A picture containing text, toy

Description automatically generated



Table of Contents

[Background Information 4](#_heading=h.gjdgxs)

[Food Delivery: Business Application 4](#_heading=h.30j0zll)

[Food Delivery: Model 5](#_heading=h.1fob9te)

[Online Food Ordering System ER Diagram 7](#_heading=h.3znysh7)

[Relationship and Cardinality Between the Entities 8](#_heading=h.2et92p0)

[Food Delivery: Logical Schema 10](#_heading=h.tyjcwt)

[Food Delivery: Creating the Database 12](#_heading=h.3dy6vkm)

[INSERT 13](#_heading=h.1t3h5sf)

[Example Query 1: Order Receipt 15](#_heading=h.4d34og8)

[Example Query 2: Customer Order History 16](#_heading=h.17dp8vu)

[Example Query 3: Issuing Discount Codes 17](#_heading=h.3rdcrjn)

A hand holding a smart phone

Description automatically generated with medium confidence

**Background Information**

| Background Information **Food Delivery: Business Application**  Food delivery has become an increasingly important aspect of modern-day living. According to a recent survey by Mintel, approximately one-third of the American population orders food online at least twice per week, highlighting the popularity of food delivery services (Hendelmann, 2023). In addition, the COVID-19 pandemic has further accelerated the growth of the food delivery industry, as more people opt for the convenience and safety of home delivery.  The global food delivery market is expected to continue its rapid expansion in the coming years, with a projected value of $182.3 billion by 2024 (Spdload, 2023). This growth is driven by the increasing availability of food delivery services across a wide range of businesses, including restaurants, grocery stores, meal kit services, catering companies, and food trucks. With the rise of mobile apps and online ordering platforms, food delivery has become more accessible and convenient than ever before, making it an important part of the overall food industry landscape. Food Delivery: Business Application Food delivery has become an essential aspect of the food industry, with businesses around the world recognizing its potential and adopting it as a valuable service. It is a type of courier service that involves transporting meals or food items from stores, restaurants, or third-party applications to customers who have placed orders and require delivery on an immediate or scheduled basis (Hendelmann, 2023).  The growth of the food delivery industry can be attributed to several factors, including technological advancements that have made it easier and more efficient to order and deliver food. Mobile apps and online platforms have transformed the way people order food, enabling them to browse menus, customize orders, and track deliveries in real-time.  In addition, changing consumer habits have played a significant role in the rise of food delivery. Consumers are increasingly seeking convenience and accessibility, and food delivery services offer a hassle-free way to enjoy their favorite meals without leaving their homes or offices. This has been especially important during the COVID-19 pandemic, which has led to an increase in demand for contactless delivery options.  Overall, food delivery has become an important business application for a range of food industry players, from small local restaurants to large multinational chains. By embracing this service, businesses can increase their reach and tap into new customer segments, while providing a valuable service that meets the evolving needs of today's consumers. | | |  |
| --- | --- | --- | --- |
|  | *“What food item do you think was delivered in the earliest documented case of food delivery?*  ***Yes, it was pizza.****”* |  |  |
|  | | |  |
| Food Delivery: Model In a food delivery system, there are typically three parties involved.   1. **The customer/ person ordering**: Customers can place their orders online via a third-party app or a restaurant's website, or they can place the order over the phone. The order usually includes details such as the type of food, the quantity, the delivery address, and the preferred delivery time. 2. **The delivery agent:** who is responsible for transporting the food from the restaurant to the customer's location. The delivery agent can be an employee of the restaurant or a third-party delivery service provider. In recent years, many third-party delivery service providers have emerged, such as Uber Eats, DoorDash, and Grubhub, which act as intermediaries between the customer and the restaurant. 3. **The restaurant where the order is placed:** The restaurant prepares the food, packages it, and hands it over to the delivery agent for transportation. The restaurant is responsible for ensuring that the food is of high quality and meets the customer's expectations. Additionally, the restaurant needs to coordinate with the delivery agent to ensure that the food is delivered on time and in good condition. In some cases, restaurants may also have their own delivery system, which means they handle both the preparation and delivery of the food.     In broader terms, below is the overview of how the model works or its operations: | | |  |
|  | | |  |
| * **Customer Order Placement**: Customers place their food orders through a mobile app or website, they browse through the available restaurant options selecting their preferred restaurant, menu items, and delivery address. Finally, they review the details of their selection on the checkout page. On the checkout page, they will then set the tip amount, payment method, delivery address, order delivery time, and if there is a promotion they wish to apply to their order. * **Restaurant Preparation**: Upon confirmation of payment, the restaurant receives the order and prepares the food, packaging it for delivery. * **Delivery Dispatch**: The delivery service matches available drivers in the area to the order, dispatches a delivery driver to pick up the food from the restaurant and deliver it to the customer. * **Delivery**: The delivery driver transports the food to the customer, following the optimized route provided to him/her by the delivery service's dispatch system. * **Payment**: The customer pays for the food, delivery fee, obtains a discount (if available) and any other applicable charges through the delivery service's mobile app or website. The customer payment details are verified prior to finalizing the order. The delivery service also pays the restaurant for the cost of the food and takes a commission for facilitating the transaction. The food delivery intermediary or service also pays the driver for the delivery of food. | | |  |

