NaplesLogic: Restaurant Management Tool

CS3520 X01 Database Theory

Group 3: Meaghan Barrett, Nat Parkinson, Kaden Kirkwood

Table of Contents

Executive Summary

NaplesLogic is a restaurant management system. It is created for handling the workflow within the restaurant industry. Designed with flexibility and clarity in mind, NaplesLogic organizes restaurant functions: such as: customer management, reservations, menu planning, ordering, staff scheduling, inventory, billing, and analytics into a dependable database system. Through automating and managing the functions mentioned, NaplesLogic reduces inefficiencies, improves customer experience, and enables restaurants to make data-driven decisions.

Vision and Objective

Vision:
Empower restaurants, regardless of size, to be able to operate more efficiently through accessible, computerized management software transforming operations into digital workflows.

Objective:
NaplesLogic aims to become the restaurant management tool for improving operational clarity and consistency. By modeling critical processes digitally, we provide structure and insight that allows restaurants to enhance quality of service, reduce waste, and increase profitability.

Mission Statement

Our mission is to simplify restaurant operations through a database system that empowers owners, managers, and staff to deliver exceptional customer experiences. NaplesLogic is built to handle the complexity of restaurant management by helping teams stay organized, informed, and responsive to customer needs.

Service

NaplesLogic is a restaurant management tool with real-world applications for understanding the complexity of what makes restaurants tick. All businesses have similar needs, and learning how to organize them correctly can lead to smoother operations. NaplesLogic is significant because it exists in a difficult industry, where budgets are tight and the number of people coming in and out of stores are numerous. NaplesLogic provides the leeway most restaurants need by structuring order information, staffing the restaurant and making sure that there is not an under or overstock of food supplies. This can save chefs and managers from the headache of constantly observing patterns during their busy shifts. The owner does not have to worry about spending too much. Now, the one who gets the most benefit from NaplesLogic is the customer. They will go to a restaurant that has enough food, fully staffed, and without an extremely long wait time.

Additionally, business owners can use the software to identify patterns and generate direct reports from the data.

Use Case

Use Case 1: Customer Makes a Reservation

Actor: Customer
Goal: Reserve a table at the restaurant.
Steps:

1. The customer logs into the restaurant website or app.
2. Selects the date, time, and party size.
3. The system checks available tables using the DinerTable table.
4. The system creates a record in the Reservation table, linking it to the Customer and selecting TableID.
5. A confirmation email is sent to the customer.

Use Case 2: Customer Places an Online Order

Actor: Customer
Goal: Order food for delivery or pickup.
Steps:

1. The customer browses the menu (using MenuItem table) and adds items to the cart.
2. The system creates a CustomerOrder and links it to the Customer.
3. The payment is processed and stored in the Payment table.
4. An OnlineOrder record is created with delivery type and address.
5. Order status is tracked and updated in real time.

Use Case 3: Manager Schedules Staff Shifts

Actor: Manager
Goal: Assign staff members to work shifts.
Steps:

1. Manager logs into the system and navigates to the ShiftSchedule table.
2. Inputs new shift details: date, start time, and end time.
3. Assigns staff members using the Staff table linked to the schedule.

4. Shift assignments are updated in the database and reflected in the staff dashboard, ensuring no scheduling conflicts .

Use Case 4: Inventory Auto Restock

Actor: System / Inventory Manager
Goal: Automatically reorder supplies when stock is low.
Steps:

1. The system checks inventory levels at scheduled intervals.
2. If QuantityInStock is less than ReorderThreshold, a SupplyOrder is created.
3. Each supply order is linked to the appropriate supplier.
4. The manager is notified of the pending delivery.
5. Upon delivery, Inventory levels are updated.

Business Requirements

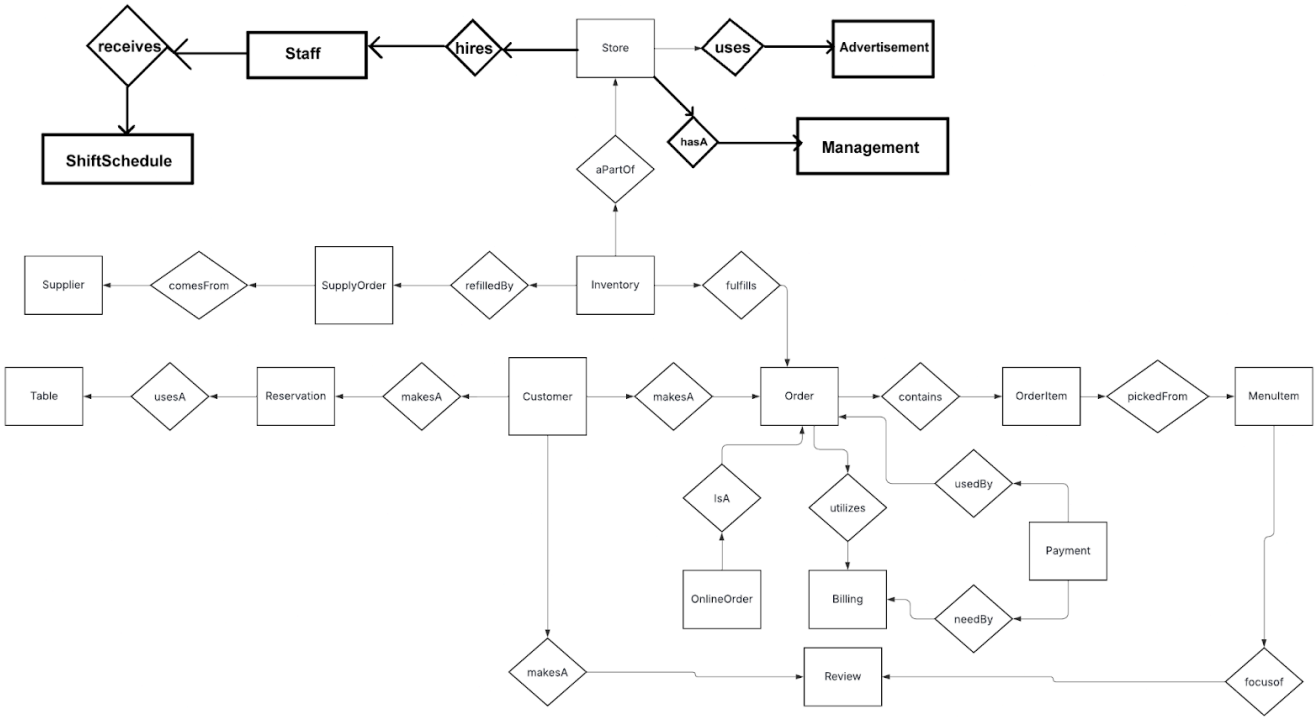
Here are some applicable business requirements NaplesLogic covers:

1. Customer Management
    a. Collects and stores customer information: name (first and last name), contact info (phone number and email), address and reservations.
    b. Maintain a customer's history of visits and online orders.
2. Menu and Order Management
    a. Manage the restaurant's menu (items, descriptions, categories, prices, and availability).
    b. Allow online order placement, with tracking details per customer.
3. Reservation System
    a. Enable customers the ability to book tables online or at the venue.
    b. Display real-time table id, table size and location.
4. Pricing and Billing
    a. Support flexible pricing.
    b. Generate and process bills, while accepting multiple payment methods (cash, card, paypal, etc.).
5. Staff and Shift Scheduling
    a. Profiles and roles of the store staff.
    b. Plan and manage work schedules for kitchen and floor staff.

6. Inventory and Supply Management
    a. Keep track of ingredients and stock levels.
    b. Automate the supply orders based on inventory.
    c. Track supplies deliveries and update inventory in real-time.
7. Reports and Analytics
    a. Generate detailed reports on sales and inventory usage.
    b. Provide managers with performance dashboards to demonstrate financial planning.

