Pet Store Database

CSCI 4370 (Database Management)
Fall 2021
Group 4
Reese Mallory, Kalyb Sanders, Joseph Nguyen, and Avranil Basu

Problem and Motivation

- The problem that we set out to overcome is to create an ecommerce web store where users can add products to their cart and checkout. This is very similar to almost all online web stores used today, so we wanted to implement something similar.
- Our motivation for this is project is to implement the web store in a way similar to how most enterprise businesses set up their web stores. This means trying to include features that some online stores use, such as store multiple cards per user, and more.

Project Description

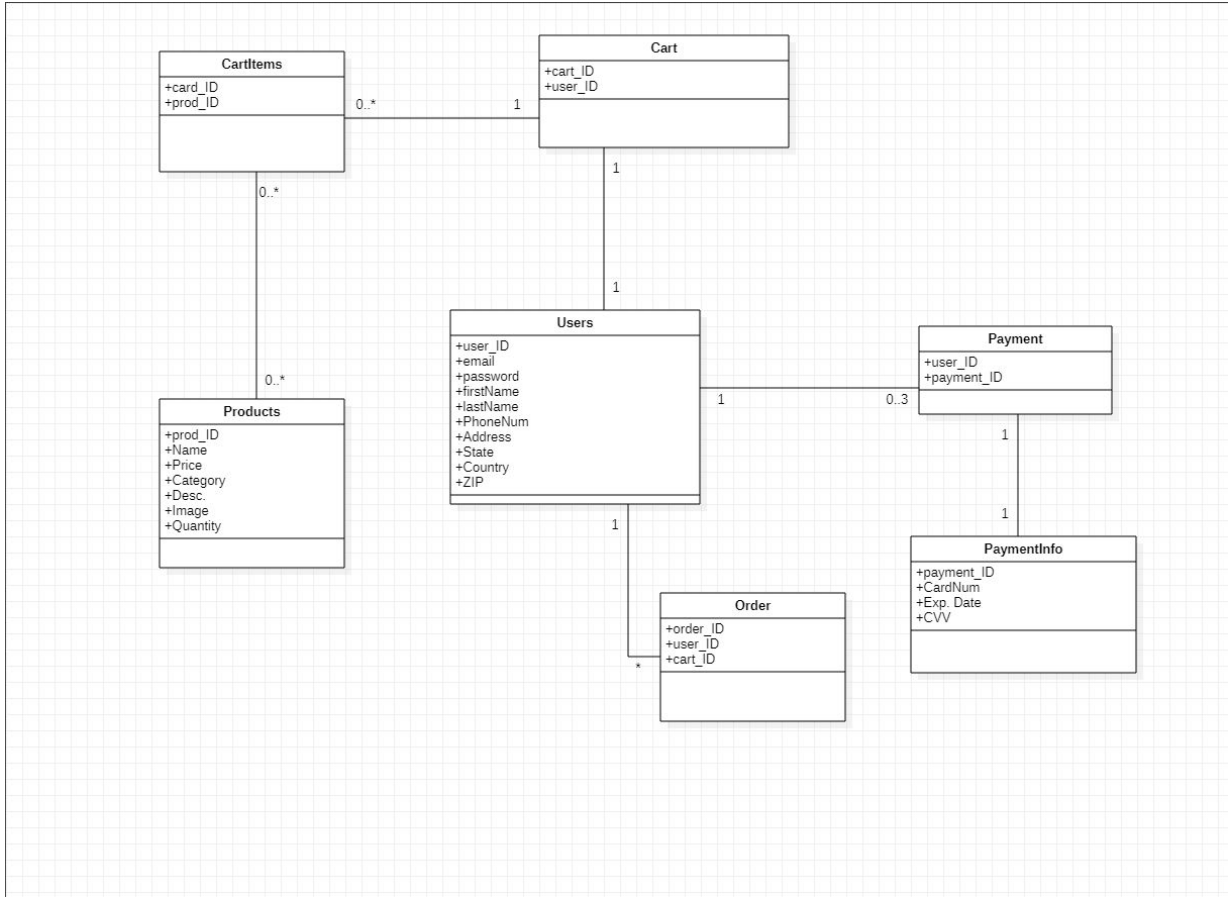
- Our Project is defining a basic online shopping system in the form of a pet supplies store. This pet store should include different functions such as editing product details, editing payment types, editing user information, querying a products list, adding to a cart, and more
- Our client in this case is a business that needs to track its products, as well as allow users to query and purchase items from the database. The users for this database in this case is both customers(normal users) and the business/employees(admins).
- Some Examples of queries needed are :
    - Query all products based on search term
    - Allow the user to change information of their account
    - Query all past transactions/orders

How the work as divided

- Work was divided evenly once we determined which forms we were going to use. Once we determined which forms needed to be created, each person was assigned a portion of the forms to complete.
- Once the forms were completed, we set up a meeting where we could implement functionality of these forms and solve problems with them as a team. This includes making sure the forms function as they are supposed to and making sure the overall design is setup properly.
- Finally, each person added sections to the presentation until it was completed.

