**Restaurant Management Database System**

***Data Management & SQL for Analytics***

by **Mahalakshmi Jayavel, Uday Naidu Gadwala, Maazin Sherif & Someshawar Suresh**

*Group 8*

Clark University

BAN5501-02-S23

Professor Theyab Alhwiti

April 20, 2023

**Project Description**

Restaurants are the backbone of American society, comprising a sizable percentage of small businesses. Many are also run by people of various ethnic backgrounds and historically have been a way for immigrants to start anew. It is an entrepreneurial way that culture is shared. The mom and pop style of restaurant is often susceptible to human error and vulnerable to macro-economic conditions. The database management system that we present is designed to be simple enough for such restaurant owners to master, yet powerful in its functionality. Our goal was to create a user-friendly and scalable solution that can be customized to meet the needs of a restaurant and promote lean operation.The system was designed to manage all basic aspects of the restaurant with possible added functionality as needed. The system includes a comprehensive customer data management, order management, ingredient inventory management, reservations management and reservation management. The system is also enabled with functionality for delivery and delivery rating. Restaurants are a small profit margin class of business and can benefit from efficient management of data to minimize error.

So how does this system function?

Once a customer makes a reservation or places a delivery order, data from that transaction can be presumed to be saved with the customer’s consent, for customer convenience. In the interest of limiting the scope to a practical extent, there are assumptions that were taken regarding the DBMS. These assumptions help fill the gaps where our system does not extend and were partially due to the absence of an ideal dataset. An example is that the dataset was limited from 500 entries to 60. That was because of missing data such as only first name, missing phone, missing email address, etc, and would have taken an unreasonable amount of time to fix. We therefore opted for a more practical approach.

The DBMS effectively handles dine-in, pick-up and delivery orders while also enabling delivery customers to leave reviews. The database design is structured in a way that ensures efficient storage, retrieval, and manipulation of data, as well as maintaining data integrity and consistency. Customers place orders for menu items, which consist of various ingredients sourced from vendors. Each menu item can have many ingredients, and an ingredient can be a part of multiple menu items. The MenuItem\_Ingredient associative entity links the MenuItem and Ingredient entities, allowing for easy management of the relationship between them. Additionally, the Employee entity stores information about the restaurant's staff, with each employee assigned to a specific job, like line cook, waiter, manager, or lead cook. Reservations can be made by registered customers, and the Reservation entity captures essential details like the number of guests and reservation date and time.

Finally, the Order and MenuItem entities are connected through the OrderItem associative entity, which represents the relationship between orders and menu items while also capturing the quantity of each menu item in an order. This structure enables efficient management of orders and facilitates generating reports and insights into the restaurant's performance. Ideally, this entity would be incorporated into the DBMS but unfortunately, it did not make it past the ERD due to limitations of time and effort.

**Database Environment**

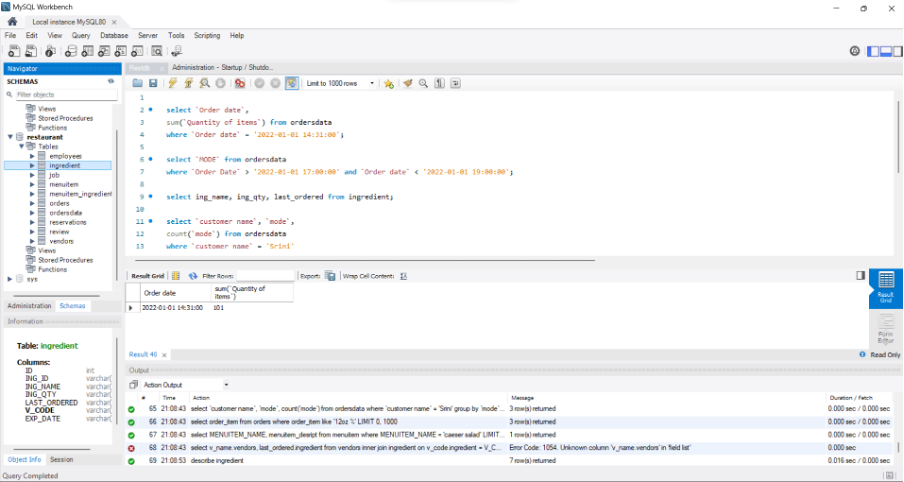
Client Profile: -

The business is a restaurant consisting of 16 employees that serves indigenous cuisine. Food menu is inspired from famous Indian and American dishes and the restaurant offers three modes such as delivery, pickup and dine-in. The database size is 728 bytes which holds employees, customers, and ingredient/menu data. Attributes such as order details, customer details, reservations, employees, vendor purchase and reviews are some of the major input data.