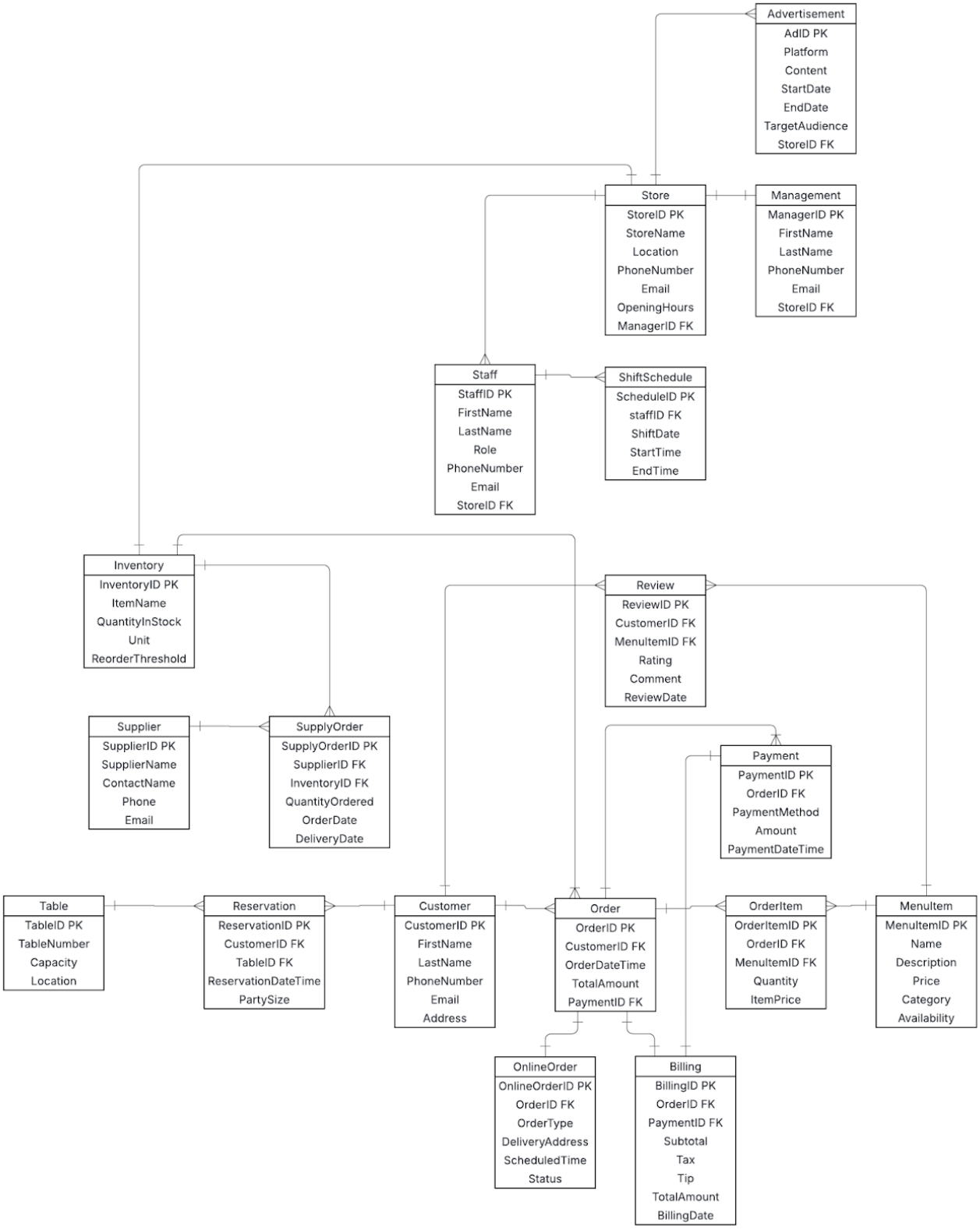
Conceptual Model Diagram

The tentative conceptual diagram of NaplesLogic shows how tables are interconnected and the nature of their relationship (one has an instance of another table, etc).



ER Diagram

The ER diagram is designed to show how tables are connected, similar to the Conceptual Model Diagram, but also to illustrate how the relationships work together (e.g., one-to-one). As for Primary Keys, that is how each table is primarily identified. The Foreign keys are optional and are used to show a connection between two tables. (For example, a connection between customer & reservation). Now, the cardinality of each column in a table is indicative of how unique that column is. All Primary keys are extremely unique, while foreign keys are less unique, and all other columns in the tables are variations. An example is that there is only 1 customer with Customer ID 1; however, customer #1 can have multiple reservations. Currently, there are no planned indexes; however, indexes could easily be added.


Advertisement
AdID PK
Platform
Content
StartDate
EndDate
TargetAudience
StoreID FK
Store
StoreID PK
StoreName
Location
PhoneNumber
Email
OpeningHours
ManagerID FK
Management
ManagerID PK
FirstName
LastName
PhoneNumber
Email
StoreID FK
Staff
StaffID PK
FirstName
LastName
Role
PhoneNumber
Email
StoreID FK
ShiftSchedule
ScheduleID PK
staffID FK
ShiftDate
StartTime
EndTime
Inventory
InventoryID PK
ItemName
QuantityInStock
Unit
ReorderThreshold
Review
ReviewID PK
CustomerID FK
MenuItemID FK
Rating
Comment
ReviewDate
Supplier
SupplierID PK
SupplierName
ContactName
Phone
Email
SupplyOrder
SupplyOrderID PK
SupplierID FK
InventoryID FK
QuantityOrdered
OrderDate
DeliveryDate
Payment
PaymentID PK
OrderID FK
PaymentMethod
Amount
PaymentDateTime
Table
TableID PK
TableNumber
Capacity
Location
Reservation
ReservationID PK
CustomerID FK
TableID FK
ReservationDateTime
PartySize
Customer
CustomerID PK
FirstName
LastName
PhoneNumber
Email
Address
Order
OrderID PK
CustomerID FK
OrderDateTime
TotalAmount
PaymentID FK
OrderItem
OrderItemID PK
OrderID FK
MenuItemID FK
Quantity
ItemPrice
MenuItem
MenuItemID PK
Name
Description
Price
Category
Availability
OnlineOrder
OnlineOrderID PK
OrderID FK
OrderType
DeliveryAddress
ScheduledTime
Status
Billing
BillingID PK
OrderID FK
PaymentID FK
Subtotal
Tax
Tip
TotalAmount
BillingDate

The Approach

Week 1: Finalize Project Scope & Database Planning

- Finalize the majority of the project scope and important features (reservations, menu, inventory, billing, etc.).
- Assign tasks to team members.
- Create an initial ER diagram to visualize how everything connects.
- Expand the ER diagram into a complete conceptual model.
- Design the relational schema, identifying all necessary tables and relationships.

Week 2: Build the Database

- Set up the database using a relational DBMS (such as: MySQL).
- Create core tables: customers, menu items, orders, staff, and inventory.
- Normalize the data to avoid redundancy and ensure consistency.
- Define all relationships using primary and foreign keys.
- Begin writing key transactional queries (such as: placing orders, billing logic).

Week 3: Add Features & Functionality

- Building system functionalities (such as: table tracking, staff scheduling, inventory management, online ordering.)
- Implement automated billing processes.
  Develop basic user interfaces or input forms for data entry.
- Start tying together different components of the system.
- Begin early testing of major functions to detect any issues early.

