DATABASE 6710

Georgia State University

Simple E-commerce Database

Team: Sravya Kambhampati, Yi Lu, Xiao Wang,

Date: 12/7/2019

Table of Contents

Introduction

The purpose of this project is to design and implement an e-commerce database with a relational database management system. The motivation behind this project is to design a realistic database and can be applied in a simple and comprehensive GUI. As we can see online shopping market is part of our daily routine nowadays, everyone is using online shopping, we hope to design a database that helps people determine their preference, as well as helping companies make better decisions suitable to customers.

Our project focuses on online shopping market with 7 categories of products: women's clothing, men's clothing, kids wear, cell phone, laptop, TV and furniture. There are 12 entities and 120 products datasets record in the database. Some of the database functions a user can perform includes:

- Browse products
- Add products to wish list
- Buy products (add them to cart)
- Manage personal information
- Manage delivery addresses
- User write product reviews

Database Structure

13 Entities:

- User
- Reviews (Product has multiple customer reviews)
- Category
- Product
- Product Photo
- Credit card
- user_card
- User address
- Order
- User_order
- Wish
- Prod_cat
- User_cart

Detail of entity relations is shown in the EER diagram in the next section.

Data Source

- Product details: manual extraction from Amazon.com
- User, card, and address details generated using online tools.

Data Reference

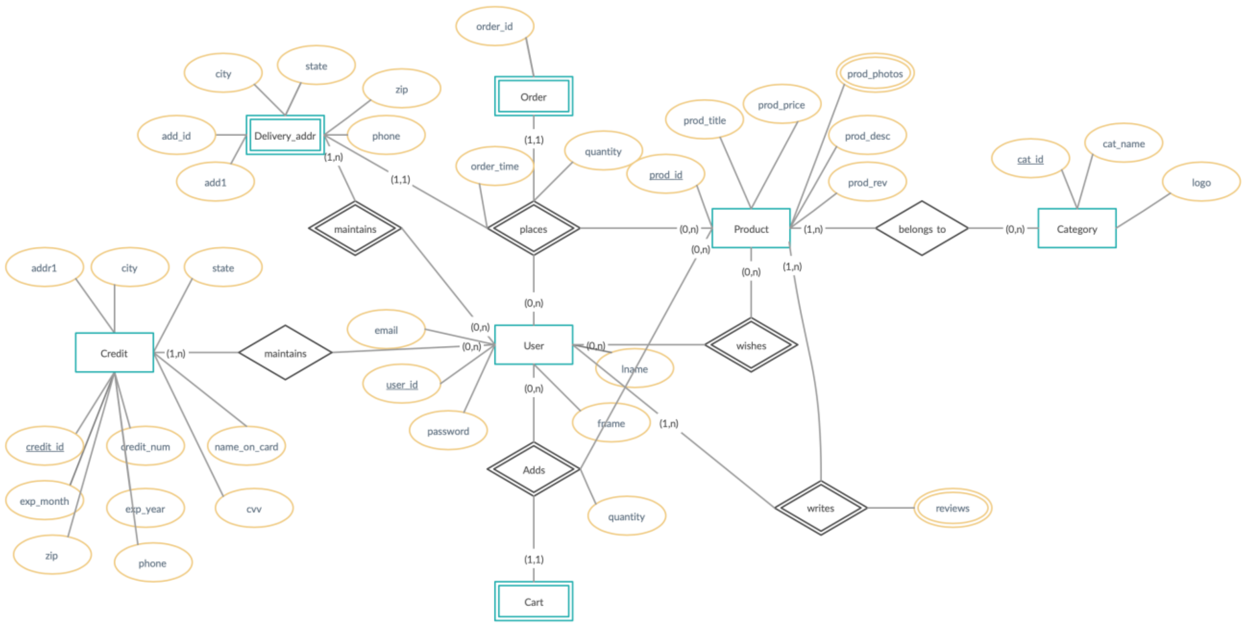
- https://www.kaggle.com/datafiniti/electronic-products-prices
- https://data.world/datafiniti/electronic-products-and-pricing-data

