Database Design Document

My E-commerce application will use Django/PostgreSQL (Relational Database) to support the functionality of the web application day-to-day operation including user management, product and inventory, shopping cart and payment management. For example, for e-commerce will have customer table and orders table which will have relationship. This will make application much more portable and scalable.

The MVP database design I will include would be core tables:

Customer table will be used to manage different types of users including admins and customers. It can also be used to relate the product creators (from the admin panel) and customer orders placed on the website. Users can track their own orders and track the status.

Customers Table - holds:

- CustomerID
- CustomerFirstName
- CustomerLastName
- CustomerStreetAddress
- CustomerCityAddress
- CustomerStateAddress
- CustomerZipCode
- CustomerCellphone
- CustomerEmail
- CustomerPassword

Category Table – hold:

- CategoryName
- CategoryDescription
- CategoryPicture
- Category Status/Active

Product table will use the column quantity to track the stock available in the product inventory to keep the design simple. It might be required to specify the quantity by several columns to cover a wide range of products.

Products Table – holds:

- ProductName
- ProductDescription
- ProductSize
- ProductColor
- ProductPrice
- ProductDiscount
- ProductPicture

Orders Detail – holds:

- OrderDetailPrice
- OrderDetailQuantity
- OrderDetailDiscount
- OrderDetailTotal
- OrderDetailSize
- OrderDetailColor
- OrderDetailShipdate

Orders Table – holds:

- OrderNumber
- OrderSalesTax
- OrderDeleted
- OrderPaid
- OrderPaymentDate
- OrderTransactionStatus

Credit Card Table – holds:

- CreditCardID
- CustomerID
- OrderID
- FirstName
- LastName
- BillingStreetAddress
- BillingCityAddress
- BillingStateAddress
- BillingZipCode
- BillingCellphone
- BillingEmail
- CreditCardNumber

Payment table will be used for transaction table to track the order payments made by the buyer and for bookkeeping. I can also use the same table to record the partial or full refund of the order.

Payment Table – holds:

- PaymentID

- OrderID
- PaymentAmount
- PaymentStatus

❖ Add Shopping Cart Database in details

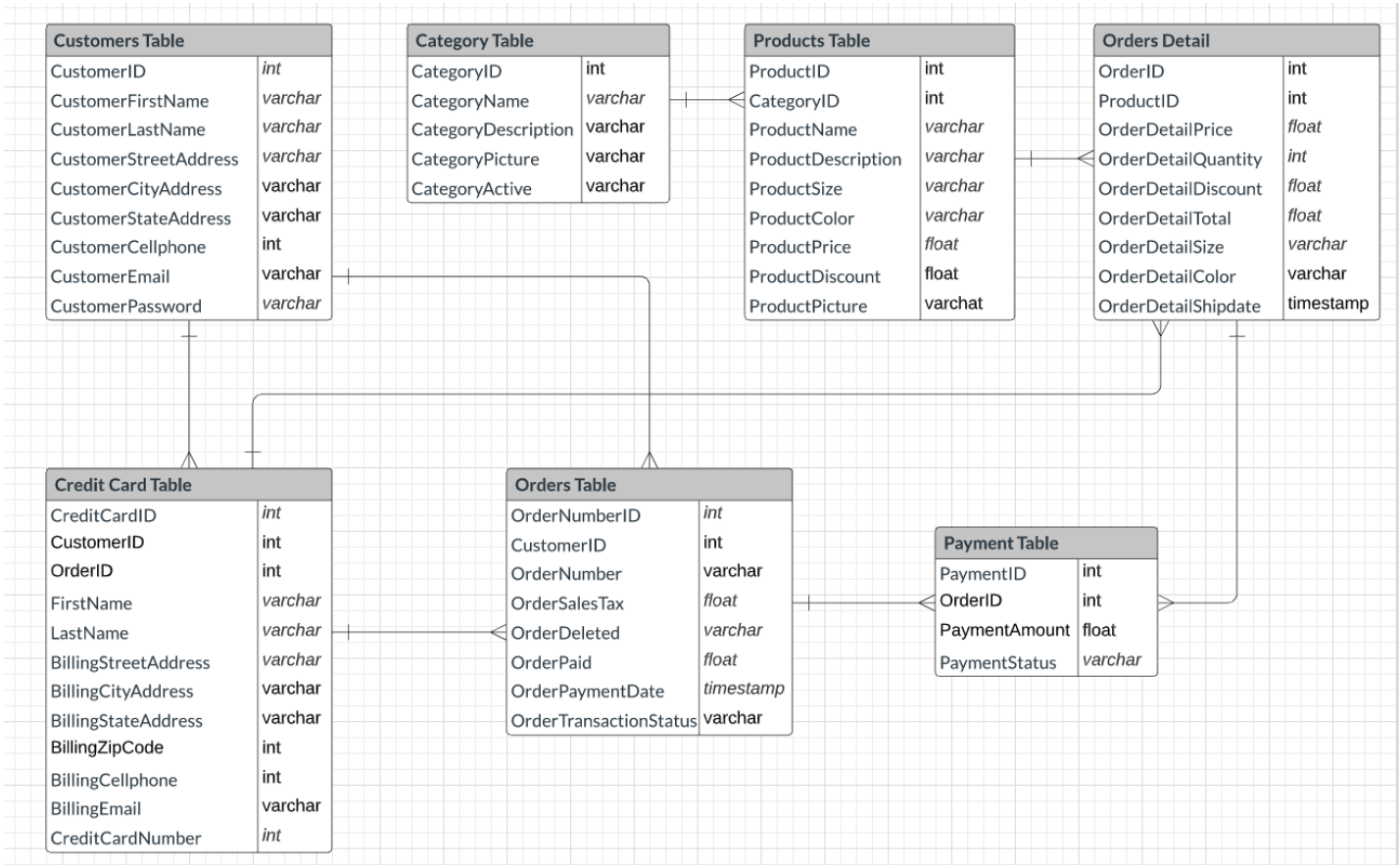
Shopping cart database will provide the tables to manage the virtual carts to store the user selection before creating the actual order. The same can be referred to create the order on payment success.