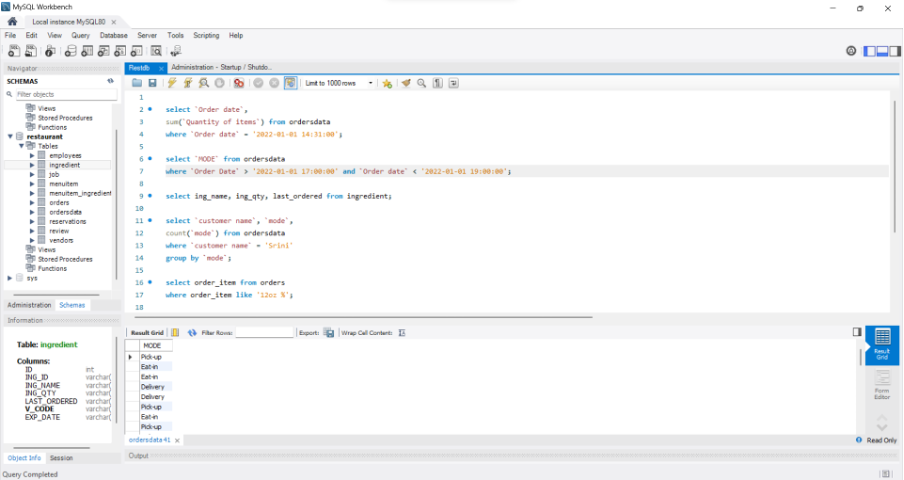
User Profile: -

The database is used by the manager to keep track of storage, menu changes, customer reviews and reservations. Managers should have a thorough knowledge of how the database works, however there should be a person employed to take care of the database and the front-end. Most used database is MongoDB for employees, companies, and items whereas MySQL is used for orders, customers, and transactions. For this project we’re using MySQL as the primary database to display desired results. The user interface is simple and easy to navigate however basic sql query knowledge is required to produce results.

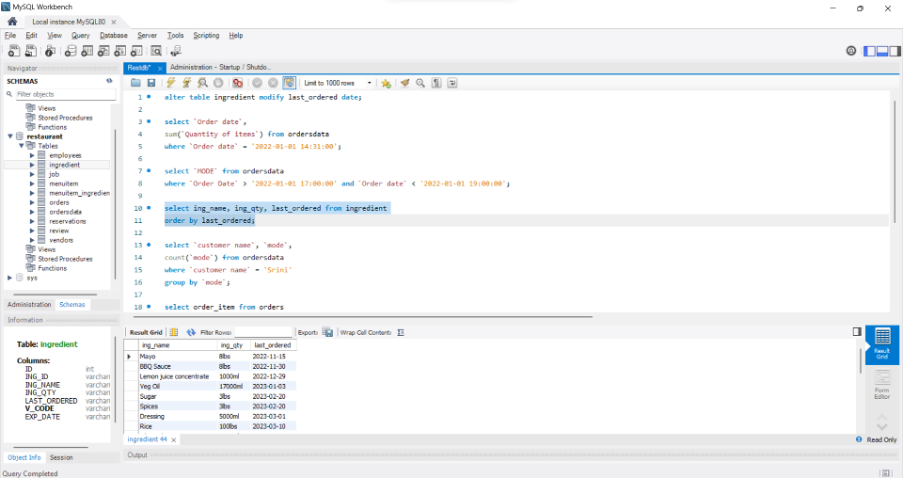
Some of the results for questions below are displayed:

- Looking up orders for number of orders placed on a particular date.**

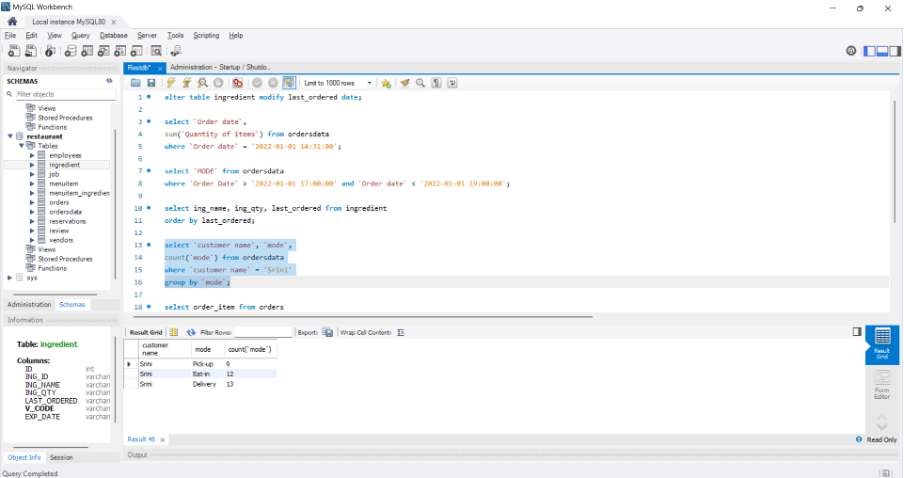
- Looking up the most common order type between 5pm and 7pm

**

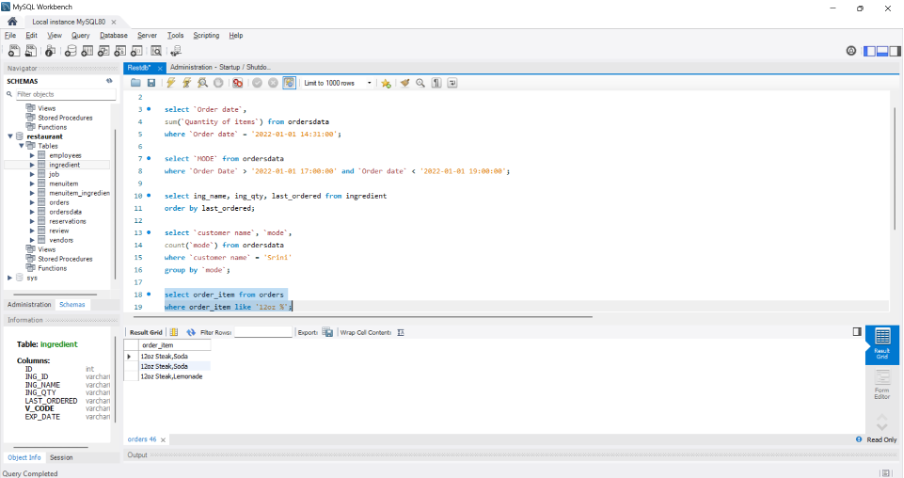
- What is the restaurant’s storage order history?



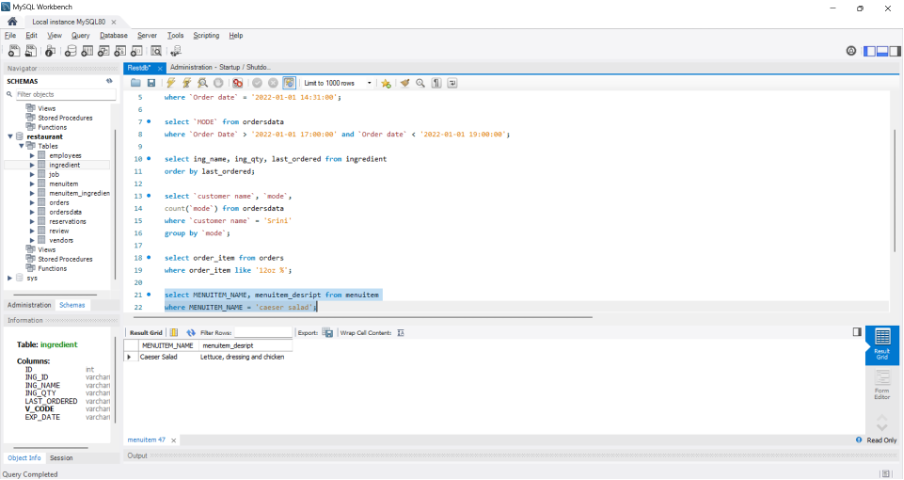
- What is specific customer’s preferred order type?

