

Homework 2

Group 2: Ujwal Joshi, Daisy Gonzalez, Fernando Ramirez, Rebecca (Becky) Santos

Presentation Link:

<https://drive.google.com/file/d/1MMeJrQFvzZoPHufMAUqLk5pBCj0qur5q/view?usp=sharing>

Interesting Queries:

- Can be seen on the Reports page (located on the footer)
- Has a few insightful queries such as the following:
 - Wealthiest Restaurants: restaurants with the most revenue

- SQL:

```
SELECT r.restaurantid, r."name", b.balance AS "wealth"
FROM restaurant r JOIN bank b ON r.bankaccountid = b.accountid
ORDER BY b.balance;
```

- Most Expensive Dishes: the most expensive dishes of the restaurant chains

- SQL:

```
SELECT r2.restaurantid, res.name AS "restaurant_name", mi.name AS
"dish_name", mi.itemid AS "dish_id", maxPrice.price
FROM (select r.restaurantid, MAX(r.price) as "price"
      FROM restaurantmenu r
      GROUP BY r.restaurantid) maxPrice, restaurantmenu r2, menuitem
mi, restaurant res
WHERE maxPrice.price = r2.price AND maxPrice.restaurantid =
r2.restaurantid AND r2.menuitemid = mi.itemid AND res.restaurantid =
r2.restaurantid
ORDER BY restaurantid;
```

- Restaurants → Restaurants's Menu Items → Restaurant ID of choice (ids 1 - 10):
the menu items of the Restaurant ID chosen

- SQL:

```
SELECT m.name, r.price, m.type, m.description, m.itemid,
r.restaurantid
FROM restaurantmenu r JOIN menuitem m ON r.menuitemid = m.itemid
WHERE r.restaurantid = ${restaurantID};
```

- Restaurants → Popular Dishes: each restaurant's popular dishes

- SQL:

```
SELECT m."name" AS "dish_name", dishStats.num_orders
FROM (SELECT c.menuitemid, COUNT(c.menuitemid) AS "num_orders"
      FROM cart c JOIN customerorder c2 ON c.orderid = c2.orderid
      WHERE c2.processed = true
      GROUP BY c.menuitemid
      ORDER BY "num_orders" DESC) dishStats JOIN menuitem m ON m.itemid
= dishStats.menuitemid;
```

- Restaurants → Popular Restaurants: best-selling restaurants

- SQL:

```
SELECT r.name AS "restaurant_name", orderStats.num_orders
FROM (SELECT c2.restaurantid, COUNT(c2.restaurantid) AS "num_orders"
      FROM customerorder c JOIN cart c2 ON c.orderid = c2.orderid
      WHERE c.processed = true
      GROUP BY c2.restaurantid
      ORDER BY "num_orders" desc) orderStats JOIN restaurant r ON
r.restaurantid = orderStats.restaurantid;
```

- Search for Customer → Last Name of choice: customer info

- SQL:

```
SELECT email, firstname, lastname, phone, zipcode, membership
FROM customer c
WHERE c.lastname LIKE '${customerLastName}';
```

Initializing the Database:

- Database can be initialized using the “Start DB” button on the Home Page

Web Pages:

- Main Components (present in all web pages):
 - Navigation Bar (from left to right):
 - Home Button: Leads you to the Home Page
 - Menu: Leads you to the Locations Page. However, if you try to click this link before signing in, you will receive a pop-up that prompts you to sign in before being able to order. You will then be led to the Sign In page to sign in
 - Reports: redirects you to the Reports Page
 - Manage Account: Leads you to the Manage Account Page. However, you will only go to the Manage Account Page if you are signed in. If you are not signed in, you will be led to the Sign In Page
 - Register: Leads you to the Register Page
 - Sign In: Leads you to the Sign In Page. When you are already signed in, the Navigation Bar will say the name of the account holder and the words “(Sign Out)” instead, and you will be signed out and led back into the Sign In Page
 - Cart: Leads you to the Order Page. If you add a menu item to your cart, a number will appear on the cart graphic (depending on the number of items in your cart)
 - Footer
 - Copyright that shows the Semester, Academic Year, and Group Number
 - Reports: redirects you to the Reports Page

