RESTAURANT MANAGEMENT DATABASE PROJECT

BY

SHIWANI RAJAGOPALAN

Restaurant Management Database project overview:

This project focusses on all the information that is collected in a restaurant from ordering till billing and maintaining a check on stock of ingredients. Storing of these information will make each and every data organized and will be helpful for the restaurant employees to pull data about anything easily.

This project will be done individually by SHIWANI RAJAGOPALAN

Purpose:

The purpose of this project is to organize data to help people working in a restaurant to be able to track anything in a given time. For example, information like number of orders that have been given out and payments made can benefit the employees in situation where they have to calculate the revenue made by the restaurant and compare revenues with other days to be able to get an idea of whether if there has been profit or not.

End User:

Employees of the restaurant who administer this work.

Data:

There are existing data to populate in the table but I will be manually populating certain entries using insert statements.

RDBMS Software:

MySQL

Step 1: Scenario and Database Requirement

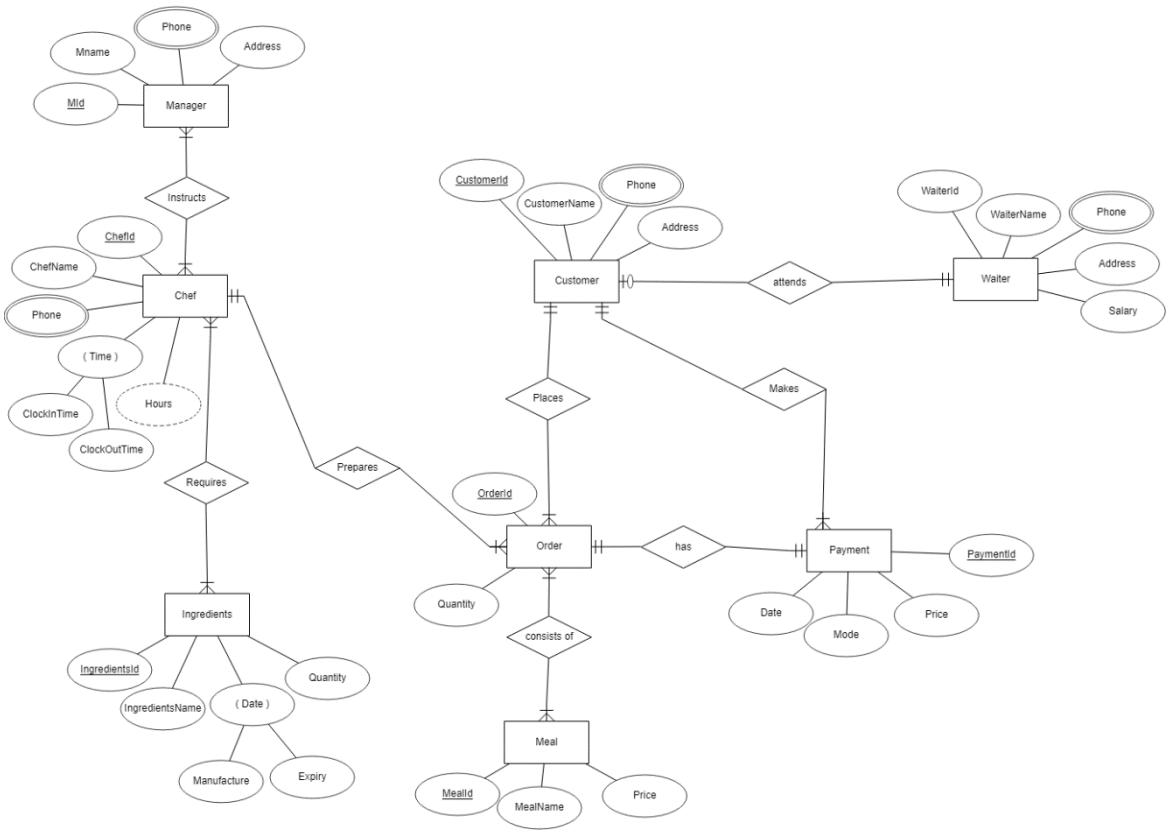
Restaurant Management System Database project focusses on maintaining records of information related customers who are ordering a meal in the restaurant, order details of the customer, transaction information, employee related like chef, waiters and managers who are working in the restaurant. In addition to all of this, record of ingredients required to make dishes are also kept track of so as to know which ingredients are going out of stock.

The main purpose of creating this database system is to help the restaurant keep track of which meals are available at any particular time, to check if the ingredients are available or not, and to have information related to the customer who has placed the order so as to map the order to the customer who has asked for it.

