



- For schema sale, I add a attribute productID, which can deal with the relation problem between Sale and Product. (Assume the deposit and payment is calculated by percent. So, the deposit and payment only have dependency with sid rather than sid and deposit determine the payment.)
- For schema Product, there is no dependency between SKU\* and ProductName, price and currency. But, productID have dependencies between ProductName, Price and Currency. (Assume products which have same name have other differences. Such as size, etc.)
- For schema CreditCard, I split it into two schemas. Because cID->number, number->{expDate, ccv, cardholder, billingZipCode}. It does not obey 3NF. (Assume one card number corresponds to one billingZipCode rather than one cardholder corresponds to one billingZipCode)

### Functional Dependencies

**Sale:** sid->productID, sid->date, sid->delNotes, sid->deposit, sid->payment

**CreditCardInfo:** cid->number

**CreditCardDetail:**

number->ccv, number->expDate, number->cardholder, number-> billingZipCode

**Product:**

productID->productName, productID->Price, productID->currency, productID->SKU

**Stock:** SKU->quantityInStock

\*: SKU, Stock Keeping Unit <https://www.thebalancesmb.com/what-is-a-sku-in-retail-terms-2890158> So, it only have dependency between SKU and quantity in stock.