- Home Page
 - Landing page of the web application
 - Contains a “Start DB” Button that will initialize the database and lead you to the Reports page
- Sign In Page
 - Page that allows you to sign into your account
 - If you do not put a registered email in the “Email” textbox, you will receive the following message: “Please enter a valid email. Email not registered.”
 - If you put a registered email in, but do not include an account pin in the “Account Pin Number” text box, you will receive the following message: “Please enter a valid pin.”
 - If you put a registered email in, but input an incorrect pin number in the “Account Pin Number” text box, you will receive the following message: “Please enter a valid pin. Incorrect account pin.”
 - Once putting a registered email and the valid pin number associated with it, pressing the “Sign In” Button will lead you to the Manage Account Page
- Register Page
 - Page that allows you to register a new account
 - If the boxes are not filled with the proper information, pressing the “Submit” button will result in an error message being displayed, depending on the text box that was not filled correctly
 - If the proper input is filled, pressing the “Submit” button will show you a prompt that says “Congratulations! You successfully created your account, you may now sign in!” and adds the new customer information to the database. Clicking the underlined “sign in!” link leads you to the Sign In page
 - Transaction that adds new customer to the database:

```
BEGIN;
INSERT INTO bank (accountID, balance)
VALUES
('${bankAccountID}', ${balance});

INSERT INTO customer (email, pin, firstName, lastName, phone, zipcode,
membership, bankAccountID)
VALUES
('${email}', ${pin}, '${fname}', '${lname}', ${phone}, ${zip}, 'yes',
'${bankAccountID}');
COMMIT;
```

- Manage Account Page
 - Page that shows the customers' information:
 - Account information includes the following: First Name, Last Name, Zip Code, Phone Number, Email, and Membership Status
 - Payment information includes the following: Card Number, CVV, Card Name, and Expiration Date
 - Contains an "Update Account Information" Button and an "Update Payment Information," where you can update your account and payment information
 - If your payment information is not set, the areas that will display your payment information will instead say "Payment Method Not Set!"
 - Contains a "Delete Account" Button that will allow you to delete your account. Clicking the button will show you a pop-up that says "Confirm Deletion of Account"
 - Navigation SideBar:
 - Order History: Leads you to the Order History Page
- Update Account Page
 - Page that allows you to update your current account information
 - Has the same inputs as the Register Page (First Name, Last Name, Email, Account Pin Number, Phone Number, Zip Code) that you can submit to update the account information
 - If the boxes are not filled with the proper information, pressing the "Submit" button will result in an error message being displayed, depending on the text box that was not filled correctly
 - If the proper input is filled, pressing the "Submit" button will show you a prompt that says "Information updated, please sign in again" and updates the information. Clicking the underlined "sign in again" will lead you to the Sign In page
 - Transaction for updating the account:

```
BEGIN;
UPDATE customer
SET email = '${newEmail}',
    pin = ${pin},
    firstname = '${fname}',
    lastname = '${lname}',
    phone = '${phone}',
    membership = '${membership}',
    zipcode = ${zip}
WHERE email = '${oldEmail}';

UPDATE paymentinformation
SET customeremail = '${newEmail}'
WHERE customeremail = '${oldEmail}';
```

```
COMMIT;
```

- Update Payment Page
 - Page that allows you to update your current payment information
 - Lets you input the following: Payment ID, Card Number, CVV, Card Name, and Expiration Date
 - If the boxes are not filled with the proper information, pressing the “Submit” button will result in an error message being displayed, depending on the text box that was not filled correctly
 - If the proper input is filled, pressing the “Submit” button will show you a prompt that says “Payment information updated!” and updates the information
 - Transaction for updating the payment information:

```
BEGIN;  
  
UPDATE paymentinformation  
SET cardnumber = '${cardNumber}',  
    cvv = ${cvv},  
    cardname = '${cardName}',  
    expiration = '${expiration}'  
WHERE customeremail = '${customerEmail}';  
  
COMMIT;
```

- Order History Page
 - Page that allows the customer to see their full order history
 - Inputting the customer’s email into the “Enter Order Email” text box will show you the customer’s full order history. Inputting this will show the order date and order time
 - Upon inputting the order date and order time into their respective boxes, you can view the following:
 - Receipt Details that show the following:
 - Menu Item
 - Restaurant
 - Price
 - Payment Restaurant Details that show the following:
 - Restaurant ID
 - Restaurant Name
 - Account ID
 - Total Amount Paid
 - Customer Payment Quantity that shows the following:
 - Subtotal
 - Tax

