Project 1 Part 1

Kunpeng Liu (kl3070) and Zichen Pan (zp2197)

1. Description

In this project, we are going to build an online shopping platform, such as eBay and Amazon. We want to provide a platform for trading goods.

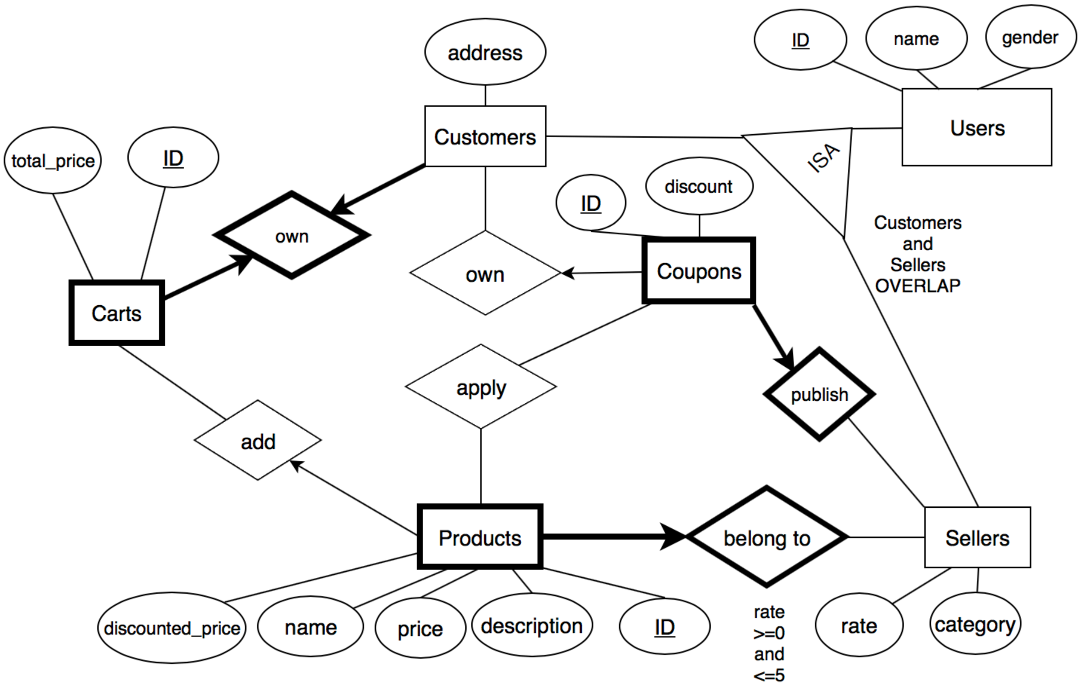
The entity sets are Customers, Sellers, Products, Carts and Coupons. Each of them has very general attributes, such as ID, name etc. We eliminate Customer Service, Record and Payment part of the platform to focus on the main service.

The relationships include adding a cart, applying a coupon etc.

Also, there are several constrains:
1. 'Products' is a weak entity of 'Sellers', since a product must be sold by a specific seller.
2. 'Coupons' is a weak entity of 'Sellers', since a coupon must be published by a specific seller.
3. One coupon can belong to at most one customer.
4. Though the TA suggested us to use a Ternary relationship on the add relationship, we finally decided to separate it into two relationships to make it more clear without eliminating reasonability. Customers and Carts are in one to one relationship and if a customer is deleted, the corresponding cart disappears.
5. One product can be added to at most one cart.

We can find a lot of data about different goods online, but we might need to make up data about sellers and customers, since it is hard to get private information online.

2. E-R Diagram


address
ID
name
gender
Customers
Users
ISA
total_price
ID
own
discount
ID
Customers
and
Sellers
OVERLAP
own
Coupons
Carts
apply
publish
add
Products
belong to
Sellers
discounted_price
name
price
description
ID
rate
>=0
and
<=5
rate
category

3. SQL Schema

```
CREATE TABLE Users(
  user_id int,
  name text NOT NULL,
  gender text,
  PRIMARY KEY (user_id)
)

CREATE TABLE Customers(
  user_id int,
  address text,
  PRIMARY KEY (user_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id)
)

CREATE TABLE Sellers(
  user_id int,
  rate float,
  category text,
  PRIMARY KEY (user_id),
  FOREIGN KEY (user_id) REFERENCES Users(user_id),
  CHECK(
    rate >= 0 and rate <= 5
  )
)

CREATE TABLE Carts(
  cart_id int,
  total_price float,
  customer_id int NOT NULL,
  PRIMARY KEY (cart_id),
  FOREIGN KEY (customer_id) REFERENCES Customers(user_id) ON DELETE CASCADE
)

CREATE TABLE Products(
  product_id int,
  name text,
  description text,
  seller_id int NOT NULL,
  cart_id int,
  price float,
  discounted_price float,
  PRIMARY KEY (product_id),
  FOREIGN KEY (seller_id) REFERENCES Sellers(user_id) ON DELETE CASCADE,
  FOREIGN KEY (cart_id) REFERENCES Carts(cart_id)
)

CREATE TABLE Coupons(
  coupon_id int,
  discount float,
  seller_id int NOT NULL,
  customer_id int,
  PRIMARY KEY (coupon_id),
  FOREIGN KEY (seller_id) REFERENCES Sellers(user_id) ON DELETE CASCADE,
  FOREIGN KEY (customer_id) REFERENCES Customers(user_id)
)

CREATE TABLE Coupon_applied(
  coupon_id int NOT NULL,
  product_id int NOT NULL,
  PRIMARY KEY(coupon_id,product_id),
  FOREIGN KEY(coupon_id) REFERENCES Coupons(coupon_id),
  FOREIGN KEY(product_id) REFERENCES Products(product_id)
)

CREATE TABLE Customers_Carts(
  customer_id int UNIQUE NOT NULL,
  cart_id int UNIQUE NOT NULL,
  PRIMARY KEY (customer_id, cart_id),
  FOREIGN KEY (customer_id) REFERENCES Customers(user_id),
  FOREIGN KEY (cart_id) REFERENCES Carts(cart_id)
)
```