

AMRANI Wassim
EL MISSIRY Tarek

Restaurant de Plage "CPP"

1. Calendrier du Projet

Le mini-projet a été réalisé durant les deux dernières séances de travaux pratiques (TP3 et TP4), chaque séance ayant une durée de 3 heures.

Déroulement :

- Séance 1 : Analyse du cahier des charges, modélisation entité-association (E/A), et schéma relationnel
- Séance 2 : Implémentation de la base de données, peuplement des tables et réalisation des scénarios de test

2. Introduction

Le restaurant de plage "CPP" souhaite mettre en place une base de données pour gérer efficacement ses activités, incluant la gestion des tables, des clients, des commandes, et de la facturation. Ce projet a pour objectif de créer une solution qui permet de centraliser et simplifier la gestion du restaurant, tout en respectant l'intégrité des données et en étant facilement extensible pour de futures fonctionnalités (telles que des services de commande en ligne).

3. Description des Étapes du Mini-Projet

3.1 Analyse du Cahier des Charges

L'analyse initiale a permis de déterminer les besoins du restaurant, qui comprennent la gestion des tables, des clients, des commandes, et de la facturation.

Nous avons identifié les contraintes suivantes :

- Le système doit permettre la gestion simultanée de plusieurs commandes.
- Le personnel doit pouvoir accéder rapidement aux informations des clients et des tables.
- Les factures doivent être générées pour les commandes terminées, avec des options de paiement variées.

Cette analyse nous a permis de définir une structure de données qui supporte à la fois les opérations quotidiennes (comme la prise de commandes) et les besoins de suivi administratif (facturation, statistiques, etc.).

3.2 Modélisation Entité-Association (E/A)

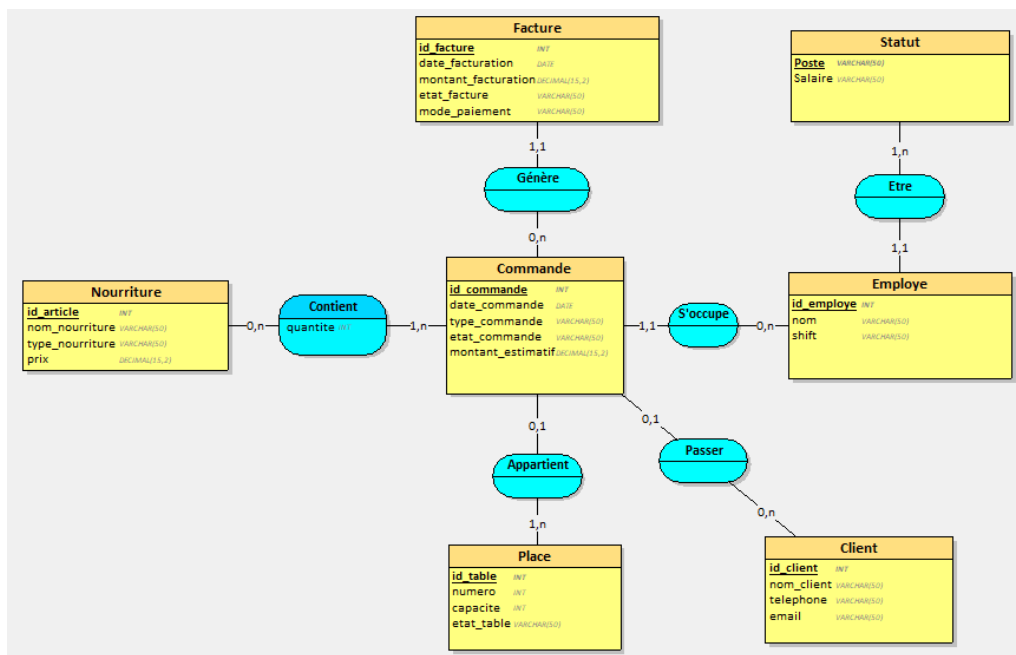
Nous avons identifié les entités suivantes :

- Client : contient des informations telles que le nom, le numéro de téléphone, et l'email.
- Place : représente les tables disponibles dans le restaurant, incluant la capacité et l'état.
- Employé : gère le personnel du restaurant avec la période de travail.
- Statut : contient les différents rôles qu'il peut y avoir au sein du restaurant (serveur, cuisinier, manager) avec leur salaires (en supposant qu'ils sont tous payé équitablement).
- Commande : inclut les détails des commandes effectuées par les clients, associées aux tables et employés.
- Facture : permet de suivre les paiements des commandes.
- Nourriture : contient les articles disponibles (nom, type, prix).

