This database has been redesigned to be what I hope is more efficient and useful. It is a relational database with a single one-to-many relationship and a single one-to-one relationship. So far in my testing I can create the views I think will be required to present data to the users along with the ability to update and delete as needed.

The intention is to create a "temporary" customer and cart until the user logs in or saves the cart. At which point the three tables: customer, cart and cartItems will be considered permanent. If the user quits before logging in or saving, I would like to delete the temporary information to save database space because the customer ID will be disassociated with anyone at that point.

The main user interface will be populated using the products database. This will be json data associated with an add-to-cart button that will be used to create a temporary cart associated with all the temporary cartItems. There will be no actual temporary cart or temporary customer table, but the information will be considered temporary until there is a way to associate the user to the customer table.

The cart will be considered an official order after the payment confirmation has been made. When this happens the state of the cart, which is a boolean/tinyint value, will be updated. In this way I intend to be able to pull all order or saved cart data if needed.

In the end the product updating needs to be done by the employees of the store so I will need to create another interface based on update api's. For now, the user interface will be my focus.