Database Requirement Analysis

1. User Login: query to check whether user has account and his credentials are correct.
2. Current User: Query to display user who is logged in currently.
3. User register: query to check where user existed or not. If he doesn't exist register him taking his details like first name, last name, email, password.
4. User profile: Query to retrieve user's orders, wishlist, account details like username, email, password, delivery addresses and credit card details if he has any previously stored.
5. Categories: Query to list all categories available.
6. Products: Listing all products based on category.
7. Product details: Listing product details like price, reviews, ratings, description etc when user clicked a product image.
8. User review: Query to list user reviews of a product.
9. Add to cart: Query to insert product details and its quantity to the cart table.
10. Delete cart: query to delete items in cart or empty cart.
11. Count items in cart: Display number of items in the cart.
12. Cart total: Query to count total amount to be paid to buy cart products.
13. Place order: Query to add order id, timestamp, delivery address and card used to purchase in order table.
14. Order Summary: To list placed order summary.
15. Past Orders: Query to retrieve particular user past orders as per date.
16. Wishlist: Query to retrieve particular user's wishlist.
17. Cards Saved: Query to display cards saved by user.
18. Addresses saved: Query to display addresses saved by user.
19. Update User details: Query to update user details in their profile.
20. Update email and password: Query to check user's old password and if it matches update his old one with new password.
21. Delete User address: Deleting user address if it is not required any more.
22. Adding new user address: Adding new user address.
23. Adding new card: Adding new card to user profile.

EER Model

The EER model of the database is also attached in the submission. The 12 entities and functions are illustrated in the EER diagram. For example, User Places Orders, User Writes Review, User Add Product, Credit card Pays Product, etc.



Relational Model

Mapping :

Step 1: Regular entities.

User

Email	**user_id**	Password	Fname	Lname

Category

Cat_id	Cat_name	logo

Product

Prod_id	Prod_title	Prod_price	Prod_desc	Prod_rev

Credit

Credit_num	**Credit_id**	Name_on_card	Exp_month	Exp_year	Cvv	Addr	City	state	zip	phone

Step 2: **Finding week entities.**

Delivery_addr

add_id	add1	city	State	zip	phone

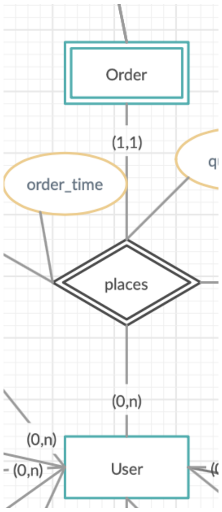
Cart

Cart_id

Order

Order_id

Step 3: Binary 1:1 relationship.
There are no one to one relationship in our project.
Step4: Binary 1:N relation ship
We have around 10 1:N relationships.



Email	**user_id**	Password	Fname	Lname

Order_id	User_id

Converted above 1:N relationship to relation. Similarly converted all other relations.
Step5: Binary M:N relation ships
We have around 5 M:N relationships. For example 1 M:N relationship converted as follows

User:

Email	**user_id**	Password	Fname	Lname

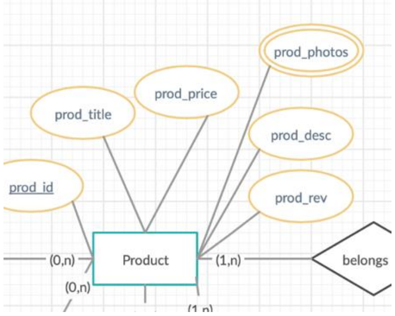
Credit

Credit_num	Credit_id	Name_on_card	Exp_month	Exp_year	Cvv	Addr	City	state	zip	phone

User_card

Userid	**creditid**

Step6: Multivalued attributes
Prod_photos and reviews are two multivalued attributes.



The above multivalued attribute converted as follows:

Product:

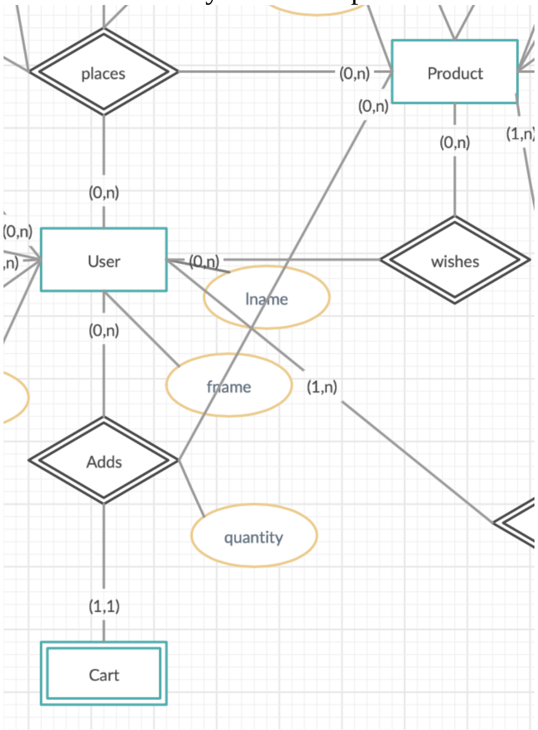
Prod_id	Prod_title	Prod_price	Prod_desc	Prod_rev

Prod_photos

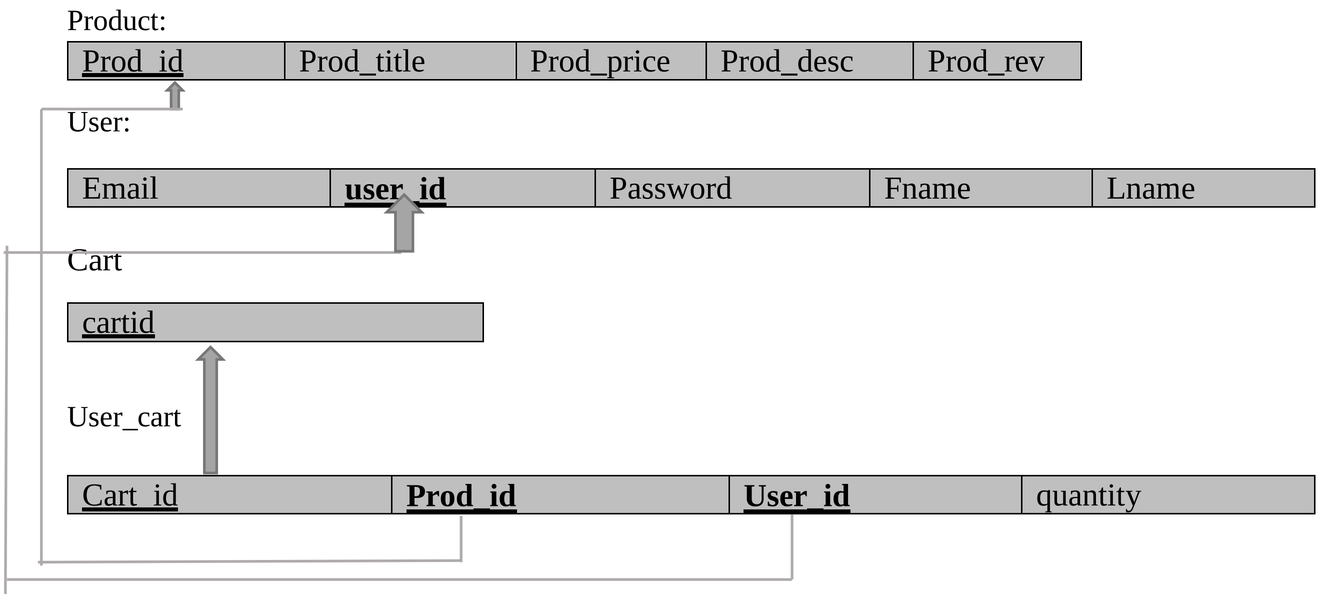
Prod_id	**Prod_img**

Step7: N-ary relationship

We have 2 ternary relationships.