- Tip
- Total

- Locations Page
 - Page that allows you to choose which restaurant you'd like to order from
 - Cannot be accessed until the customer signs in
 - Customers can search by Restaurant ID to find a specific restaurant
 - Restaurant information contains the following: Restaurant ID, Restaurant Name, Phone Number, Street Name, Revenue
 - Pressing the "Order Here!" button leads you to the Menu Page for the chosen restaurant
- Menu Page
 - Page that allows you to start selecting menu items
 - Cannot be accessed until the customer signs in and chooses a location to order from
 - Customer can select from three options on this page: "Meals," "Sides," and "Drinks"
- Meals/Sides/Drinks Page:
 - Three different pages that shows you the available menu items, depending on which page you selected as well as the location you selected
 - Customers are shown the name of the menu item, a small description of the item, and its price
 - Pressing the "Add to Cart" button adds to your cart (which is accessible in the Order Page)
- Order Page (Cart)
 - Page that allows you to see your current shopping cart
 - Must be signed in before viewing the cart.
 - Displays all menu items that the customer has added to their order
 - Transaction for adding to cart:

```
BEGIN;

INSERT INTO customerorder (customerEmail, orderDate, orderTime, tip,
processed)
SELECT '${email}', '${date}', '${time}', 0.00, false
WHERE NOT EXISTS (
    SELECT 1
    FROM customerorder
    WHERE customeremail = '${email}' and processed = false
);
```

```

WITH userOrderID AS (
    SELECT orderid AS userorder
    FROM customerorder c
    WHERE c.customeremail = '${email}' and c.processed = false
) INSERT INTO cart (orderid, menuitemid, restaurantid)
    SELECT userorder, ${dishID}, ${restaurantID}
    FROM userOrderID;

COMMIT;

```

- Customers have the ability to delete any item that they have added to their cart
- Transaction for deleting from cart:

```

BEGIN;

WITH userOrderID AS (
    SELECT orderid AS userorder
    FROM customerorder c
    WHERE c.customeremail = '${email}' and c.processed = false
) DELETE FROM cart c2
USING (
    SELECT c3.cartitemid
    FROM cart c3
    WHERE c3.orderid = (select userorder from userOrderID)
        AND c3.restaurantid = ${restaurantID}
        AND c3.menuitemid = ${dishID}
    ORDER BY c3.cartitemid
    LIMIT 1
) c4
WHERE c2.cartitemid = c4.cartitemid;

COMMIT;

```

- Subtotal, Tax, and Total are all displayed for the customer to view
- Pickup Location is displayed for the customer to view
- Pressing the “Checkout” button leads you to the Checkout Page
- Checkout Page
 - Page that allows you to check any final details of your order and provide payment
 - Provides Order Details which include the Subtotal, Tax, Tip, and Total
 - Allows the customer to provide a tip between the following values: \$2.00, \$3.00, \$5.00, and \$0.00, which will be included in the Total
 - Transaction for adding tip:

```

BEGIN;
UPDATE customerorder
SET tip = ${tip}

```

```
WHERE customeremail = '${email}' AND processed = false;
COMMIT;
```

- Allows the customer to provide payment information
- Shows more details for unprocessed orders such as the following:
 - Money owed to the restaurant that is being ordered from
 - Bank accounts of the customer that is ordering and the restaurant that is being ordered from
- Pressing the “Checkout” button will “process” the order. You will receive a prompt that says “Payment Successfully Processed!”
- Transaction for updating bank balance:

```
BEGIN;

UPDATE bank
SET balance = balance - ${customerTotal}
WHERE accountid IN (
    SELECT b.accountid
    FROM customer c JOIN bank b ON c.bankaccountid = b.accountid
    JOIN customerorder c2 ON c2.customeremail = c.email
    WHERE c2.customeremail = '${email}' AND c2.processed = false
);

UPDATE bank
SET balance = balance + total_due
FROM (
    SELECT r2.restaurantid, r2."name" AS "restaurant", b.accountid,
    rbill.total_due
    FROM (SELECT r.restaurantid, sum(r.price) AS "total_due"
    FROM customerorder c JOIN cart c2 ON c.orderid = c2.orderid
    JOIN restaurantmenu r ON c2.menuitemid = r.menuitemid AND
    c2.restaurantid = r.restaurantid
    WHERE c.processed = false AND c.customeremail = '${email}'
    GROUP BY r.restaurantid) rbill JOIN restaurant r2 ON
    rbill.restaurantid = r2.restaurantid JOIN bank b ON b.accountid =
    r2.bankaccountid
    ) AS owed
WHERE bank.accountid = owed.accountid;

UPDATE customerorder
SET processed = true,
    orderdate = '${orderDate}',
    ordertime = '${orderTime}'
WHERE customeremail = '${email}' AND processed = false;

COMMIT;
```

