Feast Fleet

Your Squad, Your Feast, Our Fleet

Database Management System for Group Food Delivery

Food Delivery Management System

MISM6201 Database Management for Business

Northeastern University

D'Amore-McKim School of Business

Submitted By:

Aditya Wanjari, Atharva Rotkar, Jensy Doshi
Prashant Singh, Shireen Bhardwaj, Vidhi Bhatia

Table of Contents

Introduction

The Feast Fleet Database project is a structured implementation of database management principles to address the operational needs of a modern food delivery service. The project's foundation lies in creating a relational database that ensures efficient storage, retrieval, and management of data while supporting the complexities of a multi-stakeholder system. The stakeholders include customers, restaurants, and delivery agents, each requiring specific functionalities enabled through precise database design.

The database architecture is built to facilitate seamless workflows, including order management, promotions, payment processing, and real-time coordination. For example, group cart management is implemented to allow multiple users to add items collaboratively, such as for office lunches or family dinners. The promotion system integrates validation rules and multi-restaurant support, while the order tracking feature ensures transparency and updates for all involved parties.

To maintain data integrity and consistency, the database employs normalization techniques and constraints like foreign keys and `check` conditions. Real-time tracking mechanisms and eco-delivery options demonstrate how the database supports modern business demands. Overall, the project highlights the application of database design and implementation to solve real-world challenges in the food delivery domain.

Overview

The Feast Fleet Database is structured to maintain detailed records of all entities involved in the food delivery process. It encompasses the following key components:

- Customer Management: Stores customer details, including their addresses and contact information.
- Restaurant Management: Keeps track of restaurant information, including cuisine types and ratings.
- Menu and Item Management: Organizes menus and individual food items offered by restaurants.
- Order Processing: Handles the creation and tracking of customer orders.
- Delivery Management: Manages delivery agent information and order assignments.
- Payment Processing: Supports various payment methods and tracks transactions.
- Promotional Offers: Implements a system for managing promotional codes and discounts.

Database Requirements:

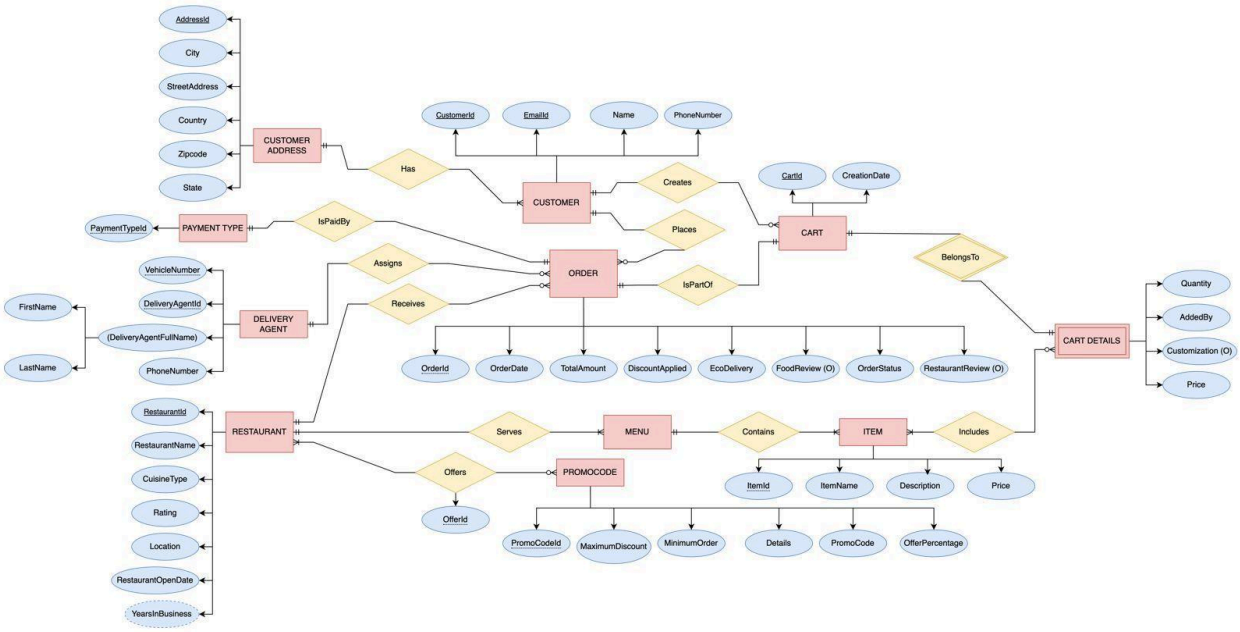
The database is designed to capture and store a wide range of information crucial for the operation of the food delivery service. Feast Fleet Database will keep track of the following:

- For each customer: CustomerID (unique), EmailID (unique), Name, and PhoneNumber.
- For each customer address: AddressID (unique), City, StreetAddress, Country, State, and Zipcode.
- For each payment type: PaymentTypeID (unique).
- For each cart: CartID (unique) and CreationDate.
- For each cart detail: Quantity, AddedBy, Customization (optional), and Price..
- For each restaurant: RestaurantID (unique), RestaurantName, CuisineType, Rating, Location, RestaurantOpenDate, and YearsInBusiness(Derived from restaurant OpenDate).
- For each Menu: MenuID (unique), MenuType
- For each item: ItemID, ItemName, Description, and Price.
- For each promo code: PromoCodeID (unique), PromoCode, OfferPercentage, MaximumDiscount, and MinimumOrder.
- For each delivery agent: DeliveryAgentID (unique), Fullname (composed from FirstName, LastName), PhoneNumber, and VehicleNumber(unique).
- For each order: OrderID (unique), OrderDate, TotalAmount, DiscountApplied, EcoDelivery (boolean), OrderStatus, FoodReview (optional), and RestaurantReview (optional).
- Each customer can place one or more orders. Each order is created by one customer.
- Each customer can have only one address but each address can belong to one or more customers
- Each cart belongs to one customer. Customer creates zero or multiple carts.
- Each order is part of one cart and vice versa.
- Each cart contains exactly one cart detail and vice versa.
- Each cart detail includes one or many items. Each menu item can appear in zero or more cart details.
- Each item is a part of exactly one menu and each menu contains one or more items.
- Each menu is served by exactly one restaurant, each restaurant serves oneor more menus
- Each promo code is offered by one or more restaurants. Each restaurant offers zero or many promo codes.
- Each promo code can be applied to zero or more orders. Each order may have one or no promo code applied.
- Each order is associated with one payment type. Each Payment type can be associated with zero or multiple orders.
- Each order is placed at one restaurant. Each restaurant receives zero or more orders.

- Each order is assigned to one delivery agent. Each delivery agent handles zero or more orders.

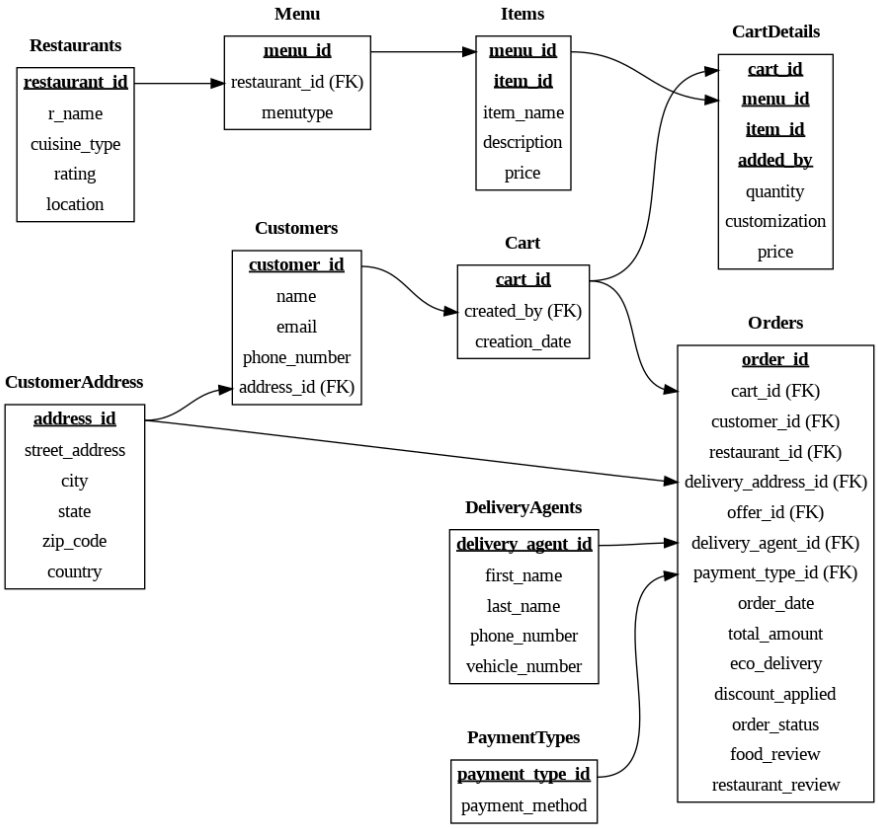
ER Diagram:

The Entity-Relationship (ER) diagram provides a visual representation of the database structure, illustrating the relationships between different entities and the attributes associated with each table.



Relational Schema:

The Relational Schema provides a visual representation of the database structure, illustrating the relationships between different entities and the attributes associated with each table.



