Junqi Qian; Yuanliang Song

IST 659 M005

Food Ordering and Delivery System

Database management project

Content

Project Summary ……………………………………………………………………………………………. 1

Entity and Attribute Table ………………………………………………………………………………. 2

Entity-Relationship Diagram …………………………………………………………………………… 7

Creation of Tables ………………………………………………………………………………………….. 8

Major Data Questions …………………………………………………………………………………… 23

Interfaces ……………………………………………………………………………………………………… 26

Report …………………………………………………………………………………………………………… 35

Project Summary

The purpose of this project is to design a database management system for customers to order food delivery easily. In recent years, besides food safety, convenience and food diversity have become main pursuits for the food industry. In each area, there are different kinds of restaurants serving different foods such as Japanese food, Italian food, and Mexican food, etc. For customers who would like to eat something but do not have time or skills to cook it, they may order meal from restaurants that could bring what he or she wants right to the doors.

Before this system been introduced, customers can only order food delivery by calling the restaurant with no idea of the restaurant’s menu or going to each restaurant’s website. It is also inconvenient for customers that if they would like to get food from different restaurants because they might have to make multiply orders.

A thorough database system would enable customers order food from one or more restaurants and track the status of their food whether it’s get prepared or whether it’s get delivered on this system. The system necessarily collected basic information from restaurants including the name of the restaurants, menu list of the restaurants, and price of each types of food and so on. A massive volumes of data would be involved in this process for the large quantity of restaurants. In addition, the system will keep updating the menu of the restaurant and the status of the restaurants.

The system is mainly structured by three entities:

* Customers
* Restaurant
* Drivers

The primary users are the customers and the restaurants which were involved in the database. It would be an online portal or a mobile application that would allow customers search and choose the restaurants they like, browse through the menus and place their orders. It is also a platform for restaurants that can be widely known by more customers. Their efficiency of processing orders will also be monitored by customers.

For assessment of the data, the customers will have part of restrict that they can view the information of all the restaurant but can only view orders placed by themselves. They will not have authority to view information about other customers. On the other side, restaurants will have certain restrict as well. They can only view the orders and customer information involving their restaurants.

In this report, we will contain all the data of the proposed database system in an Access database. The following chart is the explanation of our entities and attributes. Every attribute is at an atomic level to reduce repetition and manual effort. After that, we will show relationships between each entities and attributes in an entity relationship diagram (ERD). Our report will demonstrate the business questions, interface and report as well.

Entity and Attribute Table

1. Customer – This entity consists of information about the customer who placed the food orders from the system.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Customer | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | CustomerID | INTEGER | NOT NULL | Primary Key, Unique Identifier of every customer |
| Other Attributes | FirstName | VARCHAR(30) | NOT NULL | First Name of customer |
|  | MiddleName | VARCHAR(30) | NULL | Middle Name of the student |
|  | LastName | VARCHAR(30) | NOT NULL | Last Name of the student |
|  | PhoneNumber | VARCHAR(30) | NOT NULL | Contact information of customer |
|  | Address1 | VARCHAR(30) | NOT NULL | Address information of customer |
|  | Address2 | VARCHAR(30) | NULL | Address information of customer |
|  | City | VARCHAR(30) | NOT NULL | City information of customer |
|  | CState | CHAR(2) | NOT NULL | State information of customer |
|  | ZipCode | VARCHAR(10) | NOT NULL | ZipCode information of customer |

1. Restaurant – This entity consists of the information about the restaurants offering delivery services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Restaurant | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | RestaurantID | INTEGER | NOT NULL | Primary key, unique  identifier of every restaurant |
| Other Attributes | RestaurantName | VARCHAR(40) | NOT NULL | Name information of restaurant |
|  | PhoneNumber | VARCHAR(20) | NOT NULL | Contact information of restaurant |
|  | Address1 | VARCHAR(30) | NOT NULL | Address information of restaurant |
|  | Address2 | VARCHAR(30) | NOT NULL | Address information of restaurant |
|  | City | VARCHAR(30) | NOT NULL | City of restaurant |
|  | CState | CHAR(2) | NOT NULL | State of restaurant |
|  | ZipCode | VARCHAR(10) | NOT NULL | ZipCode of the restaurant |
|  | RestaurantStatus | VARCHAR(7) | NOT NULL | Is restaurant  open or not |

1. Food – This entity consists of the information about the food included in orders.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Food | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | FoodID | INTEGER | NOT NULL | Primary key, unique identifier of food |
| Foreign Key | RestaurantID | INTEGER | NOT NULL | Primary key associated with Restaurant Entity. |
| Other Attributes | FoodName | VARCHAR (40) | NOT NULL | The name of food |
|  | Food Description | TEXT | NOT NULL | Food description |
|  | Price | VARCHAR (10) | NOT NULL | Price of food |

1. Car – This entity consists of the information about the car used for delivery.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Car | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | PlateNumber | VARCHAR(20) | NOT NULL | Primary key, unique identifier of each car |
| Other Attributes | Make | VARCHAR(20) | NOT NULL | Brand of car |
|  | Model | VARCHAR(20) | NOT NULL | Model of car |
|  | Color | VARCHAR(20) | NOT NULL | Color of car |

1. Driver – This entity consists of the information about the driver assigned to deliver the order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Driver | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | DriverID | INTEGER | NOT NULL | Primary key, the unique identifier of each driver |
| Other Attributes | FirstName | VARCHAR(30) | NOT NULL | First Name of the driver |
|  | LastName | VARCHAR(30) | NOT NULL | Last Name of the driver |
|  | PhoneNumber | VARCHAR(20) | NOT NULL | Contact information  of the driver |
| Foreign Key | PlateNumber | VARCHAR(20) | NOT NULL | primary key associated with Car entity |

1. Orders – This entity consists of the information about the order that placed by customers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Orders | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | OrderID | INTEGER | NOT NULL | Primary key, unique identifier of each order |
| Other Attributes | PlacedTime | TIME | NOT NULL | The time that the order is placed |
|  | OrderDate | DATE | NOT NULL | The date that the order is placed |
|  | OrderStatus | VARCHAR(10) | NOT NULL | The status of the order |
| Foreign Key | CustomerID | INTEGER | NOT NULL | primary key associated with Customer entity |
| Foreign Key | RestaurantID | INTEGER | NOT NULL | primary key associated with Restaurant entity |

1. Orderline – This entity consists of the information among the order placed by the customer, the customer and the food included in the order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Orderline | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Foreign Key | FoodID | INTEGER | NOT NULL | Primary key associated with Food entity |
| Foreign Key | OrderID | INTEGER | NOT NULL | Primary key associated with Orders entity |
| Other Attributes | Quantity | INTEGER | NOT NULL | The quantity of the food in the order |

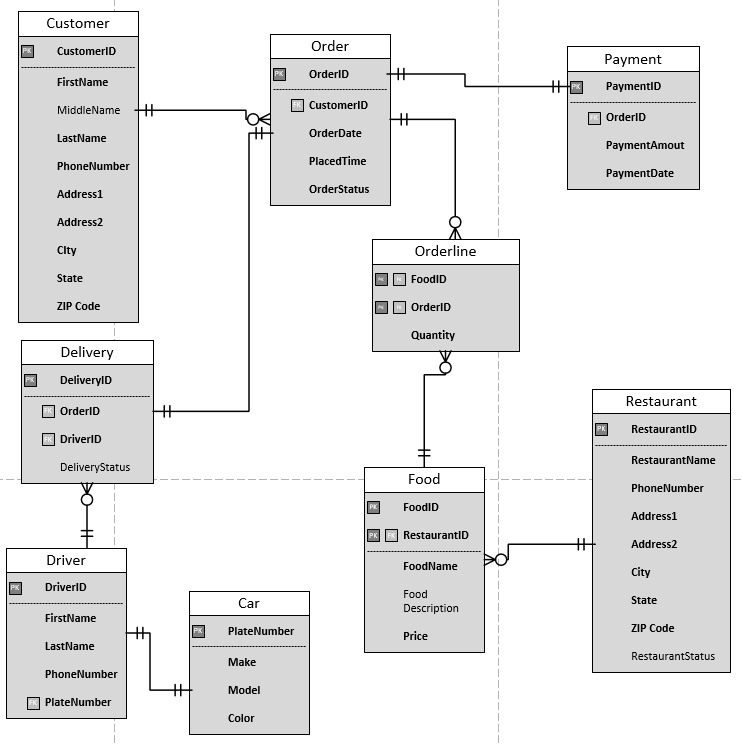
1. Payment – This entity consists of the information of payment associated with the placed order.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Payment | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | PaymentID | INTEGER | NOT NULL | Primary key, unique identifier for each  Payment |
| Foreign Key | OrderID | INTEGER | NOT NULL | Primary key associate with Orders entity |
| Other Attributes | PaymentAmount | Numeric | NOT NULL | Total amount of payment for the order |
|  | PaymentDate | DATETIME | NOT NULL | The date of payment |

1. Delivery – This entity consists of the information between the order and the driver.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ENTITY NAME:  Delivery | ATTRIBUTE NAME | FIELD TYPE | NULL/NOT  NULL | EXPLANATION |
| Primary Key | DeliveryID | INTEGER | NOT NULL | Primary key, unique identifier of every delivery |
| Foreign Key | OrderId | INTEGER | NOT NULL | Primary key associated with Order entity |
| Foreign Key | DriverID | INTEGER | NOT NULL | Primary key associated with Driver entity |
| Other Attributes | DeliveryStatus | VARCHAR(10) | NOT NULL | The status of delivery |