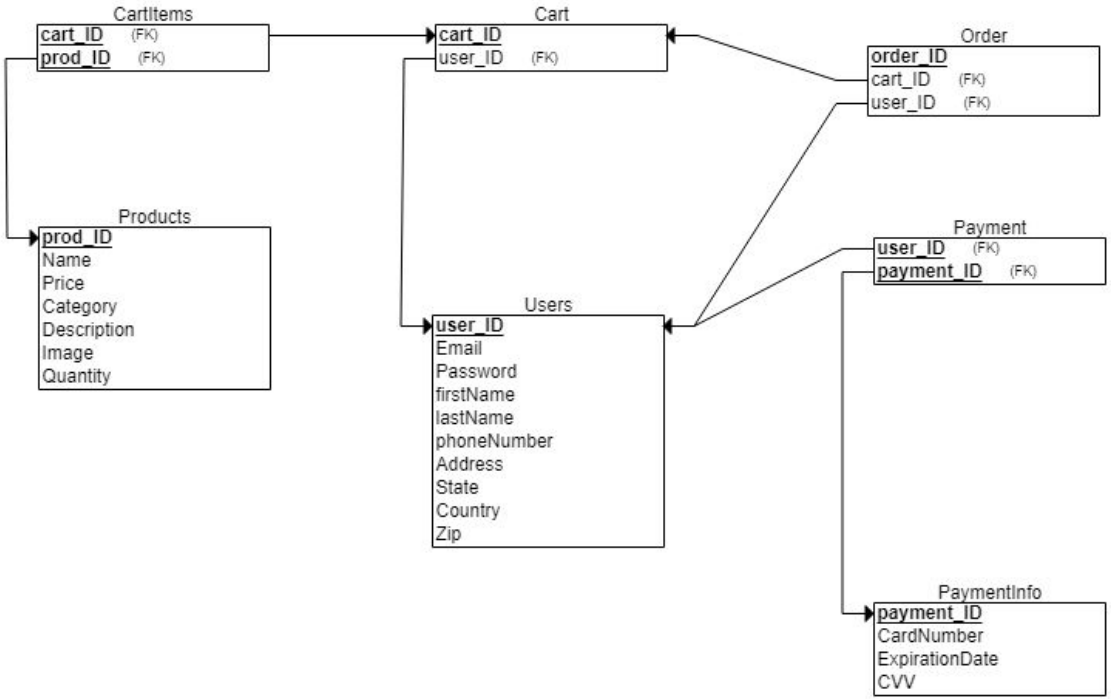
ER Diagram

- This is our ER diagram that shows the relationships between each of our tables. It shows the various Cardinalities of the relationships, as well as the columns that are included in each class



Relational Schema

- This is a diagram of our relation schema. It displays the various primary and foreign keys, and the relationship between them in each table.



CRUD Matrix: Forms versus Tables

- This is our CRUD Matrix that shows the relationships between our tables and forms for the pet store. Specifically it shows the activity and permissions with tables that the form may have.

CSCI 4370 Group 4 Project CRUD Matrix

Forms x Tables	Products	CartItems	Cart	Users	Order	Payment	Payment Info
Add Products	C						
Add Users				C			
Add Payment Cards						C	C
Add to Cart	R	C	C				
View Cart		R					
Remove Item from Cart		UD					
Checkout	U	RUD	RUD	RU	CRUD	RU	RU
Edit Product Info	RUD						

Table Descriptions - Users

- One of the main tables of our website is the ‘Users ‘ table. This table is made up of all the user accounts and their corresponding information. This includes email, location, payment info, and more

users

Column	Type	Null	Default	Links to	Comments	Media type
user_ID *(Primary)*	int(11)	No				
email	varchar(75)	No				
password	varchar(15)	No				
firstname	varchar(50)	No				
lastname	varchar(50)	No				
phoneNum	int(11)	Yes	*NULL*			
Address	varchar(75)	Yes	*NULL*			
State	varchar(20)	Yes	*NULL*			
Country	varchar(30)	Yes	*NULL*			
ZIP	varchar(10)	Yes	*NULL*			

Table Descriptions - Products

- The next most populated table is the 'Products' table. This table includes the entire catalog of products that are stocked in the store, along with the details that go with each product such as name, price, description picture, etc.

