

Rapport sur le Schéma Entité-Association et la Base de Données LivinParis

Baptiste MARSEILLE

Raphael LAURET

Germain LE GUEN

TD-K

Le projet LivinParis vise à modéliser un système de gestion des commandes et de livraisons de plats.

Le schéma Entité-Association est structuré autour de plusieurs entités. L'entité Utilisateur regroupe les informations personnelles des individus, comme le nom, le prénom, l'adresse, le téléphone, l'email et le mot de passe. Cette entité est ensuite spécialisée en Client et en Cuisinier. Un Client peut être soit un Particulier, soit une Entreprise Locale qui possède un nom et un référent. Le Cuisinier est un utilisateur chargé de la préparation des plats.

Le système de commande est organisé autour de plusieurs entités interconnectées. Une Commande possède une date et un montant total et est passée par un Client. Chaque commande est détaillée dans une ou plusieurs Lignes_Commande, qui associent des plats aux commandes et précisent la quantité, la date et le lieu de livraison. Chaque plat est défini par un nom, un type (entrée, plat principal, dessert), une date de fabrication, une date de péremption, un prix, le nombre de personnes qu'il peut servir, sa nationalité, un régime alimentaire, les ingrédients et une photo illustrative. Les plats sont préparés par des Cuisiniers et inclus dans les commandes.

Le système de livraison est géré par l'entité Livraison (clé étrangère cuisinier), qui contient une date et une zone de distribution et est effectuée par un Cuisinier. Les livraisons sont directement liées aux Lignes_Commande. Enfin, la gestion des transactions financières repose sur l'entité Transaction, qui comprend un identifiant unique, un montant et un statut de paiement, et qui est liée à une Commande.

Le script SQL permet la création de la base de données LivinParis en définissant la structure de chaque table et leurs relations en fonction du diagramme E/A. Il commence par la création de l'entité Utilisateur et de ses spécialisations. Ensuite, les commandes et leurs lignes sont mises en place avec les relations appropriées. Les plats et les livraisons sont définis en s'assurant que chaque cuisinier peut préparer

plusieurs plats et effectuer des livraisons. Enfin, les transactions sont liées aux commandes pour assurer un suivi des paiements.

Le script SQL met aussi en place des contraintes d'intégrité essentielles à la cohérence des données. Les clés primaires assurent l'unicité de chaque enregistrement, tandis que les clés étrangères garantissent l'intégrité référentielle entre les différentes tables. Des suppressions en cascade sont mises en place pour assurer que la suppression d'un enregistrement parent entraîne automatiquement celle des enregistrements enfants associés.

(NOUS AVONS FAIT DEUX PROJETS POUR LA VISUALISATION ET LES CLASSES QUI ANALYSENT LE GRAPHE CAR NOUS NOUS SOMMES RÉPARTIS LE TRAVAIL DE CETTE MANIÈRE)

Pour ce qui est de la partie Visualisation avec l'aide de l'IA. Le but était de développer une méthode capable de charger des données de graphe à partir d'un fichier texte contenant des informations structurées sous forme de matrice creuse, telle que celle issue du format MatrixMarket, et à les visualiser graphiquement. Le fichier texte en question contient des informations sur les arêtes et les nœuds d'un graphe, chaque arête étant définie par deux nœuds reliés entre eux. L'approche choisie consiste à lire le fichier ligne par ligne, en ignorant les métadonnées et commentaires (qui sont précédés par le symbole "%"). Une fois que les données pertinentes concernant le nombre de nœuds et d'arêtes ont été extraites, l'analyse continue par l'ajout des nœuds et des arêtes dans une structure de données adéquate, en s'assurant que chaque nœud est unique dans la liste des nœuds et que les arêtes sont correctement définies par des tuples représentant les connexions entre les nœuds. Après l'extraction des données brutes du fichier texte, celles-ci sont utilisées pour créer un graphe à partir d'une structure de données implémentée en C#, où les nœuds et les arêtes sont stockés dans des listes.

Une autre partie du projet concernait la visualisation du graphe. L'idée était de proposer une interface graphique en utilisant Windows Forms, permettant de dessiner les nœuds et les arêtes extraits du fichier texte. Cette interface repose sur la bibliothèque graphique de C# pour afficher les arêtes sous forme de lignes reliant les nœuds et les nœuds sous forme de cercles. Chaque nœud est identifié par un numéro, et l'arête qui le relie à un autre nœud est représentée par une ligne tracée entre deux cercles. Les informations sur les nœuds et les arêtes sont ensuite utilisées pour positionner ces éléments sur le canevas de l'interface graphique, en ajustant dynamiquement leur espacement et en rendant le graphe lisible et compréhensible pour l'utilisateur. L'intégration d'outils d'intelligence artificielle générative a permis de simplifier certaines parties du processus, notamment dans l'analyse des données et dans la génération de requêtes permettant de traiter et de visualiser efficacement les

données du fichier. Ce processus a abouti à une visualisation interactive du graphe qui permet de mieux comprendre les relations entre les différents nœuds, ce qui est particulièrement utile dans des domaines comme l'analyse de réseaux sociaux ou l'analyse de relations dans de grands ensembles de données. En conclusion, l'utilisation de l'intelligence artificielle générative a optimisé l'extraction, le traitement et la présentation des données en permettant de définir des méthodes automatisées, simples et efficaces pour manipuler des fichiers texte complexes et les visualiser sous forme graphique.