Database Management System

Semester-III(Batch-2024)

Online Food Delivering System

Group - G11(3)

Supervised By:

Sachin Garg Sir

Submitted By:

Aditya Shashi Kumar(2410990821)
Ansh Sachdeva(2410990844)
Abheyjeet Singh(2410990007)
Pavitar Modgil(2410990883)

Department of Computer Science and Engineering
Chitkara University Institute of Engineering & Technology,
Chitkara University, Punjab

Table Of Content

Project Title: Online Food Delivery System

Problem Statement:

In today's fast-paced world, people prefer convenient and quick solutions for their daily needs, including food. Traditional methods of ordering food either require visiting restaurants physically or calling them, which is time-consuming, inefficient, and prone to errors. Customers often face difficulties such as:

- Lack of a centralized platform to browse menus from multiple restaurants.
- Delays in placing orders through phone calls.
- Inability to track delivery status in real-time.
- Errors in manual order-taking and payment handling.

Restaurants also struggle with:

- Managing multiple customer orders simultaneously.
- Keeping track of order history and payments.
- Efficiently assigning delivery personnel to orders.

To overcome these challenges, an Online Food Ordering System is proposed. This system will allow customers to register, browse restaurants and their menus, place food orders, make online payments, and get their food delivered at their doorstep. Restaurants can manage incoming orders digitally, update menu items, and assign delivery persons efficiently. Delivery personnel will be able to receive order details and deliver food within time.

This system will reduce manual effort, minimize errors, and improve customer satisfaction by providing a seamless platform for customers, restaurants, and delivery staff.

Introduction:

The Online Food Ordering System is a web-based application that makes food ordering easier and faster for customers. Instead of visiting restaurants or calling them, customers can browse menus, place orders, and make payments online. Restaurants can manage orders digitally, update menus, and assign delivery staff, while delivery persons can view order details and deliver food on time.

This system reduces manual errors, saves time, and provides a convenient way for customers to get their food delivered at their doorstep. It benefits all three parties – customers, restaurants, and delivery personnel – by making the process smooth, efficient, and reliable.

Overview:

The Online Food Delivery System is designed to simplify the process of ordering food from various restaurants and delivering it to customers in an efficient manner. The system will allow customers to browse menus, place orders, make online payments, and track delivery status in real time. Restaurants can manage their menus, update availability, and process incoming orders, while delivery personnel can be assigned to fulfill deliveries. The admin team will oversee the entire system, handle user management, resolve issues, and analyze data for performance and customer feedback.

Objectives:

- Allow customers to browse restaurants and menus online.
- Enable users to place food orders and make secure payments.
- Assign delivery personnel to deliver order efficiently.
- Maintain data of customers, restaurants, menus, orders, and payments.
- Collect customer feedback and ratings after delivery.

Core Functional Requirements (Entities and Relationships):

1. Customers

- Each customer has unique ID, name, contact details, and address.
- Customers can place multiple orders.
- Customers can make payments after delivery.

2. Restaurant

- Each restaurant has a unique ID, name, location, and cuisine type.
- Restaurants manage their menus (dishes, prices, availability).
- A restaurant can receive multiple orders from different customers.

3. Menu/OrderItem

- Each food item has a unique ID, name, description and availability status.
- Food items belong to a specific restaurant.
- Customers can order multiple food items in a single order.

4. Order

- Each order has a unique order ID, customer ID, restaurant ID, order status (placed, preparing, dispatched, delivered, cancelled), and payment status.
- An order can include multiple food items.
- Each order is assigned to one delivery person.

5. Delivery Personnel

- Each delivery person has a unique ID, name, contact details, and assigned orders.
- A delivery person can handle multiple orders at different times.
- Delivery personnel update delivery status in real time.

6. Payment

- Each payment has a unique ID, order ID, payment method (cash, card, wallet, UPI), and status (pending, completed, failed).
- Payments are linked to orders and customers.

Relationship:

1. Customer – Order
   - One-to-Many relationship.
   - A customer can place many orders, but each order belongs to one customer.
2. Order- OrderItem(OrderItem)
   - One-to-Many relationship.
   - An order can include multiple food items, but each food item in the list is linked to one specific order.
3. Restaurant – Order
   - One-to-Many relationship.
   - A restaurant can receive many orders, but each order belongs to one restaurant.
4. Restaurant – OrderItems
   - One-to-Many relationship.
   - A restaurant can offer many food items, but each food item belongs to one restaurant.
5. Order – Delivery Person
   - Many-to-One relationship.
   - A delivery person can handle many orders, but each order is assigned to one delivery person.
6. Order – Payment
   - One-to-One relationship.
   - Each order has one payment, and each payment belongs to one order.