Week 4: Test, Polish & Wrap Up

- Conduct thorough testing (such as: edge cases and concurrent users).
  Validate data integrity and ACID compliance across transactions.
- Debug and fine-tune queries and processes for better performance.
- Finalize any reporting and analytics features.
- Compile the final project report for the presentation.

Current releases

The first release of the NaplesLogic proprietary software contains an SQL system capable of meeting the needs for both front-facing and back-facing elements of the restaurant industry. It allows a user to create Customers, Menu Items, Orders, Payment methods, Bills, Reservations, and Tables when dealing with front-facing elements such as customers and their orders.

However, NaplesLogic"s software also includes the logic necessary for management of Staff, Schedules, Inventory, Resuppliers, Management, Stores, and even Advertisement campaigns!

Future releases

The NaplesLogic Database software upcoming features in future releases include such improvements as: a new "Valet" table to be in a none-or-one to one relationship with Reservation, addition of a "To Go" data to the Order to show that the order is to go rather than sit down and eat, the addition of a "Benefits" table to form an optional many to one relationship with the Employees to show what Benefits an Employee has, along with the addition of a "ManagementID" to the Employee table, so you can find out who is in charge of who when the time comes. Also, an autoincrement and autoresupply feature will be added to save time to those who are constantly using the database.

Conclusion

NaplesLogic is more than just a database system. It is a comprehensive restaurant management tool designed to handle operations, improve restaurant efficiency, and further enhance the customer experience. By combining critical functions like reservations, orders, staff scheduling, and inventory into one platform, NaplesLogic provides restaurants with the structure needed to run smoothly in a demanding, fast-paced industry.

Through careful planning, designing, and an emphasis on real-world applications, this system enables restaurant owners and managers the ability to make informed decisions through the data gathered from their restaurant(s). Ultimately, NaplesLogic bridges the gap between restaurant complexity and digital simplicity, empowering restaurants to deliver consistent service, reduce waste, and boost profitability, while ensuring the best possible experience for every customer.

Appendix

Populated table samples

This is a sample of what the tables will look like when using SQL, they are populated by the data provided.

```
mysql> SELECT * FROM Advertisement;
+------+---------+-----------+-----------------------------------+------------+------------+------------------+
| AdID | StoreID | Platform  | Content                           | StartDate  | EndDate    | TargetAudience   |
+------+---------+-----------+-----------------------------------+------------+------------+------------------+
|    1 |       1 | Facebook  | Summer specials on pizza!         | 2025-07-01 | 2025-08-01 | Young adults     |
|    2 |       2 | Instagram | Fresh salads now available!       | 2025-07-05 | 2025-08-05 | Health conscious |
|    3 |       3 | Instagram | The best Italian in new york      | 2025-08-01 | 2025-09-01 | Italian Lovers   |
|    4 |       4 | Twitter   | The best Italian in new california | 2025-08-01 | 2025-09-01 | Italian Lovers   |
+------+---------+-----------+-----------------------------------+------------+------------+------------------+
4 rows in set (0.037 sec)

mysql> SELECT * FROM Billing;
+-----------+---------+-----------+----------+------+-------+-------------+-------------+
| BillingID | OrderID | PaymentID | Subtotal | Tax  | Tip   | TotalAmount | BillingDate |
+-----------+---------+-----------+----------+------+-------+-------------+-------------+
|         1 |       1 |         1 |    23.00 | 2.99 |  0.00 |       25.99 | 2025-07-05  |
|         2 |       2 |         2 |    11.00 | 1.50 |  0.00 |       12.50 | 2025-07-06  |
|         3 |       3 |         3 |     7.99 | 0.24 | 17.00 |       25.23 | 2025-08-02  |
|         4 |       4 |         4 |   155.88 | 4.68 | 89.69 |      250.25 | 2025-08-02  |
|         5 |       5 |         5 |    23.98 | 0.72 | 15.55 |       40.25 | 2025-08-02  |
|         6 |       6 |         6 |     7.99 | 0.24 | 11.00 |       19.23 | 2025-08-02  |
|         7 |       7 |         7 |    43.96 | 1.32 |  5.00 |       50.28 | 2025-08-02  |
|         8 |       8 |         8 |    56.97 | 1.71 | 17.00 |       75.68 | 2025-08-02  |
|         9 |       9 |         9 |    19.98 | 0.60 |  5.00 |       25.58 | 2025-08-02  |
|        10 |      10 |        10 |    10.99 | 0.33 |  4.61 |       15.93 | 2025-08-02  |
|        11 |      11 |        11 |   227.88 | 6.84 | 22.48 |      257.20 | 2025-08-02  |
|        12 |      12 |        12 |   143.91 | 4.32 | 11.77 |      160.00 | 2025-08-02  |
|        13 |      13 |        13 |    12.99 | 0.39 |  3.00 |       16.38 | 2025-08-02  |
|        14 |      14 |        14 |     7.99 | 0.24 |  2.02 |       10.26 | 2025-08-02  |
|        15 |      15 |        15 |    11.99 | 0.36 |  0.00 |       12.35 | 2025-08-02  |
|        16 |      16 |        16 |    39.97 | 1.20 |  9.09 |       12.35 | 2025-08-02  |
|        17 |      17 |        17 |    39.97 | 1.20 |  9.09 |       12.35 | 2025-08-02  |
+-----------+---------+-----------+----------+------+-------+-------------+-------------+
17 rows in set (0.011 sec)
```

SQL Scripts used

Data Types

```
mysql> SELECT
    ->     TABLE_NAME,
    ->     COLUMN_NAME,
    ->     DATA_TYPE,
    ->     CHARACTER_MAXIMUM_LENGTH,
    ->     NUMERIC_PRECISION,
    ->     NUMERIC_SCALE
    -> FROM
    ->     INFORMATION_SCHEMA.COLUMNS
    -> WHERE
    ->     TABLE_SCHEMA = 'napleslogic'
    -> ORDER BY
    ->     TABLE_NAME, ORDINAL_POSITION;
```

TABLE_NAME	COLUMN_NAME	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	NUMERIC_PRECISION	NUMERIC_SCALE
Advertisement	AdID	int	NULL	10	0
Advertisement	StoreID	int	NULL	10	0
Advertisement	Platform	varchar	30	NULL	NULL
Advertisement	Content	varchar	360	NULL	NULL
Advertisement	StartDate	date	NULL	NULL	NULL
Advertisement	EndDate	date	NULL	NULL	NULL
Advertisement	TargetAudience	varchar	50	NULL	NULL
Billing	BillingID	int	NULL	10	0
Billing	OrderID	int	NULL	10	0
Billing	PaymentID	int	NULL	10	0
Billing	Subtotal	decimal	NULL	10	2
Billing	Tax	decimal	NULL	6	2
Billing	Tip	decimal	NULL	6	2
Billing	TotalAmount	decimal	NULL	11	2
Billing	BillingDate	date	NULL	NULL	NULL
Customer	CustomerID	int	NULL	10	0
Customer	FirstName	varchar	30	NULL	NULL
Customer	LastName	varchar	30	NULL	NULL
Customer	PhoneNumber	char	10	NULL	NULL
Customer	email	varchar	50	NULL	NULL
Customer	address	varchar	50	NULL	NULL
CustomerOrder	OrderID	int	NULL	10	0
CustomerOrder	CustomerID	int	NULL	10	0
CustomerOrder	PaymentID	int	NULL	10	0
CustomerOrder	OrderDateTime	datetime	NULL	NULL	NULL
CustomerOrder	TotalAmount	decimal	NULL	10	2
DinerTable	TableID	int	NULL	10	0
DinerTable	TableNumber	tinyint	NULL	3	0
DinerTable	Capacity	tinyint	NULL	3	0