Les relations principales incluent :

- Passer : entre le client et la commande, chaque client pouvant passer plusieurs commandes.
- Contient : entre la commande et la nourriture, chaque commande pouvant contenir plusieurs articles de nourriture.
- S'occupe : entre l'employé et la commande, permettant de suivre quel employé gère quelle commande.
- Appartient : entre la commande et la place, chaque commande pouvant appartenir à une seule table.
- Génère : entre la commande et la facture, chaque commande pouvant générer une seule facture.
- Etre : entre l'employé et le statut, chaque employé pouvant avoir un seul statut.

Pour la modélisation E/A, nous avons utilisé l'outil Looping afin de faciliter la création et la visualisation du modèle entité-association, rendant la compréhension et la validation de notre modèle plus rapide et plus précise.



3.3 Schéma Relationnel et Normalisation

Nous avons ensuite créé le schéma relationnel en respectant les règles de normalisation. Toutes les tables sont en troisième forme normale (3NF) pour éviter la redondance et garantir l'intégrité des données. À l'aide de l'outil Looping, nous avons directement pu avoir le schéma relationnel depuis l'application que l'on a revérifié par dessus.

Exemples :

- Client(id_client, nom_client, telephone, email) : Table normalisée contenant les détails des clients sans redondance inutile.
- Commande(id_commande, date_commande, type_commande, etat_commande, montant_estimatif, id_client, id_employe, id_table) : Chaque commande est liée à un client, une table, et un employé, garantissant une intégrité relationnelle.
- Contient(id_commande, id_article, quantite) : Cette table fait le lien entre les commandes et les articles de nourriture, permettant une gestion des quantités commandées sans redondance.

3.4 Implémentation SQL

La partie SQL a consisté à créer les tables du schéma relationnel dans un SGBD (Système de Gestion de Base de Données) avec MySQL Workbench. L'outil Looping nous a également fourni le script tout en respectant les étapes de création qui sont :

- Créer les tables avec les clés primaires et étrangères appropriées pour assurer l'intégrité référentielle.
- Ajouter des contraintes de validation, telles que la non-nullité de certains champs essentiels (par exemple, id_client et id_employe dans la table Commande).
- Utiliser des types de données appropriés pour chaque colonne (par exemple, VARCHAR, DECIMAL, INT, etc.) afin d'optimiser la précision et l'efficacité du stockage des données.

3.5 Peuplement des Tables et Scénarios de Test

Une fois les tables créées, nous avons écrit des requêtes d'insertion (INSERT INTO) pour peupler chaque table avec des données fictives mais réalistes. Les scénarios de test ont été utilisés pour valider le bon fonctionnement de la base de données. Par exemple :

- Création de plusieurs clients, employés, et commandes avec des états variés (par exemple, commandes terminées, en cours, annulées).
- Association de différents articles de nourriture aux commandes, avec des quantités variées pour tester la relation Contient.

- Validation des contraintes de clés étrangères en ajoutant des employés et en les liant à des commandes.

Nous avons également utilisé des requêtes SQL pour vérifier la cohérence des données (par exemple, vérifier si toutes les commandes terminées ont bien une facture associée).

4. Conclusion

Forces et Faiblesses de la Solution

- Forces : La base de données est bien normalisée, assurant une bonne intégrité des données. Elle est extensible, ce qui permettra d'ajouter de nouvelles fonctionnalités (comme les commandes en ligne) dans le futur.
- Faiblesses : La gestion des réservations de tables n'est pas totalement implémentée, et l'ajout d'un module de gestion des disponibilités serait nécessaire pour une meilleure gestion.

Améliorations Possibles

- Gestion des Réservations : Ajouter un module permettant aux clients de réserver des tables.
- Services de Livraison : Étendre la base de données pour inclure la gestion des commandes en ligne et de la livraison.
- Interface Utilisateur (pour aller plus loin) : Créer une interface utilisateur conviviale pour permettre au personnel de gérer plus facilement les commandes, les clients, et les tables.

Ce que le Projet Nous a Appris

Ce projet nous a permis de mieux comprendre la conception et la mise en œuvre d'une base de données relationnelle complète. Nous avons acquis des compétences clés en modélisation de données, en normalisation, en création de schémas relationnels, ainsi qu'en implémentation et test de la base avec SQL. Travailler sur ce projet nous a également aidé à développer nos capacités de collaboration en équipe pour mener un projet technique de bout en bout.

Outils utilisés :

Looping : <https://www.looping-mcd.fr/>

MySQL Workbench : <https://www.mysql.com/>