Graphical user interface, icon

Description automatically generated

**Entity Relationship Diagram**



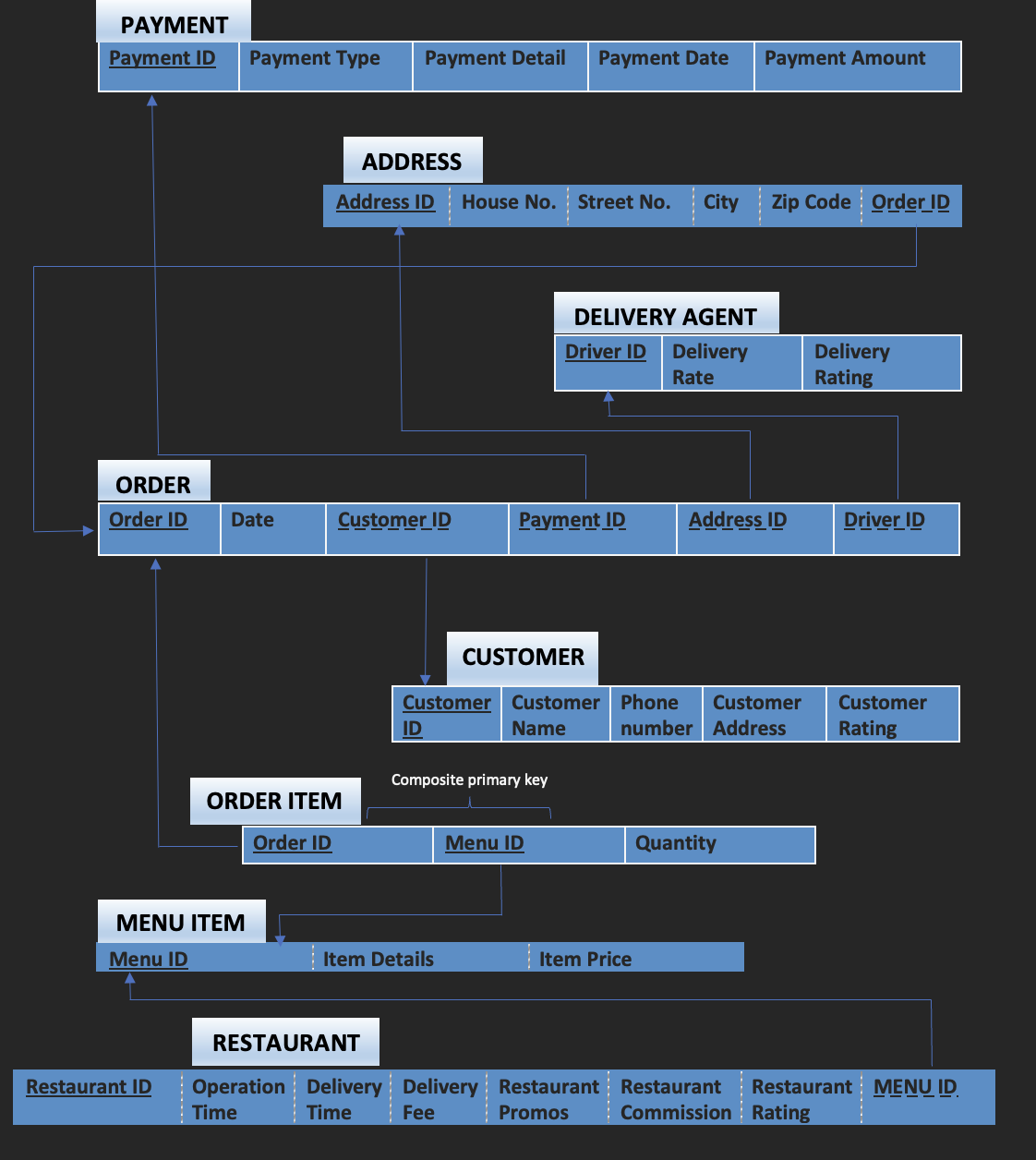
| Online Food Ordering System ER Diagram The ER Diagram for Online Food Ordering System illustrates the entity relationships between each entity.  Diagram  Description automatically generated  **Entities And Their Relationships**  **Entities in the Model:**  The Food Delivery System’s Model shall consist of the following entities.   1. **Customer**: This entity would represent the customers who use the online food delivery service to order food from restaurants. Primary key for this entity will be CustomerID. 2. **Restaurant:** This entity would represent the restaurants that provide food to customers through the online food delivery service. Primary key for this entity will be RestaurantID. 3. **Menu Item:** This entity would represent the different items available on the menus of restaurants. Primary key for this entity will be MenuID. 4. **Order:** This entity would represent the orders placed by customers for food items from restaurants. Primary key for this entity will be OrderID. 5. **Delivery Agent:** This entity would represent the agents responsible for delivering food items to customers. Primary key for this entity will be DriverID. 6. **Payment:** This entity would represent the payments made by customers for their food orders. Primary key for this entity will be PaymentID. 7. **Address:** This entity would represent the delivery addresses of customers. Primary key for this entity will be AddressID.  Relationship and Cardinality Between the Entities These entities can be related to each other in the following ways:   * A customer can place multiple orders and an order can be placed by a single customer. This is a one-to-many relationship. * A restaurant can have multiple menu items and a menu item can belong to only one restaurant. This is a one-to-many relationship. * An order can consist of multiple menu items and a menu item can be a part of multiple orders. This is a many-to-many relationship and can be resolved by creating a new entity, Order Item, to represent the relationship between Orders and Menu Items with a composite Primary Key of OrderID and MenuID. * An order can be placed for a single delivery address and a delivery address can receive multiple orders. This is a one-to-many relationship. * An order can be paid for using a single payment and a payment can be used to pay for multiple orders. This is a one-to-many relationship. * An order can be delivered by a single delivery agent and a delivery agent can deliver multiple orders. This is a one-to-many relationship. |  |
| --- | --- |
| **Entities in the Model are Customer, Restaurant, Menu Item, Order, Delivery Agent, Payment and Address** |
|  |

A picture containing timeline

Description automatically generated 

**Logical Schema 3rd Normal Form**

# Food Delivery Logical Schema



RELATIONS and ATTRIBUTES in the schema

*The logical schema defines the structure of the data itself and the relationships between the various attributes, tables, and entries.*

The above Logical Schema conforms to the ***3rd Normal Form***.

* All the database tables have atomic attributes. There are no columns in any of the tables with multiple values. Hence, the condition for the 1st Normal Form is satisfied.
* All the database tables do not have partial dependency. All columns have full functional dependencies. Hence, the condition for the 2nd Normal Form is satisfied.
* All the database tables do not have transitive dependency. Hence, all the conditions for the 3rd Normal Form are satisfied.
* Our database system consists of 8 relations PAYMENT, ADDRESS, DELIVERY AGENT, CUSTOMER, ORDER, ORDER ITEM, MENU AND RESTAURANT.
* PAYMENT has PaymentID, DELIVERY AGENT has DriverID and CUSTOMER has CustomerID as primary keys respectively.
* ADDRESS has Address ID as primary key and Order ID as foreign key.
* ORDER has OrderID as primary key and primary keys of PAYMENT, ADDRESS, DRIVER, CUSTOMER as foreign keys.
* RESTAURANT has RestaurantID as primary key and MenuID as foreign key, which is also the primary key for MENU.
* ORDER ITEM has a composite primary key made of two attributes OrderID and MenuID which are also foreign keys from the relation ORDER and MENU.



Graphical user interface

Description automatically generated

**MySQL Workbench**



|  | Food Delivery: Creating the Database |
| --- | --- |
|  |

## How to Create the Database

This SQL code creates a database schema for a food ordering system.

The schema includes tables for storing customer information, menu items and their prices, payment details, delivery agents, restaurant information, addresses, orders, and order items.