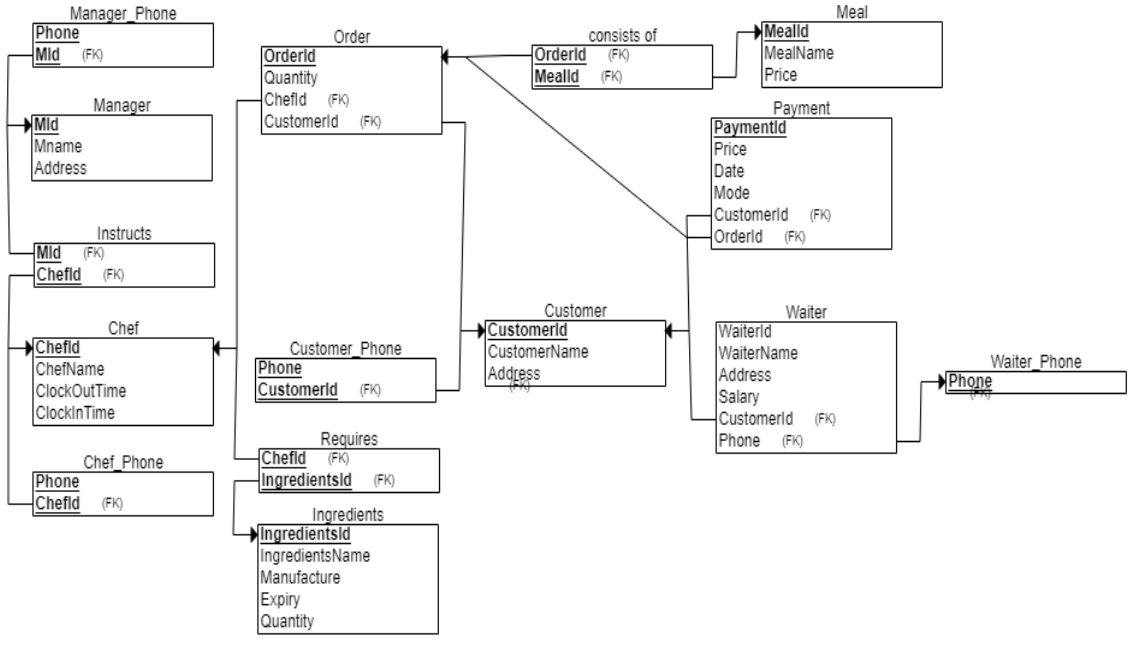
The Restaurant Management System will keep data of the following:

1) Every Manager has Unique manager id, name, phone number and address.
2) Each chef in the restaurant has a unique chef id, name, phone number, clock in and clock out times, and a derived attribute hours calculated from the total number of hours the chef has worked from clock in time to clock out time.
3) Each Ingredients in the pantry has a unique ingredient id, name, date which comprises of the date of manufacture and expiry date, and quantity of those ingredients in stock.
4) Each customer who comes to the restaurants has a unique customer id, name, phone number, and address.
5) Each order taken from the customer gets a unique order id to identify the orders made by customers and quantity of that particular meal.
6) Each meal has a unique meal id, name and price.
7) Every waiter working in the restaurants has a unique waiter id, name, phone number, address and salary information.
8) Each payment for the particular order has a payment id, price, mode of transaction, and date.
9) Each manager must instruct at least 1 chef in the restaurant and each chef must have at least 1 manager to instruct them.
10) Each chef requires at least 1 ingredient to make a dish and each ingredient must be used by at 1 or more chefs.
11) Each chef must have at least 1 order to prepare a dish and each order must be handled by only 1 chef.
12) Each customer must place at least 1 order and each order can have only 1 customer ordering it.
13) Each order must have at least 1 meal listed from menu and one of the meal must be present in at least one order.
14) Each order must have only 1 payment and each payment has only 1 order.
15) Each customer makes a payment for the order that has been placed and they have to make at least 1 payment. Each payment must have at least 1 order to process the payment.
16) Each customer is attended by only 1 waiter and each waiter can attend 1 customer or no customers at all.

Step 2: Enhanced Entity Relationship Diagram (EERD)



Step 3: Relational Schema



Step 4: Normalization

Most of the tables are in first normal form as there are multi-valued attributes like phone number in tables manager, customer, chef and waiter where separate records for each of them having multi phone numbers are created.
So far I haven’t worked on whether if the table will be in second or third normal form so depending on the records inserted in the tables, I will work on whether if the tables have transitive dependencies or not.