Customer-to-Customer Online Electronics Sales Website Database System

Group 27

Zhong Guan, Runze Zhang, Yining Zhou

1. Requirement Analysis

Introduction

This application is designed to provide an online platform for users to purchase and sell their electronics. Sellers can open several online stores to sell their products and buyers can buy more than one products from different stores. After online payment, the purchased products will be delivered to the address where the buyer provides. We limit our sphere of interest to electronics sales website and consider customer-to-customer E-Commerce application.

Database Description

Entities and their attributes

The following entities will be stored in tables of a relational database:

User: A user is someone who has registered an account for this website. A user could be either buyer or seller and is associated with several attributes including name, phone number and email address. It is identified by a unique primary key called userid.

Buyer: A buyer is someone who will buy something on the website and it is a subcategory of the user entity. It inherits all the features of a user.

Seller: A seller is someone who has opened a store and on the website and it is a subcategory of the user entity. It inherits all the features of a user.

Bank Card: A bank card is either a credit card or a debit card depends on whether is used for buyers to pay or is used for sellers to receive the payment. The database records which bank the card belongs to, the expiry date of the card, and the card number which is also the primary key of this entity.

Credit Card: A credit card is a subcategory of the more general bank card entity. it inherits all the features of a bank card and supports its own attributes organization.

Debit Card: A debit card is also a subcategory of the bank card entity. It also inherits all the features of a bank card.

Store: A store is an online shop managed by a seller that sells appliance products. The database keeps track of when the store founded, the name of a store, the address where the store located, and the customer reviews. It can be identified by its unique primary key named sid.

Product: A product is an item that is going to be sold by a seller. A product is recorded in the database with its name, price, which category it belongs to, colour, model number, inventory, and the customer review. A product can be identified by its primary key called pid.

Order item: An order item is a product that a buyer has already ordered but without payment. The database keeps track of the time and the price when the item has been ordered. An order item can be identified by its order number.

Order: An order is a list showing the products that the buyer has already ordered. An order contains an order number, the time when the order is created, the payment status showing whether the buyer has paid or not, and the total amount. An order can be identified by its primary key called oid.

Address: An address is a location to which the purchased products are delivered. It is provided by the buyer and each of addresses has its own city, street address, province, postal code, and the contact number. Each of addresses can be identified by a primary key called addrid.

Relationships

Credit Card Ownership: A credit card is owned by a buyer. This is a one-to-many relationship because a buyer can own multiple credit cards while one credit card can only be owned by one buyer.

Debit Card Ownership: A debit card is owned by a seller. This is a one-to-many relationship because a seller can own multiple debit cards while one debit card can only be owned by one seller.

Manage: A store is managed by a seller. This is a one-to-many relationship since a seller can manage multiple stores, but one store can only be managed by one seller.

Sell: A product is sold by one store. This is a one-to-many relationship since a product can only be sold in one store, but a store can sell multiple different products.

Save to shopping cart: A buyer saves a product to shopping cart. This is a many-to-many relationship because a buyer can order more than one product item or even do not order product, and a product can be ordered by multiple buyers. This relationship also has attributes including the time when the product be ordered and the quantity of the product that the buyer ordered.

Reflection: An order item corresponds to a product. This is a one-to-many relationship since a product can be multiple order items in different orders, but an order item can only correspond to one product.

Contain: An order item is contained in an order. This is a one-to-many relationship since an order item should belong to only one order while one order can contain multiple order items. This contain relationship has an attribute indicating the quantity of the order item in an order.

Deliver to: An order is delivered to an address. This is a one-to-many relationship because an order can only be delivered to one address location while one address can have multiple orders.

Payment: An order is paid by a credit card and a debit card receive payment from an order. This is a ternary one-many-many relationship. The reason is that a credit card can pay for multiple orders while an order can only be paid by one credit card and a debit card can receive multiple payments from different orders while one order can only give the payment to one debit card. This payment relationship has an attribute indicating the time when the payment is settled.

Address Ownership: An address is owned by a buyer. This is a one-to-many relationship since one buyer can have multiple different addresses, but one address can only belong to one buyer.