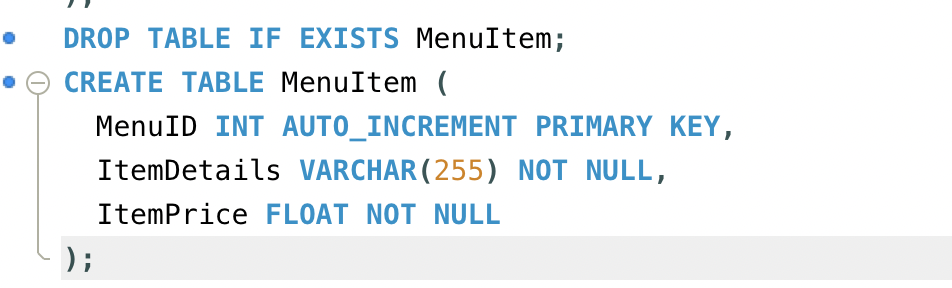
**Customer table**

The Customer table stores information about customers, including their ID, name, phone number, address, and rating and customer id is the primary key.

## 

**MenuItem table**

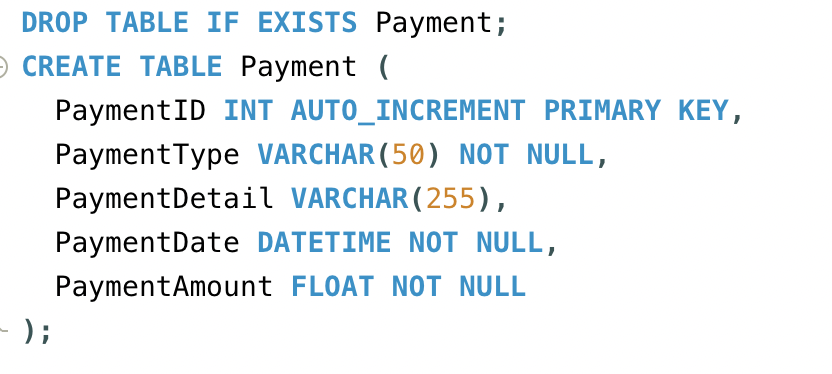
The MenuItem table stores details about the available menu items, including the ID, name, and price and menu id is the primary key.

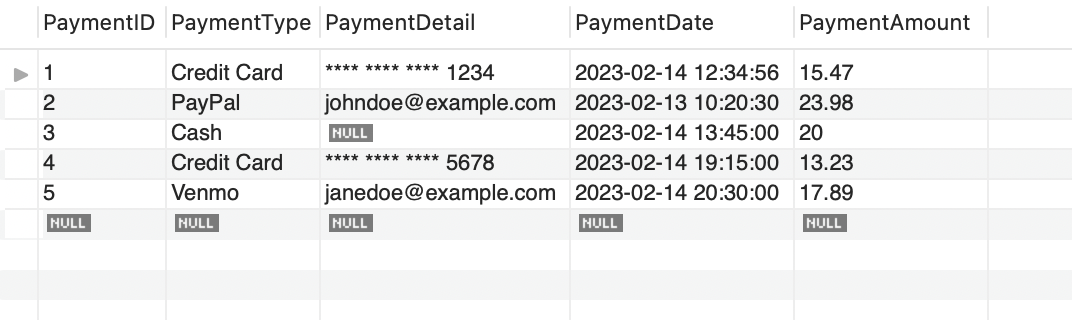




**Payment table**

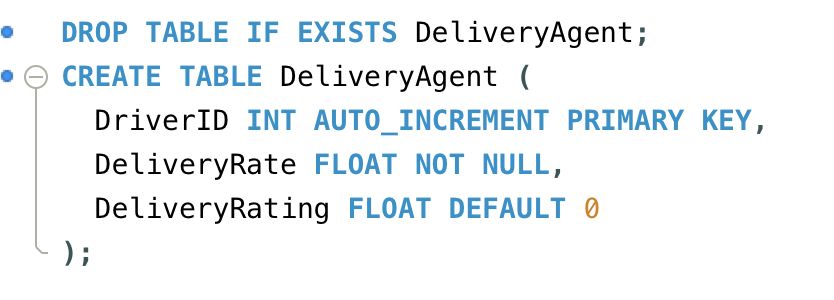
The Payment table stores details about the payments made by customers, including the ID, payment type, payment details, payment date, and amount and payment id is the primary key.

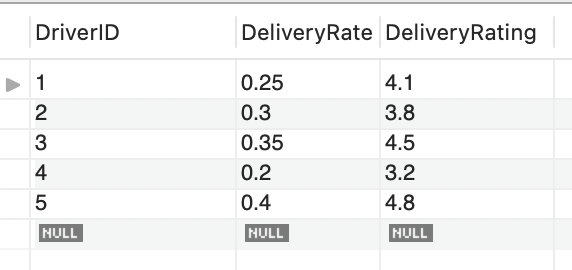




**DeliveryAgent table**

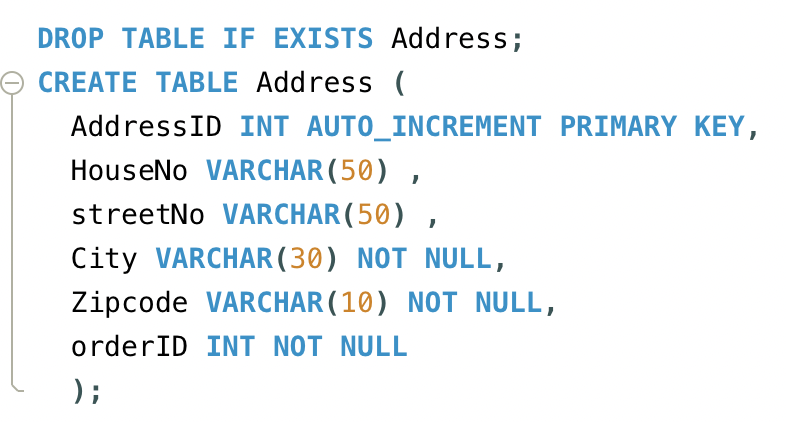
The DeliveryAgent table stores information about the delivery agents, including their ID, delivery rate, and rating and driver id is the primary key.