Entity-Relationship Diagram



Creation of Tables

CREATE TABLE Customer

(

CustomerID INTEGER NOT NULL IDENTITY PRIMARY KEY,

FirstName VARCHAR(30) NOT NULL,

MiddleName VARCHAR (30) NULL,

LastName VARCHAR(30) NOT NULL,

PhoneNumber VARCHAR(20) NOT NULL,

Address1 VARCHAR (30) NOT NULL,

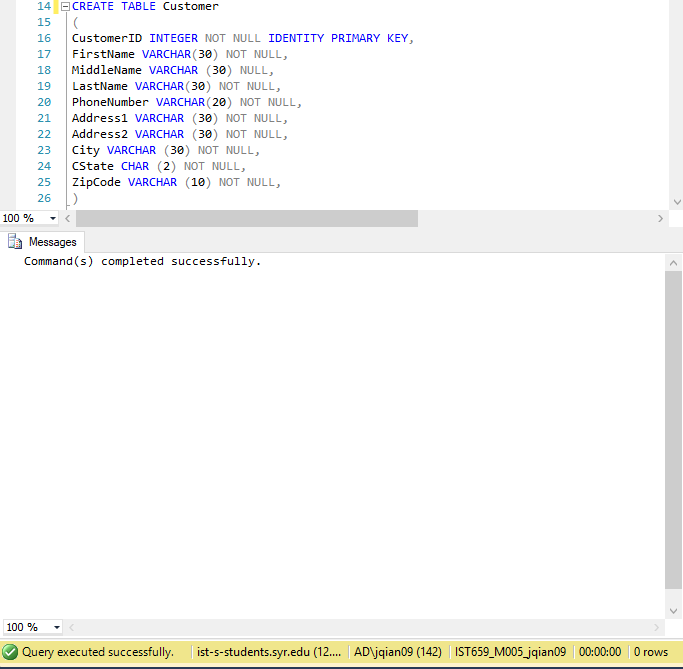
Address2 VARCHAR (30) NOT NULL,

City VARCHAR (30) NOT NULL,

CState CHAR (2) NOT NULL,

ZipCode VARCHAR (10) NOT NULL,

)



CREATE TABLE Restaurant

(

RestaurantID INTEGER NOT NULL IDENTITY PRIMARY KEY,

RestaurantName VARCHAR(40) NOT NULL,

PhoneNumber VARCHAR(20) NOT NULL,

Address1 VARCHAR (30) NOT NULL,

Address2 VARCHAR (30) NOT NULL,

City VARCHAR (30) NOT NULL,

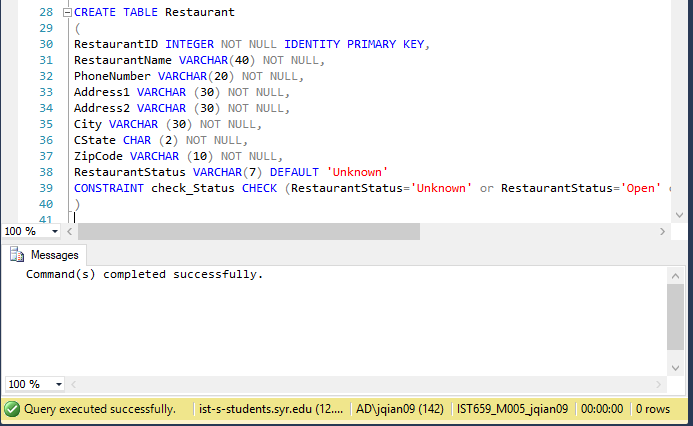
CState CHAR (2) NOT NULL,

ZipCode VARCHAR (10) NOT NULL,

RestaurantStatus VARCHAR(7) DEFAULT 'Unknown'

CONSTRAINT check\_Status CHECK (RestaurantStatus='Unknown' or RestaurantStatus='Open' or RestaurantStatus='Closed')

)



CREATE TABLE Food

(

FoodID INTEGER NOT NULL IDENTITY PRIMARY KEY,

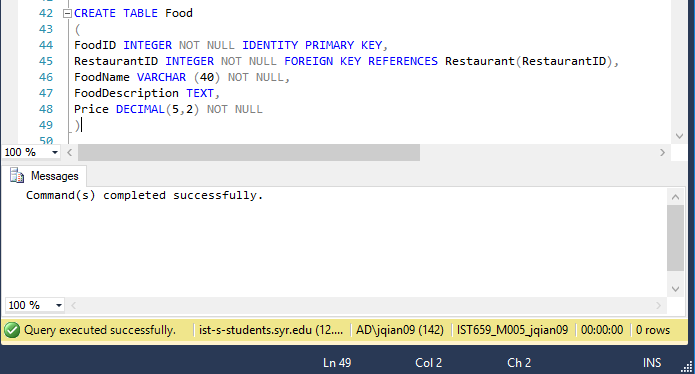
RestaurantID INTEGER NOT NULL FOREIGN KEY REFERENCES Restaurant(RestaurantID),

FoodName VARCHAR (40) NOT NULL,

FoodDescription TEXT,

Price DECIMAL(5,2) NOT NULL

)



CREATE TABLE Car

(

PlateNumber VARCHAR(20) NOT NULL PRIMARY KEY,

Maker VARCHAR(20) NOT NULL,

Model VARCHAR(20) NOT NULL,

Color VARCHAR(20) NOT NULL,

)



CREATE TABLE Driver

(

DriverID INTEGER NOT NULL IDENTITY PRIMARY KEY,

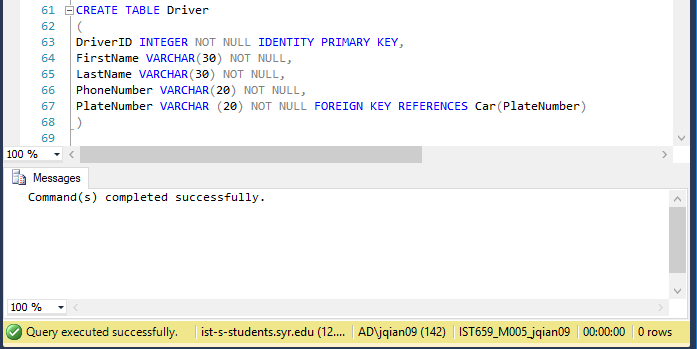
FirstName VARCHAR(30) NOT NULL,

LastName VARCHAR(30) NOT NULL,

PhoneNumber VARCHAR(20) NOT NULL,

PlateNumber VARCHAR (20) NOT NULL FOREIGN KEY REFERENCES Car(PlateNumber)

)



CREATE TABLE Orders

(

OrderID INTEGER NOT NULL IDENTITY PRIMARY KEY,

CustomerID INTEGER NOT NULL FOREIGN KEY REFERENCES Customer(CustomerID),

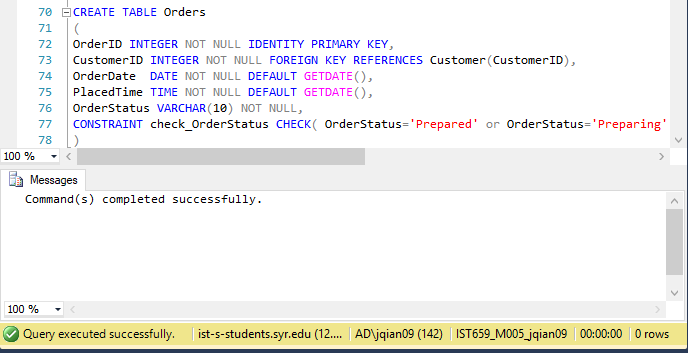
OrderDate DATE NOT NULL DEFAULT GETDATE(),

PlacedTime TIME NOT NULL DEFAULT GETDATE(),

OrderStatus VARCHAR(10) NOT NULL,

CONSTRAINT check\_OrderStatus CHECK( OrderStatus='Prepared' or OrderStatus='Preparing' or OrderStatus='Delivering' or OrderStatus='Delivered')

)



CREATE TABLE Orderline

(

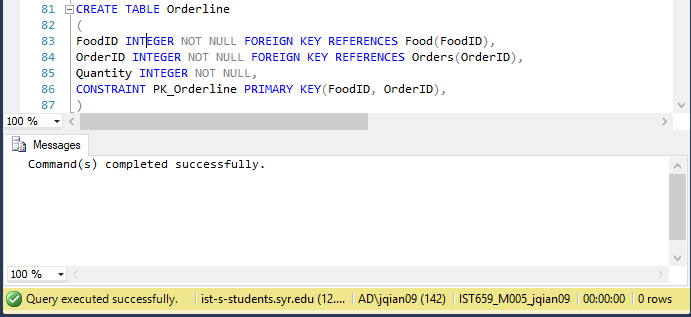
FoodID INTEGER NOT NULL FOREIGN KEY REFERENCES Food(FoodID),

OrderID INTEGER NOT NULL FOREIGN KEY REFERENCES Orders(OrderID),

Quantity INTEGER NOT NULL,

CONSTRAINT PK\_Orderline PRIMARY KEY(FoodID, OrderID),

)



CREATE TABLE Payment

(

PaymentID INTEGER NOT NULL IDENTITY PRIMARY KEY,

OrderID INTEGER NOT NULL FOREIGN KEY REFERENCES Orders(OrderID),

PaymentAmount DECIMAL(5,2) NOT NULL,

PaymentDate DATETIME NOT NULL DEFAULT GETDATE(),

)