Cardinalities:

[1]. Restaurant – OrderItem

- **Cardinality:** 1 : N
- One restaurant can list many food items, but each food item belongs to exactly one restaurant.

[2]. **Customer – Order**

- **Cardinality:** 1 : N
- One customer can place many orders, but each order belongs to exactly one customer.

[3]. Order – OrderItem (OrderItem)

- **Cardinality:** 1 : N
- One order can contain many food items, but each food item entry is linked to exactly one order.

[4]. Delivery Person – Order

- **Cardinality:** 1 : N
- One delivery person can deliver many orders, but each order is assigned to exactly one delivery person.

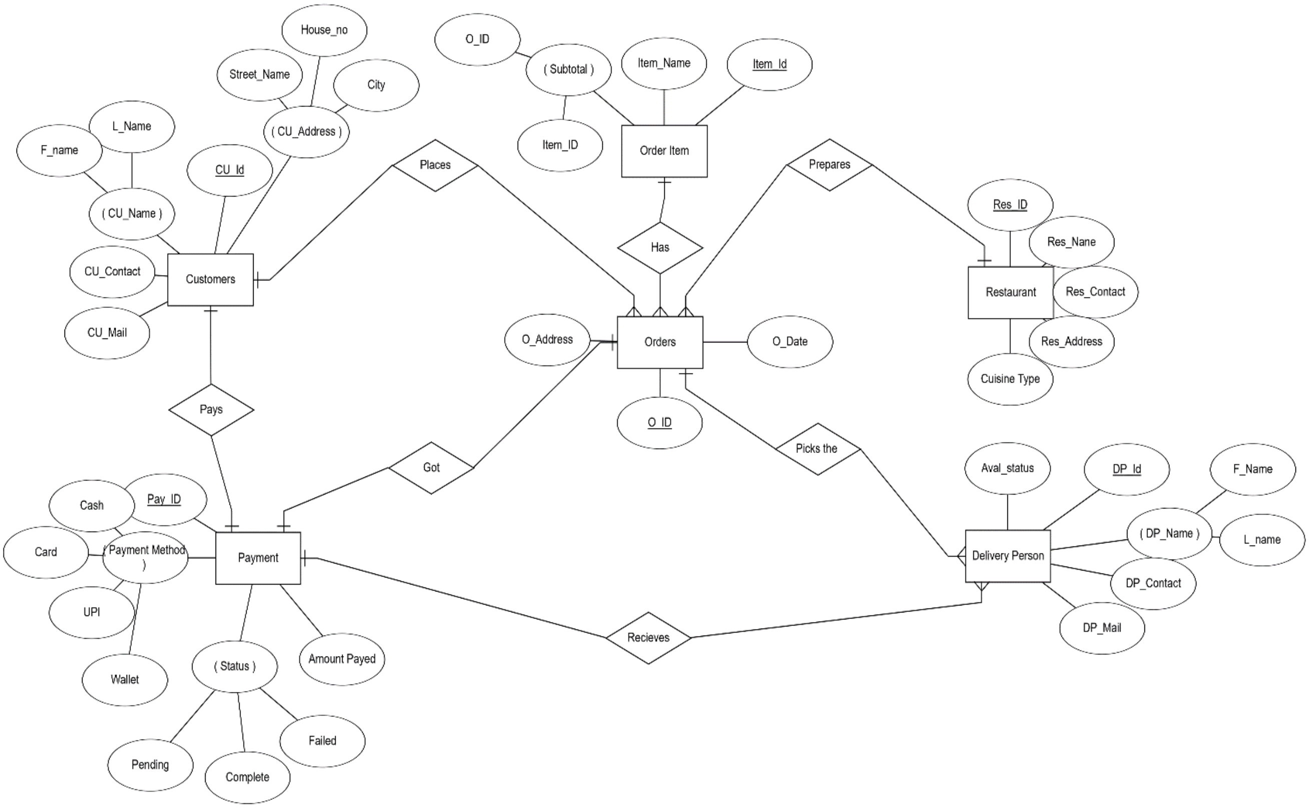
[5]. Order – Payment

- **Cardinality:** 1 : 1
- Each order has exactly one payment record, and each payment corresponds to exactly one order.