The above ternary relationship converted as follows:



Product:

Prod_id	Prod_title	Prod_price	Prod_desc	Prod_rev

User:

Email	**user_id**	Password	Fname	Lname

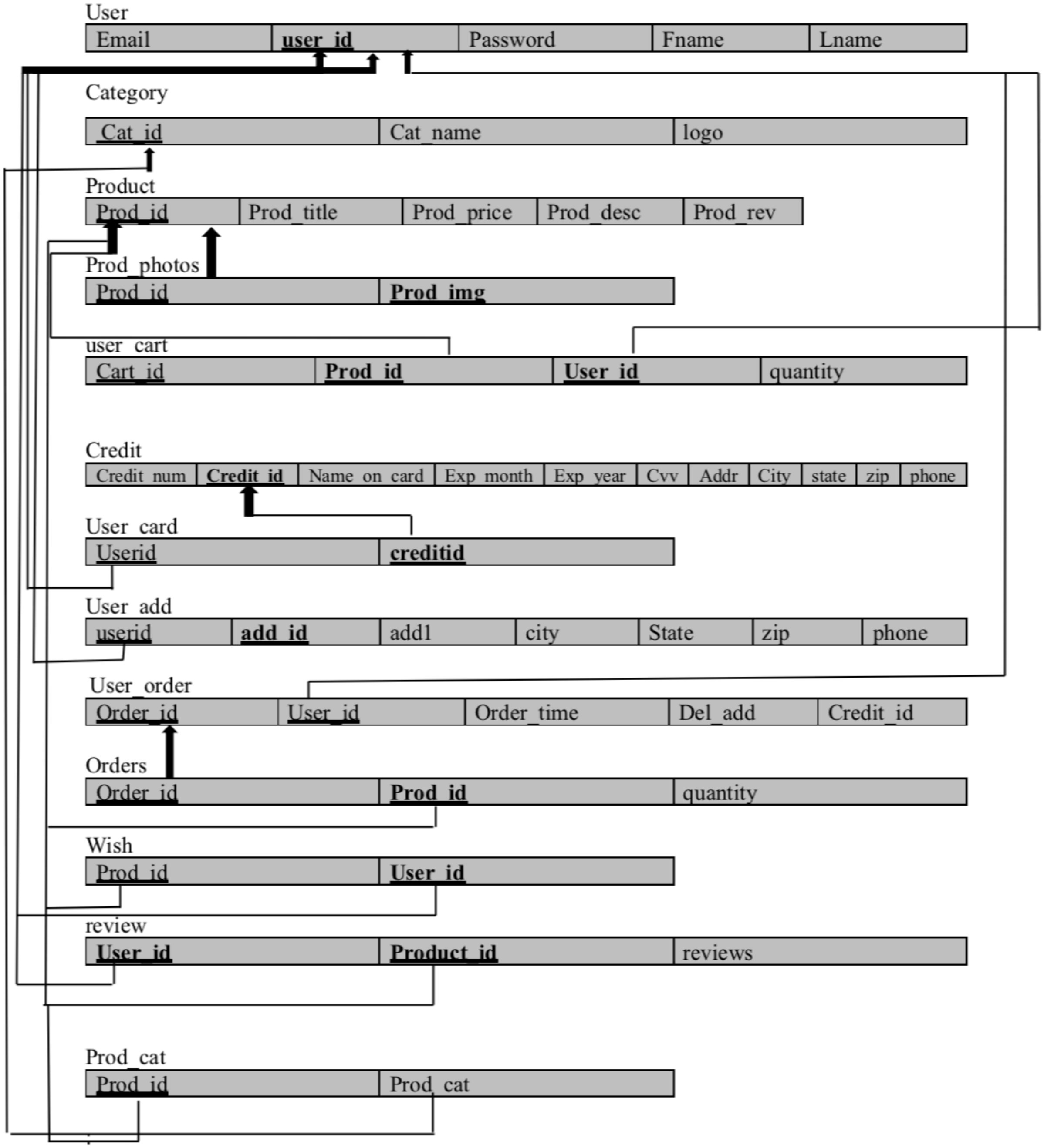
Cart

cartid

User_cart

Cart_id	**Prod_id**	**User_id**	quantity

After mapping whole EER model we got following relational model.