**

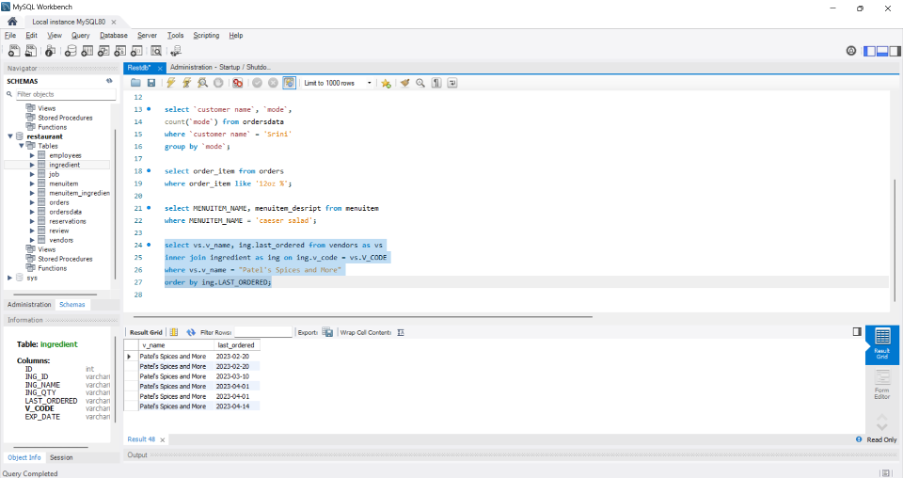
- What did people who ordered steak get for drinks?



- What are the ingredients for a Caesar salad?

**

- When was the last time the restaurant ordered from Patel’s Spices and More?

**

**Goals & Objectives**

The User learning curve expected for effortless utilization of the DBMS is not unreasonable as it was made with simplicity of use in mind. SQL is an easy language to learn, especially DML. If the client wants to do something past creating query reports and making entries, such as adding entities and crafting new relationships, they would have to have expertise with database management which might pose a risk to our goal of simplicity. These are some of the client goals we expect to have fulfilled:

* - To develop a user-friendly and intuitive interface that will enable restaurant staff to manage orders, inventory, and customer data efficiently.
* - To implement features that will improve customer service, such as online reservations and mobile ordering.
* - To provide real-time data analytics and reporting to help restaurant owners make informed business decisions.
* - To integrate with existing point of sale (POS) systems, as well as third-party services such as food delivery platforms.
* - To facilitate vendor relationships and avoid ingredient stock-outs.

**Database Profile**

*Business Rules:*

1. Reservations:

a. A customer can make one reservation per day, and must provide their name, phone number, and party size. Each reservation can only be made by one customer.

b. Reservations can only be made for future dates and times, not for past or current dates/times.

c. Reservations must be made within the restaurant's operating hours (1pm - 9pm).

d. The maximum number of guests per reservation is limited by the restaurant's capacity.

2. Ingredients:

a. Ingredients must have a name and quantity.

B. Each Ingredient must have one Vendor. Each V\_Code may be associated with multiple Ing\_ID.

c. Inventory updates should be recorded whenever ingredients are added, removed, or used in meal preparation.

3. Menu:

a. Each menu item must have a name, description and price.

b. Menu items can be categorized into different sections, such as appetizers, main courses, and desserts.

c. Menu items must be updated if ingredients are unavailable or if the recipe changes.

d. Seasonal or special menu items can be added or removed based on availability and demand.

4. Employee:

a. Each employee member must have a unique identifier, name, job code, and contact information.

b. Employee schedules should be created based on the restaurant's operating hours and employee availability.

c. Employee performance and attendance should be tracked for evaluation purposes.

5. Job:

a. Each employee member has a specific role. Each role may have multiple employees except for Lead cook and manager.

6. Customer data:

a. Customer information, such as name, contact number, and email, should be securely stored and used for marketing and service purposes, with the customer's consent.

7. Review rules:

a. Rev\_ID must be unique for each review.

b. Cust\_ID must reference a valid customer.

c. Order\_ID must reference a valid order (dine-in, pick-up, or delivery).

