CSE 305

Final Project Report

Rediet Negash: 111820799
Cogitater Sigaukae:

Submitted to Professor Art Lee
June 10, 2020

Content

Background

Domain and Description:

RC house is a shopping website where users can buy clothes, shoes and accessories. The website gives a good experience to the user as it helps them easily fiter the types of products they want by category, price and by seller. In addition to helping users browse easily, RC shops also provides a way where users can add the products they want to shop to their own cart and checkout later when they finish shopping . Apart from letting users buy products it also has a system where users can sell their products on the website. All they need to do is create an account as a seller and register their products then their product will then be part of the shop on the site.

The Goal of this project is to design an efficient database for easy queries and inserts with an interaction to the frontend to display or get data from the database, we did this using the domain of retail and ecommerce.

Our requirement in this project is to make a complete website that stores user and product information to the database and retrieve it when needed based on the requests from the user while using different web pages in the website.

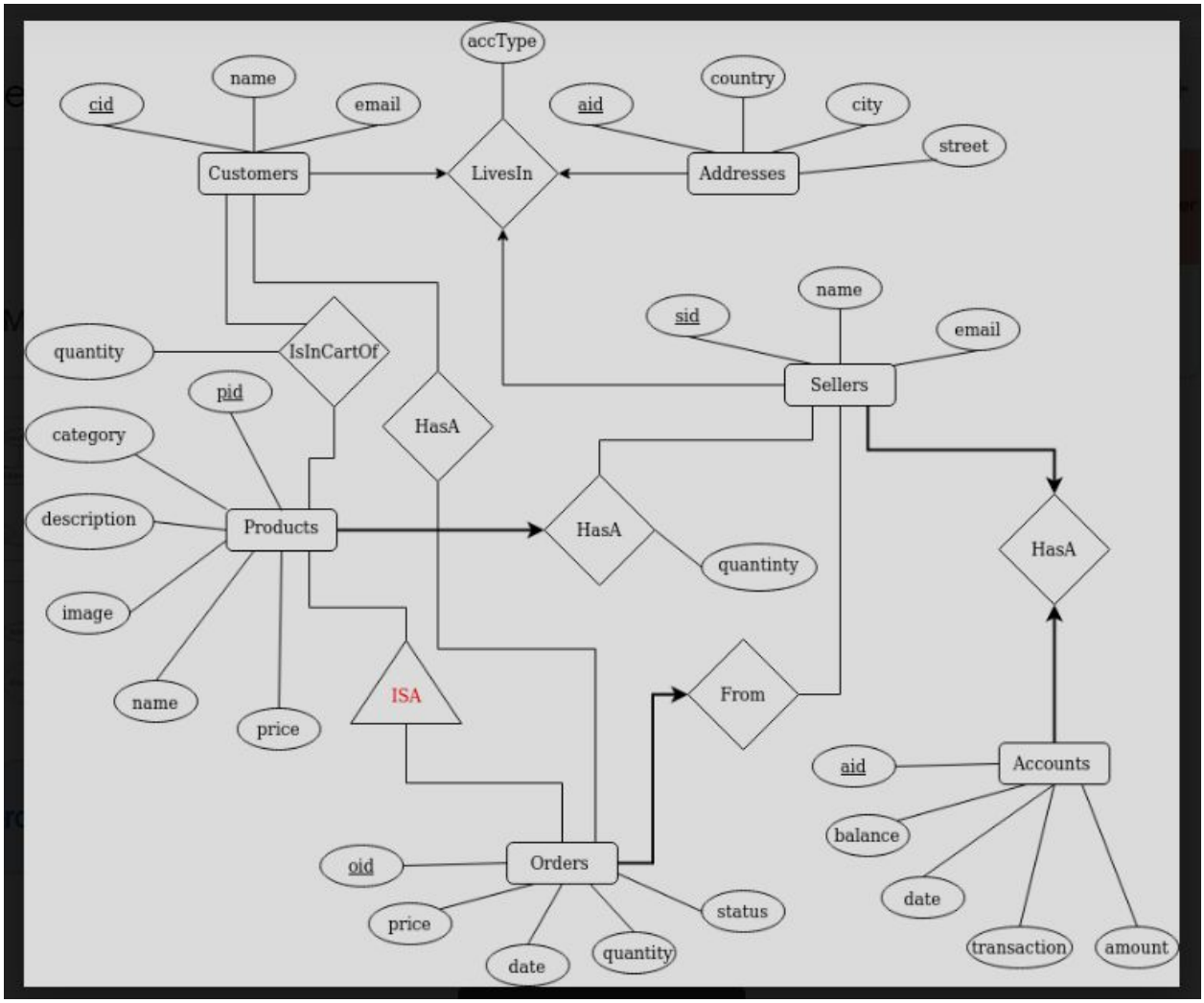
Key Features:

- Create Account as a seller or as a buyer
- Login /Logout- User authentication
- Insert product to the shopping site and see the products added
- Browse through products with different filters
- Buy a product add to cart and checkout
- Remove from cart
- Place Order
- See placed order

Use Case

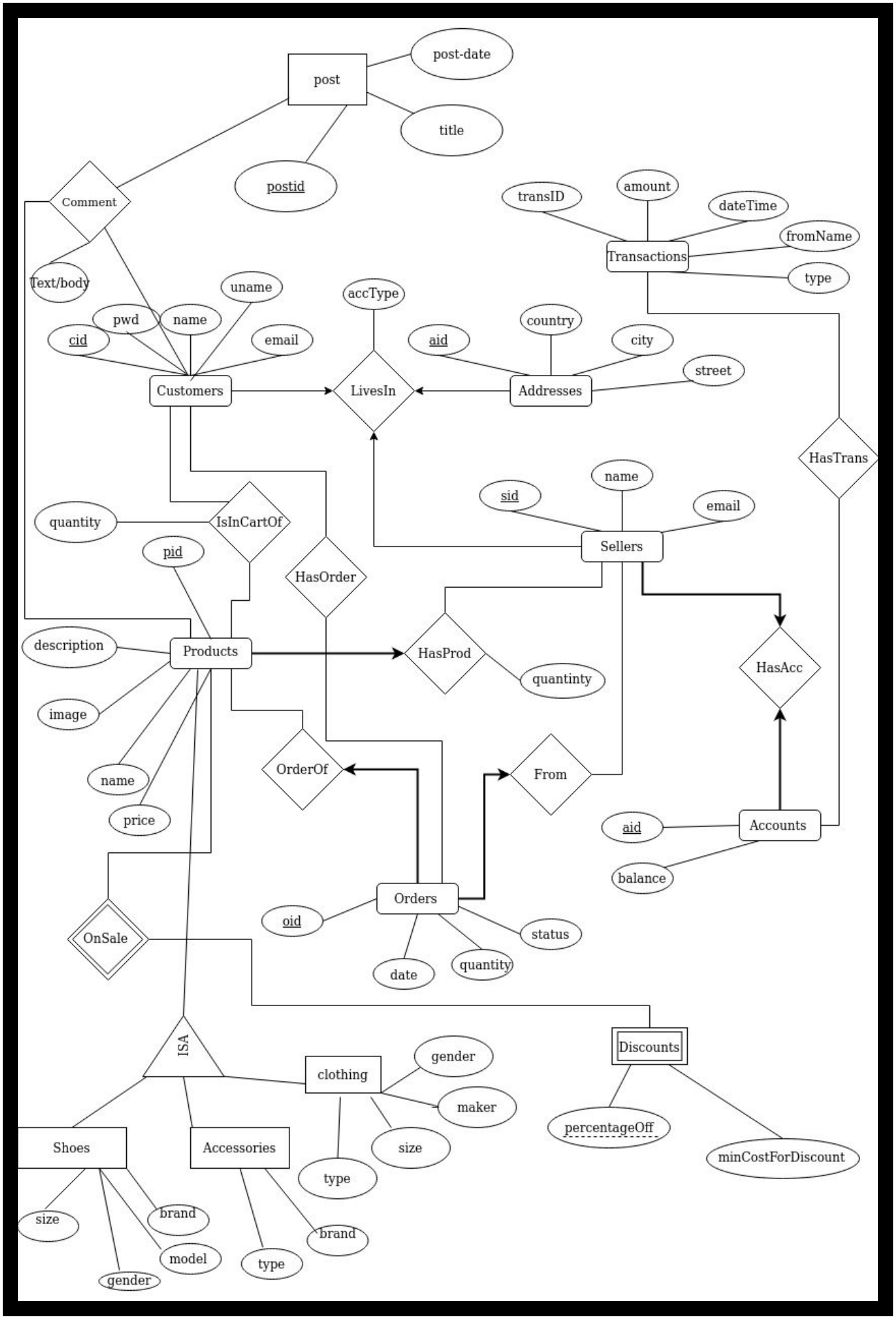
Use Case:	Create Account/Login
Primary Actor:	Seller/Buyer
Goal in Context:	A user wishes to create an account or Login .
Preconditions:	1,The user knows which account to create either a seller or a buyer. 2, The user has all the information needed to create the account or the username and password to login if they already have an account
Trigger:	A user clicks on create account/ Login and they will be displayed with a box where they can enter their information.
Scenario:	1, A user is browsing in the website and clicks a button either login or create account 2, For creating an account the user will be displayed with a sign up page that takes the users information as a form. 3, The system takes the inputs from the user and checks if the information entered is as expected in which case it inserts the information to the database. Otherwise, it will tell the user to enter again.

