# **Rapport :**

**Partie 1 : Schéma et SQL**

## **1. Introduction**

Cette partie présente la réflexion autour de la conception de la base de données en lien avec le cahier des charges. Il détaille la structure du schéma Entité-Association ainsi que la logique du script SQL qui en découle.

## **2. Conception du Modèle Relationnel**

### **a. Analyse des Besoins**

Le cahier des charges impose la gestion de plusieurs types d’utilisateurs : clients particuliers, clients entreprises et cuisiniers. Ces utilisateurs doivent être en mesure d’interagir avec un système de commandes et de sous-commandes, tout en respectant certaines contraintes comme la proximité d’une station de métro pour la livraison.

### **b. Modélisation du Schéma Entité-Association**

Le modèle repose sur plusieurs entités principales :

* Clients : Deux types d'entités sont distingués : les clients particuliers et les clients entreprises. Ils partagent des attributs communs (nom, adresse, email, etc.), mais les entreprises possèdent un référent.
* Cuisiniers : Responsables de la préparation des commandes, ils sont également identifiés par leurs coordonnées et leur proximité avec un métro.
* Commandes et Sous-Commandes : Une commande peut être composée de plusieurs sous-commandes, permettant une meilleure gestion des livraisons.
* Plats et ingrédients : Chaque plat est défini par son nom, prix, quantité, régime et nature. Il est composé d’un ou plusieurs ingrédients, chacun caractérisé par son volume.

Ce modèle permet d’assurer une gestion efficace des interactions entre les différents acteurs du système.

## **3. Implémentation SQL**

### **a. Création des Tables**

La base de données a été implémentée en suivant la structure définie par le schéma E/A. Les entités principales ont été traduites en tables avec des relations bien définies via des clés primaires et étrangères. Par exemple :

* Chaque commande est identifiée par un **numero\_commande** et est liée à plusieurs sous-commandes via une clé étrangère.
* Un plat est identifié par **id\_plat** et est relié à ses ingrédients grâce à une relation spécifique.

Afin d'assurer la cohérence des données, des contraintes ont été appliquées, les clés primaires et étrangères pour maintenir les relations entre les entités.

### **b. Peuplement de la Base**

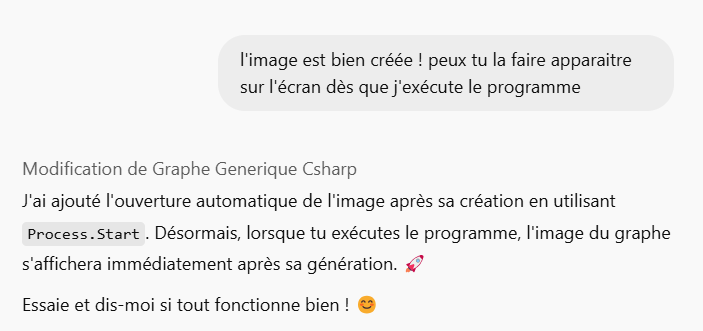
Le peuplement des tables se fait via l’importation de fichiers CSV. L’approche adoptée permet de charger efficacement un grand volume de données tout en garantissant leur conformité avec le modèle relationnel. On a utilisé le LOAD DATA INFILE.

### **c. Requêtes SQL Principales**

Plusieurs requêtes permettent d’exploiter la base de données de manière fonctionnelle :

* Consultation des clients, cuisiniers et plats.
* Comptage du nombre total de commandes.
* Ajout et suppression de commandes.
* Sélection des ingrédients nécessaires à la préparation d’un plat.





Nous avons choisi d'utiliser la bibliothèque SkiaSharp parce que nous avons créé notre projet Visual en Net6.0 et que cette bibliothèque s’adapte bien avec, contrairement à System.Drawing.

Les premiers codes que nous proposait chat GPT affichaient une image très brouillon avec les noeuds du graphe qui avaient tendance à se chevaucher, on a donc modifié légèrement le code pour rendre l’image plus lisible avec un changement de couleur et de police d’écriture, et on a espacé davantage les noeuds.