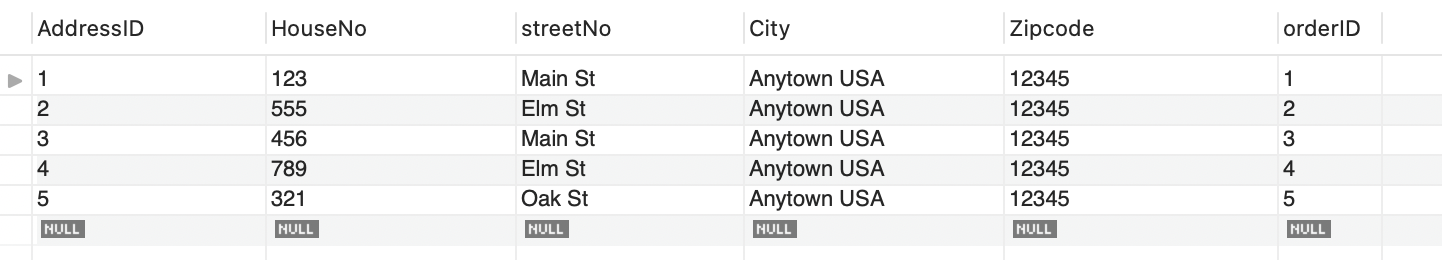




**Address table**

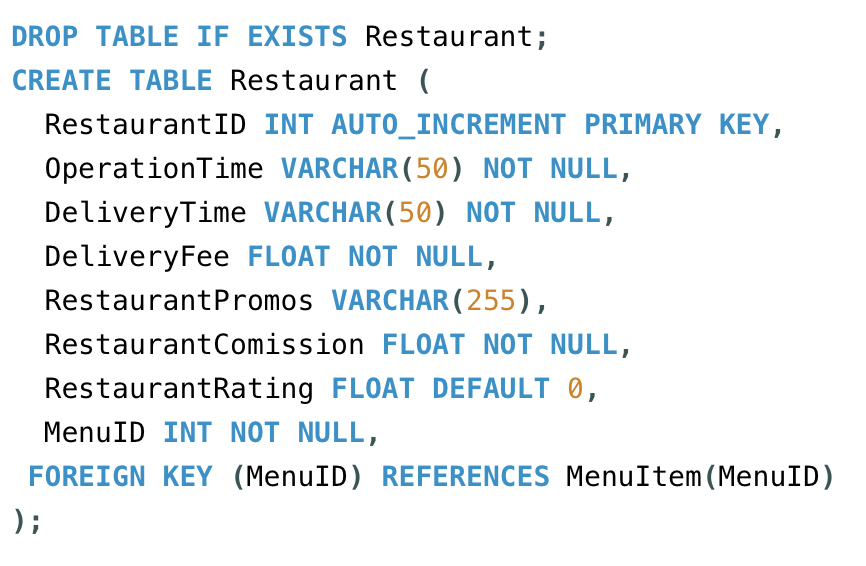
The Address table stores information about the delivery addresses, including the ID, house number, street number, city, zip code, and associated order ID and address id is the primary key.

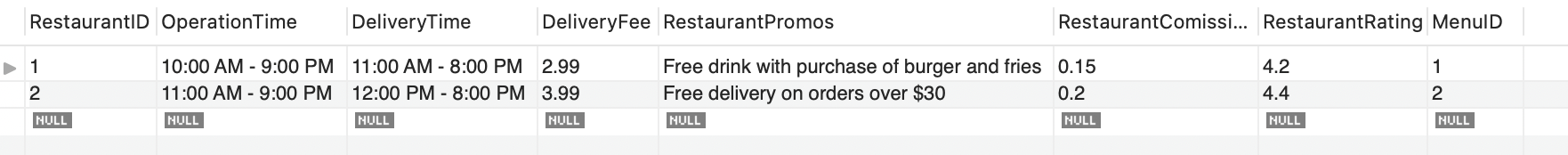




**Restaurant table**

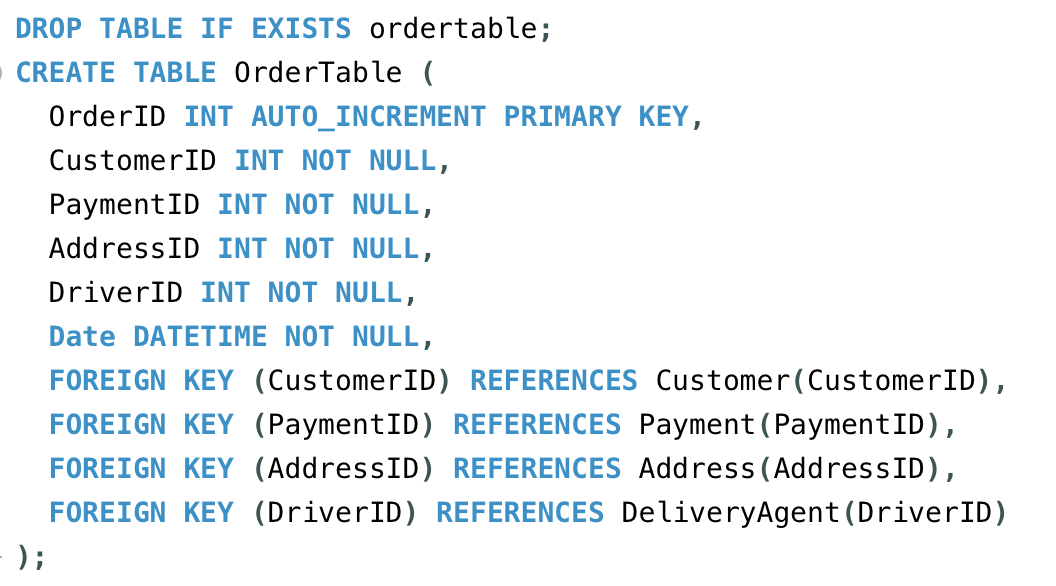
The Restaurant table stores information about the restaurants, including the ID, operation time, delivery time, delivery fee, restaurant promotions, commission rate, rating, and associated menu ID and restaurant id is the primary key.

