

P3 Logical ERD Model

Group 2:

1. Madhura Adadande
2. Kiran Sathya Sunkoji Rao
3. Harshitha Chennareddy
4. Yue Li
5. Sneh Patil

- In earlier models, we might have repeated staff information in each shift. But now, we store staff details once in the **Staff** table, and each shift simply references the relevant staff member using their StaffID. This way, you don't store the same staff data multiple times, and it makes shift tracking cleaner and easier.

- **Dish Details**

Instead of storing details like dish prices and ingredients every time an order is placed, we only store a reference to the dish (using DishID) in the **OrderDetails** table.

- **Efficient Reservation and Order Management:**

We've kept the **Reservation** and **Order** data separate. This means that reservation details are not repeated every time an order is made. Orders reference the reservation, but they don't store all the reservation data again, keeping things streamlined.

- **Centralized Supplier Information:**

We store supplier details once in the **Supplier** table and link it to **Inventory** using the SupplierID foreign key. This keeps supplier data from being repeated across every inventory item and ensures the supplier details are accurate and easy to update.

Additional Entities:

- **New Entity - OrderDetails:**

The **OrderDetails** table was added to capture specific line items in an order. This helps track which dishes are ordered, their quantities, and the total price for each order. Instead of mixing all this information into the **Order** table, we now keep the details separate, which makes managing large orders much easier.

- **New Entity - Shifts:**

The **Shifts** table was introduced to track staff shifts separately. This means we can keep track of which staff member worked which shift, without cluttering the **Staff** table. This approach also makes managing shift schedules and payroll much simpler.

- **Detailed Dish Information:**

We've added more detailed attributes for dishes, such as **IsAvailable**, **IsVegetarian**, and **PreparationTime**. These new attributes give a clearer picture of each dish, which is essential for managing the menu and ensuring customers have the information they need when ordering.

- **Supplier and Inventory Management:**

We recognized the need to keep supplier data separate from inventory details, so the **Supplier** table was added. This keeps the inventory related to the supplier without repeating supplier information for each inventory item, which makes inventory management more efficient.

- **Enhanced Staff Scheduling:**

The **Shifts** entity is a new addition to track the exact timing of staff shifts, which wasn't previously captured as clearly. This allows better scheduling and ensures we don't have to repeatedly store shift times for each staff member.

1. New Attributes Added:

- **Shifts Entity:**
 - **ShiftDate:** Specifies the date of the shift.
 - **StartTime:** The starting time of the shift.
 - **EndTime:** The ending time of the shift.
- **Dish Entity:**
 - **IsAvailable:** Indicates if the dish is available for serving.
 - **IsVegetarian:** Indicates if the dish is vegetarian.
 - **PreparationTime:** The time it takes to prepare the dish.
- **Feedback Entity:**
 - **Primary Key:**
 - **FeedbackID:** Unique identifier for each feedback entry.
 - **Foreign Keys:**
 - **CustomerID (FK):** References Customer.CustomerID, linking feedback to a specific customer.
 - **OrderID (FK):** References Order.OrderID, linking feedback to a specific order.

2. New Foreign Keys Added:

- **Staff Entity:**

RestaurantID (FK): Links each staff member to a specific restaurant.

- **Shifts Entity:**

StaffID (FK): Connects shifts to the staff member working them.

- **Dish Entity:**

MenuID (FK): Associates each dish with a specific menu.

- **OrderDetails Entity:**

OrderID (FK): Links order details to specific orders.

DishID (FK): Connects order details to specific dishes.

- **Foreign Keys:**

CustomerID (FK): References Customer.CustomerID, linking feedback to a specific customer.

OrderID (FK): References Order.OrderID, linking feedback to a specific order.

3. Changes in Relationships:

- **Staff Entity:**

- The **Staff** entity is now connected to the **Restaurant** entity through the RestaurantID foreign key.

- **Shifts Entity:**

- The **Shifts** entity is now connected to the **Staff** entity through the StaffID foreign key.

- **Dish Entity:**

- The **Dish** entity is now linked to the **Menu** entity through the MenuID foreign key.

- **OrderDetails Entity:**

- The **OrderDetails** entity now links both the **Order** and **Dish** entities through the OrderID and DishID foreign keys.

