

Projet Base de données

La transformation du MCD vers MLD :

Hotel (id_hotel, ville, pays, code_postal)

TypeChambre (id_typechambre, type, tarif)

Client (id_client, nom_complet, adresse, ville, code_postal, email, numero_telephone) // ici on ajoute id_client pour que être spécifique

Chambre (id_chambre, etage, fumeur, #id_hotel, #id_typechambre)

Reservation(id_reservation, date d'arrivee, date de depart, #id_client)

Evaluation (id_evaluation, date d'arrivee, la note, texte_descriptif, #id_client, #id_hotel)

Prestation (id_prestation, prix)

Offre (#id_hotel, #id_prestation)

Concerner (#id_typechambre, #id_reservation)

Partie 1 :

a.

```
CREATE DATABASE IF NOT EXISTS HotelDB;  
USE HotelDB;
```

```
CREATE TABLE Hotel (  
    id_hotel INT PRIMARY KEY,  
    ville VARCHAR(100),  
    pays VARCHAR(100),  
    code_postal INT  
);
```

```

CREATE TABLE Client1 (
    id_client INT PRIMARY KEY,
    nom_complet VARCHAR(100),
    adresse VARCHAR(255),
    ville VARCHAR(100),
    code_postal INT,
    email VARCHAR(100),
    telephone VARCHAR(20)
);

CREATE TABLE Prestation (
    id_prestation INT PRIMARY KEY,
    prix INT
);

CREATE TABLE TypeChambre (
    id_typechambre INT PRIMARY KEY,
    nom VARCHAR(50),
    prix FLOAT
);

CREATE TABLE Chambre (
    id_chambre INT PRIMARY KEY,
    numero INT,
    etage INT,
    fumeur BOOLEAN,
    id_typechambre INT,
    id_hotel INT,
    FOREIGN KEY (id_typechambre) REFERENCES TypeChambre(id_typechambre),
    FOREIGN KEY (id_hotel) REFERENCES Hotel(id_hotel)
);

CREATE TABLE Reservation (
    id_reservation INT PRIMARY KEY,
    date_debut DATE,
    date_fin DATE,
    id_client INT,
    FOREIGN KEY (id_client) REFERENCES Client1(id_client)
);

CREATE TABLE Evaluation (
    id_evaluation INT PRIMARY KEY,
    date1 DATE,
    note INT,
    commentaire TEXT,
    id_client INT,
    FOREIGN KEY (id_client) REFERENCES Client1(id_client)
);

```

b.

```
INSERT INTO Hotel VALUES
```

```
(1, 'Paris', 'France', 75001),  
(2, 'Lyon', 'France', 69002);
```

```
INSERT INTO Client1 VALUES
```

```
(1, '12 Rue de Paris', 'Paris', 75001, 'jean.dupont@email.fr',  
'0612345678', 'Jean Dupont'),  
(2, '5 Avenue Victor Hugo', 'Lyon', 69002, 'marie.leroy@email.fr',  
'0623456789', 'Marie Leroy'),  
(3, '8 Boulevard Saint-Michel', 'Marseille', 13005,  
'paul.moreau@email.fr', '0634567890', 'Paul Moreau'),  
(4, '27 Rue Nationale', 'Lille', 59800, 'lucie.martin@email.fr',  
'0645678901', 'Lucie Martin'),  
(5, '3 Rue des Fleurs', 'Nice', 06000, 'emma.giraud@email.fr',  
'0656789012', 'Emma Giraud');
```

```
INSERT INTO Prestation VALUES
```

```
(1, 15, 'Petit-déjeuner'),  
(2, 30, 'Navette aéroport'),  
(3, 0, 'Wi-Fi gratuit'),  
(4, 50, 'Spa et bien-être'),  
(5, 20, 'Parking sécurisé');
```

```
INSERT INTO TypeChambre VALUES
```

```
(1, 'Simple', 80),  
(2, 'Double', 120);
```

```
INSERT INTO Chambre VALUES
```

```
(1, 201, 2, 0, 1, 1),  
(2, 502, 5, 1, 1, 2),  
(3, 305, 3, 0, 2, 1),  
(4, 410, 4, 0, 2, 2),  
(5, 104, 1, 1, 2, 2),  
(6, 202, 2, 0, 1, 1),  
(7, 307, 3, 1, 1, 2),  
(8, 101, 1, 0, 1, 1);
```

```
INSERT INTO Reservation VALUES
```

```
# Client 1 (Jean Dupont)
```

```
(1, '2025-06-15', '2025-06-18', 1),
```

```
# Client 2 (Marie Leroy)
```

```
(2, '2025-07-01', '2025-07-05', 2),
```

```
(7, '2025-11-12', '2025-11-14', 2),
```

```
(10, '2026-02-01', '2026-02-05', 2),
```

```
# Client 3 (Paul Moreau)
```

```
(3, '2025-08-10', '2025-08-14', 3),
```

```
# Client 4 (Lucie Martin)
```

```
(4, '2025-09-05', '2025-09-07', 4),
```

```
(9, '2026-01-15', '2026-01-18', 4),
```

```
# Client 5 (Emma Giraud)
```

```
(5, '2025-09-20', '2025-09-25', 5);
```

```
INSERT INTO Evaluation VALUES
```

```
(1, '2025-06-15', 5, 'Excellent séjour, personnel très accueillant', 1),
```

```
(2, '2025-07-01', 4, 'Chambre propre, bon rapport qualité/prix.', 1),
```

```
(3, '2025-08-10', 3, 'Séjour correct mais bruyant la nuit.', 3),
```

```
(4, '2025-09-05', 5, 'Service impeccable, je recommande.', 4),
```

```
(5, '2025-09-20', 4, 'Très bon petit-déjeuner, hôtel bien situé.', 1),
```