DinerTable	Capacity	tinyint	NULL	3	0
DinerTable	location	varchar	50	NULL	NULL
Inventory	InventoryID	int	NULL	10	0
Inventory	ItemName	varchar	50	NULL	NULL
Inventory	QuantityInStock	int	NULL	10	0
Inventory	Unit	smallint	NULL	5	0
Inventory	ReorderThreshold	varchar	30	NULL	NULL
Management	ManagerID	int	NULL	10	0
Management	FirstName	varchar	30	NULL	NULL
Management	LastName	varchar	30	NULL	NULL
Management	PhoneNumber	char	10	NULL	NULL
Management	Email	varchar	50	NULL	NULL
MenuItem	MenuItemID	tinyint	NULL	3	0
MenuItem	ItemName	varchar	20	NULL	NULL
MenuItem	ItemDescription	varchar	180	NULL	NULL
MenuItem	price	decimal	NULL	6	2
MenuItem	category	varchar	30	NULL	NULL
MenuItem	availability	tinyint	NULL	3	0
OnlineOrder	OnlineOrderID	int	NULL	10	0
OnlineOrder	OrderID	int	NULL	10	0
OnlineOrder	OrderType	varchar	10	NULL	NULL
OnlineOrder	DeliveryAddress	varchar	50	NULL	NULL
OnlineOrder	ScheduledTime	datetime	NULL	NULL	NULL
OnlineOrder	OrderStatus	varchar	10	NULL	NULL
OrderItem	OrderItemID	int	NULL	10	0
OrderItem	OrderID	int	NULL	10	0
OrderItem	MenuItemID	tinyint	NULL	3	0
OrderItem	Quantity	tinyint	NULL	3	0
OrderItem	ItemPrice	decimal	NULL	6	2
Payment	PaymentID	int	NULL	10	0
Payment	PaymentMethod	varchar	10	NULL	NULL
Payment	Amount	decimal	NULL	10	2
Payment	PaymentDateTime	datetime	NULL	NULL	NULL
Reservation	ReservationID	int	NULL	10	0
Reservation	CustomerID	int	NULL	10	0
Reservation	TableID	int	NULL	10	0
Reservation	ReservationDateTime	datetime	NULL	NULL	NULL
Reservation	PartySize	tinyint	NULL	3	0
Review	ReviewID	tinyint	NULL	3	0
Review	CustomerID	int	NULL	10	0
Review	MenuItemID	tinyint	NULL	3	0
Review	Rating	decimal	NULL	2	1
Review	ReviewComment	varchar	180	NULL	NULL
Review	ReviewDate	date	NULL	NULL	NULL
ShiftSchedule	ScheduleID	int	NULL	10	0
ShiftSchedule	ShiftDate	date	NULL	NULL	NULL
ShiftSchedule	StartTime	time	NULL	NULL	NULL
ShiftSchedule	EndTime	time	NULL	NULL	NULL
Staff	StaffID	int	NULL	10	0
Staff	ScheduleID	int	NULL	10	0
Staff	StoreID	int	NULL	10	0
Staff	FirstName	varchar	30	NULL	NULL
Staff	LastName	varchar	30	NULL	NULL

```
| Staff         | FirstName          | varchar   |                  30 |            NULL |          NULL |
| Staff         | LastName           | varchar   |                  30 |            NULL |          NULL |
| Staff         | EmployeeRole       | varchar   |                  20 |            NULL |          NULL |
| Staff         | PhoneNumber        | char      |                  10 |            NULL |          NULL |
| Staff         | Email              | varchar   |                  50 |            NULL |          NULL |
| Store         | StoreID            | int       |                NULL |              10 |             0 |
| Store         | ManagerID          | int       |                NULL |              10 |             0 |
| Store         | StoreName          | varchar   |                  50 |            NULL |          NULL |
| Store         | Location           | varchar   |                  50 |            NULL |          NULL |
| Store         | PhoneNumber        | char      |                  10 |            NULL |          NULL |
| Store         | Email              | varchar   |                  50 |            NULL |          NULL |
| Store         | OpeningHours       | time      |                NULL |            NULL |          NULL |
| Supplier      | SupplierID         | int       |                NULL |              10 |             0 |
| Supplier      | SupplierName       | varchar   |                  30 |            NULL |          NULL |
| Supplier      | Contactname        | varchar   |                  30 |            NULL |          NULL |
| Supplier      | Phone              | char      |                  10 |            NULL |          NULL |
| Supplier      | Email              | varchar   |                  50 |            NULL |          NULL |
| SupplyOrder   | SupplyOrderID      | int       |                NULL |              10 |             0 |
| SupplyOrder   | SupplierID         | int       |                NULL |              10 |             0 |
| SupplyOrder   | InventoryID        | int       |                NULL |              10 |             0 |
| SupplyOrder   | QuantityOrdered    | smallint  |                NULL |               5 |             0 |
| SupplyOrder   | OrderDate          | date      |                NULL |            NULL |          NULL |
| SupplyOrder   | DeliveryDate       | date      |                NULL |            NULL |          NULL |
+---------------+--------------------+-----------+---------------------+-----------------+---------------+
102 rows in set (0.045 sec)

mysql> SHOW TABLES;
+-----------------------+
| Tables_in_napleslogic |
+-----------------------+
| Advertisement         |
| Billing               |
| Customer              |
| CustomerOrder         |
| DinerTable            |
| Inventory             |
| Management            |
| MenuItem              |
| OnlineOrder           |
| OrderItem             |
| Payment               |
| Reservation           |
| Review                |
| ShiftSchedule         |
| Staff                 |
| Store                 |
| Supplier              |
| SupplyOrder           |
+-----------------------+
18 rows in set (0.009 sec)
```

