

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.

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema, including tables like `employees`, `ingredient`, `orders`, and `ordersdata`. The main editor contains a SQL query with three parts: a summary of orders for a specific date, a list of ingredients for a specific order, and a count of orders by mode for a specific customer.

```

1
2 * select 'Order date',
3   sum('Quantity of items') from ordersdata
4   where 'Order date' = '2022-01-01 14:31:00';
5
6 * select 'MODE' from ordersdata
7   where 'Order Date' > '2022-01-01 17:00:00' and 'Order date' < '2022-01-01 19:00:00';
8
9 * select ing_name, ing_qty, last_ordered from ingredient;
10
11 * select 'customer name', 'mode',
12   count('mode') from ordersdata
13   where 'customer name' = 'Srinil';

```

The **Result Grid** shows the output of the first query:

Order date	sum('Quantity of items')
2022-01-01 14:31:00	101

The **Output** pane shows the execution progress of the queries:

Time	Action	Message	Duration / Fetch
65 21:08:43	select 'customer name', 'mode', count('mode') from ordersdata where 'customer name' = 'Srinil' group by 'mode'...	3 row(s) returned	0.000 sec / 0.000 sec
66 21:08:43	select order_item from orders where order_item like '12oz %' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
67 21:08:43	select MENUITEM_NAME, menuitem_description from menuitem where MENUITEM_NAME = 'casser salad' LIMIT...	1 row(s) returned	0.000 sec / 0.000 sec
68 21:08:43	select v_name,vendors, last_ordered ingredient from vendors inner join ingredient on v_code ingredient = V_C...	Error Code: 1054. Unknown column 'v_name,vendors' in field list	0.000 sec
69 21:08:53	describe ingredient	7 row(s) returned	0.016 sec / 0.000 sec

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

The screenshot shows the MySQL Workbench interface. The left sidebar displays the database schema, including tables like `employees`, `ingredient`, `orders`, and `ordersdata`. The main editor contains a SQL query with three parts: a summary of orders for a specific date, a list of ingredients for a specific order, and a count of orders by mode for a specific customer.

```

1
2 * select 'Order date',
3   sum('Quantity of items') from ordersdata
4   where 'Order date' = '2022-01-01 14:31:00';
5
6 * select 'MODE' from ordersdata
7   where 'Order Date' > '2022-01-01 17:00:00' and 'Order date' < '2022-01-01 19:00:00';
8
9 * select ing_name, ing_qty, last_ordered from ingredient;
10
11 * select 'customer name', 'mode',
12   count('mode') from ordersdata
13   where 'customer name' = 'Srinil'
14   group by 'mode';
15
16 * select order_item from orders
17   where order_item like '12oz %';
18

```

The **Result Grid** shows the output of the first query:

Order date	sum('Quantity of items')
2022-01-01 14:31:00	101

The **Output** pane shows the execution progress of the queries:

Time	Action	Message	Duration / Fetch
65 21:08:43	select 'customer name', 'mode', count('mode') from ordersdata where 'customer name' = 'Srinil' group by 'mode'...	3 row(s) returned	0.000 sec / 0.000 sec
66 21:08:43	select order_item from orders where order_item like '12oz %' LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
67 21:08:43	select MENUITEM_NAME, menuitem_description from menuitem where MENUITEM_NAME = 'casser salad' LIMIT...	1 row(s) returned	0.000 sec / 0.000 sec
68 21:08:43	select v_name,vendors, last_ordered ingredient from vendors inner join ingredient on v_code ingredient = V_C...	Error Code: 1054. Unknown column 'v_name,vendors' in field list	0.000 sec
69 21:08:53	describe ingredient	7 row(s) returned	0.016 sec / 0.000 sec

- What is the restaurant's storage order history?

The screenshot shows the MySQL Workbench interface. On the left, the 'SCHEMAS' pane displays a tree view of the database structure. The 'restaurant' schema is expanded, showing tables: employees, ingredient, job, menuitem, menuitem_ingredient, orders, ordersdata, reservations, review, vendors, and views. The 'ingredient' table is selected, and its columns are listed: ID, ING_ID, ING_NAME, ING_QTY, LAST_ORDERED, V_CODE, and EXP_DATE.

The main editor window shows a SQL script with 18 lines of code. The script includes an ALTER TABLE statement, a SELECT statement with a subquery, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, a SELECT statement with a WHERE clause, and a SELECT statement with a WHERE clause.