c.

a. En SQLite :

```
SELECT R.id_reservation, C.nom_complet, H.ville
FROM Reservation AS R
JOIN Client1 AS C ON R.nom_complet = C.id_reservation
JOIN Chambre AS CH ON CH.id_reservation = R.id_reservation
JOIN Hotel H ON CH.id_hotel = H.id_reservation;
```

En algèbre relationnel :

π R.id, C.nom, H.ville (Reservation \bowtie R.id_client = C.id
Client \bowtie R.id = CH.id Chambre \bowtie CH.id_hotel = H.id
Hotel)

b. En SQLite :

```
SELECT * FROM Client WHERE ville = 'Paris';
```

En algèbre relationnel :

σ ville = 'Paris' (Client)

c. En SQLite :

```
SELECT C.nom_complet, COUNT(R.id_reservation) AS nb_reservations:
FROM Client1 AS C
LEFT JOIN Reservation AS R ON C.id_reservation = R.nom_complet
GROUP BY C.nom_complet;
```

En algèbre relationnel :

γ nom, COUNT(R.id) \rightarrow nb_reservations (Client \bowtie Client.id
= Reservation.id_client)

d. En SQLite :

```
SELECT T.nom_complet, COUNT(C.id_typechambre) AS nb_chambres
FROM TypeChambre T
JOIN Chambre AS C ON T.id_typechambre = C.id_chambre
GROUP BY T.nom_complet;
```

En algèbre relationnel :

γ T.nom, COUNT(C.id) \rightarrow nb_chambres (TypeChambre \bowtie
TypeChambre.id = Chambre.id_type)

e. En SQLite :

```
SELECT * FROM Chambre
WHERE id_chambre NOT IN (
    SELECT id_reservation FROM Reservation
    WHERE NOT (
        date_fin < '2025-06-01' OR date_debut > '2025-06-10'
    )
);
```

En algèbre relationnel :

Chambre – π id_chambre (
 σ NOT(date_fin < '2025-06-01' OR date_debut > '2025-06-10') (Reservation)
)

4. SQLite est un Système de Gestion de Base de Données relationnelle léger et embarqué et qui est stocké dans un seul fichier .db sur le disque, il est gratuit, open source et ultra portable, il est écrit en C , intégré directement dans l'application.

La différence entre SQLite et MySQL :

Critère	MySQL	SQLite
Type	Serveur client / serveur	Bibliothèques embarquée
Installation	Nécessite l'installation d'un serveur	Pas d'installation, juste 1 fichier
Multi-utilisateur	Gere plusieurs connexions	Limitée
Fonctionnalités	Avancée	Plus limitée
Performances	Optimisée pour gros volumes	Optimisée pour légèreté et lecture rapide
Utilisation typique	Sites web, systèmes complexes	Application, script
Sécurité	Gere droits utilisateurs	Pas de gestion d'utilisateur