SQL CODE: Creating Tables

This section describes the creation of the primary database tables needed to manage customer orders, restaurants, and promotions. A total of 12 tables were designed, each with specific columns, data types, and constraints:

- **CustomerAddress** and **Customers** tables store address details and customer information. Primary and foreign keys are used to link addresses to customers, with `VARCHAR` and `INTEGER` as key data types and constraints like `NOT NULL`and `UNIQUE` applied.
- **Restaurants** and **Menu** tables manage restaurant information and menu types. Ratings are stored using `NUMERIC`, and foreign keys establish the necessary relationships.
- **Promocodes** and **RestaurantPromocodeRelation** contain promotional details, with constraints such as `CHECK` to validate offer percentages and foreign key links to restaurants.
- The **Items** table defines menu items with composite keys for identification and uses `NUMERIC` for pricing.
- **Cart** and **CartDetails** tables track shopping cart data, including timestamps and references to customers and items.

- **DeliveryAgents** and **PaymentTypes** store information about delivery agents and payment methods.
- The **Orders** table connects all entities—customers, carts, and agents—and enforces constraints to manage order statuses, delivery modes, and reviews.

SQL Code: Inserting values

The database is populated to mimic real-world scenarios and enable thorough testing. Data has been inserted into various tables such as Customers, Addresses, Restaurants, Menus, Items, Promocodes, DeliveryAgents, Carts, and Orders.

Each dataset is carefully designed with unique identifiers and logical relationships between tables to ensure consistency and accuracy. For example, customer addresses are linked to their respective customers, menus are associated with specific restaurants, and orders combine information from carts, payment methods, and delivery agents. Promocode applicability is captured using a relational table.

This setup allows testing key functionalities like placing orders, applying promotions, and managing deliveries. The inclusion of different cuisines, item customizations, and order statuses ensures a wide range of scenarios can be tested effectively.

SQL Code: Execution

In this phase of the project, SQL operations were used to simulate real-world scenarios and dynamically update the database. Key activities included the addition of two new customers, complete with their address and contact details. A new restaurant was introduced to the platform, accompanied by a promotional offer linked to several restaurants. Additionally, existing restaurants updated their offerings with new menus and items, including detailed descriptions and pricing. Two new delivery agents were onboarded, and their details were integrated into the system. Payment methods such as "Apple Pay" and "Klarna" were also introduced to the platform. Finally, new shopping cart sessions were created for both existing and newly added customers. These operations showcased the database's flexibility and scalability, effectively incorporating new users, entities, and features while preserving data integrity and supporting future growth.

Analysis (Data Retrieval for Business Relevant information)

The analysis looks at important business metrics to give a clear picture of how things are going. It dives into customer behavior, how well restaurants are doing, and how effective promotions have been. We look at total revenue for each restaurant, which items are selling the most, and the average spend per customer. We also examine how much top customers are spending, how discounts are affecting sales, and how customers are rating the restaurants. Other aspects include how often promo codes are used, how revenue is split by payment method, and which menu items are the most popular in each category. Some deeper dives include identifying customers who spend more than the average. All of this helps businesses make smarter choices, improve their operations, and boost their revenue with smarter marketing and better resource planning.

Challenges and Solutions

Challenge	Solution
Complex Cart Management in Group Ordering	• Implemented composite primary key (CartID, ItemID, AddedBy) for unique identification • Tracked individual customizations per user • Maintained price consistency through direct item reference.
Order Status Management	• Implemented status constraints using CHECK • Added tracking fields like FoodReview and RestaurantReview • Integrated delivery agent status updates • Defined a clear status progression path.
Multiple Restaurant Menu Management	• Created a hierarchical menu structure • implemented menu type categorization (Veg, NonVeg, Vegan) • Added foreign key constraints for data integrity • Enabled flexible menu item management

Promotion System Complexity	• Created an intermediate table for promo-restaurant relationships • Implemented validation rules (e.g., MinimumOrder, MaximumDiscount) • Added constraints for offer percentage, and supported multi-restaurant promotions.
Delivery Assignment and Tracking	• Developed a dedicated DeliveryAgents table for detailed tracking • Implemented an eco-delivery option • Added vehicle tracking support, and integrated delivery status updates.

Conclusion

The Feast Fleet Database project offers a practical approach to managing the data needs of a food delivery service. With a well-organized structure and clear relationships between entities, the system lays a solid foundation for a food delivery application. Features like promotional code management and support for various payment methods highlight the database's ability to meet current business requirements.

The project, which includes creating tables, inserting sample data, and simulating real-world scenarios, demonstrates the database's capacity to handle essential operations and adapt to business needs. As the food delivery industry evolves, the structure provides a good starting point for future adjustments and improvements.