User

Email	**user_id**	Password	Fname	Lname

Category

Cat_id	Cat_name	logo

Product

Prod_id	Prod_title	Prod_price	Prod_desc	Prod_rev

Prod_photos

Prod_id	**Prod_img**

user_cart

Cart_id	**Prod_id**	**User_id**	quantity

Credit

Credit_num	**Credit_id**	Name_on_card	Exp_month	Exp_year	Cvv	Addr	City	state	zip	phone

User_card

Userid	**creditid**

User_add

userid	**add_id**	add1	city	State	zip	phone

User_order

Order_id	User_id	Order_time	Del_add	Credit_id

Orders

Order_id	**Prod_id**	quantity

Wish

Prod_id	**User_id**

review

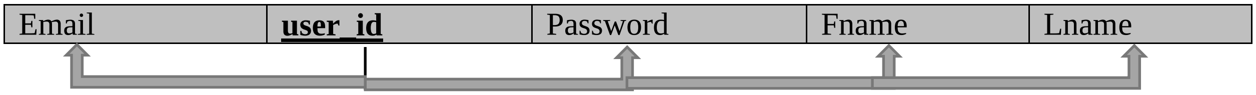
User_id	**Product_id**	reviews

Prod_cat

Prod_id	Prod_cat

Normalization

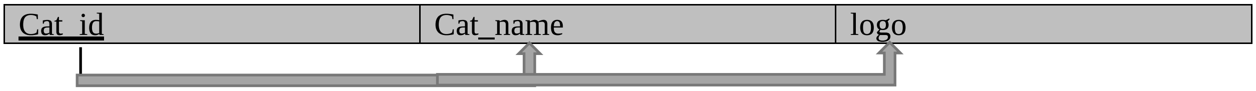
User: It is 3NF



Email	**user_id**	Password	Fname	Lname

There are no multivalued attributes hence it is in 1NF. There is no partial dependency user_id is only primary key. Email is candidate key. Every non-prime attribute is fully functionally dependent on the primary key. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

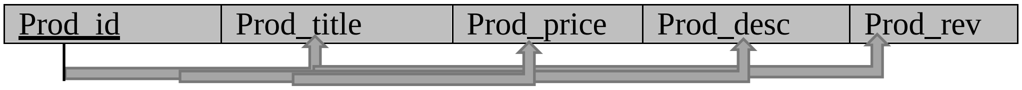
Category:



Cat_id	Cat_name	logo

There are no multivalued attributes hence it is in 1NF. There is no partial dependency cat_id is only primary key. Every non-prime attribute is fully functionally dependent on the primary key. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Product:



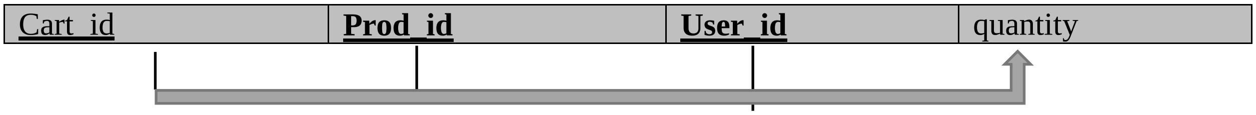
Prod_id	Prod_title	Prod_price	Prod_desc	Prod_rev

There are no multivalued attributes hence it is in 1NF. There is no partial dependency prod_id is only primary key. Every non-prime attribute is fully functionally dependent on the primary key. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Prod_photos

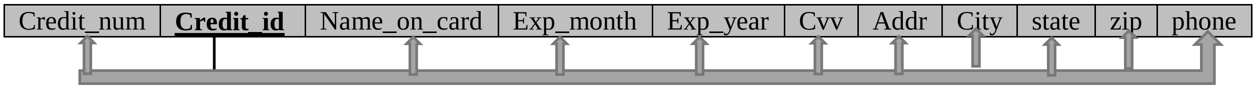
Prod_id	**Prod_img**

There are no multivalued attributes hence it is in 1NF. There is no partial dependency. Prod_id and prod_img combinely called primary key. Just prod_id can not determine prod_img hence there is no partial dependency. No non-prime attribute is here. So it is in 2NF. It is in 3NF.



Cart_id	**Prod_id**	**User_id**	quantity

There are no multivalued attributes hence it is in 1NF. There is no partial dependency. Cart_id Prod_id and user_id combinely forms primary key. Just prod_id cannot determine quantity hence there is no partial dependency. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.