Application Description

Overview

The database is designed to provide support for a Customer-to-Customer(C2C) electronic device sales website. The website allows both buyer and seller to register as a user, and seller can register their own online stores, which will be allowed to sell different kinds of electronic devices through the Internet and express delivery. The database is build based on the basic function of the website including querying products, adding to the shopping cart, and performing payment, etc.

Functions

When the buyers access to the website and type in the key words of the product, the website ought to provide the products information through the retrieval of the product name or other information among the entity *Product*. The electronic product has their product name, model number and colour, which will also be provided to buyers. When buyers see the products from different *Stores* they want to buy, they can add them to the shopping cart. In this situation, the relationship *Saving to shopping Cart* between the buyer and the product are built. Buyers can add products to the shopping cart for many times and the quantity of the product can also be selected.

After buyers finish selecting products, they can select to pay for the products. At this moment, some *Order Items* are created. These *Order Items* reflects original product and they will record the price and the product id at the time when they are created. The *Order Item* is used to record the statement of the product of the order, to make sure that even the price of the product has been changed, there will still be a cache of the product for both buyers and sellers to review. The *Order Item* cannot be modified after they are created.

When the order is created, a *Payment* is required. It is a relationship between the buyer's credit card and the seller's debit card, and the order. The order will provide the total amount to the *Payment* and the *Payment* will give a feedback to the order which implies whether the payment is performed successfully or not. After successful paid, the money will be deducted from the credit card and transferred to the account of the website. The products paid will be delivered to the address selected by the buyer afterwards. After the buyer receives the product, the payment of goods will be transferred to the debit card of the seller's.

This database system is not a complete website selling system. It must work with the system in the sales website, like working with the banking system. As mentioned, its function is to provide data for the website system to finish the query, create and delete operation. The total amount in the *Order* also requires some summation method in the website system.

2. Relations

Entities:

User (<u>userid</u>, name, phoneNum)

buyer (<u>userid)</u>

Seller (userid)

Store (sid, name, startTime, customerGrade, streetaddr, city, province)

Product (pid, name, type, inventory, price, colour, customerReview, modelNumber)

Order Item (itemid, price, creationTime)

Order (orderNumber, creationTime, PaymentStatus, totalAmount)

Address (addrid, name, city, postalCode, streetaddr, province, contactPhoneNumber)

Bank Card (cardNumber, bank, expirydate)

Credit Card (cardNumber, organization)

Debit Card (cardNumber)

***Relationships*:**

Credit Card Ownership (userid, cardnum) (userid ref Customer, cardnum ref CreditCard)

Debit Card Ownership (userid, cardnum) (userid ref Seller, cardnum ref DebitCard)

Manage (userid, sid) (userid ref Seller, sid ref Store)

Sell (sid,pid) (sid ref Sotre, pid ref Product)

Reflection (itemid, pid) (itemid ref Order Item, pid ref Product)

Save to Shopping Cart (userid, pid, quantity, addtime) (userid ref Buyer, pid ref Product)

Contain (oid, itemid, quantity) (oid ref Order, itemid ref Order Item)