-- <u>Tables Created</u>

```
-- Drop database if exists
DROP DATABASE IF EXISTS NaplesLogic;


-- Create database
CREATE DATABASE NaplesLogic;


-- Use the new database
USE NaplesLogic;


-- Create tables --
-- Customer TABLE
```

```
CREATE TABLE Customer (
    CustomerID INT UNSIGNED PRIMARY KEY,
    FirstName VARCHAR(30),
    LastName VARCHAR(30),
    PhoneNumber CHAR(10),
    email VARCHAR(50),
    address VARCHAR(50)
);

-- DinerTable TABLE
CREATE TABLE DinerTable (
    TableID INT UNSIGNED PRIMARY KEY,
    TableNumber TINYINT UNSIGNED,
    Capacity TINYINT UNSIGNED,
    location VARCHAR(50)
);

-- Reservation TABLE
CREATE TABLE Reservation(
    ReservationID INT UNSIGNED PRIMARY KEY,
    CustomerID INT UNSIGNED NOT NULL,
    TableID INT UNSIGNED NOT NULL,
    ReservationDateTime DATETIME,
    PartySize TINYINT UNSIGNED,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (TableID) REFERENCES DinerTable(TableID)
);

-- MenuItem TABLE
CREATE TABLE MenuItem (
    MenuItemID TINYINT UNSIGNED PRIMARY KEY,
```

```
    ItemName VARCHAR(20) NOT NULL,
    ItemDescription VARCHAR(180),
    price DECIMAL(6, 2) NOT NULL,
    category VARCHAR(30) NOT NULL,
    availability BOOLEAN NOT NULL
);

-- Review TABLE
CREATE TABLE Review (
    ReviewID TINYINT UNSIGNED PRIMARY KEY,
    CustomerID INT UNSIGNED NOT NULL,
    MenuItemID TINYINT UNSIGNED NOT NULL,
    Rating DECIMAL(2,1) NOT NULL,
    ReviewComment VARCHAR(180),
    ReviewDate DATE NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (MenuItemID) REFERENCES MenuItem(MenuItemID)
);

-- Payment TABLE
CREATE TABLE Payment (
    PaymentID INT UNSIGNED PRIMARY KEY,
    PaymentMethod VARCHAR(10) NOT NULL,
    Amount DECIMAL(10, 2) NOT NULL,
    PaymentDateTime DATETIME NOT NULL
);

-- CustomerOrder TABLE
CREATE TABLE CustomerOrder (
    OrderID INT UNSIGNED PRIMARY KEY,
    CustomerID INT UNSIGNED NOT NULL,
```

```
    PaymentID INT UNSIGNED NOT NULL,
    OrderDateTime DATETIME NOT NULL,
    TotalAmount DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID),
    FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)
);

-- OnlineOrder TABLE
CREATE TABLE OnlineOrder(
    OnlineOrderID INT UNSIGNED PRIMARY KEY,
    OrderID INT UNSIGNED NOT NULL,
    OrderType VARCHAR(10) NOT NULL,
    DeliveryAddress VARCHAR(50) NOT NULL,
    ScheduledTime DATETIME,
    OrderStatus VARCHAR(10),
    FOREIGN KEY (OrderID) REFERENCES CustomerOrder(OrderID)
);

-- Billing TABLE
CREATE TABLE Billing (
    BillingID INT UNSIGNED PRIMARY KEY,
    OrderID INT UNSIGNED NOT NULL,
    PaymentID INT UNSIGNED NOT NULL,
    Subtotal DECIMAL(10, 2) NOT NULL,
    Tax DECIMAL(6, 2) NOT NULL,
    Tip DECIMAL(6, 2),
    TotalAmount DECIMAL(11, 2) NOT NULL,
    BillingDate DATE NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES CustomerOrder(OrderID),
    FOREIGN KEY (PaymentID) REFERENCES Payment(PaymentID)
);
```

```
-- OrderItem TABLE
CREATE TABLE OrderItem (
    OrderItemID INT UNSIGNED PRIMARY KEY,
    OrderID INT UNSIGNED NOT NULL,
    MenuItemID TINYINT UNSIGNED,
    Quantity TINYINT UNSIGNED NOT NULL,
    ItemPrice DECIMAL(6, 2) NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES CustomerOrder(OrderID),
    FOREIGN KEY (MenuItemID) REFERENCES MenuItem(MenuItemID)
);

-- Supplier TABLE
CREATE TABLE Supplier(
    SupplierID INT UNSIGNED PRIMARY KEY,
    SupplierName VARCHAR(30),
    Contactname VARCHAR(30),
    Phone CHAR(10),
    Email VARCHAR(50)
);

-- Inventory TABLE
CREATE TABLE Inventory(
    InventoryID INT UNSIGNED PRIMARY KEY,
    ItemName VARCHAR(50),
    QuantityInStock INT UNSIGNED,
    Unit SMALLINT UNSIGNED,
    ReorderThreshold VARCHAR(30)
);

-- SupplyOrder TABLE
```

```
CREATE TABLE SupplyOrder(
    SupplyOrderID INT UNSIGNED PRIMARY KEY,
    SupplierID INT UNSIGNED NOT NULL,
    InventoryID INT UNSIGNED NOT NULL,
    QuantityOrdered SMALLINT UNSIGNED NOT NULL,
    OrderDate DATE,
    DeliveryDate DATE,
    FOREIGN KEY (SupplierID) REFERENCES Supplier(SupplierID),
    FOREIGN KEY (InventoryID) REFERENCES Inventory(InventoryID)
);

-- ShiftSchedule TABLE
CREATE TABLE ShiftSchedule(
    ScheduleID INT UNSIGNED PRIMARY KEY,
    ShiftDate DATE NOT NULL,
    StartTime TIME NOT NULL,
    EndTime TIME
);

-- Management TABLE
CREATE TABLE Management(
    ManagerID INT UNSIGNED PRIMARY KEY,
    FirstName VARCHAR(30),
    LastName VARCHAR(30),
    PhoneNumber CHAR(10),
    Email VARCHAR(50)
);

-- Store TABLE
CREATE TABLE Store(
    StoreID INT UNSIGNED PRIMARY KEY,
```

```
    ManagerID INT UNSIGNED NOT NULL,
    StoreName VARCHAR(50),
    Location VARCHAR(50),
    PhoneNumber CHAR(10),
    Email VARCHAR(50),
    OpeningHours TIME,
    FOREIGN KEY (ManagerID) REFERENCES Management(ManagerID)
);

-- Staff TABLE
CREATE TABLE Staff(
    StaffID INT UNSIGNED PRIMARY KEY,
    ScheduleID INT UNSIGNED NOT NULL,
    StoreID INT UNSIGNED NOT NULL,
    FirstName VARCHAR(30),
    LastName VARCHAR(30),
    EmployeeRole VARCHAR(20),
    PhoneNumber CHAR(10),
    Email VARCHAR(50),
    FOREIGN KEY (ScheduleID) REFERENCES ShiftSchedule(ScheduleID),
    FOREIGN KEY (StoreID) REFERENCES Store(StoreID)
);

-- Advertisement TABLE
CREATE TABLE Advertisement(
    AdID INT UNSIGNED PRIMARY KEY,
    StoreID INT UNSIGNED NOT NULL,
    Platform VARCHAR(30),
    Content VARCHAR(360),
    StartDate DATE NOT NULL,
    EndDate DATE NOT NULL,
```

```
    TargetAudience VARCHAR(50),
    FOREIGN KEY (StoreID) REFERENCES Store(StoreID)
);
```

