Vanessa Aguilar z1846838
Assignment 4
CSCI 466 Section 1

† = Foreign Key
___ = Primary Keys

Step 1 a: Handle The entities

Schema of new relations

CustomerAccount(customerID, fullname, address, balance)
PersonalBankAccount(bankID, bankName, routingNum, balance, withdraw, deposit, currentdate)
Orders(OrderID, typeofOrder, exchangeRequestAmount)
OrderStatus(orderstatusID, orderType)
StockExchange(exchangeID, customerOrders)
StockTransaction(TransactionID, exchangedStock, exchangedWith, recipient, currentDate)
Stock(stockSymbol, nameOfCompany, share)

Step 2a: Handle Relationships

One-to-one Relationships

To not break 1NF I will represent one-to-one relationships by putting a new foreign key into the entity of the other side. Side picked does not matter.

- Relationship between Order and OrderStatus. Adding OrderStatus foreign key into entity LimitOrder to represent the dependency

  Order(OrderID, exchangeRequestAmount, OrderType, orderStatusID†)

Step 2b: Handle Relationships

One-to-many Relationships

To not break 1NF I will represent one-to-many relationships by putting a new foreign key into the entity that carries the many cardinality.

- Relationship between **CustomerAccount** entity and **Order** entity

  Orders(OrderID, OrderType, exchangeRequestAmount, customerID†)

- Relationship between **CustomerAccount** entity and **StockTransaction** entity
  StockTransaction(TransactionID, exchangedStock, exchangedWith, recipient, currentDate, customerID†)

- Relationship between **Order** entity and **StockExchange** entity

  StockExchange(exchangeID, exchangeOrders, OrderID†)

Step 2c: Handle Relationships

Binary many-to-many Relationship

The many-to-many relationship requires a new relation. Its foreign key will be the concatenation of the primary keys of each of the entity relations, which will be used as foreign keys to the corresponding tables. Any intersection data is put into this new relation as a non-prime attribute.

- Relationship between **CustomerAccount** and **PersonalBankAccount** entity
  Intersecting Data = Amount
  Account(customerID†, bankID†, Amount)

  customerID†: Foreign key, home relation CustomerAccount
  bankID†: Foreign key, home relation PersonalBankAccount

- Relationship between **StockExchange** and **Stock** entity

Market(exchangeID†, StockSymbol†)

- Relationship between **StockExchange** and **StockTransaction** entity
  Exchange(exchangeID†, TransactionID†)

Final Results:

CustomerAccount(customerID, fullName, address, balance)
PersonalBankAccount(bankID, bankName, routingNum, balance, withdraw, deposit, currentdate)
Orders(OrderID, exchangeRequestAmount, OrderType, customerID† ,orderStatusID†)
OrderStatus(orderstatusID, orderType)
Stock(stockSymbol, nameOfCompany, share)
StockTransaction(TransactionID, exchangedStock, exchangedWith, recipient, currentDate, customerID†)
StockExchange(exchangeID, exchangeOrders, OrderID†)
Account(customerID†, bankID†, Amount)
Market(exchangeID†, StockSymbol†)
Exchange(exchangeID†, TransactionID†)