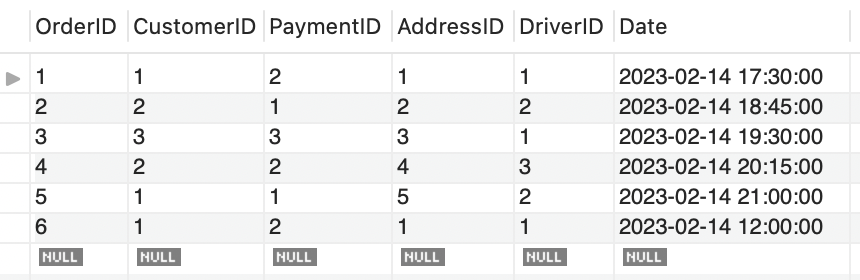




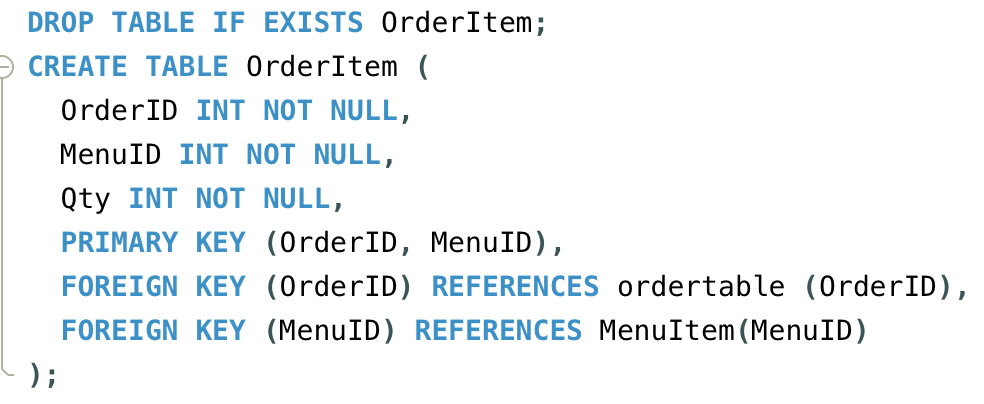
**OrderTable table**

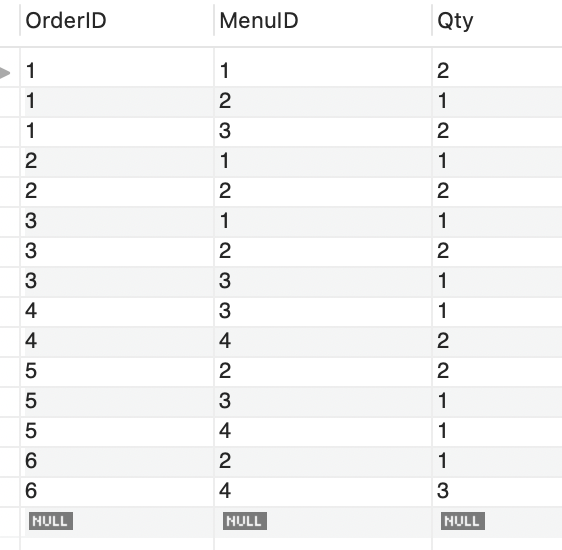
The OrderTable table stores information about the orders made by customers, including the ID, customer ID, payment ID, address ID, delivery agent ID, and date and order id is the primary key.





**OrderItem table**

The OrderItem table stores information about the items in each order, including the order ID, menu ID, and quantity. Order id and menu id are primary keys.



# 

# MYSQL MODIFICATION QUERIES –

## How to Modify the Database

**SQL code**

**Results in MySQL**

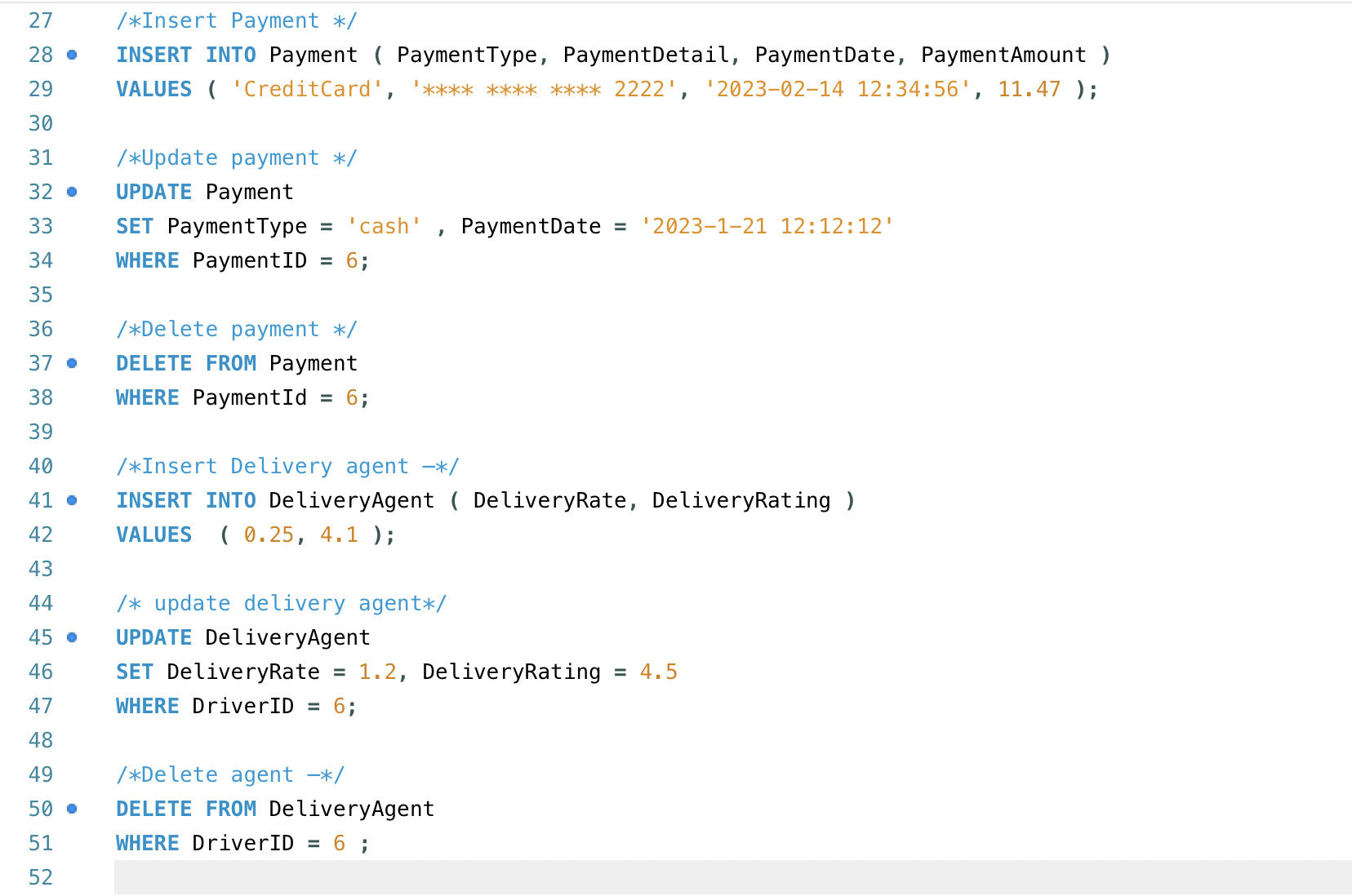
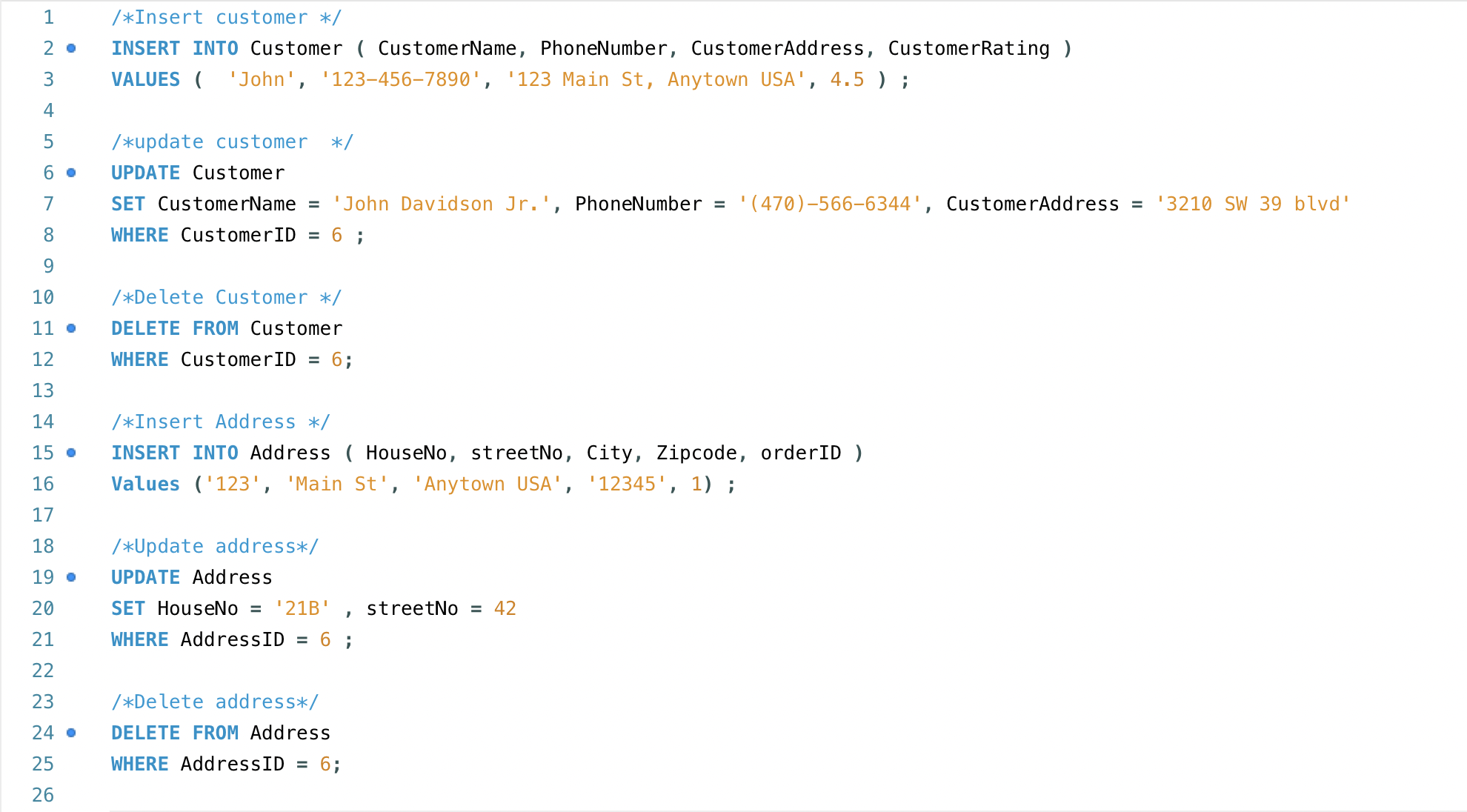
* Following are the the modification queries for the MYSQL i.e -