The 'Result Grid' pane at the bottom shows the results of the query. It contains a table with 3 columns: ing_name, ing_qty, and last_ordered. The data is as follows:

ing_name	ing_qty	last_ordered
Mayo	8lbs	2022-11-15
BBQ Sauce	8lbs	2022-11-20
Lemon juice concentrate	1000ml	2022-12-29
Veg Oil	17000ml	2023-01-03
Sugar	3lbs	2023-02-20
Spices	3lbs	2023-02-20
Dressing	5000ml	2023-03-01
Rice	100lbs	2023-03-10

- What is specific customer's preferred order type?

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Views

Stored Procedures

Functions

restaurant

Tables

employees

ingredient

job

menuitem

menuitem_ingredient

orders

ordersdata

reservations

review

vendors

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Table: ingredient

Columns:

ID int

ING_ID varchar

ING_NAME varchar

ING_QTY varchar

LAST_ORDERED varchar

V_CODE varchar

EXP_DATE varchar

Result Grid

Filter Rows

Exports

Wrap Cell Contents

customer name	mode	count(mode)
Seri	Pick-up	9
Seri	Takein	12
Seri	Delivery	13

Result 45 x

Output

Query Completed

```

1 * alter table ingredient modify last_ordered date;
2
3 * select 'Order date',
4   sum('Quantity of items') from ordersdata
5   where 'Order date' = '2022-01-01 14:31:00';
6
7 * select 'MODE' from ordersdata
8   where 'Order Date' > '2022-01-01 17:00:00' and 'Order date' < '2022-01-01 19:00:00';
9
10 * select ing_name, ing_qty, last_ordered from ingredient
11   order by last_ordered;
12
13 * select 'customer name', 'mode',
14   count('mode') from ordersdata
15   where 'customer name' = 'Seri';
16   group by 'mode';
17
18 * select order_item from orders

```

- What did people who ordered steak get for drinks?

MySQL Workbench

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

Views

Stored Procedures

Functions

restaurant

Tables

employees

ingredient

job

menuitem

menuitem_ingredient

orders

ordersdata

reservations

review

vendors

Views

Stored Procedures

Functions

sys

Administration Schemas

Information

Table: ingredient

Columns:

ID int

ING_ID varchar

ING_NAME varchar

ING_QTY varchar

LAST_ORDERED varchar

V_CODE varchar

EXP_DATE varchar

Result Grid

Filter Rows

Exports

Wrap Cell Contents

order_item
12oz Steak,Soda
12oz Steak,Soda
12oz Steak,Lemonade

Result 46 x

Output

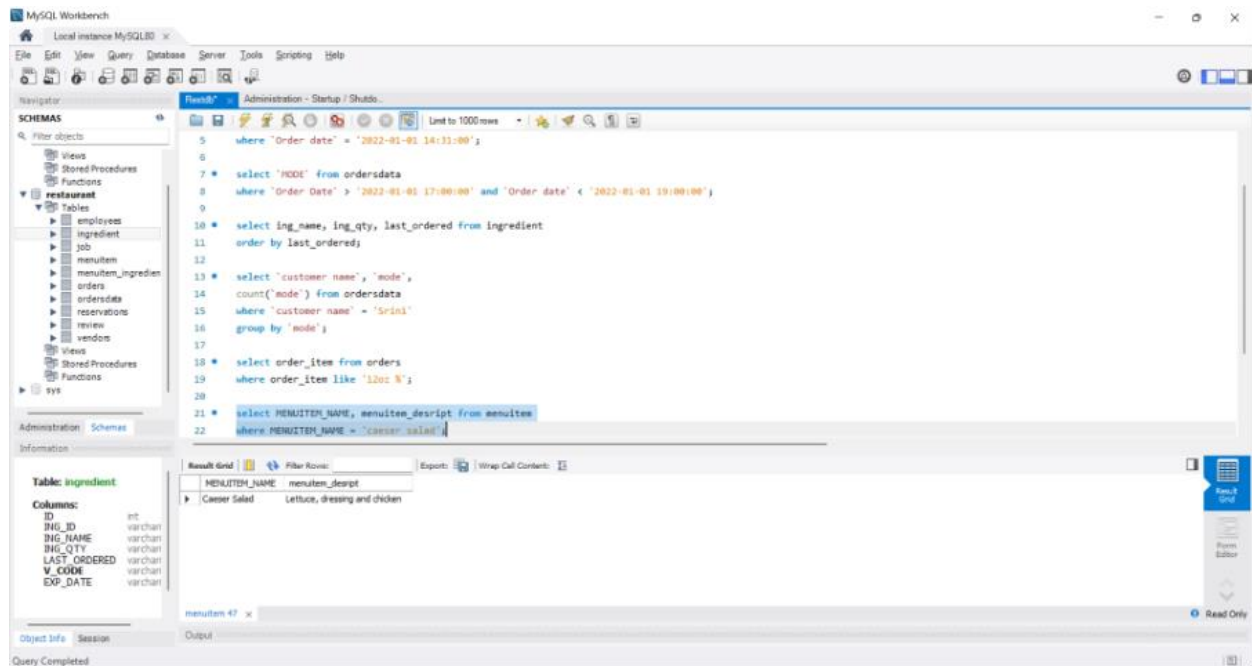
Query Completed

```

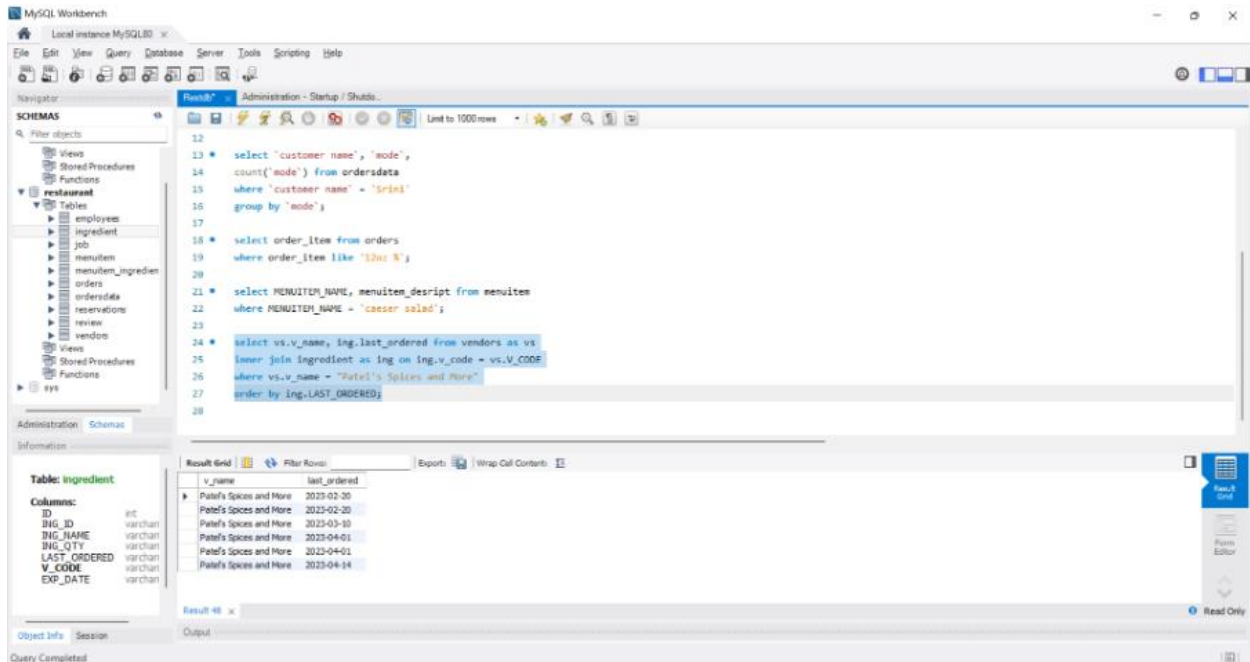
2
3 * select 'Order date',
4   sum('Quantity of items') from ordersdata
5   where 'Order date' = '2022-01-01 14:31:00';
6
7 * select 'MODE' from ordersdata
8   where 'Order Date' > '2022-01-01 17:00:00' and 'Order date' < '2022-01-01 19:00:00';
9
10 * select ing_name, ing_qty, last_ordered from ingredient
11   order by last_ordered;
12
13 * select 'customer name', 'mode',
14   count('mode') from ordersdata
15   where 'customer name' = 'Seri';
16   group by 'mode';
17
18 * select order_item from orders
19   where order_item like '12oz %';

```

- 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:

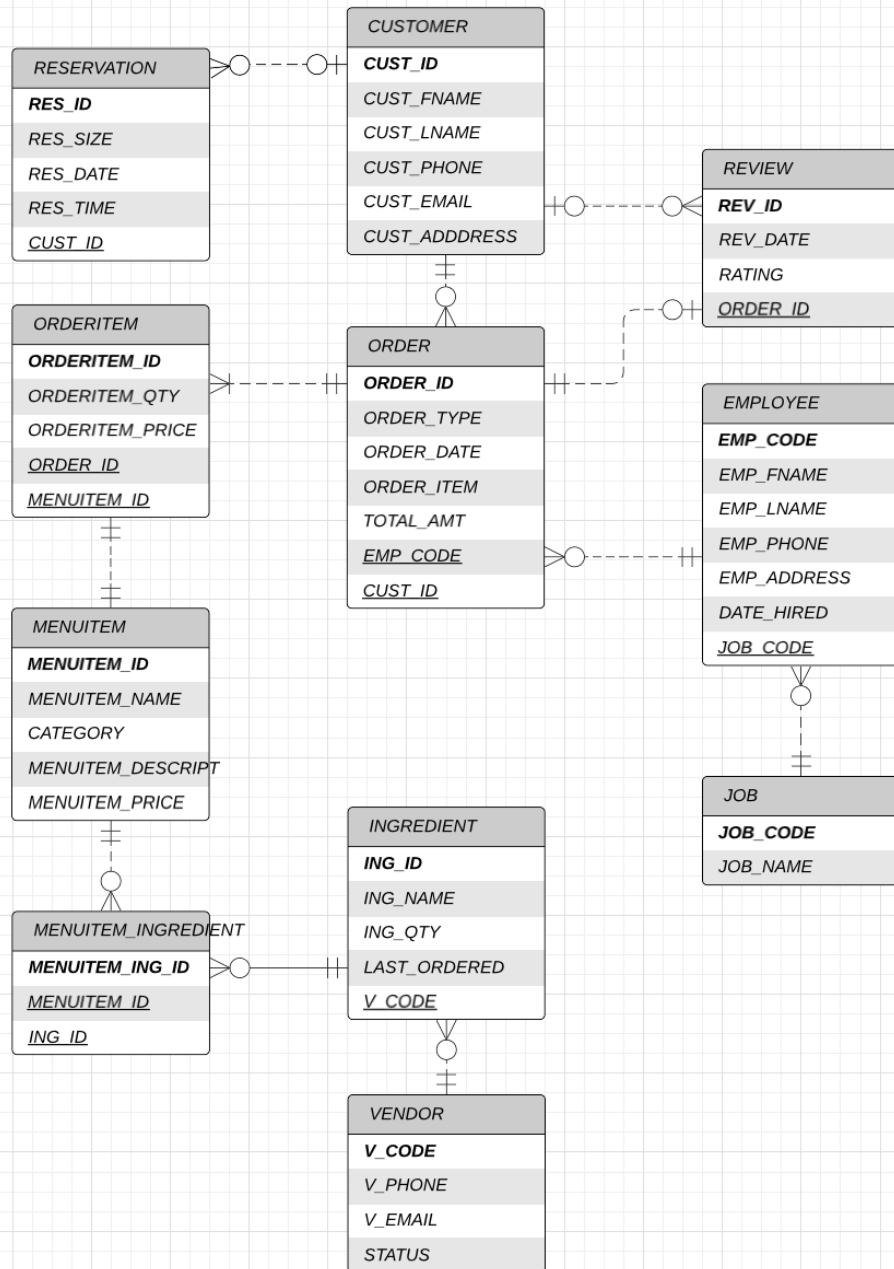
- a. Rev_ID must be unique for each review.
- b. Cust_ID must reference a valid customer.
- c. Order_ID must reference a valid order (dine-in, pick-up, or delivery).
- d. Rating must be within a predefined range (e.g., 1 to 5).
- e. Rev_Date must be a valid date, ideally set to the date when the review 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

