

Rapport TP SQL : Gestion de Réservations

Étudiant : [Ton Nom]

19 janvier 2026

1 Introduction

L'objectif de ce TP est de mettre en place une base de données relationnelle nommée **RestaurantDB** pour gérer les clients, les tables et les réservations. Nous utilisons le langage SQL pour la définition des données (DDL) et leur manipulation (DML/DQL).

2 Création de la Base de Données

Nous avons créé trois tables liées par des clés étrangères pour assurer l'intégrité référentielle.

```
1 CREATE TABLE Client (
2     idClient INT PRIMARY KEY AUTO_INCREMENT,
3     nom VARCHAR(50),
4     telephone VARCHAR(20)
5 );
6
7 CREATE TABLE TableRestaurant (
8     idTable INT PRIMARY KEY AUTO_INCREMENT,
9     numero INT,
10    capacite INT
11 );
12
13 CREATE TABLE Reservation (
14     idReservation INT PRIMARY KEY AUTO_INCREMENT,
15     idClient INT,
16     idTable INT,
17     dateReservation DATE,
18     heure TIME,
19     nombrePersonnes INT,
20     FOREIGN KEY (idClient) REFERENCES Client(idClient),
21     FOREIGN KEY (idTable) REFERENCES TableRestaurant(idTable)
22 );
```

Listing 1 – Structure des tables

3 Requêtes et Solutions

3.1 Sélection conditionnelle

Question 4 : Afficher les réservations du client "Sara".

Note : Une jointure est nécessaire car la table Reservation ne contient que l'ID du client.

```

1 SELECT R.* FROM Reservation R
2 JOIN Client C ON R.idClient = C.idClient
3 WHERE C.nom = 'Sara';

```

Question 5 : Tables avec capacité ≥ 4 .

```

1 SELECT * FROM TableRestaurant WHERE capacite >= 4;

```

3.2 Requêtes avec Jointures

Question 6 : Afficher les détails complets (Nom Client et Numéro Table).

Cette requête nécessite une double jointure.

```

1 SELECT
2     C.nom AS "Client",
3     T.numero AS "Table",
4     R.dateReservation,
5     R.heure
6 FROM Reservation R
7 JOIN Client C ON R.idClient = C.idClient
8 JOIN TableRestaurant T ON R.idTable = T.idTable;

```

3.3 Agrégations (GROUP BY)

Question 8 : Clients ayant fait plus d'une réservation.

```

1 SELECT C.nom, COUNT(R.idReservation) AS Total
2 FROM Reservation R
3 JOIN Client C ON R.idClient = C.idClient
4 GROUP BY C.idClient, C.nom
5 HAVING COUNT(R.idReservation) > 1;

```

4 Conclusion

Ce TP a permis de valider la compréhension des clés étrangères et des jointures, éléments essentiels pour lier les informations entre les tables Client, Table et Réservation.