CREATE TABLE Delivery

(

DeliveryID INTEGER NOT NULL IDENTITY PRIMARY KEY,

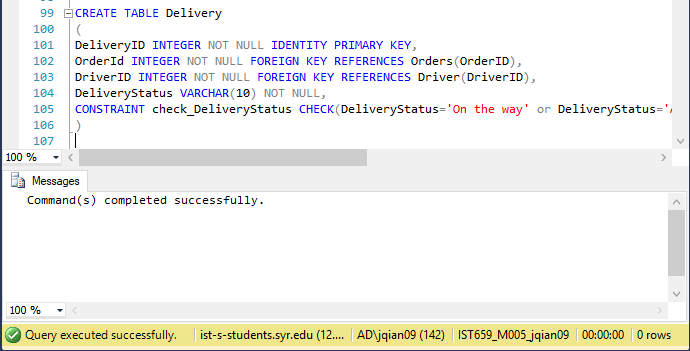
OrderId INTEGER NOT NULL FOREIGN KEY REFERENCES Orders(OrderID),

DriverID INTEGER NOT NULL FOREIGN KEY REFERENCES Driver(DriverID),

DeliveryStatus VARCHAR(10) NOT NULL,

CONSTRAINT check\_DeliveryStatus CHECK(DeliveryStatus='On the way' or DeliveryStatus='Arrived')

)



-- populating the Customer table with data

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Daenerys','E','Targaryen','315-455-3928','505','Walnut Ave.','Syracuse','NY','13210')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Jon','K','Snow','315-455-5465','325','E Genesee St.','Syracuse','NY','13244')

INSERT INTO Customer (FirstName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Yuanliang','Song','315-455-0530','505','Walnut Ave.','Syracuse','NY','13210')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Lily','M','Turner','315-455-3928','3','Ostrom Ave','Syracuse','NY','13225')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Olivia','A','Wallace','315-555-8888','4248','Nottingham Rd','Syracuse','NY','13244')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

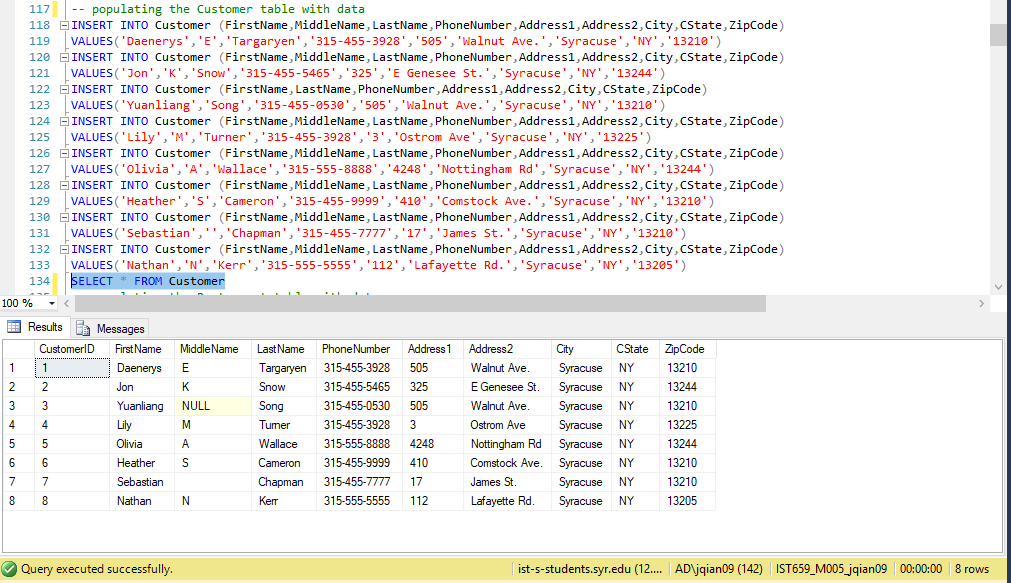
VALUES('Heather','S','Cameron','315-455-9999','410','Comstock Ave.','Syracuse','NY','13210')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Sebastian','','Chapman','315-455-7777','17','James St.','Syracuse','NY','13210')

INSERT INTO Customer (FirstName,MiddleName,LastName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Nathan','N','Kerr','315-555-5555','112','Lafayette Rd.','Syracuse','NY','13205')



-- populating the Restaurant table with data

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

VALUES('Bleu Monkey Cafe','315-741-3342','163','Marshall St','Syracuse','NY','13210','Open')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

VALUES('Johnny Rockets','315-474-7406','1','Destiny USA','Syracuse','NY','13204','Open')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

VALUES('Antonios Liverpool Pizzeria','315-123-1111','7608','Oswego','Liverpool','NY','13090','Open')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

VALUES('Taco Bell','315-123-3333','8095','Oswego Rd','Liverpool','NY','13090','Open')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

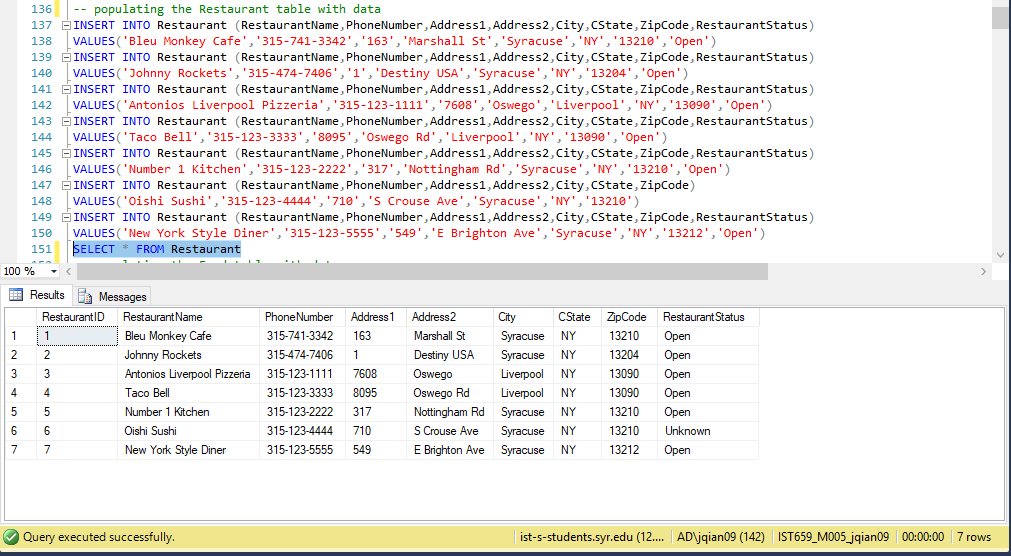
VALUES('Number 1 Kitchen','315-123-2222','317','Nottingham Rd','Syracuse','NY','13210','Open')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode)

VALUES('Oishi Sushi','315-123-4444','710','S Crouse Ave','Syracuse','NY','13210')

INSERT INTO Restaurant (RestaurantName,PhoneNumber,Address1,Address2,City,CState,ZipCode,RestaurantStatus)

VALUES('New York Style Diner','315-123-5555','549','E Brighton Ave','Syracuse','NY','13212','Open')



-- populating the Food table with data

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(1,'Beef Bowl','Served with corn,seaweed and scallions.',9.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(1,'Grilled Chicken with Teriyaki Sauce','Served with rice.',9.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(1,'Lizard Roll','Tempura shrimp and avocado topped with salmon, avocado, teriyaki and spicy sauce.',9.99)

INSERT INTO Food(RestaurantID,FoodName,Price)

VALUES(2,'Street Fries',9.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(2,'Garden Salad','Crisp seasonal greens topped with fresh diced tomatoes, shredded Wisconsin cheddar cheese and choice of dressing.',9.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(2,'The Original Burger','A certified Angus beff patty with crisp shredded lettuce, fresh tomato, chopped onion',9.99)

INSERT INTO Food(RestaurantID,FoodName,Price)

VALUES(3,'10 Traditional Wings',9.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(3,'10 Boneless Wings','Served with 1 bleu cheese',8.49)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(4,'General Tsos Chicken','Spicy.',8.5)

INSERT INTO Food(RestaurantID,FoodName,Price)

VALUES(4,'Fried Dumplings',6.8)

INSERT INTO Food(RestaurantID,FoodName,Price)

VALUES(4,'Sweet and Sour Chicken Combo',8.5)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(5,'Chalipa Cravings Box','Includes a Chalupa Supreme, Beefy 5-Layer Burrito, Crunchy Taco, Cinnamon Twists, and a Drink.',5)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(5,'Fiesta Taco Salad Combo','Served with adrink and 3 tacos',5)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(5,'XXL Grilled Stuft Burrito Combo','Served with a drink and a crunchy taco.',7.99)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

