     START WITH 1
  5     INCREMENT BY 1;
```

Sequence created.

Les fonction d'insertion : Trajet

```
SQL> BEGIN
  2     FOR i IN (
  3         SELECT DISTINCT DateTrajet
  4         FROM master.Trajet
  5     ) LOOP
  6         INSERT INTO Temps (CodeT, Jour, LibJour, Mois, LibM
ois, Annee)
  7         VALUES (
  8             seq_Temps.NEXTVAL,
  9             TO_CHAR(i.DateTrajet, 'DD'),
 10             TO_CHAR(i.DateTrajet, 'DAY'),
 11             TO_CHAR(i.DateTrajet, 'MM'),
 12             TO_CHAR(i.DateTrajet, 'MONTH'),
 13             TO_CHAR(i.DateTrajet, 'YYYY')
 14         );
 15     END LOOP;
 16     COMMIT;
 17 END;
 18 /
```

PL/SQL procedure successfully completed.

SQL>

Temps

```
SQL> --table Temps
SQL> BEGIN
2   FOR i IN (
3       SELECT DISTINCT DateTrajet
4       FROM master.Trajet
5   ) LOOP
6       INSERT INTO Temps (CodeT, Jour, LibJour, Mois, LibM
ois, Annee)
7       VALUES (
8           seq_Temps.NEXTVAL,
9           TO_CHAR(i.DateTrajet, 'DD'),
10          TO_CHAR(i.DateTrajet, 'DAY'),
11          TO_CHAR(i.DateTrajet, 'MM'),
12          TO_CHAR(i.DateTrajet, 'MONTH'),
13          TO_CHAR(i.DateTrajet, 'YYYY')
14      );
15  END LOOP;
16  COMMIT;
17 END;
18 /
```

PL/SQL procedure successfully completed.

Vehicule

```
SQL> BEGIN
2   FOR i IN (
3       SELECT
4           v.NumVehicule,
5           v.NumModele,
6           (SELECT m.CodeMarque
7            FROM masterrr.Modele m
8            WHERE m.NumModele = v.NumModele) AS CodeMarque,
9           (SELECT m.NomModele
10            FROM masterrr.Modele m
11            WHERE m.NumModele = v.NumModele) AS NomModele,
12           (SELECT mr.NomMarque
13            FROM masterrr.Marque mr
14            WHERE mr.CodeMarque =
15                (SELECT m.CodeMarque
16                 FROM masterrr.Modele m
17                 WHERE m.NumModele = v.NumModele)) AS NomMarque
18       FROM masterrr.Vehicule v
19   ) LOOP
20       INSERT INTO Vehicule (NumVehicule, NumModele, NomModèle
,NumMarque, Marque)
21       VALUES (i.NumVehicule, i.NumModele, i.NomModele, i.CodeM
arque, i.NomMarque);
22  END LOOP;
23  COMMIT;
24 END;
25 /
```

PL/SQL procedure successfully completed.

Ville

```
SQL> BEGIN
2   FOR i IN (
3     SELECT
4       v.CodeVille,
5       v.NomVille
6     FROM masterrr.Ville v
7   ) LOOP
8     INSERT INTO VilleDepart (CodeVilleD, NomVilleD)
9     VALUES (i.CodeVille, i.NomVille);
10  END LOOP;
11  COMMIT;
12 END;
13 /
```

PL/SQL procedure successfully completed.

Performance

```
PL> BEGIN
2   FOR i IN (
3     SELECT
4       t.CodeVehicule,
5       t.CodeDepart AS CodeVilleD, -- Utilisation de CodeDepart directement
6       t.CodeArrivee AS CodeVilleA, -- Utilisation de CodeArrivee directement
7       SUM(t.KilometrageParcours) AS Kilometrage,
8       SUM(t.DureeTrajet) AS DureeObligatoire,
9       SUM(t.CarburantConsomme) AS CarburantConsomme
10      COUNT(t.CodeTrajet) AS Nbtrajet, -- Nombre de trajets entre les villes
11      TO_CHAR(t.DateTrajet, 'DD') AS Jour,
12      TO_CHAR(t.DateTrajet, 'MM') AS Mois,
13      TO_CHAR(t.DateTrajet, 'YYYY') AS Annee
14    FROM masterrr.Trajet t
15   GROUP BY
16     t.CodeVehicule,
17     t.CodeDepart,
18     t.CodeArrivee,
19     TO_CHAR(t.DateTrajet, 'DD'),
20     TO_CHAR(t.DateTrajet, 'MM'),
21     TO_CHAR(t.DateTrajet, 'YYYY')
22  ) LOOP
23    DECLARE
24      v_CodeT INT;
25      v_NbPannes INT;
26    BEGIN
27      Sélectionner le CodeT basé sur la date du trajet
28      SELECT CodeT
29      INTO v_CodeT
30      FROM Temps
31      WHERE Jour = i.Jour
32      AND Mois = i.Mois
33      AND Annee = i.Annee;
34
35
36      SELECT COUNT(p.CodePanne)
37      INTO v_NbPannes
38      FROM masterrr.Panne p
39      WHERE p.CodeTrajet IN (
40        SELECT t.CodeTrajet
41        FROM masterrr.Trajet t
42        WHERE t.CodeDepart = i.CodeVilleD
43        AND t.CodeArrivee = i.CodeVilleA
44        AND t.CodeVehicule = i.CodeVehicule
45      );
46
47      -- Insertion des données dans la table Performance
48      INSERT INTO Performance (
49        CodeT,
50        CodeVilleD,
51        CodeVilleA,
52        NumVehicule,
53        Nbtrajet,
54        Kilometrage,
55        DureeObligatoire,
56        CarburantConsomme,
57        NbPannes
58      )
59      VALUES (
60        v_CodeT,
61        i.CodeVilleD,
62        i.CodeVilleA,
63        i.CodeVehicule,
64        i.Nbtrajet, -- Nombre de trajets entre ces villes
65        i.Kilometrage, -- Kilométrage total pour ce trajet
66        i.DureeObligatoire, -- Durée totale pour ce trajet
67        i.CarburantConsomme, -- Carburant total consommé
68
69        v_NbPannes -- Nombre total de pannes pour ce trajet
70      );
71    END LOOP;
72  COMMIT;
73 /ND;
74 /
```

/SQL procedure successfully completed.

Conclusion

Ce projet a permis d'explorer les fondamentaux de la gestion des bases de données relationnelles en SQL, allant de la conception du schéma relationnel jusqu'à l'insertion et la manipulation des données. Les compétences acquises, telles que la maîtrise des boucles PL/SQL et des jointures complexes, sont essentielles pour toute application pratique en gestion des données.