-- <u>Data population</u> (Since Every Restaurant is Different So Are All The Values)

```
-- Insert examples for each table --
-- INSERT VALUES INTO Customer
INSERT INTO Customer VALUES
(1, 'John', 'Deer', '1234567890', 'john@example.com', '123 Elm Street'),
(2, 'Jane', 'Smiths', '0987654321', 'jane@example.com', '456 Oak Avenue');

-- INSERT VALUES INTO DinerTable
INSERT INTO DinerTable VALUES
(1, 10, 4, 'Window side'),
(2, 11, 6, 'Center Hall');

-- INSERT VALUES INTO Reservation
INSERT INTO Reservation VALUES
(1, 1, 1, '2025-08-01 19:00:00', 3),
(2, 2, 2, '2025-08-02 20:00:00', 5);

-- INSERT VAlUES INTO MenuItem
INSERT INTO MenuItem VALUES
(1, 'Margherita Pizza', 'Classic cheese and tomato pizza', 9.99, 'Pizza', TRUE),
(2, 'Caesar Salad', 'Romaine lettuce with Caesar dressing', 6.50, 'Salad', TRUE);

-- INSERT VAlUES INTO Review
INSERT INTO Review VALUES
(1, 1, 1, 4.5, 'Delicious and Fresh!', '2025-07-01'),
(2, 2, 2, 3.8, 'Good but could be fresher', '2025-07-02');

-- INSERT VALUES INTO Payment
INSERT INTO Payment VALUES
(1, 'Credit', 25.99, '2025-07-05 13:30:00'),
(2, 'Cash', 12.50, '2025-07-06 14:00:00');

-- INSERT VALUES INTO CustomerOrder
INSERT INTO CustomerOrder VALUES
(1, 1, 1, '2025-07-05 13:25:00', 25.99),
```

```
(2, 2, 2, '2025-07-06 13:55:00', 12.50);

-- INSERT VALUES INTO OnlineOrder
INSERT INTO OnlineOrder VALUES
(1, 1, 'Delivery', '789 Pine Street', '2025-07-05 14:00:00', 'Pending'),
(2, 2, 'Pickup', 'N/A', NULL, 'Completed');

-- INSERT VALUES INTO Billing
INSERT INTO Billing VALUES
(1, 1, 1, 23.00, 2.99, 0.00, 25.99, '2025-07-05'),
(2, 2, 2, 11.00, 1.50, 0.00, 12.50, '2025-07-06');

-- INSERT VALUES INTO OrderItem
INSERT INTO OrderItem VALUES
(1, 1, 1, 2, 9.99),
(2, 2, 2, 1, 6.50);

-- INSERT VALUES INTO Supplier
INSERT INTO Supplier VALUES
(1, 'Fresh Farms', 'Alice Greens', '5551234567', 'alice@freshfarms.com'),
(2, 'Ocean Supplies', 'Bobbie Blues', '5559876543', 'bobbie@oceansupplies.com');

-- INSERT VALUES INTO Inventory
INSERT INTO Inventory VALUES
(1, 'Tomatoes', 100, 1, '50'),
(2, 'Lettuce', 50, 1, '20');

-- INSERT VAlUES INTO SupplyOrder
INSERT INTO SupplyOrder VALUES
(1, 1, 1, 30, '2025-07-01', '2025-07-03'),
(2, 2, 2, 20, '2025-07-02', '2025-07-04');

-- INSERT VALUES INTO ShiftSchedule
INSERT INTO ShiftSchedule VALUES
(1, '2025-07-10', '09:00:00', '17:00:00'),
(2, '2025-07-11', '13:00:00', '21:00:00');

-- INSERT VAlUES INTO Management
INSERT INTO Management VALUES
(1, 'Mary', 'Jones', '5551112222', 'mary.j@example.com'),
```

```
(2, 'Steven', 'Willis', '5553334444', 'steven.w@example.com');

-- INSERT VALUES INTO Store
INSERT INTO Store VALUES
(1, 1, 'Downtown Naples', 'Downtown', '5552223333', 'downtown@naples.com',
'09:00:00'),
(2, 2, 'Uptown Naples', 'Uptown', '5554445555', 'uptown@naples.com', '10:00:00');

-- INSERT VALUES INTO Staff
INSERT INTO Staff VALUES
(1, 1, 1, 'Alice', 'Brown', 'Chef', '5556667777', 'alice.brown@example.com'),
(2, 2, 2, 'Billy', 'White', 'Waiter', '5558889999', 'billy.white@example.com');

-- INSERT VALUES INTO Advertisement
INSERT INTO Advertisement VALUES
(1, 1, 'Facebook', 'Summer specials on pizza!', '2025-07-01', '2025-08-01', 'Young adults'),
(2, 2, 'Instagram', 'Fresh salads now available!', '2025-07-05', '2025-08-05', 'Health
conscious');

INSERT INTO customer VALUES
(3, "Loki", "Laufeyson", "801892701", "trickery@gmail.com", "1234 Asgardian ST"),
(4, "Thor", "Odinson", "801982701", "thunder@gmail.com", "1234 Asgardian ST"),
(5, "Peter", "Parker", "801892711", "friendlyNeighbor@gmail.com", "1234 Wilson
Ave"),
(6, "Bruce", "Banner", "801892721", "rage@gmail.com", "1234 Green Door"),
(7, "Bruce", "Wayne", "801892731", "money@gmail.com", "1234 Gotham"),
(8, "Balfog", "Stormbringer", "801893701", "frost@gmail.com", "1234 Plaguelands"),
(9, "Sun", "Mistnight", "801894701", "celestial@juno.com", "1234 Equestria"),
(10, "Abra", "Cadabra", "801895701", "magic@juno.com", "1234 Kanto ST"),
(11, "Ugradsh", "Dragonson", "801992701", "chaos@gmail.com", "1234 Beelzebub
Ave"),
(12, "Kelly", "Churchbearer", "801192701", "sheep@gmail.com", "1234 Not Britan ST"),
(13, "Heather", "Henderson", "801292701", "tired@gmail.com", "1234 Artist St"),
(14, "Andy", "Adderson", "801812701", "inventive@gmail.com", "1234 Toy ST"),
(15, "Jim", "Cartborn", "801822701", "argumentive@gmail.com", "1234 Televsion ST");

INSERT INTO DinerTable VALUES
(3, 1, 4, "section 1 first window"),
(4, 2, 12, "party table 1"),
(5, 3, 4, "section 1 second window"),
```

```
(6, 4, 4, "section 2 first window"),
(7, 5, 4, "section 1 corner booth"),
(8, 6, 4, "section 2 central table"),
(9, 7, 4, "section 3 first window"),
(10, 8, 4, "section 4 first window"),
(11, 9, 4, "section 5 first window"),
(12, 12, 12, "party table 2"),
(13, 13, 12, "party table 3"),
(14, 14, 4, "section 5 second window"),
(15, 15, 6, "section 5 corner booth");

INSERT INTO Reservation VALUES
(3, 3, 14, "2025-08-02 21:00:00", 1),
(4, 4, 4, "2025-08-02 21:03:00", 12),
(5, 5, 15, "2025-08-02 21:07:13", 2),
(6, 6, 5, "2025-08-02 21:09:24", 1),
(7, 7, 6, "2025-08-02 21:00:00", 4),
(8, 8, 7, "2025-08-02 21:10:00", 3),
(9, 9, 8, "2025-08-02 21:11:34", 2),
(10, 10, 9, "2025-08-02 21:12:32", 1),
(11, 11, 10, "2025-08-02 21:13:44", 12),
(12, 12, 11, "2025-08-02 21:16:46", 9),
(13, 13, 12, "2025-08-02 21:17:00", 1),
(14, 14, 13, "2025-08-02 21:19:02", 1),
(15, 15, 3, "2025-08-02 21:51:30", 1);

INSERT INTO MenuItem VALUES
(3, "penne alla vodka", "A very delicious italian dish made using fresh pasta and a tomato
based sauce.", 7.99, "pasta", True),
(4, "shrimp linguini", "A simple dish made using a linguini noodles and shrimp, all
covered in a white sauce", 12.99, "pasta", True),
(5, "chicken alfredo", "A classic that is sure to please", 10.99, "pasta", False),
(6, "zuppa soup", "A unique soup that many will", 11.99, "soup", False),
(7, "chicken noodle soup", "A dish made to warm the soul", 7.99, "soup", True),
(8, "beef stew", "high quality beef slow cooked in the holy trinity of carrots, celery, and
onions", 15.99, "stew", True),
(9, "away from home", "pick one pasta dish and one soup with your choice between
breadstick and a salad", 18.99, "combo", True),
(10, "chicken tenders", "Handbreaded breast meat which is double fried for the ultimate
crunch", 9.99, "entree", True),
```

(11, "spaghetti&meatballs", "made with high quality spaghetti and ground in house beef.", 10.99, "pasta", True);

INSERT INTO Review VALUES
(3, 3, 3, 4.5, "authentic penne alla vodka", "2025-08-02"),
(4, 4, 4, 2, "shrimp was not well prepared", "2025-08-02"),
(5, 5, 6, 3, "Unique flavor but not my style", "2025-08-02"),
(6, 6, 7, 5.0, "really did warm the soul", "2025-08-02"),
(7, 7, 5, 3, "it was okay", "2025-08-03"),
(8, 8, 9, 4.5, "I got the shrimp linguini, zupa soup, and breadsticks, and each one of them was great.", "2025-08-03"),
(9, 9, 10, 3.5, "the chicken tenders were a bit dry but the crunch was incredible", "2025-08-03"),
(10, 10, 11, 4.0, "simple dish but tasted great", "2025-08-03"),
(11, 11, 9, 4.0, "Fantastic deal with the away from home meal", "2025-08-04"),
(12, 12, 8, 4.0, "pricey but the flavor was very rich", "2025-08-04"),
(13, 13, 4, 3.0, "the shrimp linguini was just okay but the service was great", "2025-08-04"),
(14, 14, 3, 1.0, "got food poisoning after eating here", "2025-08-07"),
(15, 15, 6, 2, "not very good", "2025-08-07");