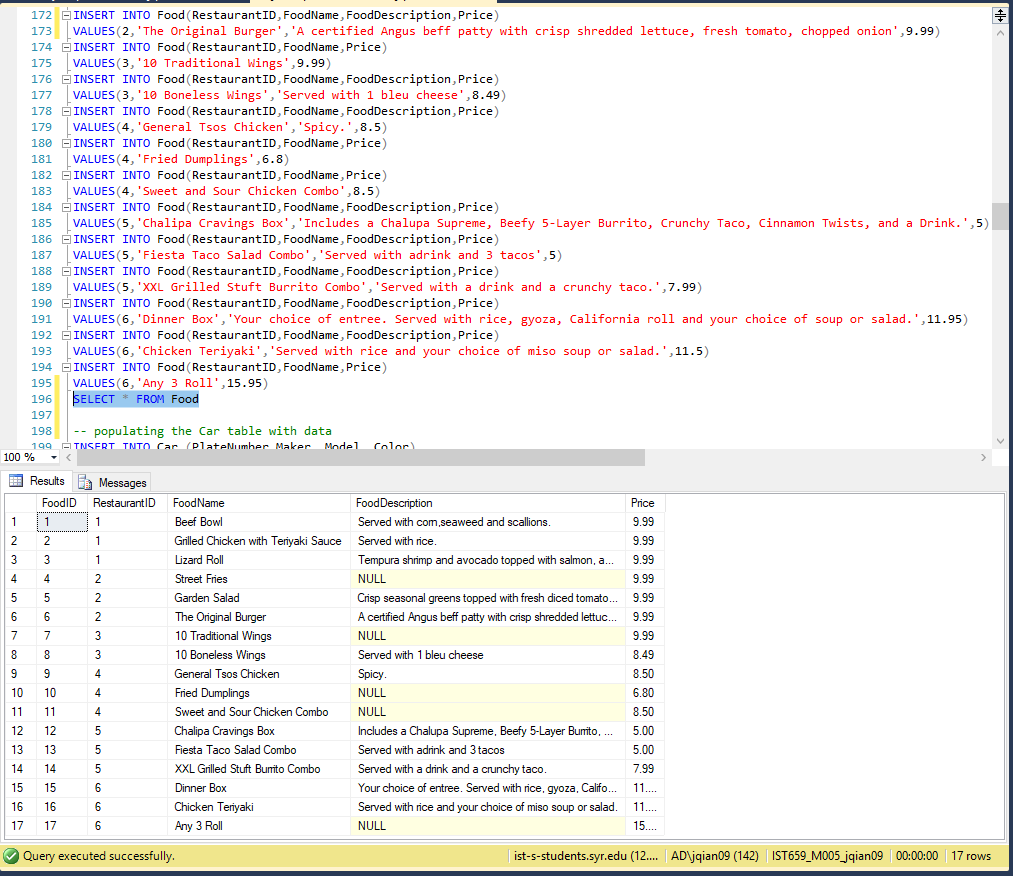
VALUES(6,'Dinner Box','Your choice of entree. Served with rice, gyoza, California roll and your choice of soup or salad.',11.95)

INSERT INTO Food(RestaurantID,FoodName,FoodDescription,Price)

VALUES(6,'Chicken Teriyaki','Served with rice and your choice of miso soup or salad.',11.5)

INSERT INTO Food(RestaurantID,FoodName,Price)

VALUES(6,'Any 3 Roll',15.95)



-- populating the Car table with data

INSERT INTO Car (PlateNumber,Maker, Model, Color)

VALUES('XYZ-1234','Honda','Civic','Black')

INSERT INTO Car (PlateNumber,Maker, Model, Color)

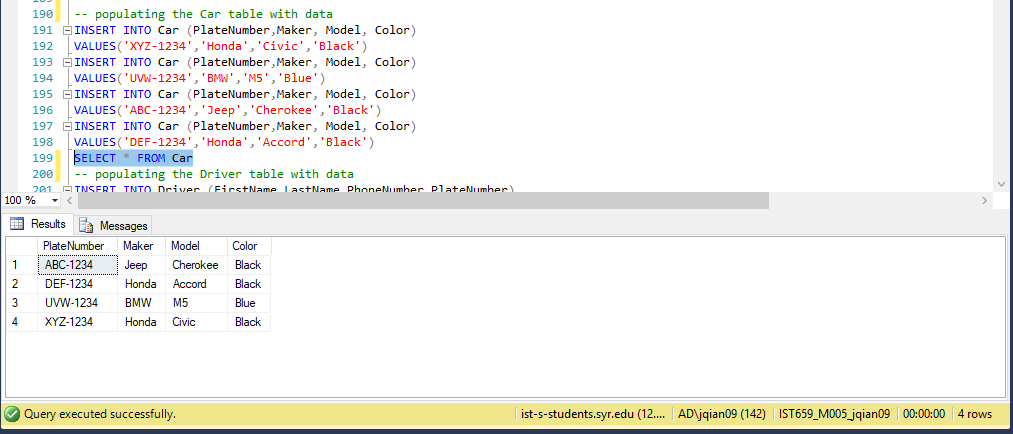
VALUES('UVW-1234','BMW','M5','Blue')

INSERT INTO Car (PlateNumber,Maker, Model, Color)

VALUES('ABC-1234','Jeep','Cherokee','Black')

INSERT INTO Car (PlateNumber,Maker, Model, Color)

VALUES('DEF-1234','Honda','Accord','Black')



-- populating the Driver table with data

INSERT INTO Driver (FirstName,LastName,PhoneNumber,PlateNumber)

VALUES('Dorothy','Paige','315-555-0126','XYZ-1234')

INSERT INTO Driver (FirstName,LastName,PhoneNumber,PlateNumber)

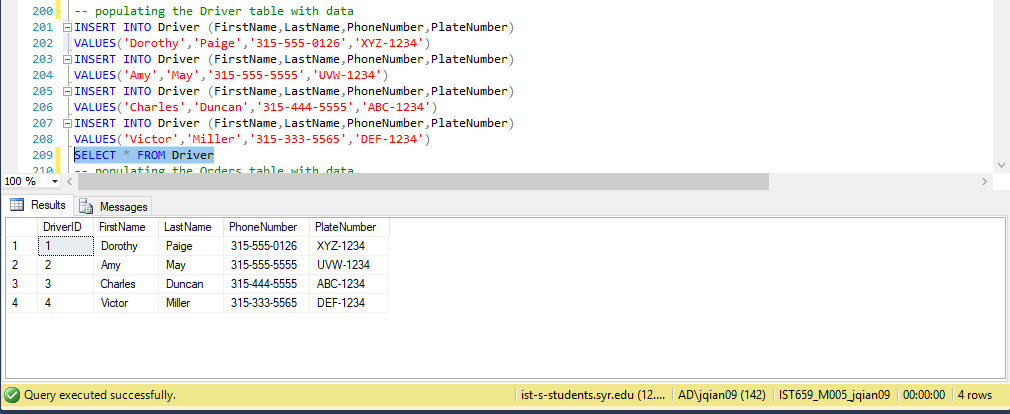
VALUES('Amy','May','315-555-5555','UVW-1234')

INSERT INTO Driver (FirstName,LastName,PhoneNumber,PlateNumber)

VALUES('Charles','Duncan','315-444-5555','ABC-1234')

INSERT INTO Driver (FirstName,LastName,PhoneNumber,PlateNumber)

VALUES('Victor','Miller','315-333-5565','DEF-1234')



-- populating the Orders table with data

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(1,'2019-04-16','12:15:24.000','Delivered')

INSERT INTO Orders(CustomerID,OrderStatus)

VALUES(2,'Delivering')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(3,'2019-04-15','17:15:00.000','Delivered')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(4,'2019-04-14','18:20:00.000','Delivered')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(5,'2019-04-13','11:00:00.000','Delivered')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

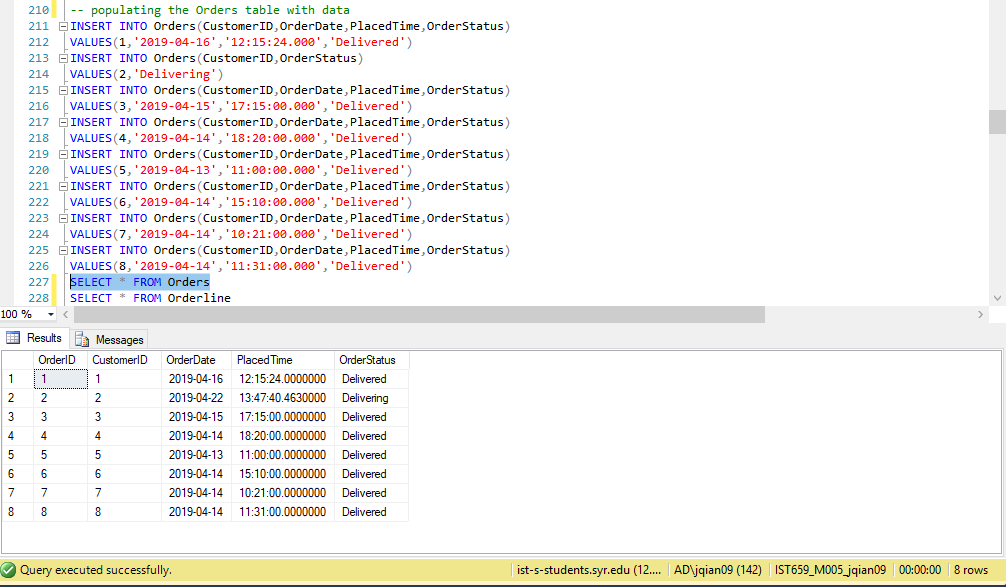
VALUES(6,'2019-04-14','15:10:00.000','Delivered')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(7,'2019-04-14','10:21:00.000','Delivered')

INSERT INTO Orders(CustomerID,OrderDate,PlacedTime,OrderStatus)

VALUES(8,'2019-04-14','11:31:00.000','Delivered')



-- populating the Orderline table with data

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(1,1,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(2,1,2)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(15,1,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(4,1,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(10,1,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(3,2,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(6,2,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(12,2,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(10,3,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(10,4,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(11,5,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(7,5,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(6,5,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(8,6,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(9,6,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(4,6,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(10,7,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(5,7,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