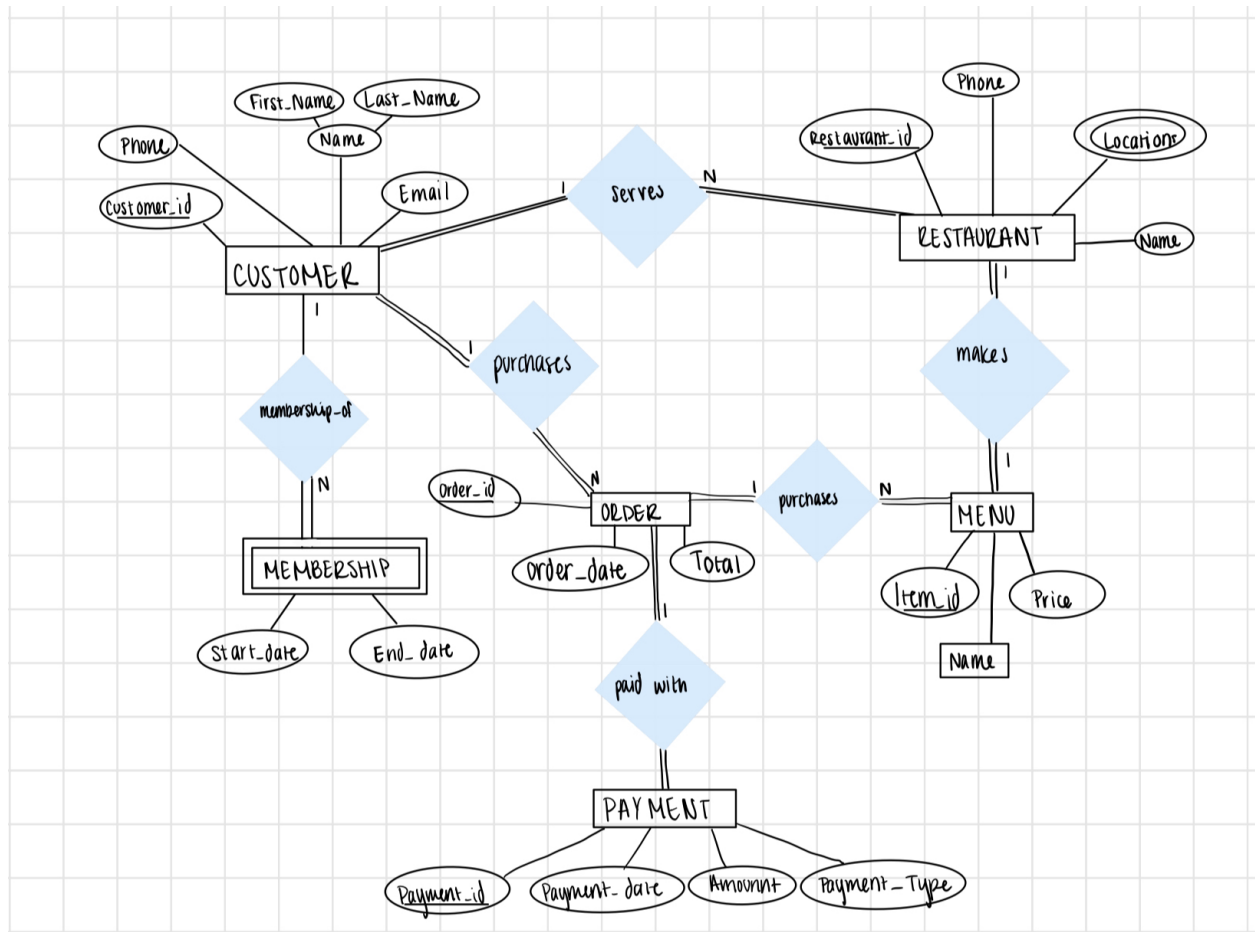

- Reports Page
 - Allows you to see the output of table data based off of joins and has a few other interesting queries
 - Has five buttons that will display the following:
 - Restaurants:
 - SQL:

```
SELECT * FROM restaurant r
ORDER BY r.restaurantid;
```
 - Customers:
 - SQL:

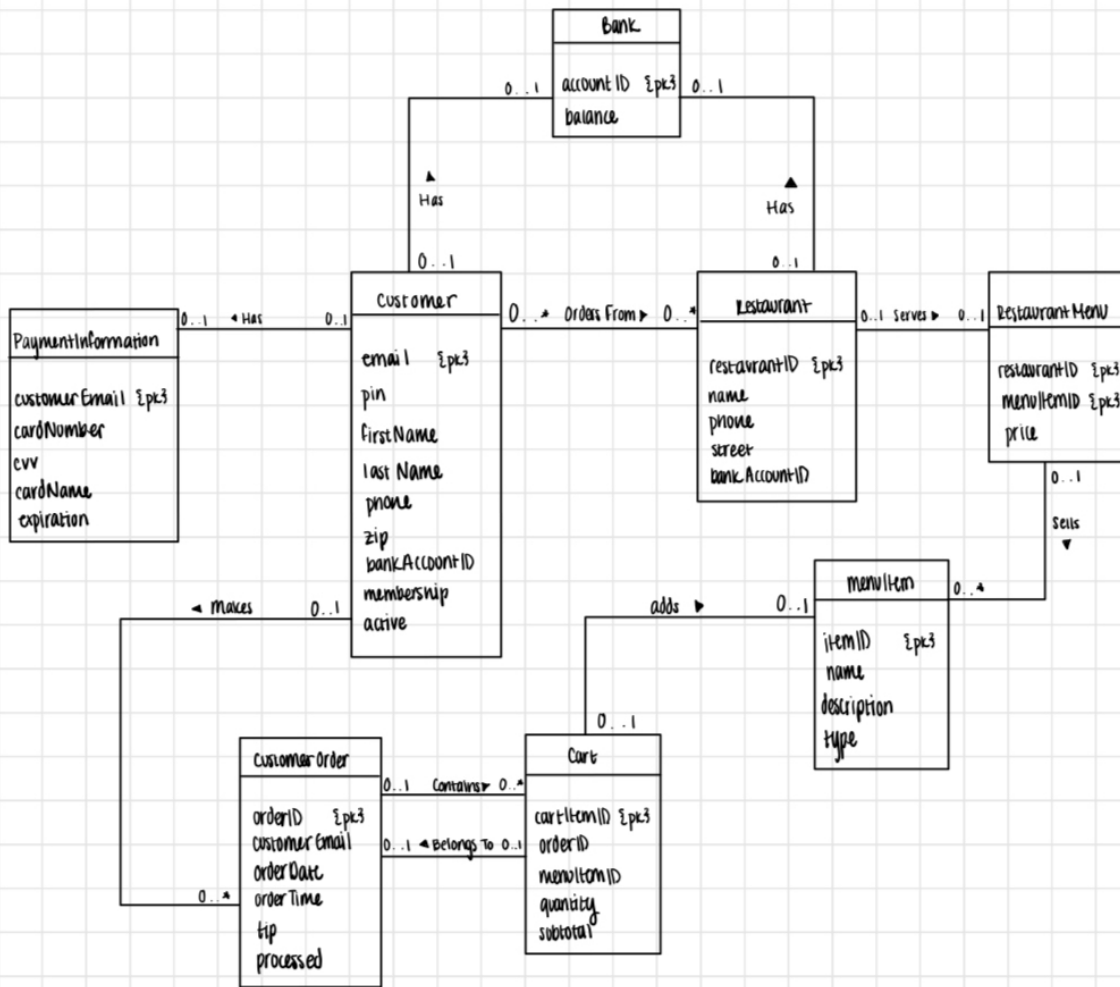
```
SELECT * FROM customer;
```
 - Gives you the option to add a new customer by clicking the “Register!” link
 - Has an “Order as customer” button that leads you to the Manage Account page
 - Orders: leads you to the Order History Page
 - Most Expensive Dishes:
 - SQL: Listed under “Interesting Queries” section of the pdf
 - Restaurants by Wealth:
 - SQL: Listed under “Interesting Queries” section of the pdf
 - Search for Customer:
 - SQL: Listed under “Interesting Queries” section of the pdf

ER Diagrams:

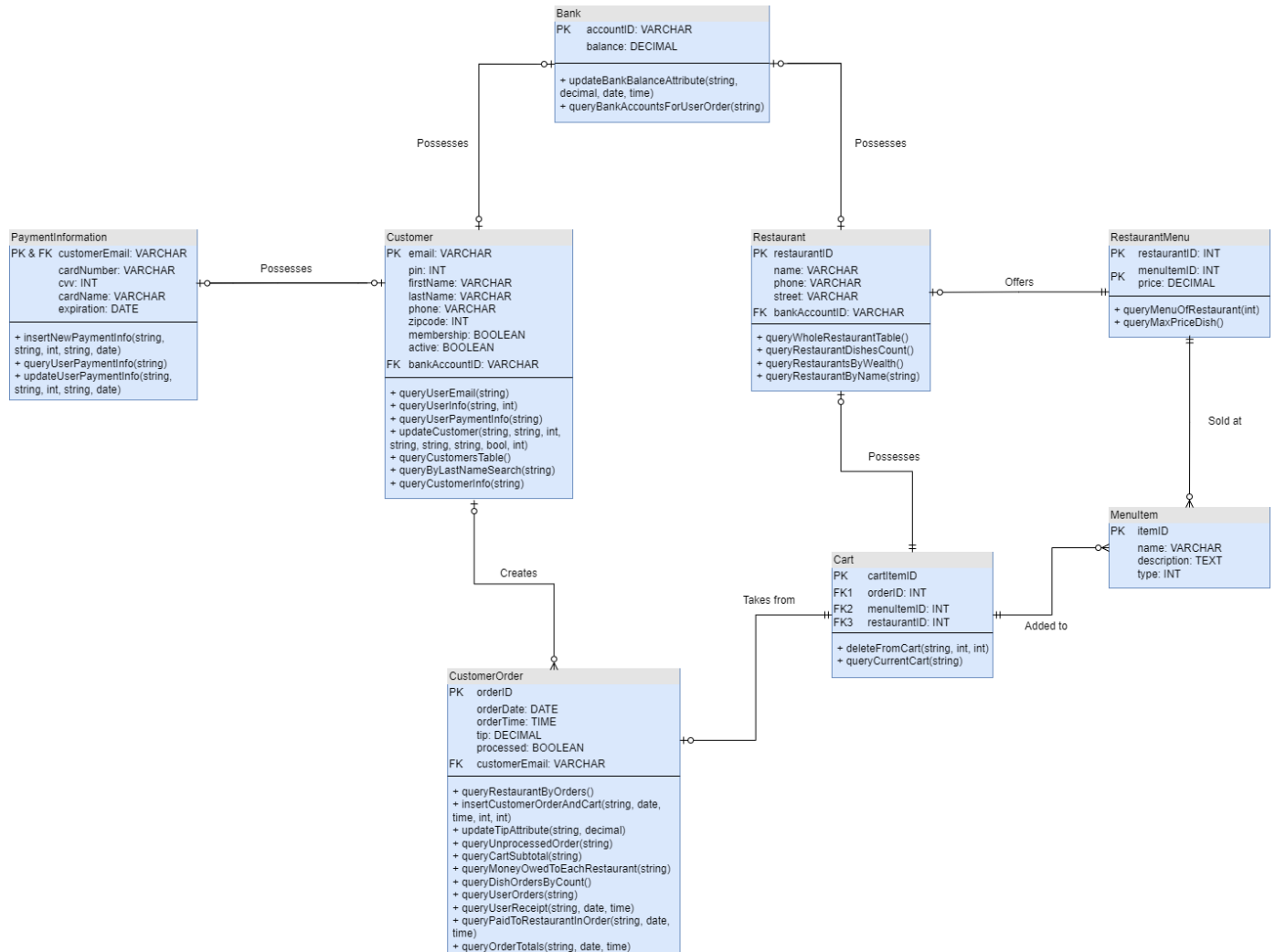
Preliminary:



Intermediate:



Final (Using draw.io):



- File for final Diagram:

https://drive.google.com/file/d/1Lp1-fmH1UqZDDjN8R0XNu9_yBiUQuNBD/view?usp=sharing

References:

- <https://www.chipotle.com/home>
- <https://legacy.reactjs.org/docs/getting-started.html>
- <https://blog.logrocket.com/how-to-use-axios-post-requests/>