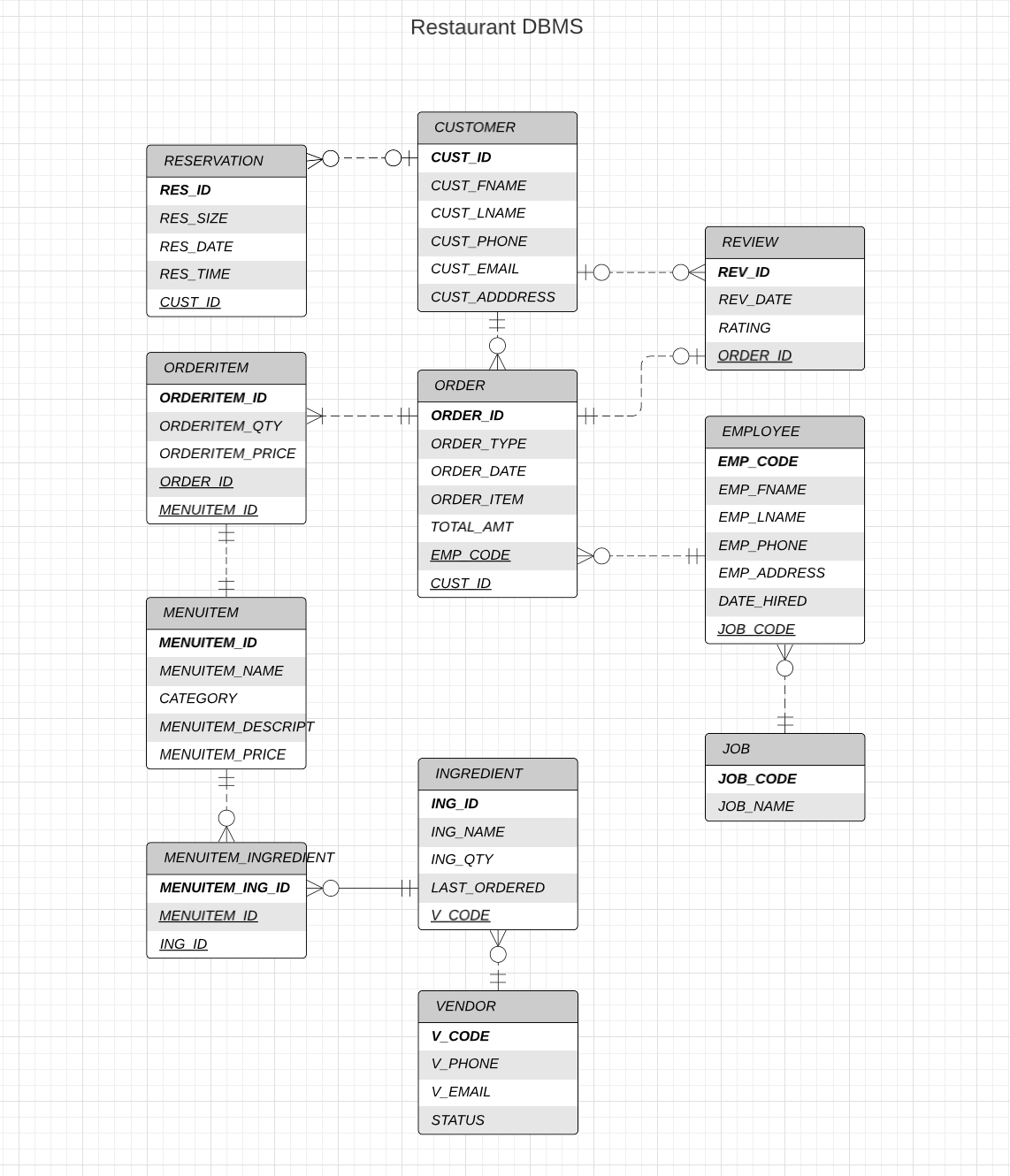
d. Rating must be within a predefined range (e.g., 1 to 5).

e. Rev\_Date must be a valid date, ideally set to the date when the review is submitted.