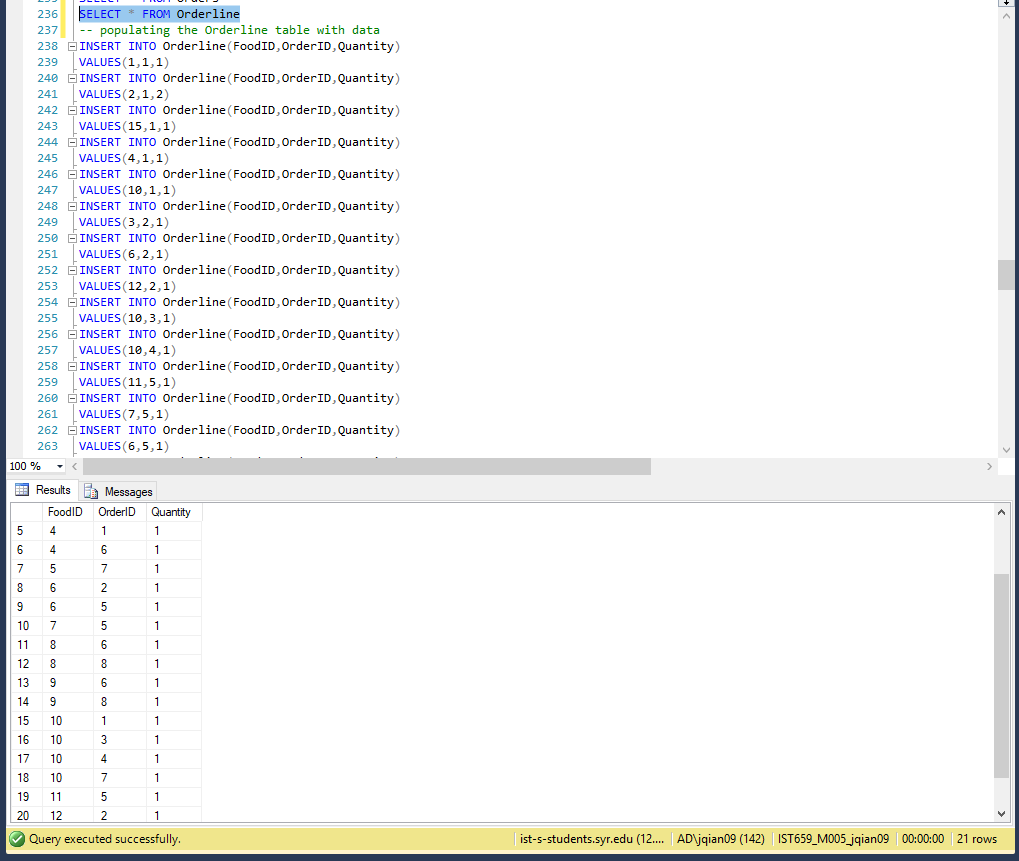
VALUES(3,8,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(8,8,1)

INSERT INTO Orderline(FoodID,OrderID,Quantity)

VALUES(9,8,1)



-- populating the Delivery table with data

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (1,1,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (2,2,'On the way')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (3,3,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (4,4,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (5,1,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

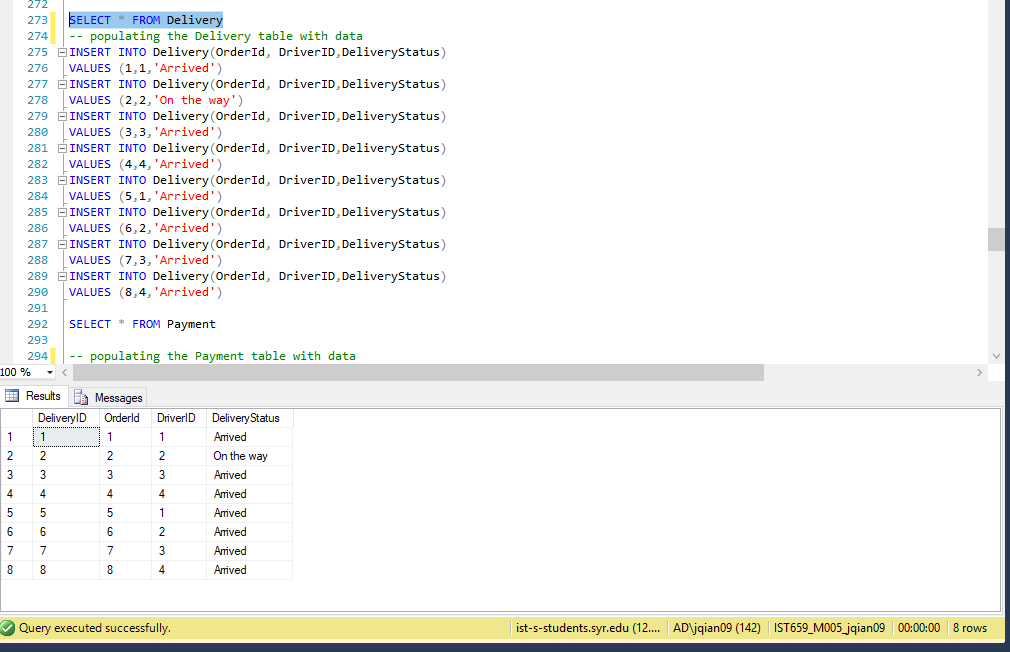
VALUES (6,2,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (7,3,'Arrived')

INSERT INTO Delivery(OrderId, DriverID,DeliveryStatus)

VALUES (8,4,'Arrived')



-- populating the Payment table with data

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(1,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=1))

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(2,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=2))

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(3,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=3))

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(4,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=4))

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(5,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=5))

INSERT INTO Payment (OrderID,PaymentAmount )

VALUES(6,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=6))

INSERT INTO Payment (OrderID,PaymentAmount )

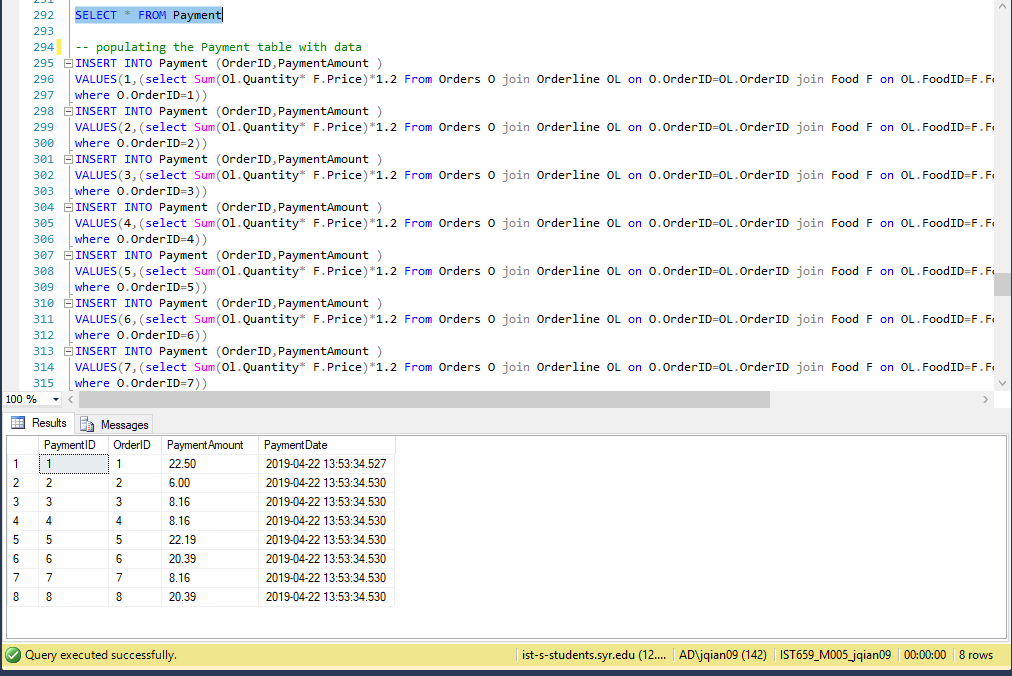
VALUES(7,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=7))

INSERT INTO Payment (OrderID,PaymentAmount )

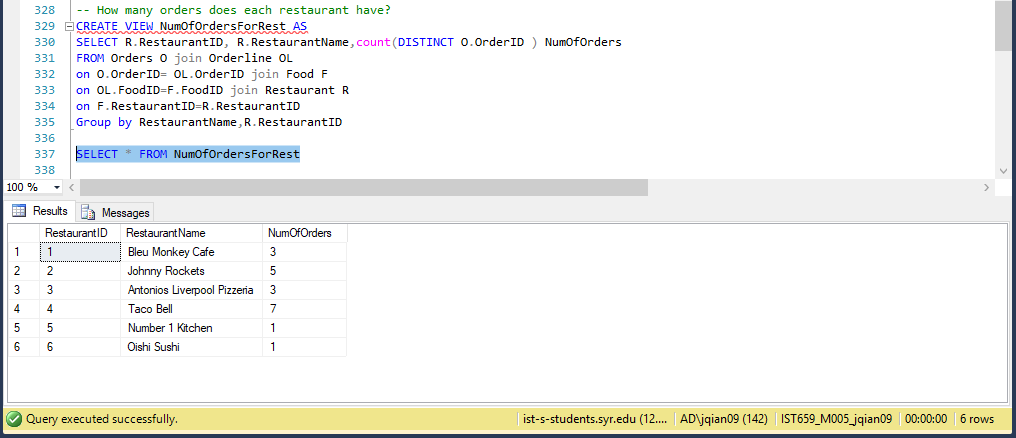
VALUES(8,(select Sum(Ol.Quantity\* F.Price)\*1.2 From Orders O join Orderline OL on O.OrderID=OL.OrderID join Food F on OL.FoodID=F.FoodID

where O.OrderID=8))