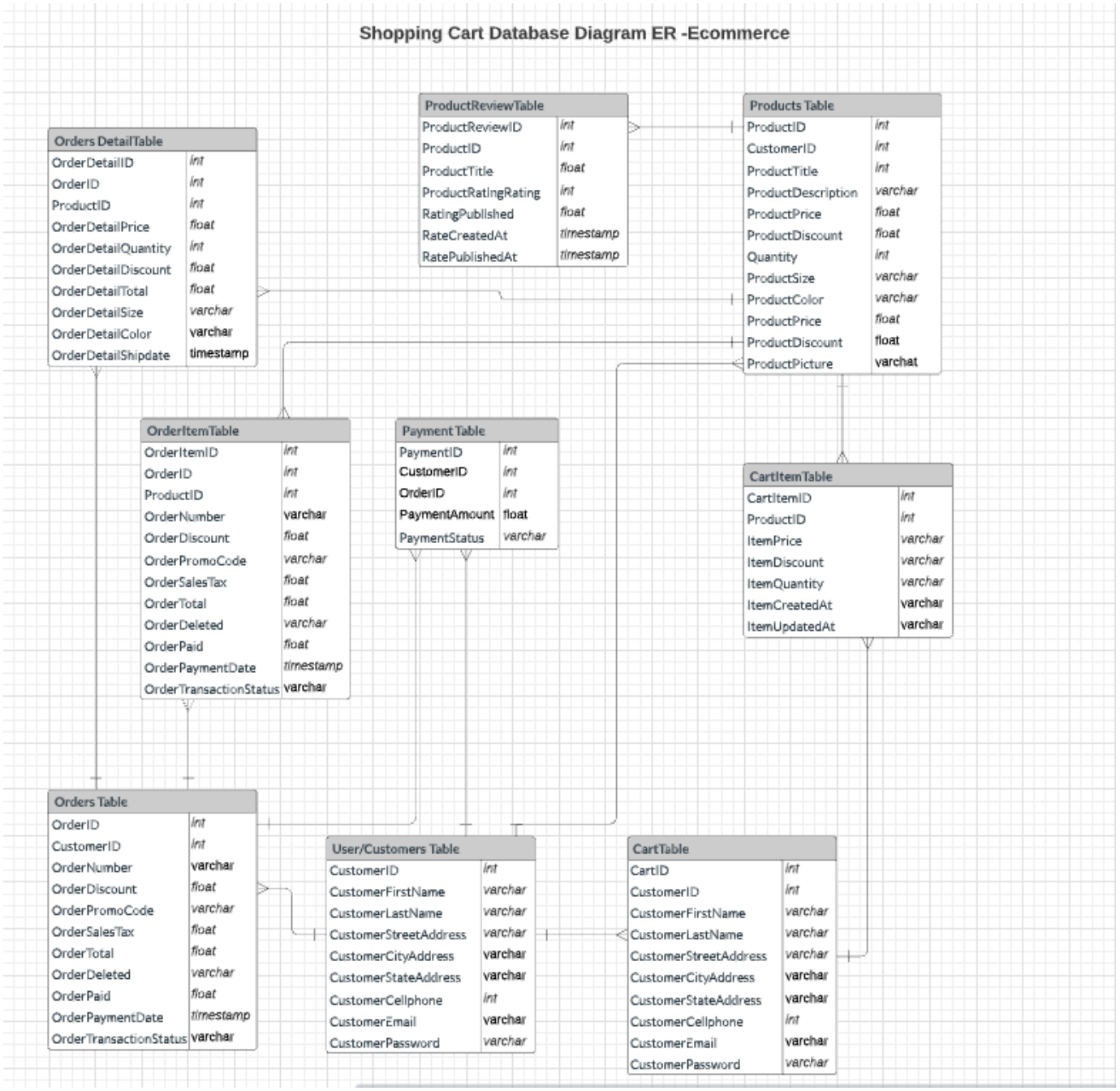
CartTable – holds:

- CartID
- CustomerID
- CustomerFirstName
- CustomerLastName
- CustomerStreetAddress
- CustomerCityAddress
- CustomerStateAddress
- CustomerZipCode

CartItemTable – holds:

- CartItemID
- ProductID
- ItemPrice
- ItemDiscount
- ItemQuantity


Customers Table
CustomerID int
CustomerFirstName varchar
CustomerLastName varchar
CustomerStreetAddress varchar
CustomerCityAddress varchar
CustomerStateAddress varchar
CustomerCellphone int
CustomerEmail varchar
CustomerPassword varchar
Category Table
CategoryID int
CategoryName varchar
CategoryDescription varchar
CategoryPicture varchar
CategoryActive varchar
Products Table
ProductID int
CategoryID int
ProductName varchar
ProductDescription varchar
ProductSize varchar
ProductColor varchar
ProductPrice float
ProductDiscount float
ProductPicture varchat
Orders Detail
OrderID int
ProductID int
OrderDetailPrice float
OrderDetailQuantity int
OrderDetailDiscount float
OrderDetailTotal float
OrderDetailSize varchar
OrderDetailColor varchar
OrderDetailShipdate timestamp
Credit Card Table
CreditCardID int
CustomerID int
OrderID int
FirstName varchar
LastName varchar
BillingStreetAddress varchar
BillingCityAddress varchar
BillingStateAddress varchar
BillingZipCode int
BillingCellphone int
BillingEmail varchar
CreditCardNumber int
Orders Table
OrderNumberID int
CustomerID int
OrderNumber varchar
OrderSalesTax float
OrderDeleted varchar
OrderPaid float
OrderPaymentDate timestamp
OrderTransactionStatus varchar
Payment Table
PaymentID int
OrderID int
PaymentAmount float
PaymentStatus varchar


Shopping Cart Database Diagram ER -Ecommerce
ProductReviewTable
ProductReviewID int
ProductID int
ProductTitle float
ProductRatingRating int
RatingPublished float
RateCreatedAt timestamp
RatePublishedAt timestamp
Products Table
ProductID int
CustomerID int
ProductTitle int
ProductDescription varchar
ProductPrice float
ProductDiscount float
Quantity int
ProductSize varchar
ProductColor varchar
ProductPrice float
ProductDiscount float
ProductPicture varchat
Orders DetailTable
OrderDetailID int
OrderID int
ProductID int
OrderDetailPrice float
OrderDetailQuantity int
OrderDetailDiscount float
OrderDetailTotal float
OrderDetailSize varchar
OrderDetailColor varchar
OrderDetailShipdate timestamp
OrderItemTable
OrderItemID int
OrderID int
ProductID int
OrderNumber varchar
OrderDiscount float
OrderPromoCode varchar
OrderSalesTax float
OrderTotal float
OrderDeleted varchar
OrderPaid float
OrderPaymentDate timestamp
OrderTransactionStatus varchar
Payment Table
PaymentID int
CustomerID int
OrderID int
PaymentAmount float
PaymentStatus varchar
CartItemTable
CartItemID int
ProductID int
ItemPrice varchar
ItemDiscount varchar
ItemQuantity varchar
ItemCreatedAt varchar
ItemUpdatedAt varchar
Orders Table
OrderID int
CustomerID int
OrderNumber varchar
OrderDiscount float
OrderPromoCode varchar
OrderSalesTax float
OrderTotal float
OrderDeleted varchar
OrderPaid float
OrderPaymentDate timestamp
OrderTransactionStatus varchar
User/Customers Table
CustomerID int
CustomerFirstName varchar
CustomerLastName varchar
CustomerStreetAddress varchar
CustomerCityAddress varchar
CustomerStateAddress varchar
CustomerCellphone int
CustomerEmail varchar
CustomerPassword varchar
CartTable
CartID int
CustomerID int
CustomerFirstName varchar
CustomerLastName varchar
CustomerStreetAddress varchar
CustomerCityAddress varchar
CustomerStateAddress varchar
CustomerCellphone int
CustomerEmail varchar
CustomerPassword varchar