INSERT INTO Payment VALUES
(3, "cash", 25.23, "2025-08-02 22:00:00"),
(4, "credit", 250.25, "2025-08-02 22:03:00"),
(5, "check", 40.25, "2025-08-02 22:07:13"),
(6, "credit", 19.23, "2025-08-02 22:09:24"),
(7, "cash", 50.28, "2025-08-02 22:00:00"),
(8, "check", 75.68, "2025-08-02 22:10:00"),
(9, "cash", 25.58, "2025-08-02 22:11:34"),
(10, "credit", 15.93, "2025-08-02 22:12:32"),
(11, "credit", 257.20, "2025-08-02 22:13:44"),
(12, "check", 160.00, "2025-08-02 22:16:46"),
(13, "cash", 16.38, "2025-08-02 22:17:00"),
(14, "check", 10.26, "2025-08-02 22:19:02"),
(15, "cash", 12.35, "2025-08-02 22:51:30"),
(16, "credit", 50.26, "2025-08-12 11:19:02"),
(17, "credit", 50.26, "2025-08-12 11:21:30");

INSERT INTO CustomerOrder VALUES
(3, 3, 3, "2025-08-02 21:10:00", 25.23),

```
(4, 4, 4, "2025-08-02 21:13:00", 250.25),
(5, 5, 5, "2025-08-02 21:17:13", 40.25),
(6, 6, 6, "2025-08-02 21:19:24", 19.23),
(7, 7, 7, "2025-08-02 21:10:00", 50.28),
(8, 8, 8, "2025-08-02 21:20:00", 75.68),
(9, 9, 9, "2025-08-02 21:21:34", 25.58),
(10, 10, 10, "2025-08-02 21:22:32", 15.93),
(11, 11, 11, "2025-08-02 21:23:44", 28.54),
(12, 12, 12, "2025-08-02 21:26:46", 24.78),
(13, 13, 13, "2025-08-02 21:17:00", 12.23),
(14, 14, 14, "2025-08-02 21:19:02", 10.26),
(15, 15, 15, "2025-08-02 21:51:30", 12.23),
(16, 3, 16, "2025-08-02 21:51:30", 50.26),
(17, 4, 17, "2025-08-02 21:51:30", 50.26);

INSERT INTO OnlineOrder VALUES
(3, 16, "pickup", "1234 Asgardian ST", "2025-08-12 12:30:00", "pending"),
(4, 17, "delivery", "1234 Asgardian ST", "2025-08-12 12:30:00", "canceled");

INSERT INTO Billing VALUES
(3,3,3, 7.99, 0.24, 17.00, 25.23, "2025-08-02"),
(4,4,4, 155.88, 4.68, 89.69, 250.25, "2025-08-02"),
(5,5,5, 23.98, 0.72, 15.55, 40.25, "2025-08-02"),
(6,6,6, 7.99, 0.24, 11.00, 19.23, "2025-08-02"),
(7,7,7, 43.96, 1.32, 5.00, 50.28, "2025-08-02"),
(8,8,8, 56.97, 1.71, 17.00, 75.68, "2025-08-02"),
(9,9,9, 19.98, 0.60, 5.00, 25.58, "2025-08-02"),
(10,10,10, 10.99, 0.33, 4.61, 15.93, "2025-08-02"),
(11,11,11, 227.88, 6.84, 22.48, 257.20, "2025-08-02"),
(12,12,12, 143.91, 4.32, 11.77, 160.00, "2025-08-02"),
(13,13,13, 12.99, 0.39, 3.00, 16.38, "2025-08-02"),
(14,14,14, 7.99, 0.24, 2.02, 10.26, "2025-08-02"),
(15,15,15, 11.99, 0.36, 0.00, 12.35, "2025-08-02"),
(16,16,16, 39.97, 1.20, 9.09, 12.35, "2025-08-02"),
(17,17,17, 39.97, 1.20, 9.09, 12.35, "2025-08-02");

INSERT INTO OrderItem VALUES
(3, 3, 3, 1, 7.99),
(4, 4, 4, 12, 12.99),
(5, 5, 6, 2, 11.99),
```

```
(6, 6, 7, 1, 7.99),
(7, 7, 5, 4, 10.99),
(8, 8, 9, 3, 18.99),
(9, 9, 10, 2, 9.99),
(10, 10, 11, 1, 10.99),
(11, 11, 9, 12, 18.99),
(12, 12, 8, 9, 15.99),
(13, 13, 4, 1, 12.99),
(14, 14, 3, 1, 7.99),
(15, 15, 6, 1, 11.99),
(16, 16, 8, 2, 15.99),
(17, 16, 3, 1, 7.99),
(18, 17, 8, 2, 15.99),
(19, 17, 3, 1, 7.99);

INSERT INTO Supplier VALUES
(3, "Tonys Delivery", "Tony Stark", "8012836437", "heavyweapons@gmail.com"),
(4, "Joe Biten", "Sleepy Joe", "8017773782", "mintchoclatechip@gmail.com");

INSERT INTO Inventory VALUES
(3, "linguini", 75, 1, "50"),
(4, "top round", 50, 1, "15"),
(5, "pepper", 200, 1, "100"),
(6, "carrots", 200, 1, "120"),
(7, "onions", 200, 1, "120"),
(8, "celery", 200, 1, "120"),
(9, "chicken breast", 40, 1, "75"),
(10, "garlic", 150, 1, "60"),
(11, "shrimp", 120, 1, "40"),
(12, "italian sausage", 32, 1, "60");

INSERT INTO SupplyOrder VALUES
(3, 3, 9, 100, "2025-8-28", "2025-9-1"),
(4, 4, 12, 100, "2025-8-28", "2025-9-1");

INSERT INTO ShiftSchedule VALUES
(3, "2025-8-02", "9:00:00", "13:00:00"),
(4, "2025-8-02", "13:00:00", "20:00:00"),
(5, "2025-8-02", "8:00:00", "13:00:00"),
(6, "2025-8-02", "13:00:00", "20:00:00");
```

```
INSERT INTO Management VALUES
(3, "Kurt", "Lenon", "801778901", "artist@bang.com"),
(4, "John", "cobain", "8016414377", "bucky@juno.com");

INSERT INTO Store VALUES
(3, 3, "New Yorks Napals", "New York 54 Street", "8011112222",
"NewyorkNapals@gmail.com", "9:00:00"),
(4, 4, "California Napals", "Route 58", "8011112223", "CaliforniaNapals@gmail.com",
"9:00:00");

INSERT INTO Staff VALUES
(3, 3, 3, "Luigi", "Mario", "Bartender", "8013864833", "Innovator@gmail.com"),
(4, 3, 3, "Ludwig", "Herber", "Bus Boy", "8013864834", "Noliscence@gmail.com"),
(5, 3, 3, "John", "Server", "Waiter", "8013864835", "Server@gmail.com"),
(6, 3, 3, "Nota", "Molerat", "Waiter", "8013864836", "RealPerson@gmail.com"),
(7, 3, 3, "Carlos", "Gartner", "chef", "8013864837", "Gartner@gmail.com"),
(8, 4, 3, "Mario", "Luigi", "Bartender", "8013864838", "NotInnovative@gmail.com"),
(9, 4, 3, "Herber", "Ludwig", "Bus Boy", "8013864839", "Yesliscence@gmail.com"),
(10, 4, 3, "Server", "John", "Waiter", "8013864840", "NotServer@gmail.com"),
(11, 4, 3, "Isa", "Molerat", "Waiter", "8013864841", "FakePerson@gmail.com"),
(12, 4, 3, "Gartner", "Carlos", "chef", "8013864842", "Carlos@gmail.com"),
(13, 5, 4, "Waluigi", "Wario", "Bartender", "8013864843", "Innovator@gmail.com"),
(14, 5, 4, "Gregory", "House", "Bus Boy", "8013864844", "NoThanks@gmail.com"),
(15, 5, 4, "Jane", "Ruler", "Waiter", "8013864845", "notruler@gmail.com"),
(16, 5, 4, "Nota", "Person", "Waiter", "801386484", "Realmolerat@gmail.com"),
(17, 5, 4, "Hans", "Gartner", "chef", "8013864847", "Hans@gmail.com"),
(18, 6, 4, "Wario", "Waluigi", "Bartender", "8013864848", "Innovator@gmail.com"),
(19, 6, 4, "House", "Gregory", "Bus Boy", "8013864849", "Thanks@gmail.com"),
(20, 6, 4, "Ruler", "Jane", "Waiter", "8013864850", "Ruler@gmail.com"),
(21, 6, 4, "Isa", "Person", "Waiter", "8013864851", "FakemoleRatn@gmail.com"),
(22, 6, 4, "Hans", "Garcia", "chef", "8013864852", "Garcia@gmail.com");

INSERT INTO Advertisement VALUES
(3, 3, "Instagram", "The best Italian in new york", "2025-8-01", "2025-9-01", "Italian
Lovers"),
(4, 4, "Twitter", "The best Italian in new california", "2025-8-01", "2025-9-01", "Italian
Lovers");

-- Insert into customer values
```

```
Insert into customer values (467, "Kyle", "Jameson", "1234567890", "kj@fakemail.com",
"phoneyBaloney Street");
Insert into customer values (487, "Brian", "Michaels", "1234567809",
"BrianM@fakemail.com", "phoneyMahoney Boulevard");

-- Insert into dinertable values
Insert into dinertable values (963, 15, 4, "Antonio's");
Insert into dinertable values (966, 17, 3, "Antonio's");

-- Insert into reservation values
Insert into reservation values (28, 467, 963, "2020-06-06 15:30:00", 6);
Insert into reservation values (29, 487, 966, "2020-06-06 12:00:00", 6);

-- Insert into menuitem
Insert into MenuItem values (89, "Cacio e pepe", "Spaghetti with a pepper and cheese
sauce", 14.99, "Dinner", 1);
Insert into MenuItem values (90, "Fettuccine Alfredo", "Pasta with a creamy cheese
sauce", 12.99, "Dinner", 1);

-- Insert into review values
Insert into review values (120, 467, 89, 4.1, "It's good", "2021-07-18");
Insert into review values (121, 487, 90, 2.6, "It's not good", "2022-11-08");

-- Insert into payment values
Insert into payment values (129, "Cash", 44.95, "2019-01-01 11:29:30");
Insert into payment values (128, "Credit", 34.50, "2018-11-12 08:28:00");

-- Insert into customerorder values
insert into customerorder values (500, 467, 129, "2019-02-02 15:15:10", 15.99);
insert into customerorder values (510, 487, 128, "2020-12-22 16:18:30", 25.99);

-- Insert into onlineorder values
insert into onlineorder values(678, 500, "Dinner", "phoneyBaloney Street", "2019-01-01
23:11:10", "delivered");
insert into onlineorder values(679, 510, "Dinner", "phoneyBaloney Street", "2019-01-01
12:12:30", "canceled");

-- Insert into billing values
insert into billing values(1000, 500, 129, 20.50, 2.50, 2.00, 25.00, "2019-01-01");
insert into billing values(2000, 510, 128, 20.50, 2.50, 2.00, 25.00, "2019-01-01");
```

```
-- Insert into orderitem
insert into orderitem values(987, 500, 89, 1, 12.99);
insert into orderitem values(988, 510, 90, 2, 24.99);

-- Insert into supplier
insert into supplier values(39, "Supplier Guy", "Jean Guye", "0987654321",
"supplierguye@fakemail.com");
insert into supplier values(30, "ACME Restaurant", "Wil.E. Mann", "9871234560",
"acmeMann@fakemail.com");

-- Insert into inventory values
insert into inventory values(76, "Potato", 23, 109, "At 0");
insert into inventory values(75, "Gnocchi", 26, 119, "10");

-- Insert into supplyorder
insert into supplyorder values(890, 39, 76, 15, "2025-12-12", "2025-12-24");
insert into supplyorder values(880, 30, 75, 15, "2025-11-12", "2025-11-24");

-- Insert into shiftschedule values
insert into shiftschedule values(555, "2022-08-08", "12:34:22", "15:00:00" );
insert into shiftschedule values(565, "2022-07-08", "12:34:22", "15:00:00" );

-- Insert into management values
insert into management values(245, "Charles", "Dixon", "1237890456",
"cd@fakemail.com");
insert into management values(235, "Michael", "Smith", "7123890456",
"ms@fakemail.com");

-- Insert into store values
insert into store values(368, 245, "Chuck's", "Dallas", "1357924680",
"chucks@fakemail.com", "09:00:00");
insert into store values(358, 235, "Mike's Bikes", "Dallas", "1537924680",
"mikesbikes@fakemail.com", "09:00:00");

-- Insert into staff values
insert into staff values(101, 555, 368, "Bob", "Richards", "Clerk", "2468135790",
"bobbyrick@fakemail.com");
insert into staff values(111, 565, 358, "Richard", "Roberts", "Mechanic", "1357946820",
"rickyrob@fakemail.com");
```

```
-- Insert into advertisement values
insert into advertisement values(222, 368, "Social Media", "Video", "2020-05-05",
"2021-05-05", "Boomers");
insert into advertisement values(232, 358, "TV", "Video", "2020-06-15", "2022-05-05",
"GenZ");
```