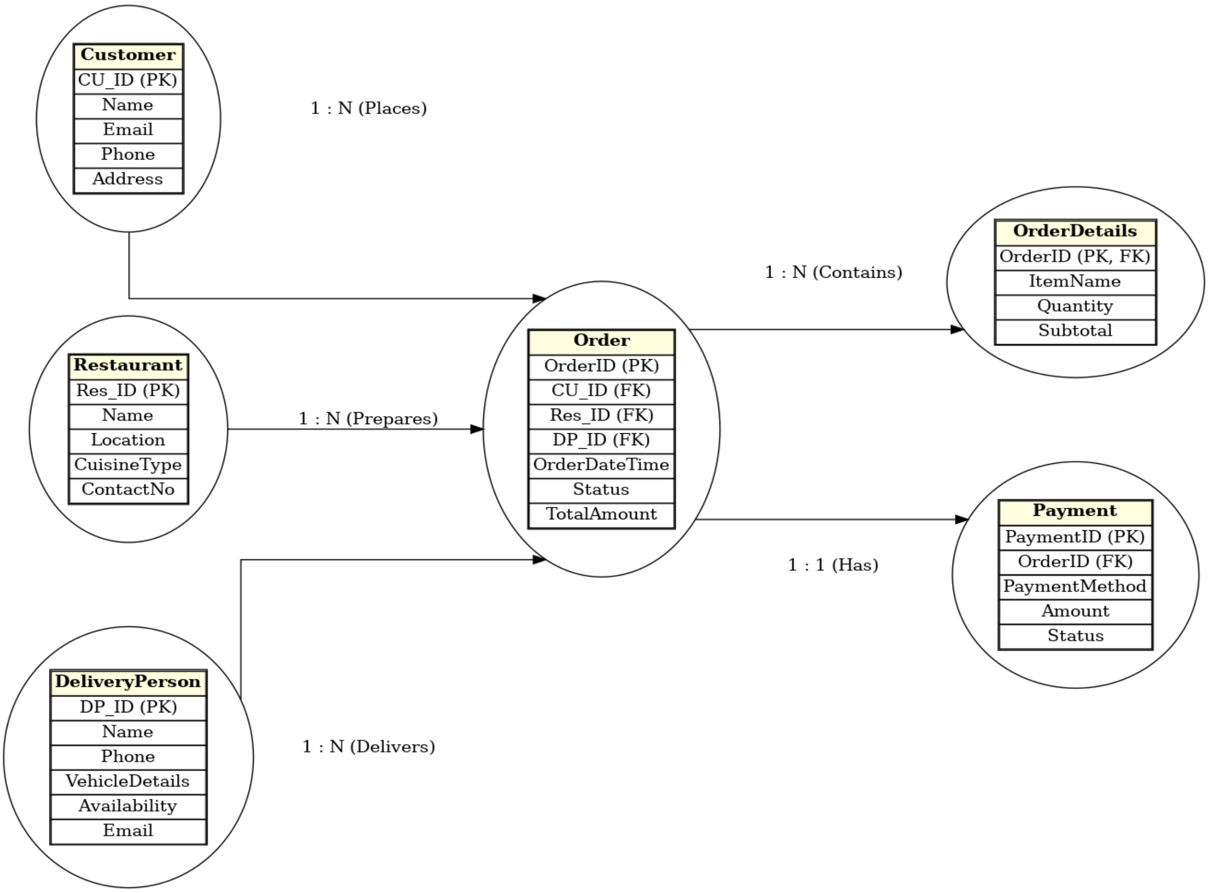
Example Attributes to Include in ERD:

Entity	Attributes
Customer	C U_ID(PK), C U_Name(F_name,L_name) ,CU_ Email, CU_Phone, CU_Address(Street_name,House_no,City).
Restaurant	Res_ID(PK), Res_Name, Res_Address, Cuisine Type, Res_Contact.
Order	OrderID(PK), CustomerID(FK), RestaurantID(FK), DeliveryPersonID(FK), OrderDateTime,TotalAmount.
OrderItem	O_ID(FK), Item_Name(Fk), Subtotal(Composite: O_ID + Item_ID).
DeliveryPerson	DP_I (PK), DP_Name(Composite), Phone, AvailabilityStatus,DP_Mail.
Payment	Pay_ID(PK), OrderID(FK), PaymentMethod(Cash/Card/UPI/Wallet),Amount,Status(Pending/Completed/Failed).

