IE 6700
Case study
Group 4

Team information

Shraman ARYA	NUID: - 002101258
Sai Gowtham Babu AMBURI	NUID: - 002190843

Business problem

Tasty Restaurant needs a new database system to support its Restaurant database. In their restaurant there are many customers ordering food daily and many employees working at the same time. They want to create a database that store the customer details, items ordered and employee details. This helps them record the quantity of food and raw materials they need to prepare.

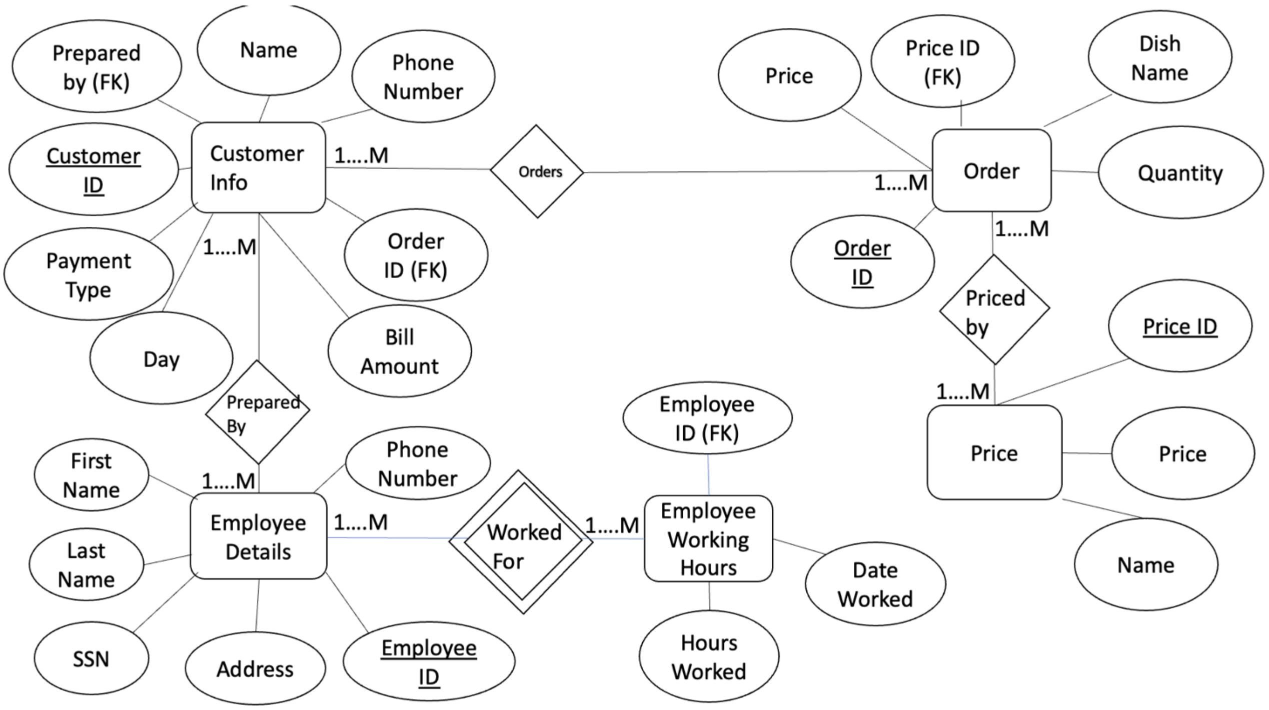
In implementing this database, the following rules must be followed:

1. An employee must take orders from a customer and then make a separate entity about customer details and the items they ordered.
2. A Customer can place one order at a time, and orders may include one to many products.
3. The food can be prepared by one or many chefs.
4. At a particular time, there can be one to many employees working.

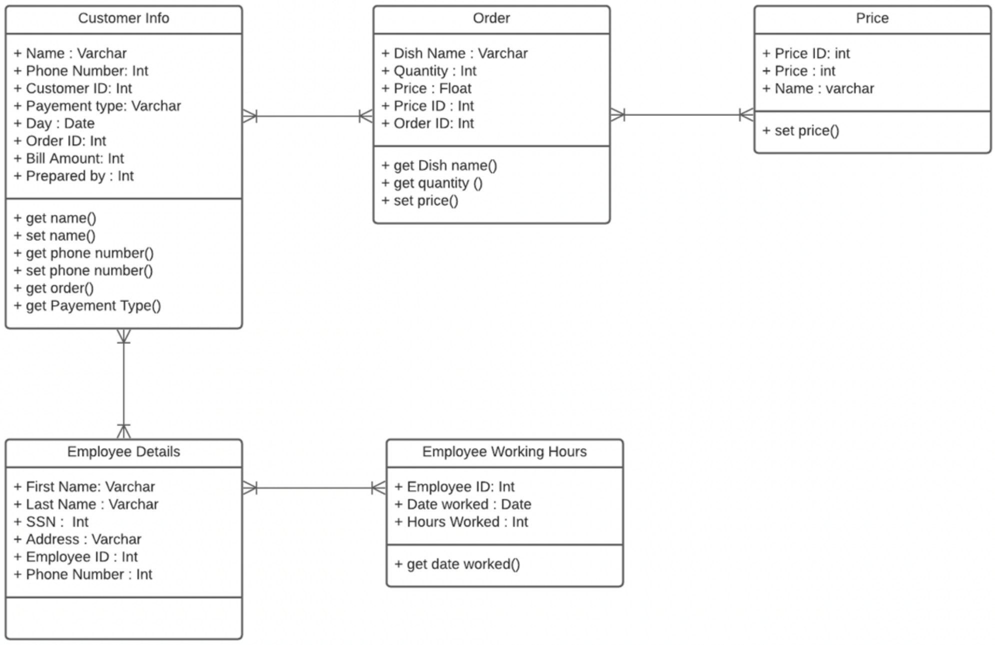
Design

To create a database that meets the requirements of the problem statement, we first need to identify the required entities such as Customer details, order details, price of items, employee details and their work hours. Since, there can be one to many customers ordering food at the same time, there will be a one – many relationship between customers and order. In the same way, two different customers can order same item and a customer can order items multiple times, there will also be a one – many relationship. In order entity, it contains the dish name, quantity and relative price based on dish name and quantity. The total sum of the price will be reflected in customer entity. The price of each item will be taken from price entity where each dish will have a unique ID and relative price associated with it. This also contains a one – many relation as many dishes can be ordered at the same time and vice versa. Coming to the employer information, an entity contains all the details of the employee and maps it to the customer entity. In customer entity, an attribute is created detailing who have made the dish (a dish can be made by one - many chefs). Another entity details the time and date each employee worked.

Entity Relationship Diagram:



Unified Modeling Language (UML) Class Diagram:



Relational Model:

Customer Information (Name, Phone Number, Prepared by (FK), Customer ID (PK), Payment type, Day, Order ID (FK), Bill amount)

Order (Order ID (PK), Price, Price ID (FK), Quantity, Dish Name)

Price (Price ID (PK), Price, Dish name)

Employee Details (First name, Last name, SSN, Address, Employee ID (PK), Phone number)

Employee Working Hours (Employee ID (FK), Date Worked, Hours Worked)