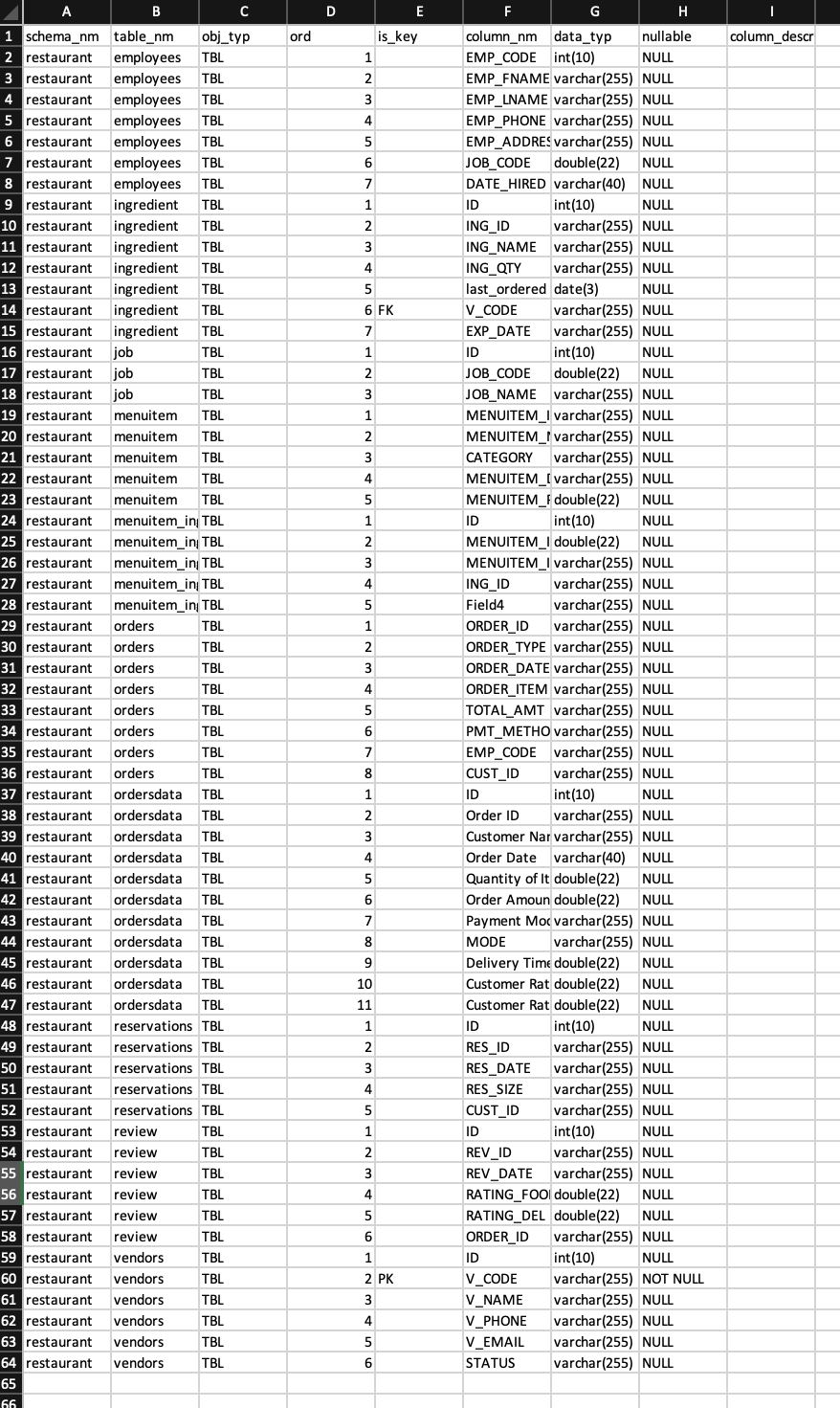
**Revised list of Entities**

The tables discussed in this report are as follows:

* - Customer
* - MenuItem
* - MenuItem\_Ingredient
* - Ingredient
* - Order
* - OrderItem
* - Employee
* - Job
* - Vendor
* - Reservation
* - Review



Data Dictionary



**References**

DePaul University. (n.d.). GIS Data Resources: Chicago. Retrieved from<https://gis.depaul.edu/shwang/teaching/geog495/LN10.htm>

GeneratorMix. (n.d.). Random Phone Number Generator. Retrieved from<https://www.generatormix.com/random-phone-numbers>

Harris, M. (n.d.). Restaurant Order Details. Kaggle. Retrieved from<https://www.kaggle.com/datasets/mohamedharris/restaurant-order-details?resource=download>

Mobilefish. (n.d.). Random Test Data Generator. Retrieved from<https://www.mobilefish.com/services/random_test_data_generator/random_test_data_generator.php>

Random Name Generator. (n.d.). Retrieved from<http://random-name-generator.info/index.php?n=10&g=1&st=2>

Software Ideas Modeler. (n.d.). Food Ordering System ER Diagram. Retrieved from<https://www.softwareideas.net/a/1597/food-ordering-system-er-diagram->