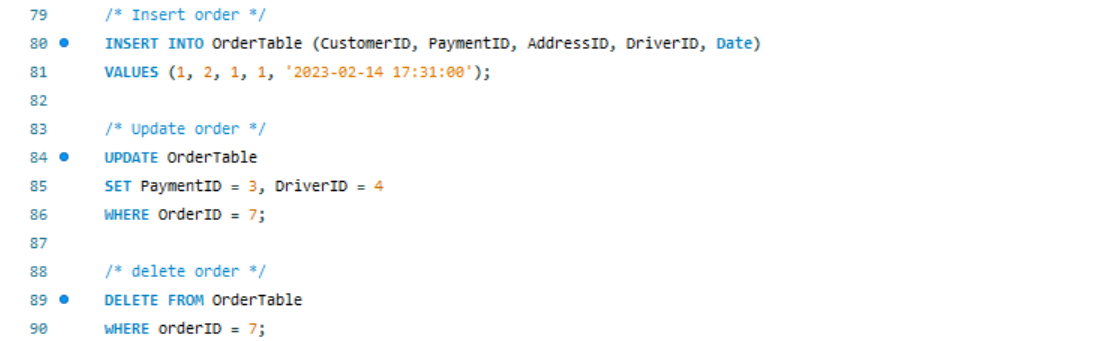
1. Insert
2. Update
3. Delete

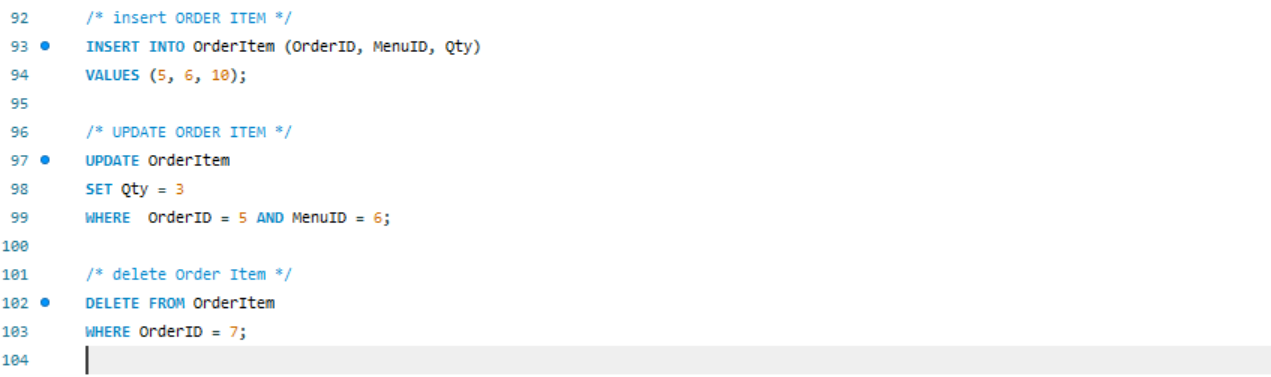
* In total there are 8 entities in the project and for the same tables have been created assigning the values to them.

The modification codes allows to make changes in the table and that could be done either by:

1. Inserting some data in the table through the insert query
2. Secondly, any data that needs to be updated within the table can be updated through the update query.
3. Thirdly the delete query helps in removing all the data that is not required.

* In the code below we have selected a particular entity starting from customer to orderitem and made all the three modifications to them ( i.e - insert, update and delete ). 