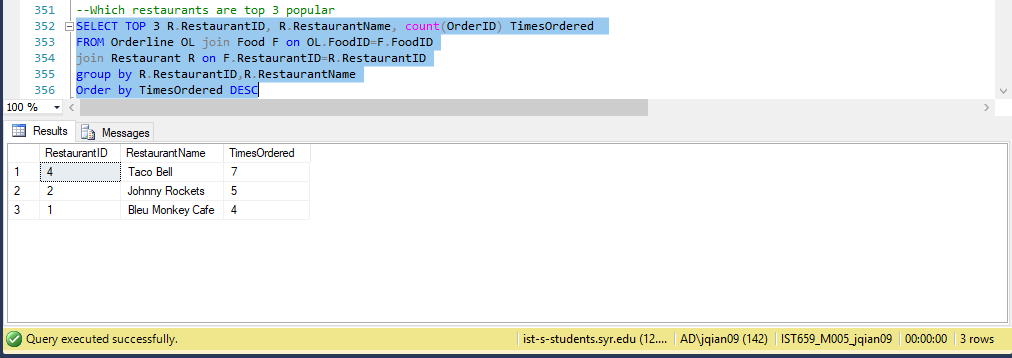


Major Data Questions

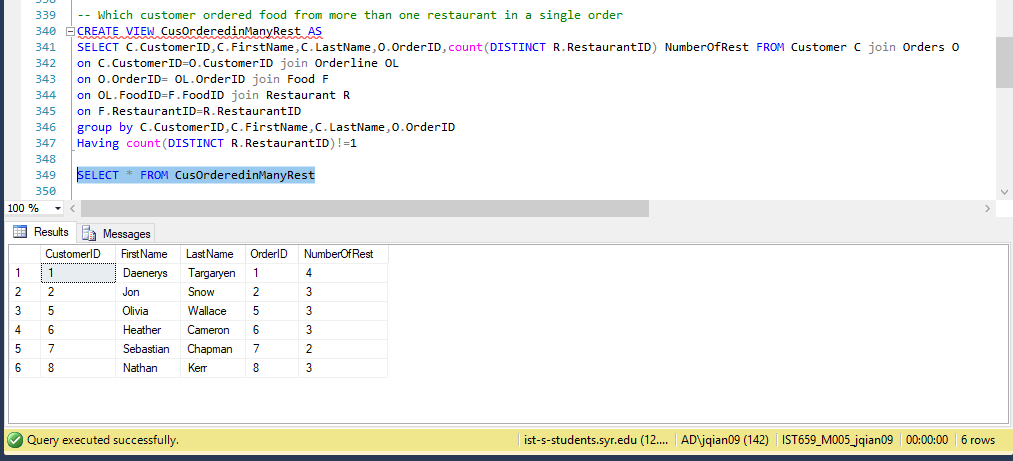
1. How many orders did a specific restaurant have?



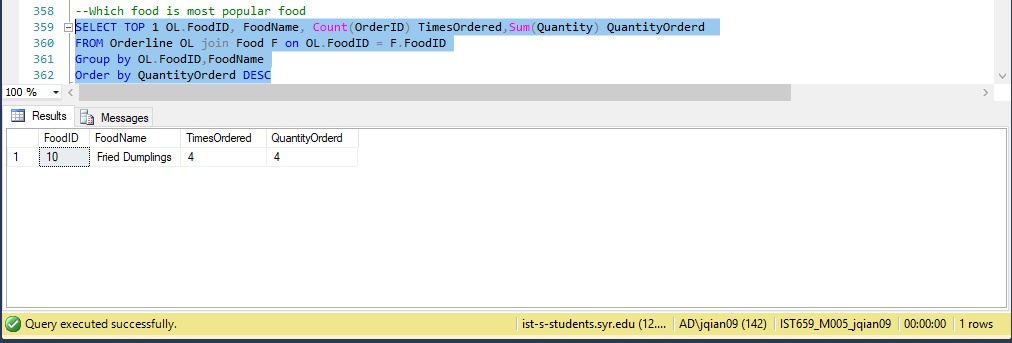
1. Which restaurants are top 3 popular?



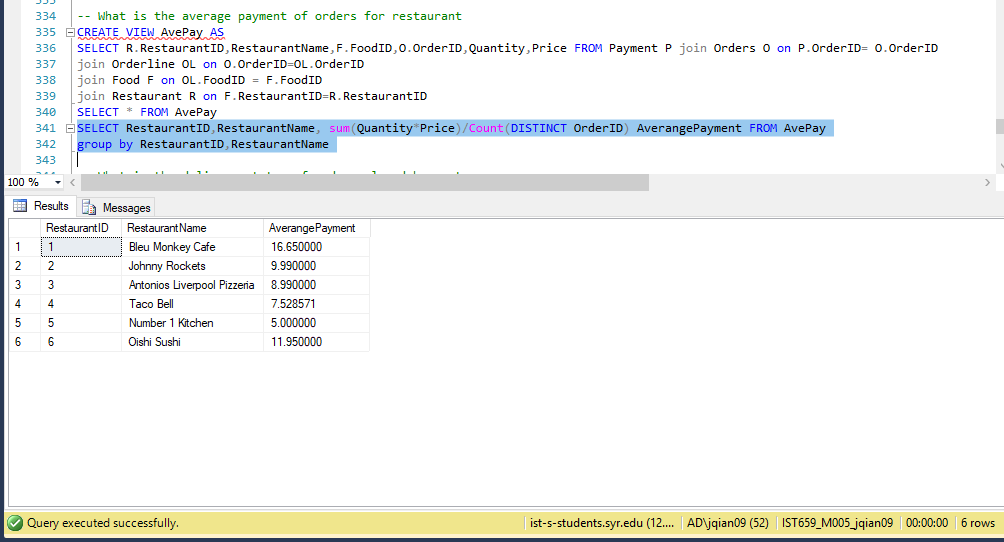
1. Which customer ordered food from more than one restaurant in a single order



1. Which food is most popular food



1. What is the average payment of orders for restaurant



1. What is the delivery status of orders placed by customers?

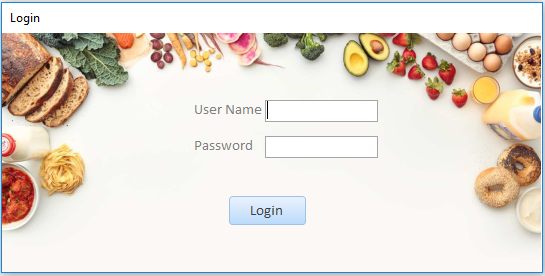


1. What is the order status of orders placed by customers

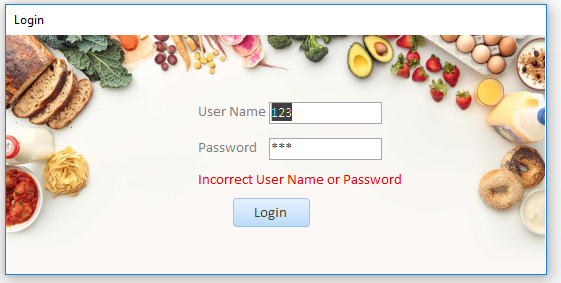


Interface

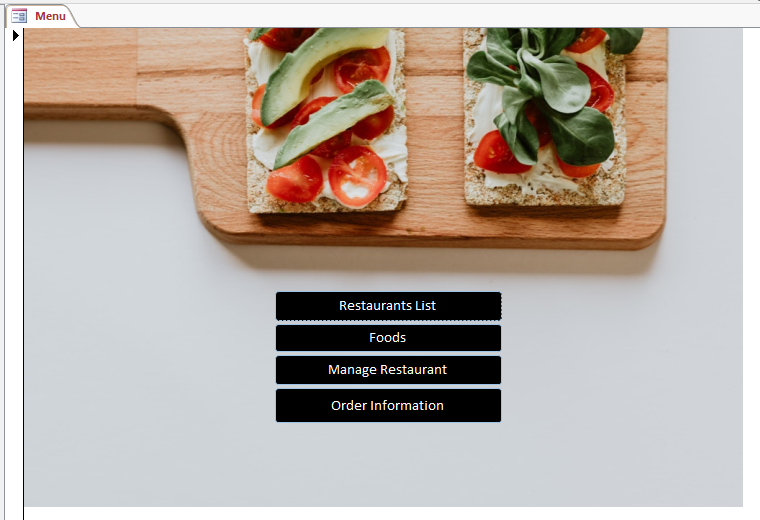
Firstly, there will be a Login page that requires customers and restaurants to input their user name and password.



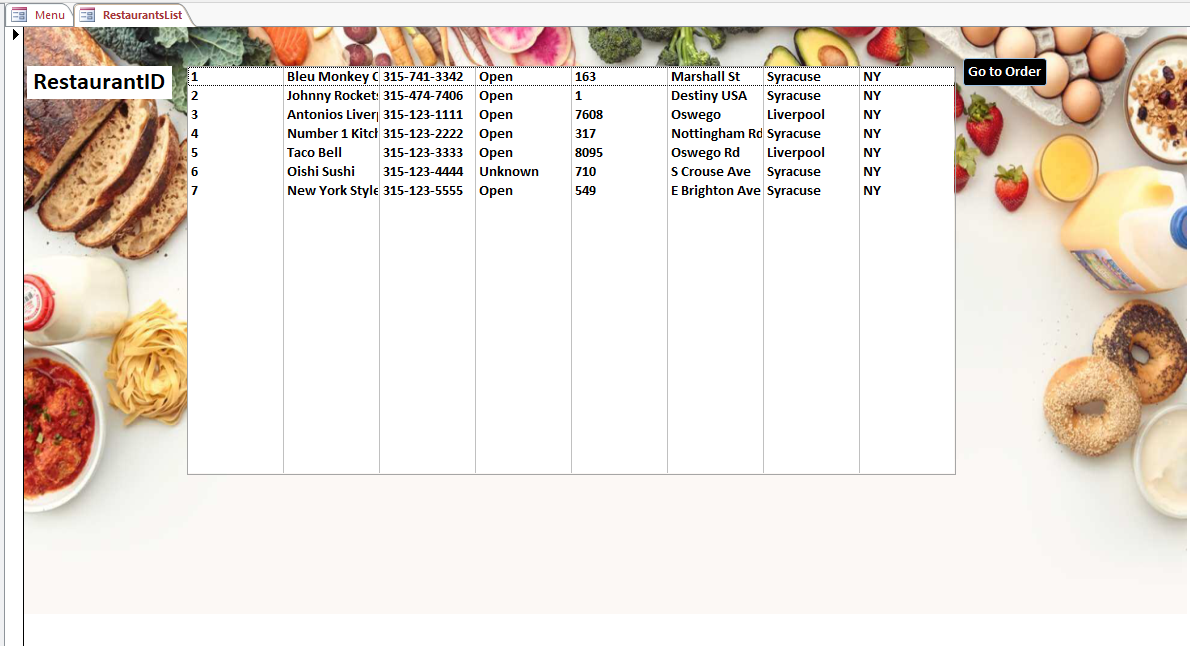
If customers or restaurants click the “Login” button with the username or the password unmatched, the system will show an error message with “Incorrect User Name or Password”.

