

TP SQL – Gestion d'un Petit Système de Réservations

1^{ère} Année Cybersécurité – Euromed University of Fes

Introduction

Ce TP a pour objectif de concevoir et manipuler une petite base de données pour un restaurant. Le système permet d'enregistrer les clients, les tables du restaurant et les réservations effectuées. L'objectif pédagogique est de pratiquer les commandes SQL de base (DDL, DML, DQL) et de comprendre le fonctionnement des **clés primaires** et **étrangères**, ainsi que les **jointures** entre tables.

1. Création de la base de données

Dans cette première étape, nous créons la base RestaurantDB qui contiendra toutes les tables nécessaires à notre système de réservation.

```
1 CREATE DATABASE RestaurantDB;  
2 USE RestaurantDB;
```

Listing 1 – Création de la base RestaurantDB

2. Création des tables

Nous créons ensuite les trois tables : Client, TableRestaurant et Reservation. Les relations entre les tables sont définies grâce aux clés étrangères.

```
1 CREATE TABLE Client (  
2     ID_client INT AUTO_INCREMENT PRIMARY KEY,  
3     nom VARCHAR(50) NOT NULL,  
4     telephone VARCHAR(20)  
5 );  
6  
7 CREATE TABLE TableRestaurant (  
8     ID_table INT AUTO_INCREMENT PRIMARY KEY,  
9     numero INT NOT NULL,  
10    capacite INT NOT NULL  
11 );  
12  
13 CREATE TABLE Reservation (  
14     ID_reservation INT AUTO_INCREMENT PRIMARY KEY,  
15     ID_client INT,  
16     ID_table INT,  
17     dateReservation DATE NOT NULL,  
18     heure TIME NOT NULL,  
19     nombrePersonnes INT NOT NULL,  
20     FOREIGN KEY (ID_client) REFERENCES Client(ID_client),  
21     FOREIGN KEY (ID_table) REFERENCES TableRestaurant(ID_table)  
22 );
```

Listing 2 – Création des tables Client, TableRestaurant et Reservation

3. Insertion des données

Insertion des données pour les clients, les tables et les réservations. Nous insérons au minimum 3 clients, 3 tables et 5 réservations.

```
1 INSERT INTO Client (nom, telephone) VALUES
2 ('Sara', '0612345678'),
3 ('mariyem', '0698765432'),
4 ('manal', '0623456789');
5
6 INSERT INTO TableRestaurant (numero, capacite) VALUES
7 (1, 2),
8 (2, 4),
9 (3, 6);
10
11 INSERT INTO Reservation (ID_client, ID_table, dateReservation, heure,
12     nombrePersonnes) VALUES
13 (1, 2, '2025-11-26', '19:00:00', 4),
14 (2, 1, '2025-11-26', '18:30:00', 2),
15 (3, 3, '2025-11-27', '20:00:00', 5),
16 (1, 3, '2025-11-28', '21:00:00', 3),
17 (2, 2, '2025-11-28', '19:30:00', 4);
```

Listing 3 – Insertion des clients, tables et réservations

4. Requêtes SELECT simples

Nous affichons toutes les données des trois tables pour vérifier leur contenu.

```
1 SELECT * FROM Client;
2 SELECT * FROM TableRestaurant;
3 SELECT * FROM Reservation;
```

Listing 4 – Afficher tous les clients, toutes les tables et toutes les réservations

5. Requêtes avec conditions

Exemples de filtrage des données avec des conditions spécifiques.

```
1 SELECT *
2 FROM Reservation R
3 INNER JOIN Client C ON R.ID_client = C.ID_client
4 WHERE C.nom = 'Sara';
5
6 SELECT *
7 FROM TableRestaurant
8 WHERE capacite >= 4;
```

Listing 5 – Réservations du client nommé "Sara" et tables de capacité ≥ 4

6. Requêtes avec jointures

Exemple de jointures pour lister les réservations avec les informations du client et de la table. Nous pouvons aussi trier les résultats et compter les réservations par client.

```
1 SELECT C.nom AS NomClient, T.numero AS NumeroTable, R.dateReservation, R
   .heure
2 FROM Reservation R
3 INNER JOIN Client C ON R.ID_client = C.ID_client
4 INNER JOIN TableRestaurant T ON R.ID_table = T.ID_table;
5
6 -- Trier les r servations par date puis heure
7 SELECT C.nom AS NomClient, T.numero AS NumeroTable, R.dateReservation, R
   .heure
8 FROM Reservation R
9 INNER JOIN Client C ON R.ID_client = C.ID_client
10 INNER JOIN TableRestaurant T ON R.ID_table = T.ID_table
11 ORDER BY R.dateReservation ASC, R.heure ASC;
12
13 -- Clients ayant plus d'une r servation
14 SELECT C.nom, COUNT(R.ID_reservation) AS NombreReservations
15 FROM Reservation R
16 INNER JOIN Client C ON R.ID_client = C.ID_client
17 GROUP BY C.nom
18 HAVING COUNT(R.ID_reservation) > 1;
```

Listing 6 – Liste des réservations avec jointures et tri

7. Requête bonus

Calcul du nombre total de réservations par jour pour analyse statistique.

```
1 SELECT dateReservation, COUNT(*) AS TotalReservations
2 FROM Reservation
3 GROUP BY dateReservation
4 ORDER BY dateReservation;
```

Listing 7 – Nombre total de réservations par jour