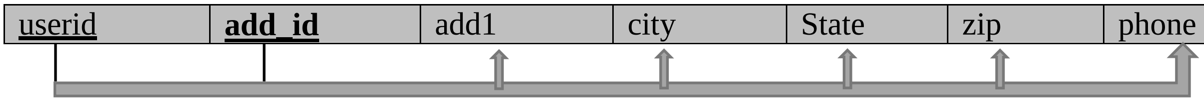
Credit_num	**Credit_id**	Name_on_card	Exp_month	Exp_year	Cvv	Addr	City	state	zip	phone

There are no multivalued attributes hence it is in 1NF. There is no partial dependency. Cart_id is only primary key. Credit_num is candidate key. It is in 2NF since every nonprime attribute is fully functional dependent on credit_id. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Userid	**creditid**

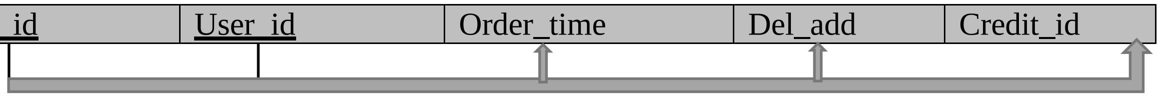
There are no multivalued attributes hence it is in 1NF. There is no partial dependency. useridand creditid combinely called primary key. Just userid cannot determine creditid because single user can have multiple cards. Hence there is no partial dependency. No non-prime attribute is here. So it is in 2NF. It is in 3NF.

userid	**add_id**	add1	city	State	zip	phone



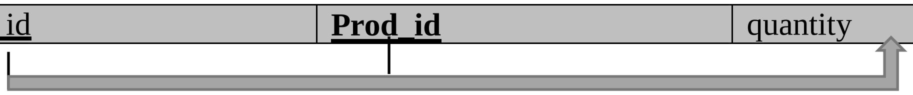
There are no multivalued attributes hence it is in 1NF. There is no partial dependency. userid addid combinely forms primary key. Just userid cannot determine nonprime attributes hence there is no partial dependency. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Order_id	User_id	Order_time	Del_add	Credit_id



There are no multivalued attributes hence it is in 1NF. There is no partial dependency. order_id and user_id together forms primary key. It is in 2NF since every nonprime attribute is fully functional dependent on order_id and user_id because user can place order to different addresses and using different cards. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Order_id	**Prod_id**	quantity



There are no multivalued attributes hence it is in 1NF. There is no partial dependency. Order_id prod_id together forms primary key. It is in 2NF since every nonprime attribute is fully functional dependent on order_id and prod_id. Here again quantity is stored because cart quantity might exist or not while placing order. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Wish:

Prod_id	**User_id**

There are no multivalued attributes hence it is in 1NF. There is no partial dependency. User_id prod_id together forms primary key. There are no nonprime attributes. User_id cannot determine prod_id because same user can have multiple products in their wish list. No non-prime attribute exists and is transitively dependent on the primary key. Hence it is in 3NF.

User_id	**Product_id**	reviews

There are no multivalued attributes hence it is in 1NF. There is no partial dependency. User_id prod_id together forms primary key. It is in 2NF since every nonprime attribute is fully functional dependent on user_id and prod_id. User_id and product_id together can determine reviews. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

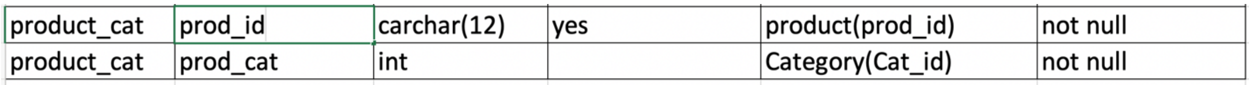
Prod_id	Prod_cat

There are no multivalued attributes hence it is in 1NF. There is no partial dependency prod_id is only primary key. Every non-prime attribute is fully functionally dependent on the primary key. So it is in 2NF. No non-prime attribute is transitively dependent on the primary key. Hence it is in 3NF.

Data Dictionary

A data dictionary is also created to define the basic organization of the tables which are to be stored in the database as shown below. This file is also attached in the submission file.