Restaurant DBMS



Data Dictionary

	A	B	C	D	E	F	G	H	I
1	schema_nm	table_nm	obj_typ	ord	is_key	column_nm	data_type	nullable	column_descr
2	restaurant	employees	TBL		1	EMP_CODE	int(10)	NULL	
3	restaurant	employees	TBL		2	EMP_FNAME	varchar(255)	NULL	
4	restaurant	employees	TBL		3	EMP_LNAME	varchar(255)	NULL	
5	restaurant	employees	TBL		4	EMP_PHONE	varchar(255)	NULL	
6	restaurant	employees	TBL		5	EMP_ADDRESS	varchar(255)	NULL	
7	restaurant	employees	TBL		6	JOB_CODE	double(22)	NULL	
8	restaurant	employees	TBL		7	DATE_HIRED	varchar(40)	NULL	
9	restaurant	ingredient	TBL		1	ID	int(10)	NULL	
10	restaurant	ingredient	TBL		2	ING_ID	varchar(255)	NULL	
11	restaurant	ingredient	TBL		3	ING_NAME	varchar(255)	NULL	
12	restaurant	ingredient	TBL		4	ING_QTY	varchar(255)	NULL	
13	restaurant	ingredient	TBL		5	last_ordered	date(3)	NULL	
14	restaurant	ingredient	TBL		6 FK	V_CODE	varchar(255)	NULL	
15	restaurant	ingredient	TBL		7	EXP_DATE	varchar(255)	NULL	
16	restaurant	job	TBL		1	ID	int(10)	NULL	
17	restaurant	job	TBL		2	JOB_CODE	double(22)	NULL	
18	restaurant	job	TBL		3	JOB_NAME	varchar(255)	NULL	
19	restaurant	menuitem	TBL		1	MENUITEM_ID	varchar(255)	NULL	
20	restaurant	menuitem	TBL		2	MENUITEM_NAME	varchar(255)	NULL	
21	restaurant	menuitem	TBL		3	CATEGORY	varchar(255)	NULL	
22	restaurant	menuitem	TBL		4	MENUITEM_PRICE	varchar(255)	NULL	
23	restaurant	menuitem	TBL		5	MENUITEM_PRICE	double(22)	NULL	
24	restaurant	menuitem_in	TBL		1	ID	int(10)	NULL	
25	restaurant	menuitem_in	TBL		2	MENUITEM_PRICE	double(22)	NULL	
26	restaurant	menuitem_in	TBL		3	MENUITEM_PRICE	varchar(255)	NULL	
27	restaurant	menuitem_in	TBL		4	ING_ID	varchar(255)	NULL	
28	restaurant	menuitem_in	TBL		5	Field4	varchar(255)	NULL	
29	restaurant	orders	TBL		1	ORDER_ID	varchar(255)	NULL	
30	restaurant	orders	TBL		2	ORDER_TYPE	varchar(255)	NULL	
31	restaurant	orders	TBL		3	ORDER_DATE	varchar(255)	NULL	
32	restaurant	orders	TBL		4	ORDER_ITEM	varchar(255)	NULL	
33	restaurant	orders	TBL		5	TOTAL_AMT	varchar(255)	NULL	
34	restaurant	orders	TBL		6	PMT_METHO	varchar(255)	NULL	
35	restaurant	orders	TBL		7	EMP_CODE	varchar(255)	NULL	
36	restaurant	orders	TBL		8	CUST_ID	varchar(255)	NULL	
37	restaurant	ordersdata	TBL		1	ID	int(10)	NULL	
38	restaurant	ordersdata	TBL		2	Order ID	varchar(255)	NULL	
39	restaurant	ordersdata	TBL		3	Customer Name	varchar(255)	NULL	
40	restaurant	ordersdata	TBL		4	Order Date	varchar(40)	NULL	
41	restaurant	ordersdata	TBL		5	Quantity of It	double(22)	NULL	
42	restaurant	ordersdata	TBL		6	Order Amount	double(22)	NULL	
43	restaurant	ordersdata	TBL		7	Payment Mode	varchar(255)	NULL	
44	restaurant	ordersdata	TBL		8	MODE	varchar(255)	NULL	
45	restaurant	ordersdata	TBL		9	Delivery Time	double(22)	NULL	
46	restaurant	ordersdata	TBL		10	Customer Rating	double(22)	NULL	
47	restaurant	ordersdata	TBL		11	Customer Rating	double(22)	NULL	
48	restaurant	reservations	TBL		1	ID	int(10)	NULL	
49	restaurant	reservations	TBL		2	RES_ID	varchar(255)	NULL	
50	restaurant	reservations	TBL		3	RES_DATE	varchar(255)	NULL	
51	restaurant	reservations	TBL		4	RES_SIZE	varchar(255)	NULL	
52	restaurant	reservations	TBL		5	CUST_ID	varchar(255)	NULL	
53	restaurant	review	TBL		1	ID	int(10)	NULL	
54	restaurant	review	TBL		2	REV_ID	varchar(255)	NULL	
55	restaurant	review	TBL		3	REV_DATE	varchar(255)	NULL	
56	restaurant	review	TBL		4	RATING_FOO	double(22)	NULL	
57	restaurant	review	TBL		5	RATING_DEL	double(22)	NULL	
58	restaurant	review	TBL		6	ORDER_ID	varchar(255)	NULL	
59	restaurant	vendors	TBL		1	ID	int(10)	NULL	
60	restaurant	vendors	TBL		2 PK	V_CODE	varchar(255)	NOT NULL	
61	restaurant	vendors	TBL		3	V_NAME	varchar(255)	NULL	
62	restaurant	vendors	TBL		4	V_PHONE	varchar(255)	NULL	
63	restaurant	vendors	TBL		5	V_EMAIL	varchar(255)	NULL	
64	restaurant	vendors	TBL		6	STATUS	varchar(255)	NULL	
65									
66									

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->