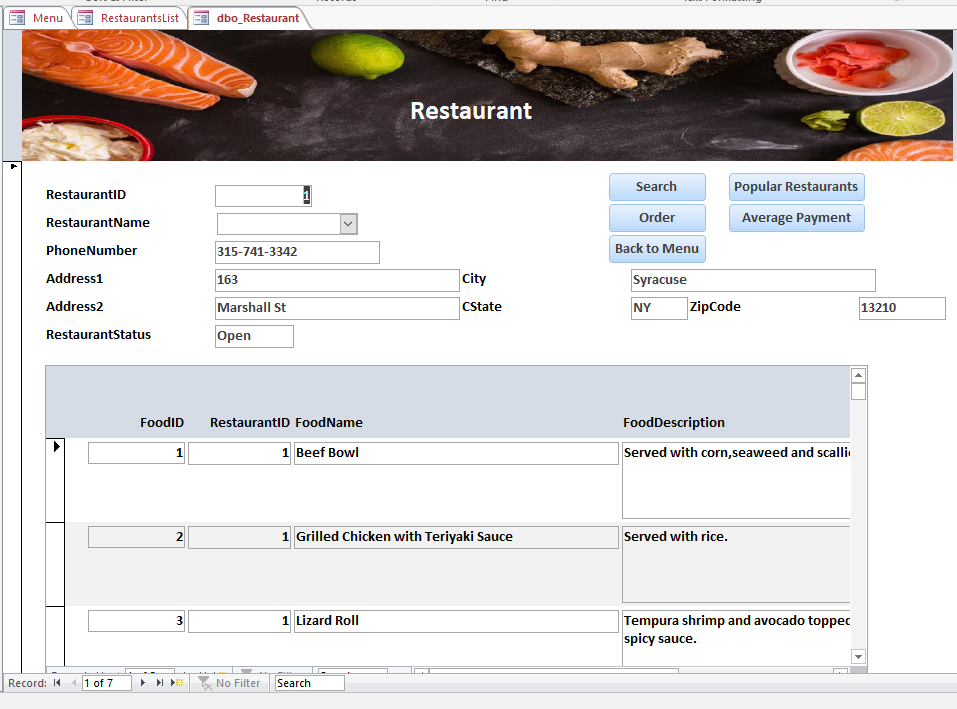


Once the user logs in with correct user name and password, a menu page will show up. There are four choices to navigate the user: Restaurants list, Food list, Manage Restaurant and Order Information.

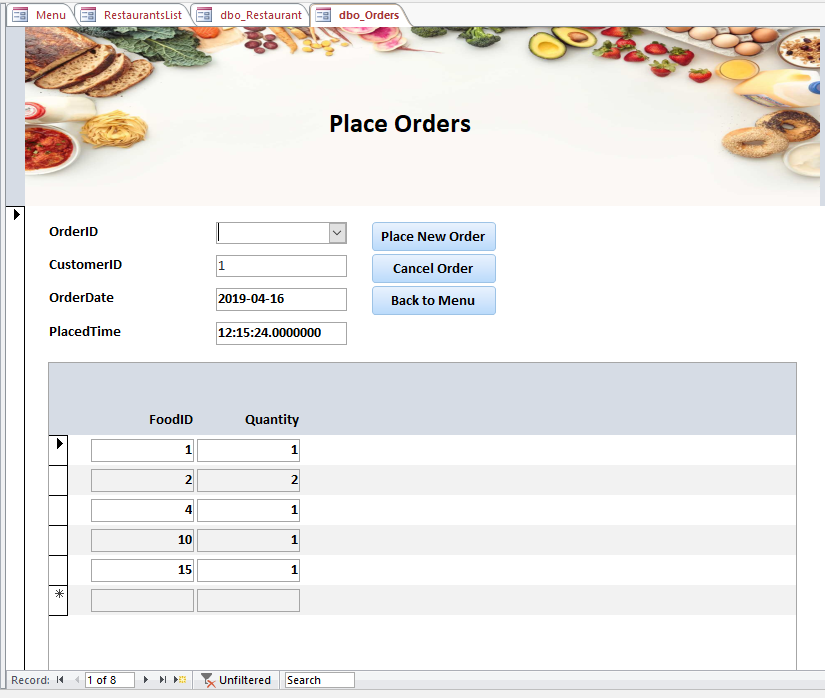


If the user would like to order food sorted by restaurants, he or she can click on the “Restaurant List” button and the following page would show up. There is a “Go to Order” button on the upright corner that users can click on it to next order page.

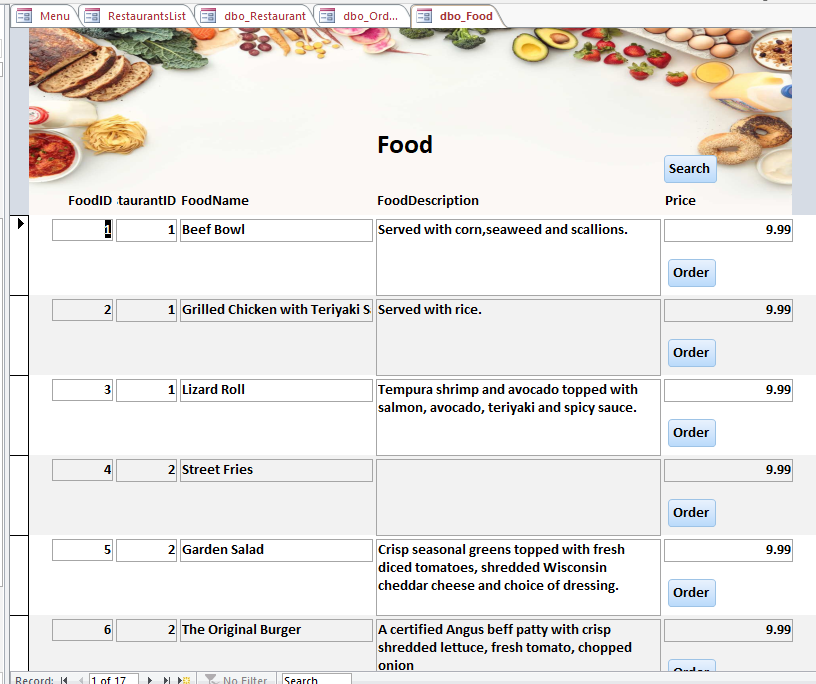


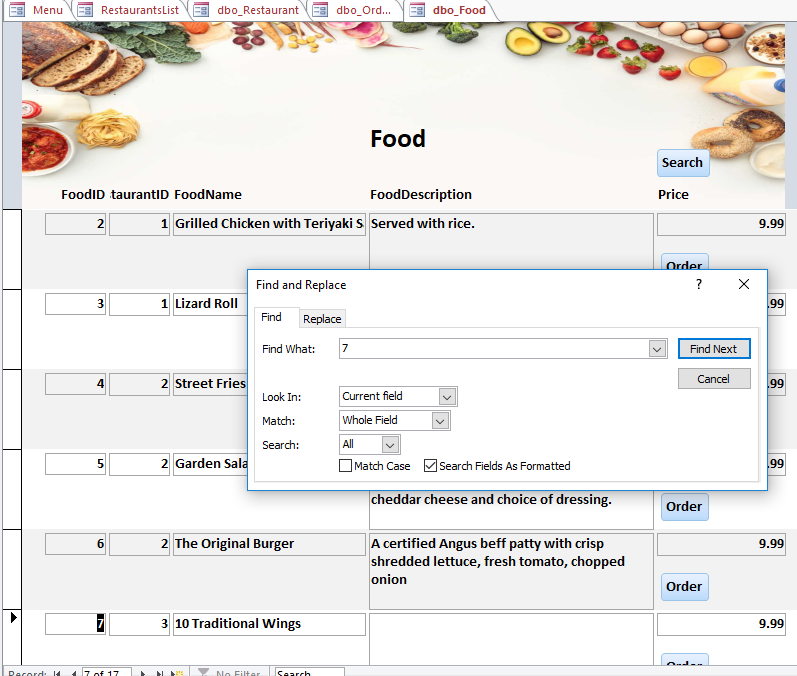
If users click on the “Go to Order” button and the following restaurant page will show up. Users can use the “Search” button to search the restaurant they like by restaurant ID or they can use drop-down list to search restaurant. Also, the users can check the popular restaurants and average payment for each restaurant by clicking the relevant buttons. “Back to Menu” button is for jumping back to menu page and “Order” button is for jumping to the next place order page.