- **Customer ↔ Feedback:**

A **customer** can provide multiple **feedback** entries. This is a one-to-many relationship where each feedback is linked to a specific customer via the **CustomerID** foreign key.

- **Order ↔ Feedback:**

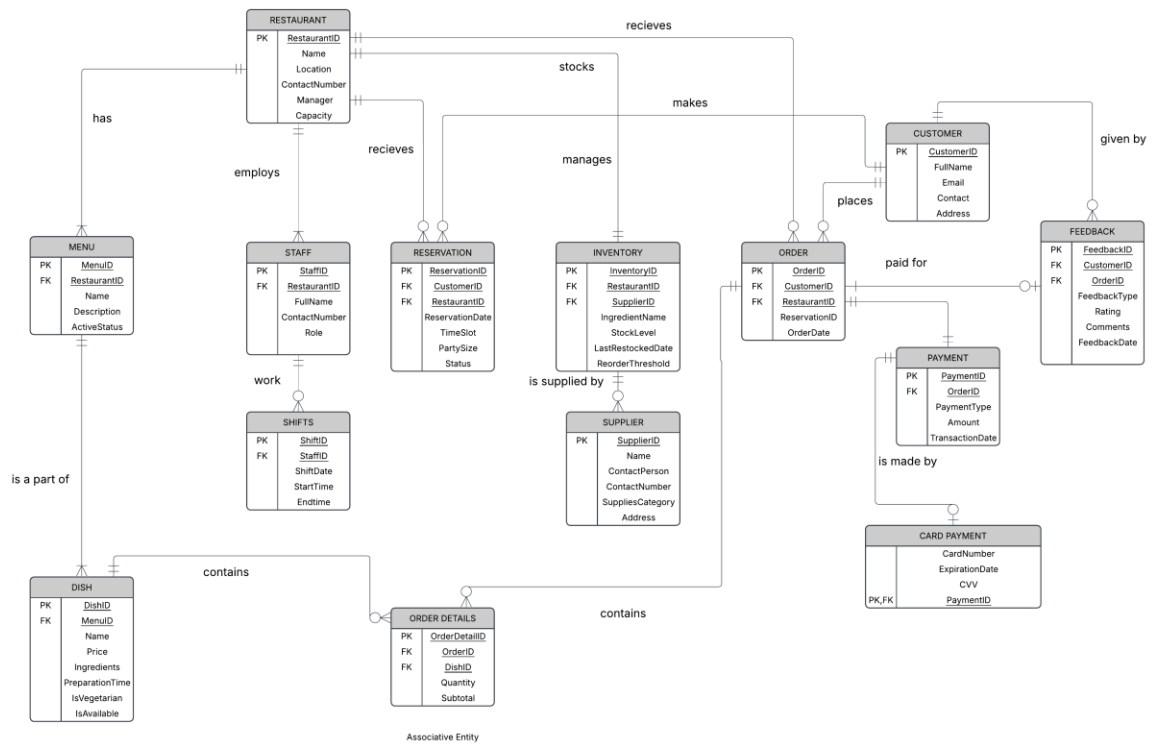
An **order** can have multiple **feedback** entries, but each piece of feedback is

associated with a single order. This relationship is defined by the **OrderID** foreign key.

Summary

- **Staff** entity and ShiftDate, StartTime, and EndTime in the **Shifts** entity provides more detailed tracking of staff scheduling.
- New attributes in the **Dish** entity like IsAvailable, IsVegetarian, and PreparationTime enhance the ability to manage the availability and characteristics of dishes.
- Foreign keys were introduced to better link the entities: **Staff** now links to **Restaurant**, **Shifts** links to **Staff**, **Dish** links to **Menu**, and **OrderDetails** links both **Order** and **Dish**.
- These changes result in a more normalized, efficient, and well-connected database schema, enhancing the overall organization and management of the restaurant's data.
- These updates ensure better tracking and clarity of operations across different entities, such as orders, payments, feedback, and inventory management. New foreign keys were added to ensure the relationships between entities are correctly represented in the logical data model. This helps maintain referential integrity across the schema, which is crucial for normalizing the database structure.

Logical ERD:



Lucid Chart Link:

https://lucid.app/lucidchart/6e4a63b2-08cb-448e-a531-cbb57812cea6/edit?invitationId=inv_5fa0b4f7-f707-41ca-947b-58429fbe277e&page=0_0#