

Soit La base de données **Le\_Traditionnel** définie par la représentation textuelle de ses tables suivante :

Client (email, tel, nom)

Réservation (idReservation, espace, numTable, email#, dateR, heureP, nbPersonnes)

Produit (idProduit, nomProduit, prix, qteStock)

Commande (idCommande, idReservation#)

LigneCommande (idCommande#, idProduit#, qte, caracteristiques)

Tableau de description des colonnes

Champs	Type	Description	Contrainte
email	Chaine de 30 caractères	Email d'un client	Clé primaire
tel	Numérique de 8 chiffres	Téléphone d'un client	
nom	Chaine de 20 caractères	Nom d'un client	
idReservation	entier	Identifiant d'une réservation	Clé primaire auto-incrémenté
espace	Chaine de 10 caractères	Nom d'un espace	
numTable	Entier de 2 chiffres	Numéro de table dans un espace	
dateR	Date	Date d'une réservation	
heureP	heure	Heure d'une réservation	
nbPersonnes	Entier de 2 chiffres	Nombre de personnes pour une réservation	
idProduit	entier	Identifiant d'un produit	Clé primaire auto-incrémenté
nomProduit	Chaine de 20 caractères	Nom d'un produit	
prix	décimale (10,3)	Prix d'un produit	Un réel strictement positif
qteStock	Entier de 3 chiffres	Quantité de stock d'un produit	Un entier positif
idCommande	entier	Identifiant d'une commande	Clé primaire auto-incrémenté
qte	entier de 3 chiffres	Quantité d'un produit commandée	Un entier strictement positif
caracteristiques	Chaine de 40 caractères	Caractéristiques d'un produit commandé	

```

-- Création de la base de données
CREATE DATABASE Le_Traditionnel;

-- Table n°1 : Client
-- Contient les informations des clients.
CREATE TABLE Client (
    email VARCHAR(30) PRIMARY KEY,
    tel VARCHAR(8),
    nom VARCHAR(20)
);

-- Table n°2 : Produit
-- Contient le catalogue des produits.
CREATE TABLE Produit (
    idProduit INT PRIMARY KEY AUTO_INCREMENT,
    nomProduit VARCHAR(20),
    prix DECIMAL(10, 3) CHECK (prix > 0),
    -- Contrainte : Le prix doit être strictement positif
    qteStock INT CHECK (qteStock >= 0 AND qteStock <= 999)
    -- Contrainte : La quantité en stock doit être un entier positif (0-999)
);

-- Table n°3 : Reservation
-- Enregistre les réservations des clients. Dépend de la table 'Client'.
CREATE TABLE Reservation (
    idReservation INT PRIMARY KEY AUTO_INCREMENT,
    espace VARCHAR(10),
    -- Contrainte : "Entier de 2 chiffres" pour la table (0-99)
    numTable INT CHECK (numTable >= 0 AND numTable <= 99),
    email VARCHAR(30),
    dateR DATE,
    heureP TIME,
    -- Contrainte : "Entier de 2 chiffres" pour le nombre de personnes (1-99)
    nbPersonnes INT CHECK (nbPersonnes > 0 AND nbPersonnes <= 99),
    -- CLÉ ÉTRANGÈRE NOMMÉE vers le client qui a réservé
    FOREIGN KEY FK_Reservation_Client (email) REFERENCES Client(email)
    ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

-- Table n°4 : Commande
-- Fait le lien entre une réservation et une commande passée. Dépend de la table
  'Reservation'.
CREATE TABLE Commande (
    idCommande INT PRIMARY KEY AUTO_INCREMENT,
    idReservation INT,
    -- CLÉ ÉTRANGÈRE NOMMÉE vers la réservation associée
    FOREIGN KEY FK_Commande_Reservation (idReservation) REFERENCES Reservation(idReservation)
);

-- Table n°5 : LigneCommande
-- Table de jonction qui détaille les produits commandés. Dépend des tables 'Commande' et
  'Produit'.
CREATE TABLE LigneCommande (
    idCommande INT,
    idProduit INT,
    -- Contrainte : "Entier de 3 chiffres" et "strictement positif" (1-999)
    qte INT CHECK (qte > 0 AND qte <= 999),
    caracteristiques VARCHAR(40),
    -- Clé primaire composite
    PRIMARY KEY (idCommande, idProduit),
    -- CLÉS ÉTRANGÈRES NOMMÉES
    FOREIGN KEY FK_LigneCommande_Commande (idCommande) REFERENCES Commande(idCommande),
    FOREIGN KEY FK_LigneCommande_Produit (idProduit) REFERENCES Produit(idProduit)
);

```