ER Diagram



First Draft

Final Draft for ER Diagram


post
post-date
title
postid
Comment
Text/body
uname
pwd
name
cid
email
Customers
accType
LivesIn
aid
country
city
street
Addresses
transID
amount
dateTime
fromName
type
Transactions
HasTrans
name
sid
email
Sellers
quantity
IsInCartOf
pid
HasOrder
description
Products
HasProd
quantinty
HasAcc
image
name
price
OrderOf
From
aid
Accounts
balance
Orders
oid
status
date
quantity
OnSale
ISA
Discounts
clothing
gender
maker
size
type
Shoes
Accessories
percentageOff
minCostForDiscount
size
brand
model
gender
brand
type

Normalization

Before Normalization, things we wondered around

- Given our tables do we have any partial dependencies, transitive dependencies,
- Does it satisfy 1st Normal form, 2nd NF, 3rd NF, how about optimizing to BCNF if necessary.
- Do we have any Anomalies

Step by Step normalization

Address Table

Original table

sid	name	uname	pwd	email	country	city	street

Inorder to normalize the table we first have to check if the table satisfies 1NF.
Condition for normal Form

- Single valued attribute
- No change in attribute domain
- Unique name for Attributes/ Columns
- Order doesn't matter

Address Table and Buyer table- Violation of 2nd NF

Before normalization: Original User Table

bid	name	uname	pwd	email	country	city	street

After Normalization

User, Address, Lives In

bid	name	userName	password	email

aid	Country	City	Street

uid	aid	type

User-Buyer Vs Seller

Original table: has issues as we can have the same user with the same information having different account types either a seller or buyer. This causes anomalies in update, delete and insert.

After Normalization:

uid	name	uname	pwd	email	AccountType

sid	name	uname	pwd	email

bid	name	uname	pwd	email

Products Table and Seller table- Violation of 1NF, 2NF, 3NF, BCNF

Before normalization: Original User Table

sid	pwd	name	uname	email	pid	description	image	pname	price	percent	num	address

Seller

sid	pwd	name	uname	email	country	city	street

Products

pid	description	name	image	description	price	num	discount

Inheritance for Product categories

Inheritance: Product

pid	name	des	img	price	prdisc	numPrdi	category

Clothing

pid	name	des	img	price	prdiscount	numPrDis

Accessories

pid	size	gender	model	brand

Buyer and Orders

If we were to put orders of different products in buyer, the table would have violated Normal forms starting from 2nd.

To comply with 3rd and 2nd Normal Forms, we could put order ids as a foreign key in buyers table and link those unique order ids to order table that has all the details including date and status, However, we further decomposed

the table to BCNF form where we separated both Buyer and Order table and we wrote a relation table that links a buyer with the order.

Some of the main Tables we have

Accessories

Buyer

Seller

Order

Cart

Address

Product -> Clothing, shoes accessories

Relational Tables:

LivesIn- user & address

HasOrder- Seller & product

OrderOf- links

fromSeller: Order ~ Seller

User Manual

Conclusion

Improvements