

SQL concrètement

1. Création de la Base de Données et des Tables

```
-- Crée la base de données si elle n'existe pas et la sélectionne pour
utilisation.
CREATE DATABASE IF NOT EXISTS restaurant;
USE restaurant;

-- Crée la table Client.
CREATE TABLE Client (
    id_client INT PRIMARY KEY,
    nom VARCHAR(50),
    prenom VARCHAR(50),
    email VARCHAR(100)
);

-- Crée la table Commande.
CREATE TABLE Commande (
    id_commande INT PRIMARY KEY,
    id_client INT,
    date_commande DATE,
    total DECIMAL(10, 2),
    FOREIGN KEY (id_client) REFERENCES Client(id_client)
);

-- Crée la table Plat.
CREATE TABLE Plat (
    id_plat INT PRIMARY KEY,
    nom_plat VARCHAR(50),
    id_categorie INT,
    prix DECIMAL(10, 2),
    FOREIGN KEY (id_categorie) REFERENCES Categorie(id_categorie)
);

-- Suppression puis recréation de la table Categorie.
DROP TABLE IF EXISTS Categorie;
CREATE TABLE Categorie (
    id_categorie INT PRIMARY KEY,
    nom_categorie VARCHAR(50)
);

-- Données pour la table `Client`

INSERT INTO Client (id_client, nom, prenom, email) VALUES
(1, 'Martin', 'Lucie', 'lucie.martin@example.com'),
(2, 'Bernard', 'Julien', 'julien.bernard@example.com'),
(3, 'Thomas', 'Émilie', 'emilie.thomas@example.com'),
(4, 'Petit', 'Sophie', 'sophie.petit@example.com'),
```

```
(5, 'Robert', 'Christophe', 'christophe.robert@example.com');

-- Données pour la table `Categorie`

INSERT INTO Categorie (id_categorie, nom_categorie) VALUES
(1, 'Entrée'),
(2, 'Plat principal'),
(3, 'Dessert'),
(4, 'Boisson');

-- Données pour la table `Plat`

INSERT INTO Plat (id_plat, nom_plat, id_categorie, prix) VALUES
(1, 'Salade Niçoise', 1, 12.50),
(2, 'Steak Frites', 2, 18.90),
(3, 'Mousse au Chocolat', 3, 6.50),
(4, 'Poulet Basquaise', 2, 16.50),
(5, 'Tarte Tatin', 3, 8.00),
(6, 'Limonade Maison', 4, 3.75);

-- Données pour la table `Commande`

INSERT INTO Commande (id_commande, id_client, date_commande, total) VALUES
(1, 1, '2023-09-01', 45.90),
(2, 2, '2023-09-02', 24.25),
(3, 3, '2023-09-03', 19.75),
(4, 4, '2023-09-04', 34.75),
(5, 5, '2023-09-05', 22.50);
```

2. Requêtes SQL

```
-- 1. Sélectionner toutes les colonnes de la table Client.
SELECT * FROM Client;

-- 2. Sélectionner le nom et l'email des clients dont le nom commence par "A".
SELECT nom, email FROM Client WHERE nom LIKE 'A%';

-- 3. Sélectionner le nombre total de clients.
SELECT COUNT(*) AS nombre_clients FROM Client;

-- 4. Sélectionner le plat le moins cher.
SELECT * FROM Plat ORDER BY prix LIMIT 1;

-- 5. Sélectionner les plats dont le prix est supérieur à 10 euros et dans
une catégorie spécifique.
SELECT * FROM Plat WHERE prix > 10 AND id_categorie = (SELECT id_categorie
FROM Categorie WHERE nom_categorie = 'Plat principal');
```

```
-- 6. Sélectionner les commandes effectuées par un client donné (ID 1).
SELECT * FROM Commande WHERE id_client = 1;

-- 7. Sélectionner le montant total de toutes les commandes.
SELECT SUM(total) AS montant_total FROM Commande;

-- 8. Mettre à jour le nom d'un client (ID 1).
UPDATE Client SET nom = 'Nouveau Nom' WHERE id_client = 1;

-- 9. Insérer un nouveau plat dans la table Plat.
INSERT INTO Plat (nom_plat, prix, id_categorie) VALUES ('Nouveau Plat',
15.99, (SELECT id_categorie FROM Categorie WHERE nom_categorie =
'Entrée'));

-- 10. Supprimer un plat de la table Plat (ID 1).
DELETE FROM Plat WHERE id_plat = 1;

-- 11. Sélectionner les plats en ordre décroissant de prix.
SELECT * FROM Plat ORDER BY prix DESC;

-- 12. Sélectionner les clients ayant passé une commande avec un montant
supérieur à 50 euros.
SELECT DISTINCT c.* FROM Client c INNER JOIN Commande cm ON c.id_client =
cm.id_client WHERE cm.total > 50;

-- 13. Sélectionner les plats avec leur catégorie respective.
SELECT p.nom_plat AS plat, c.nom_categorie AS categorie FROM Plat p INNER
JOIN Categorie c ON p.id_categorie = c.id_categorie;

-- 14. Sélectionner les clients n'ayant jamais passé de commande.
SELECT * FROM Client WHERE id_client NOT IN (SELECT DISTINCT id_client FROM
Commande);

-- 15. Sélectionner les plats et leur nombre de commandes respectif, triés
par nombre de commandes décroissant.
SELECT p.nom_plat AS plat, COUNT(cm.id_commande) AS nombre_commandes FROM
Plat p LEFT JOIN Commande cm ON p.id_plat = cm.id_plat GROUP BY p.id_plat
ORDER BY nombre_commandes DESC;
```