ER Diagram



ER Model



Customer
CU_ID (PK)
Name
Email
Phone
Address

OrderDetails
OrderID (PK, FK)
ItemName
Quantity
Subtotal

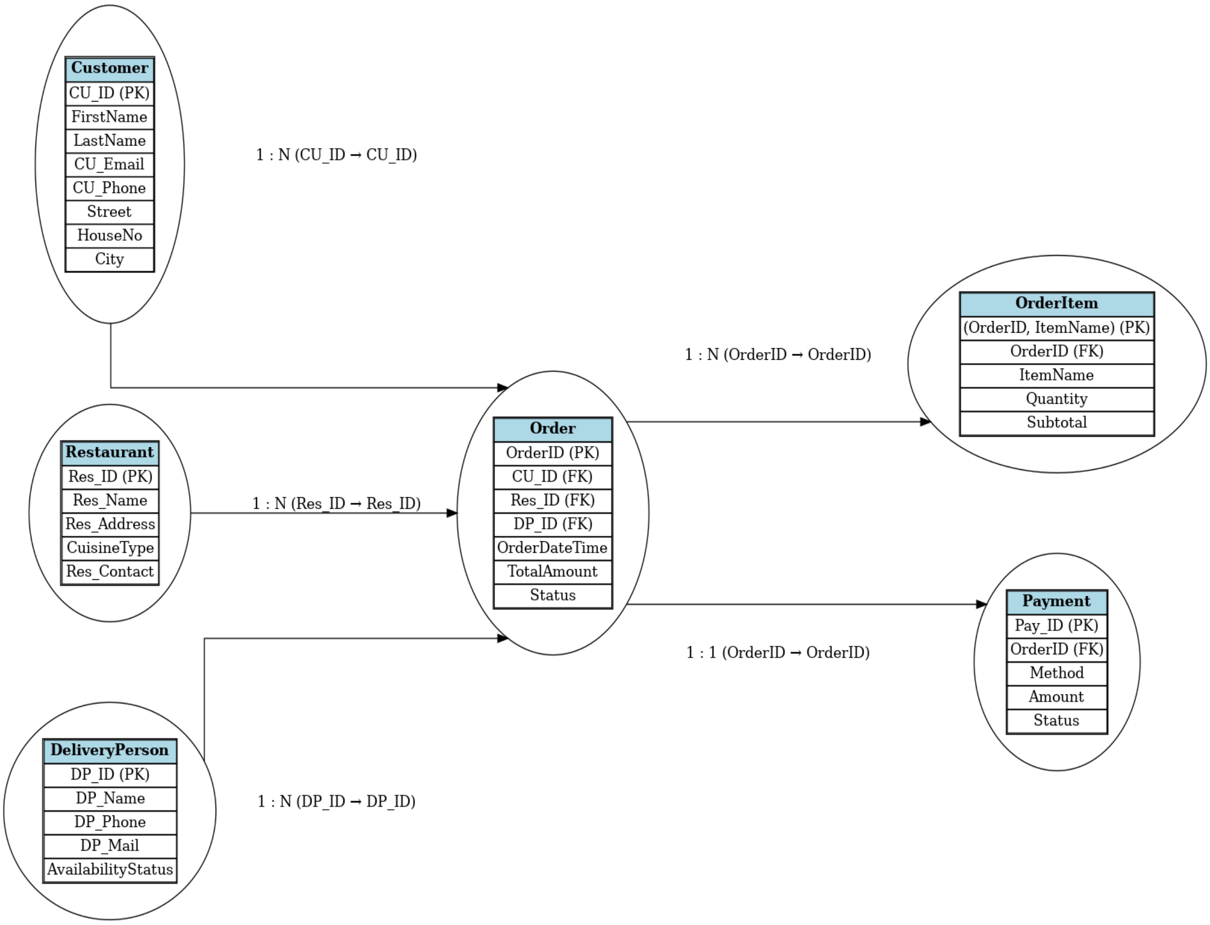
Restaurant
Res_ID (PK)
Name
Location
CuisineType
ContactNo