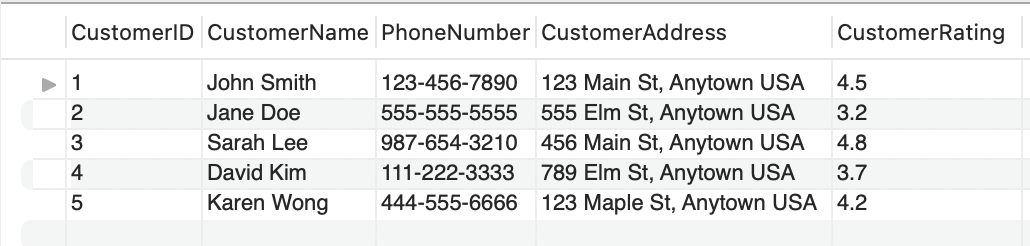
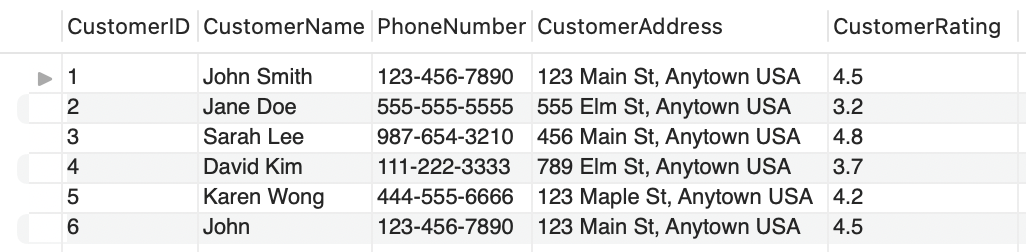


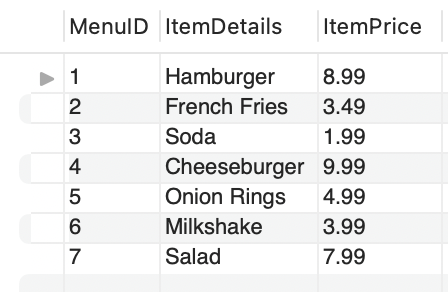
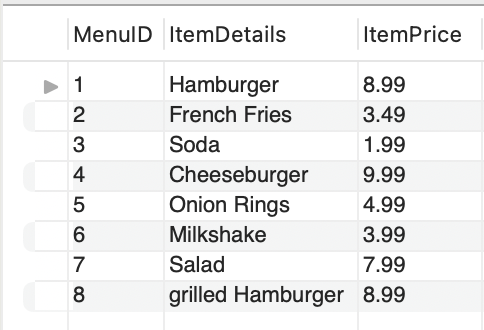
**Insert Query -**

Following are the examples of changes that were been made to the tables after the insert query was used in the MYSQL

1. For the first table i.e customer in total there were 5 values and after the insert query a 6th value was added( John)..
2. Secondly, for the Menu table the 8th value has been included in the table (i.e is the grilled hamburger).

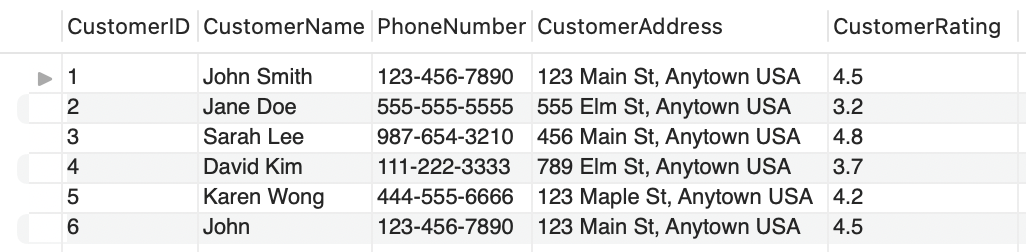
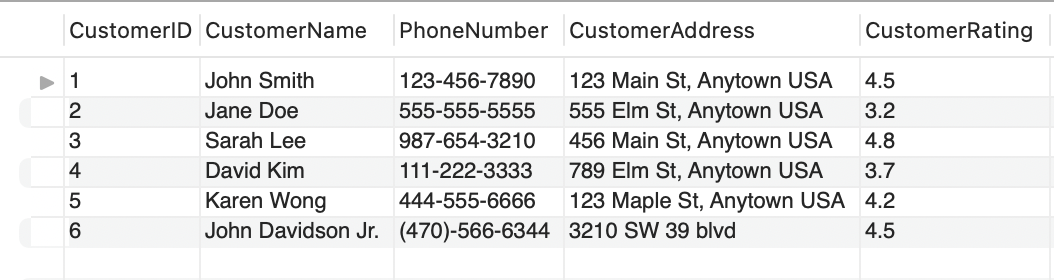
**Before After**

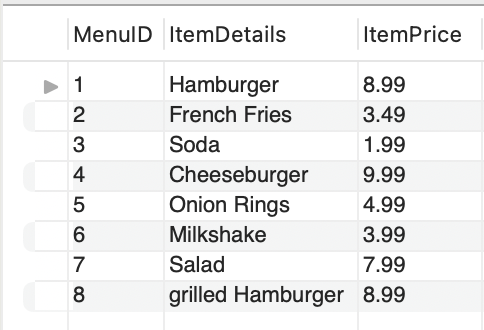
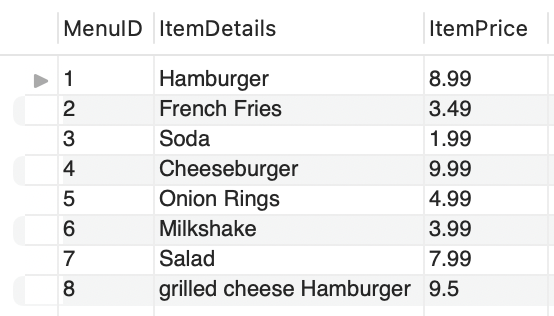
 

**Update Query -**

1. For the update query for the customer table the data for “john” changed with all other details.
2. Secondly, for the menu table the name gets updated for the 8th value after the update query is executed

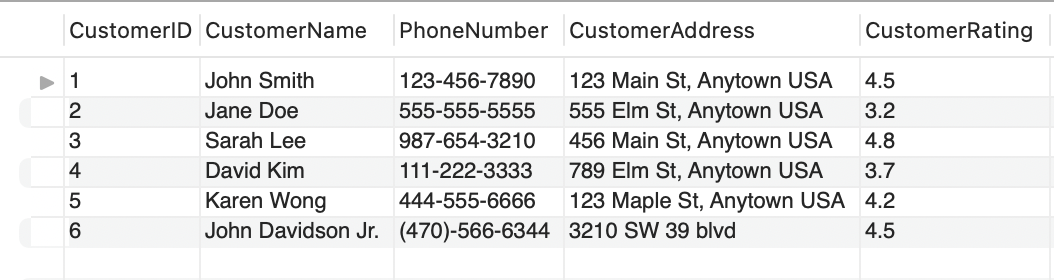
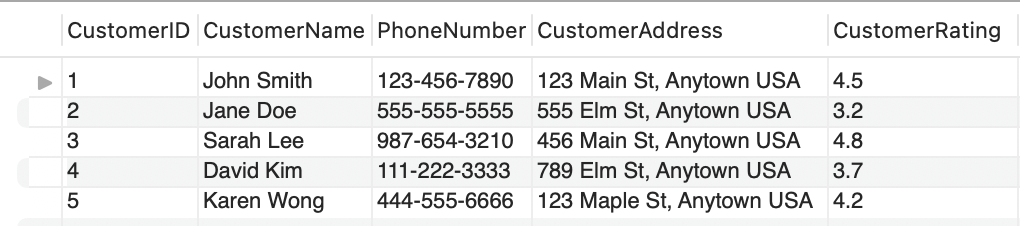
 