Table	Attribute	Data type	Primary key	Foreign key	Constraints
User	Email	varchar(35)			not null
User	user_id	int	yes		not null
User	Password	varchar(30)			not null
User	Fname	varchar(20)			not null
User	Lname	varchar(20)			not null
Category	Cat_id	int	yes		not null
Category	Cat_name	varchar(30)			not null
Category	logo	varchar(20)			
Product	Prod_id	varchar(12)	yes		not null
Product	Prod_title	varchar(200)			
Product	Prod_price	decimal(6,2)			not null
Product	Prod_des	text			not null
Product	Prod_rev	int			
Prod_photos	Prod_id	varchar(12)	yes	Product(prod_id)	not null
Prod_photos	Prod_img	varchar(12)	yes		not null
Cart	Cart_id	int	yes	Category(Cat_id)	not null
Cart	Prod_id	varchar(12)	yes	Product(prod_id)	not null
Cart	user_id	int	yes	User(user_id)	not null
Cart	quantity	int			not null
Credit	credit_num	varchar(5)			not null
Credit	credit_id	varchar(20)	yes		unique, not null
Credit	Name_on_card	varchar(30)			not null
Credit	Exp_month	int			not null
Credit	Exp_year	int			not null
Credit	CVV	int			not null
Credit	Addr	varchar(50)			not null
Credit	City	varchar(25)			not null
Credit	state	varchar(3)			not null
Credit	zip	varchar(15)			not null
Credit	phone	varchar(15)			not null
User_card	Userid	int	yes	User(user_id)	not null
User_card	creditid	varchar(5)	yes	Credit(creditid)	not null
User_add	userid	int	yes	User(user_id)	not null
User_add	add_id	varchar(4)	yes		not null
User_add	add1	varchar(5)			not null
User_add	city	varchar(25)			not null
User_add	state	varchar(3)			not null
User_add	zip	varchar(15)			not null
User_add	phone	varchar(15)			not null
User_order	Order_id	varchar(15)	yes		unique, not null
User_order	User_id	int	yes	User(user_id)	not null
User_order	Order_time	datetime			
User_order	Del_add	varchar(50)			not null
User_order	Credit_id	varchar(5)		Credit(creditid)	not null
Orders	Order_id	varchar(15)	yes	User_order(Order_id)	not null
Orders	Prod_id	varchar(12)	yes	Product(prod_id)	not null
Orders	quantity	int			not null
Wish	Prod_id	varchar(12)	yes	Product(prod_id)	not null
Wish	user_id	int	yes	User(user_id)	not null
review	User_id	int	yes	User(user_id)	not null
review	Product_id	varchar(12)	yes	Product(prod_id)	
review	reviews	text			



product_cat	prod_id	carchar(12)	yes	product(prod_id)	not null
product_cat	prod_cat	int		Category(Cat_id)	not null

Implementation

The database system is implemented in the form of a website. The GUI is built with HTML, CSS, bootstrap and PHP. The source code and web page files included in the submission can be run on local machines, however, the website is not hosted on any server at this point. Through the GUI, users will be able to create, edit personal information, browse, search, save and check out items, at the same time, their payment information and reviews of items are also stored in the database system. The website demo was provided during class on Dec. 3. The detailed code can be found in the file submission.

Teamwork

In the beginning, everyone participated in the research of ideas and the datasets of the project. Once we have our project set up and have all 13 entities ready and there are 120 product's record and total 8 different categories of the products. Originally, as a group of four people, we decided to divide 8 different categories equally to four of us, so each person got 2 categories to search their data and import to excel, write the query of create the table and insert the data for our own categories assigned for. After that. Xiao Wang and Yi Lu mainly focus on the presentation and the reports and the database preparation, and Sravya Kambhampati focus on the organization of the database and front-end part with build the web demo. However, due to the drop-out of one team member, the final team consists of three people as titled on the first page. A summary of percentage of work is roughly 40% for Sravya Kambhampati and 30% each for Xiao Wang and Yi lu because Sravya Kambhampati took on more work as one of the original members dropped out.

Summary

This project aims to build a simple to use e-commerce database and implement it in a functional GUI form. Through research, we designed 13 entities to hold all attributes in the database. The relationship among entities such as user, product, credit cards, order and reviews were illustrated in the EER diagram to explain the functions of each. After that, these relationships were mapped into the relational schema as initial 13 tables, and the normalization process is performed to ensure the resilient table structure. Then queries were added to create these tables in MySQL, along with a data dictionary to organize the attributes. Finally, the database was implemented with a client application in the form of a simple website. The demo of the GUI part was performed in class on Dec. 3. All the queries and source code of the project, as well as the illustration of EER diagram, relational schema and data dictionary are attached in the final submission.