Order
OrderID (PK)
CU_ID (FK)
Res_ID (FK)
DP_ID (FK)
OrderDateTime
Status
TotalAmount

Payment
PaymentID (PK)
OrderID (FK)
PaymentMethod
Amount
Status

DeliveryPerson
DP_ID (PK)
Name
Phone
VehicleDetails
Availability
Email

Relational Table(Schema)



Customer
CU_ID (PK)
FirstName
LastName
CU_Email
CU_Phone
Street
HouseNo
City

Restaurant
Res_ID (PK)
Res_Name
Res_Address
CuisineType
Res_Contact

DeliveryPerson
DP_ID (PK)
DP_Name
DP_Phone
DP_Mail
AvailabilityStatus

Order
OrderID (PK)
CU_ID (FK)
Res_ID (FK)
DP_ID (FK)
OrderDateTime
TotalAmount
Status

OrderItem
(OrderID, ItemName) (PK)
OrderID (FK)
ItemName
Quantity
Subtotal

Payment
Pay_ID (PK)
OrderID (FK)
Method
Amount
Status

Queries

1. List all customer names and phone numbers.
Ans: п CU_Name, CU_Phone(Customer)

2. Retrieve all restaurants located in "Delhi".
Ans: σ Res_Address='Delhi'(Restaurant)

3. Find all food items with price less than 200.
Ans: σ Subtotal<200(OrderItem)

4. Get names of customers who placed orders.
Ans: п CU_Name(Customer⋈Customer.Cu_ID=Order.Cu_ID (Order))

5. Find details of food items ordered with a specific OrderID = 101.
Ans: σ O_ID=101(OrderDetails)⋈OrderItem

6. List all delivery persons who delivered orders.
Ans: п CU_Name(DeliveryPerson⋈Order)

7. Find all restaurants that offer "Pizza".
Ans: п Restaurant.Res_Name
(Restaurant⋈OrderItemσOrderItem.Name='Pizza'

8. Retrieve all orders with status "Delivered".
Ans: σ Status='Delivered'(Order)

9. Find all payments that are still pending.
Ans: σ Status='Pending'(Payment)

10. Get names of customers who made payments using UPI.
Ans: п CU_Name(Customer ⋈ Order ⋈ Payment σ PaymentMethod='UPI')

11. Find customers who have placed more than one order.
Ans: п Cu_ID(σ Count(O_ID)>1(Order))

12. List food items that belong to RestaurantID = 5.
Ans: σ Res_ID=5(OrderItem)

13. List delivery persons who are currently available.
Ans: σ AvailabilityStatus='Available'(DeliveryPerson)

14. Retrieve customers who ordered food items costing more than 500.

Ans: п Customer.CUName(Customer ⋈Order ⋈OrderDetails ⋈OrderItem σ Price>500)

15. Find restaurant names from which a customer with CustomerID=10 has ordered.

Ans: п Restaurant.CU_Name(Restaurant ⋈ Order σ CustomerID=10)

16. Get payment details of orders handled by DeliveryPersonID=3.

Ans: п Payment (Payment ⋈ Order σ DeliveryPersonID=3)

17. Find customers who have never placed an order.

Ans: п CU_ID, CU_Name(Customer) – п CU_ID,CU_Name(Customer ⋈ Order)

18. Retrieve food items that were ordered at least once.

Ans: п OrderItemID,Name(OrderItem ⋈OrderDetails)

19. Show all restaurants.

Ans: п Res_ID,Res_Name,Res_Address(Restaurant)

20.Find all food items with price less than 300.

Ans: σ Price<300(OrderItem)

21. Find names of customers who placed an order OR made a payment

Ans: п CU_Name(Customer ⋈ Order) ∪ п CU_Name(Customer ⋈ Payment)

22. Find customers who placed orders but did not make any payment.

Ans: п CU_ID,CU_Name(Customer ⋈ Order) – п Cu_ID,Name(Customer ⋈ Payment)

23. Find customers who have ordered all food items from RestaurantID = 5**.**

Ans: п CU_ID(Order ⋈ OrderItem) ÷ п FoodItemID(σ RestaurantID = 5(FoodItem))

24. Show all restaurants serving "Chinese" cuisine.

Ans: σ CuisineType=’Chinese’(Restaurant)

25. Show all orders placed on "2025-08-24".

Ans: σ OrderDateTime=’2025−08−24’(Order)