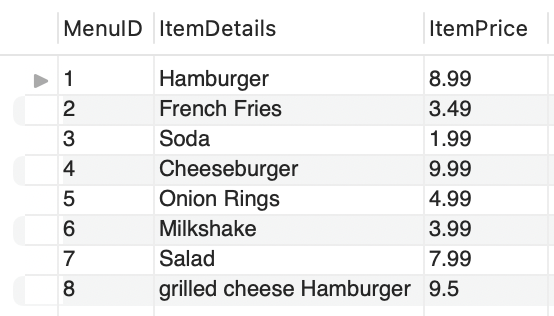
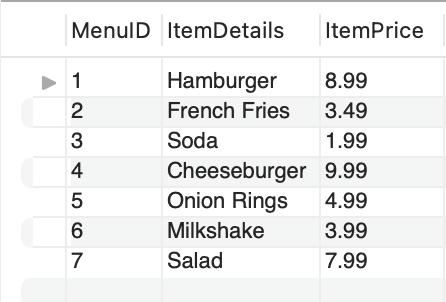
 

**Delete Query -**

1. For the delete query the customer table the 6th value(i.e - john davidson jr) gets deleted after we execute the delete query.
2. And for the Menu table it's the same thing where in the 8th value ( i.e - grilled cheese hamburger ) it gets deleted after the query is executed.

**Before After**

Graphical user interface

Description automatically generated 

**Example SQL Queries**

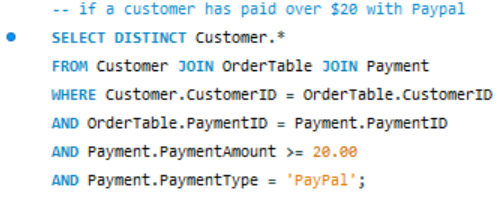
# Example Query 1: Order Receipt

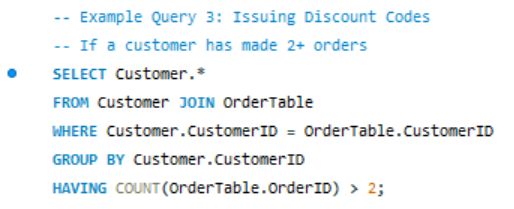
Querying a customer’s account for an order receipt can be considered one of the most common queries in a food delivery service app. Many customers check their receipt after they have placed their order to double check if all of their information and charges have gone through correctly. A customer’s order receipt can include their name, the restaurant’s name, order date and time, name and quantity of food items, payment information and total, delivery address, and delivery driver. Though, more or less of these items can be included. A mockup example of a food delivery service receipt is shown below:

| .    SQL Code & Query Result:    Our example of an order receipt contains the order ID, date/time stamp, the customer’s name, the customer’s address, payment type, payment amount, quantity and names of food items ordered, each food item’s price, and the delivery fee. Generating an order receipt takes information from six different tables: OrderTable, Customer, Payment, OrderItem, MenuItem, and Restaurant. Therefore, to produce all of this information into one query, many table joins are used. Then to select a receipt from a specific order, the condition OrderItem.OrderID = 1 is used to specify the retrieval of information from only OrderID 1. | | | | |  |
| --- | --- | --- | --- | --- | --- |
|  | | | |  |  |
| Example Query 2: Customer Order History Similar to an order receipt, a customer may wish to view all of their previous orders made on their account. To do this, a query for customer order history is enabled within the customer’s account on the app. This function is enabled so that a customer can view their history to recreate one of their past orders. In another case, they may want to check how many orders they have made within the past week, for example. In the example, order history includes all past orders made by one customer, quantity of items ordered, restaurant names, and order dates. Though, order history can include more or less of these items. See the image to the right, demonstrating the information queried when a customer views past orders made under their account. |  |  |
| SQL Code & Query Result:  Our example of a query of a customer’s order history includes the customer’s id, customer’s name, order id, date/time stamp, quantity and names of the food items ordered, and total payment amount. Producing a customer’s order history includes information from five different tables: OrderTable, Customer, OrderItem, MenuItem, and Payment; thus many table joins were utilized. The condition OrderTable.CustomerID = 2 is also specified to retrieve only information from CustomerID 2. In the code above, you can see in Jane’s order history she has made two orders, OrderID = 2 and 4, with the first order containing one hamburger and 2 french fries, and the second containing one soda and two cheeseburgers. |  |  |

# Example Query 3: Issuing Discount Codes

In the backend of the app, certain queries are designed so that once a customer’s account reaches certain conditions, discount codes are issued. From a business perspective, marketing to customers that they can earn certain discounts will incentivise them to use, for example, a different payment associated with a third party such as Paypal. This not only earns the third party a commission, but also the food delivery service company by encouraging customers to spend more on their app in the future, once they receive the discount code and place a new order with it.

SQL Code Examples:



Query Results Before & After:

| .  Our two examples of what a query’s conditions might look like when a customer gets a discount code issued to them are: when a customer makes two or more orders and when a customer pays over $20 with Paypal. The former of the two, joins the table OrderTable to Customer in order to return a list of customers that have made two or more orders with the condition COUNT(OrderTable.OrderID ) > 2. The latter joins the tables Customer, OrderTable, and Payment with the conditions Payment.PaymentAmount >= 20.00 and Payment.PaymentType = ‘Paypal’, and produces a list of customer information where those conditions apply. |  |
| --- | --- |
|  |  |
|  |  |

**Bibliography:**

*[1] grubtech (Ed.). (2021, March 8). How did food delivery start? history of food delivery. How did food delivery start? History of food delivery. Retrieved February 13, 2023, from https://blog.grubtech.com/history-of-food-delivery*

*[2] Hendelmann, V. (2023, January 2). The Food Delivery Business Model – A Complete Guide. productmint. Retrieved February 13, 2023, from https://productmint.com/the-food-delivery-business-model-a-complete-guide/*

*[3] Spdload. (2023, February 9). Food Delivery Business Model 101 for founders. SpdLoad. Retrieved February 13, 2023, from https://spdload.com/blog/food-delivery-business-model/*