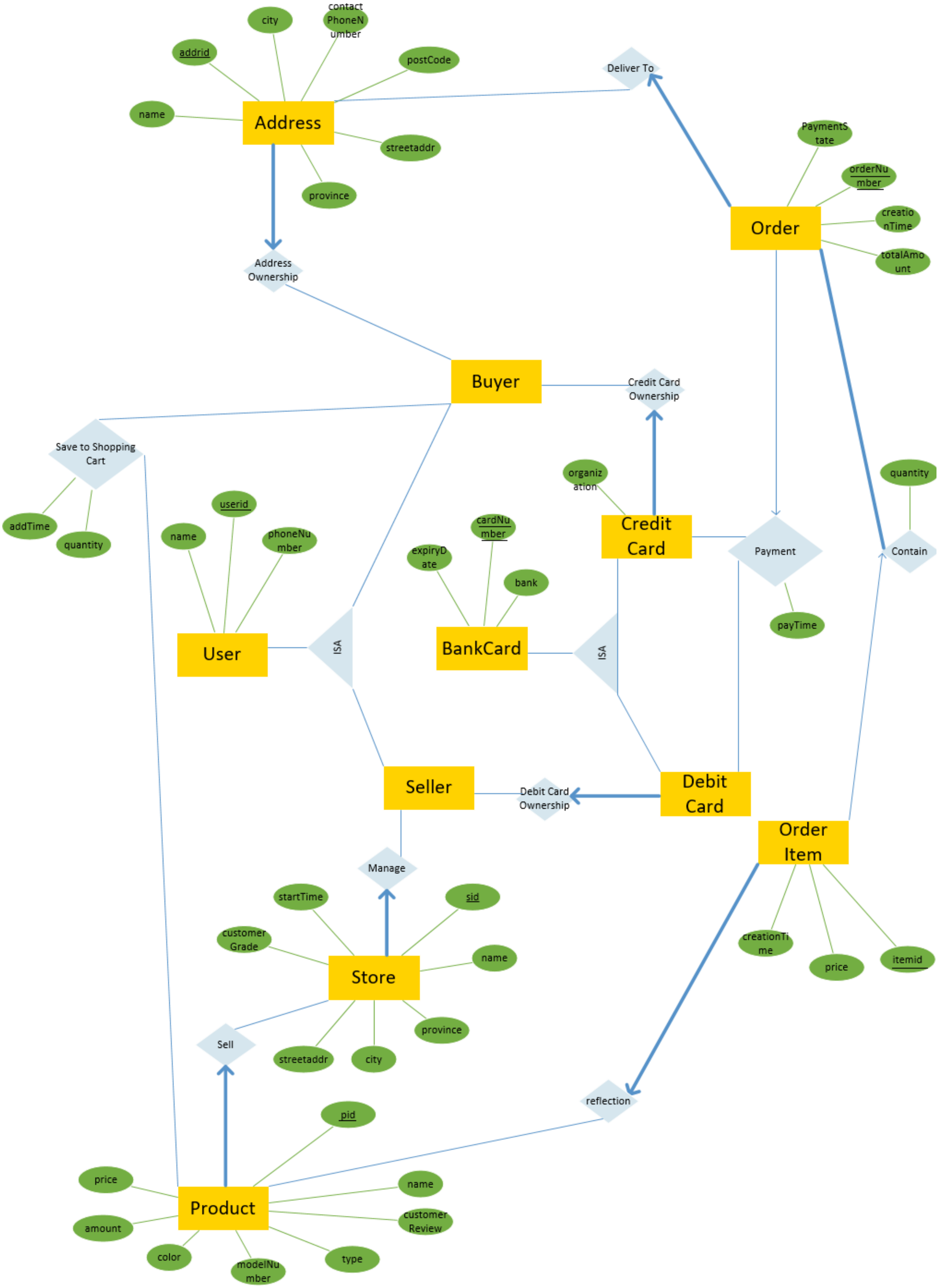
Deliver To (addrid, oid) (addrid ref Address, oid ref Order)

Address Ownership (userid, addrid) (userid ref Buyer, addrid ref Address)

Payment (C.cardNumber, D.cardNumber, oid, paytime) (C.cardNumber ref Credit Card, D.cardNumber ref Debit Card, oid ref Order)

We think there is an opportunity to combine the Credit Card and Debit Card together. Because credit card has its own attribute *organization*, like visa or master card while debit card does not have such attribute, it is good to save the storage space if these two entities are not combined. If these two entities are combined, it can save one entity without redundancy but waste the storage space since debit card would use storage to record the *organization* attribute. In this project, we prefer to separate the credit card and debit card so that save the storage space as well as make the diagram much clearer.

3. E/R Diagram


city
contact PhoneNumber
addrid
postCode
name
Address
streetaddr
province
Deliver To
PaymentState
orderNumber
Order
creationTime
totalAmount
Address Ownership
Buyer
Credit Card Ownership
Save to Shopping Cart
addTime
quantity
organization
Credit Card
quantity
userid
name
phoneNumber
cardNumber
expiryDate
Payment
Contain
bank
payTime
User
ISA
BankCard
ISA
Seller
Debit Card Ownership
Debit Card
Order Item
Manage
startTime
sid
customer Grade
name
Store
creationTime
price
itemid
province
Sell
streetaddr
city
reflection
pid
price
name
Product
customer Review
amount
color
modelNumber
type