This is an order placing page. Users need to input food ID and quantity of food and click on “Place new order”. After that the order will be placed by this user. To revise the order, the user can click on the “cancel order” which would delete the placed order. Otherwise, back to the main menu by “back to menu” button.

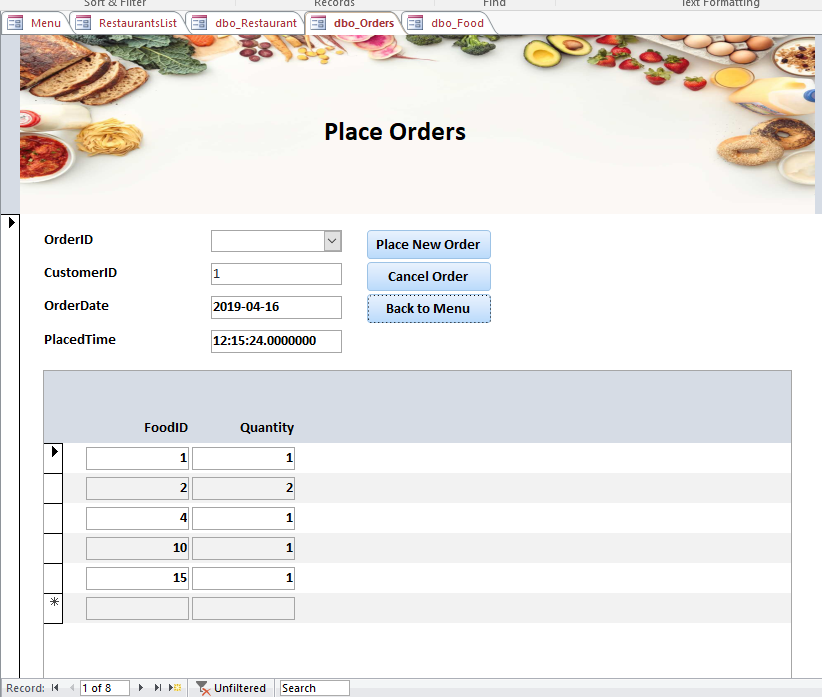


Back to the menu page, and click on “Food” button, the following page will show up. Our system allows customers to order food from different restaurants in one order. Therefore, the user can search whatever food they like no matter they are served by one or more restaurants.

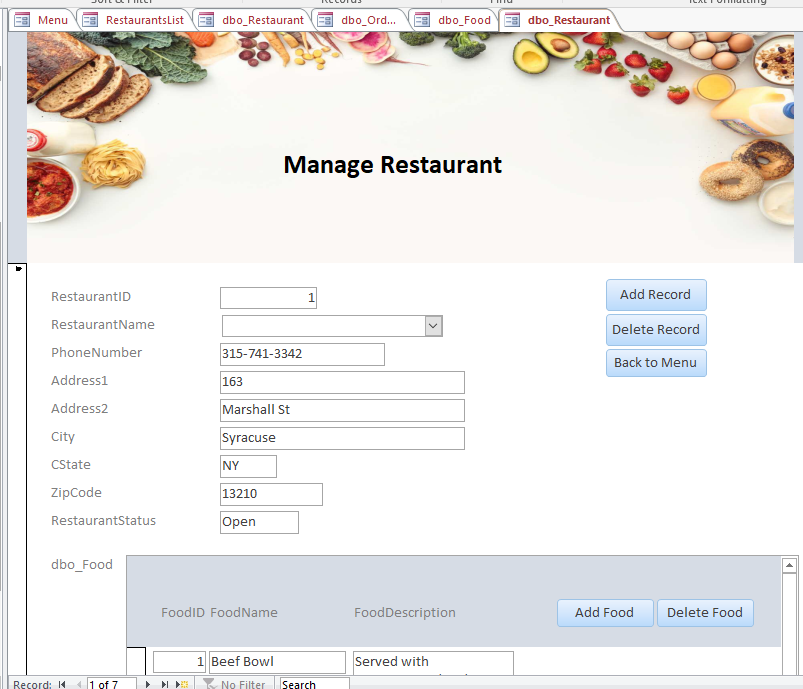


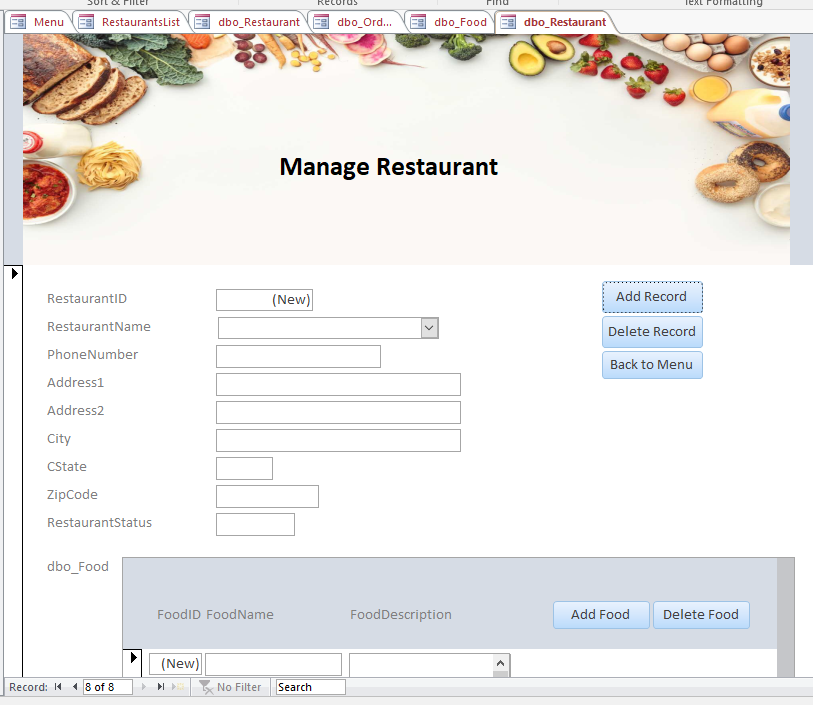
User can search food by clicking on “search” button.

Click on “Order” button to the ordering page. Same way to place order.

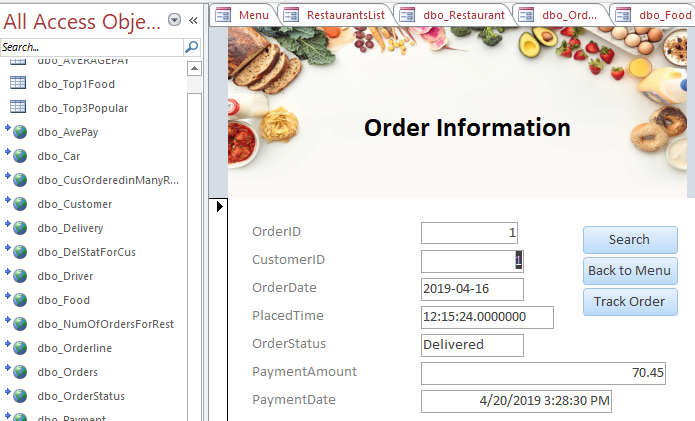


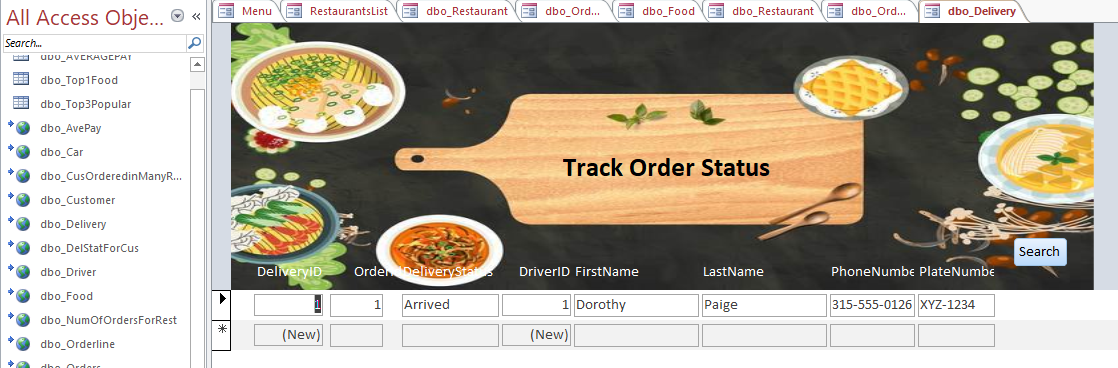
Back to the menu page and click on “Manage Restaurant” button. The following page will show up. This page is for restaurant managers. User can click on the “Add Record” to add a new restaurant to the system. Or they can click on the “Delete record” to delete the restaurant from the system. Similarly, click on “add food” and “delete food” to add or delete a type of food in that restaurant.

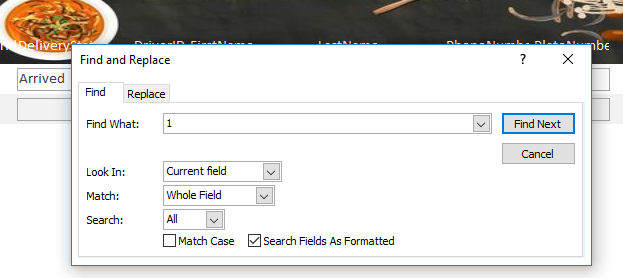




Back to menu again and click on “Order Information” button, the following page will show up. Customers can track the order status by click on “Track order” button. Also, they can search specific order in Order information page and order status in Track Order Status page.







Report