Time Log

Date	Team Member	Hours Spent	Description of work	Additional Comments

07/12/25 Meaghan: 3 hours Putting Final Proposal: Relevant Business Requirements and Expected Timelines together

07/14/25 Meaghan, Kaden, Nat: 1.5 hours Talking about assigned tasks for the Final Project: Kaden is writing the significance, Nat is working on the conceptual design, and Meaghan is adjusting the timetable of the project.

7/22/25: Nat: 4 hours putting together diagrams, ERD & Bubble Graph (conceptual diagram).
7/23/25: Nat: 1 hour putting together the Bubble graph for the final project.

07/24/25 Nat: (0.5 hours), Meaghan: 1.5 hours going through the final project, creating table of contents

7/28/25 Nat: 1 hour putting together the SQL script to make all tables necessary for the project.

07/29/25 Nat, Kaden, Meaghan: 1 hour talking about how the rest of the work will be divided up.

Each member will create 2 examples for the sql code. Also, Meaghan and Kaden will work on the rest of the **Executive Summary, Vision and Objective, Mission Statement, Service, and Use Case and Conclusion**

07/29/25 Kaden: 0.5 hours making the tables more readable. Nat: 0.5 hours writing the Current Release summary & Future Releases plans.

07/29/25 Meaghan: 0.5 hours double checking SQL script and creating inserts.

07/30/25 Meaghan: 0.5 hour Use cases and Executive Summary

07/31/25 Meaghan: 0.5 hour Creating Presentation

8/3/25 Kaden: 2 hours building out data for the database.

8/4/25 Kaden: 3.5 hours build out data for the database.

8/5/25 Meaghan: 2.5 hour Moving Kaden's work, double checking the code, editing, writing Vision and Objective, Mission Statement, and Conclusion.