products

Column	Type	Null	Default	Links to	Comments	Media type
prod_ID *(Primary)*	int(11)	No				
Name	varchar(20)	No				
Price	decimal(10,2)	No				
Category	varchar(20)	No				
Description	varchar(250)	No				
Image	varchar(250)	No				
qty	int(11)	No				

Table Descriptions - Payment & PaymentInfo

- The ‘payment’ and ‘paymentinfo’ classes store information about the user’s inputted payment info. The ‘payment’ table first links a userID to a paymentID, and payment info stores the actually info of a specific card. This type of table would allow the owner of the site to allow the users to store multiple cards without much editing to the database

payment

Column	Type	Null	Default	Links to	Comments	Media type
user_ID *(Primary)*	int(11)	No				
payment_ID *(Primary)*	int(11)	No				

paymentinfo

Column	Type	Null	Default	Links to	Comments	Media type
payment_ID *(Primary)*	int(11)	No				
CardNum	bigint(16)	No				
Exp. Date	date	No				
CVV	int(3)	No				

Table Descriptions - Cart & Cartitems

- ‘Cart’ and ‘Cartitems’ work in a very similar way as payment and payment info. The cart table links a userID to a cardID, and the cartitems table stores the actual items in a specific user’s cart.

cart

Column	Type	Null	Default	Links to	Comments	Media type
cart_ID *(Primary)*	int(11)	No				
user_ID	int(11)	No		users -> user_ID		

cartitems

Column	Type	Null	Default	Links to	Comments	Media type
cart_ID *(Primary)*	int(11)	No		cart -> cart_ID		
prod_ID *(Primary)*	int(11)	No				

Table Descriptions - Orders

- The final table we are using is a orders table. It links userID, and cartID. This allows users to go through their order history and look at past orders for example.

orders

Column	Type	Null	Default	Links to	Comments	Media type
order_ID *(Primary)*	int(11)	No				
user_ID	int(11)	No		users -> user_ID		
cart_ID	int(11)	No		cart -> cart_ID		

List of Example Queries

- SELECT * FROM Products;

prod_ID	Name	Price	Category	Description	Image	qty
0	Apples	2.00	Food	Good Apples	/appleimage	100
1	Flying Dog	150.00	Dogs	Dog that flies	/flydogimage	5
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- SELECT * FROM Products WHERE Name = ‘Apples’;

prod_ID	Name	Price	Category	Description	Image	qty
0	Apples	2.00	Food	Good Apples	/appleimage	100
NULL	NULL	NULL	NULL	NULL	NULL	NULL

- SELECT * FROM CartItems WHERE cart_ID = ‘1’;

cart_ID	prod_ID
1	0
1	1
NULL	NULL

List of Example Queries

- SELECT * FROM Users WHERE user_ID = ‘1’;

user_ID	email	password	firstname	lastname	phoneNum
1	josephnguyen902@gmail.com	joseph123	Joseph	Nguyen	2147483647
NULL	NULL	NULL	NULL	NULL	NULL

- INSERT INTO Products VALUES (1, ‘Carrots’, 1.00, ‘Food’, ‘A Stick of Carrot’, ‘/carrotimage’, 100);
-

79	21:40:16	INSERT INTO Products VALUES (2, 'Carrots', 1.0...	1 row(s) affected	0.031 sec

- DELETE FROM CartItems WHERE cart_ID = ‘1’;
-

92	21:52:35	DELETE FROM CartItems WHERE cart_ID = '1'	2 row(s) affected	0.032 sec

- UPDATE Products SET qty = qty - 1 WHERE prod_ID = ‘1’ AND qty > 0;

80	21:42:59	UPDATE Products SET qty = qty - 1 WHERE pro...	1 row(s) affected Rows matched: 1 Changed: 1 ...	0.015 sec

Forms

ADD PRODUCT

Product Information

Name
German Shepherd
Price
4
Category
Dog
Description
Fluffy
Image
Choose File No file chosen
Quantity

Add Product

ADD USER

User Information

First Name
John
Last Name
Smith
Phone Number
1234567890
Address
2222 Pet Drive
State
Georgia
Country
United States
ZIP
30078
Email
johnsmith@gmail.com
Password
****

Add User

Forms

ADD PAYMENT CARD

Add To Cart

Payment Information

Card Number

1234123412341234

Expiration Date

08/23

CVV

123

Add Payment Card

Add Product to Cart

ProductID

Quantity

0

Email

johnsmith@gmail.com

Password

****

Add to Cart

Forms

SHOPPING CART

Cart ID: 1 | Product ID: 1
Cart ID: 1 | Product ID: 2
Cart ID: 1 | Product ID: 3

Cart ID

cart_ID

Checkout

Remove From Cart

Remove Product from Cart

ProductID

Quantity

0

Email

johnsmith@gmail.com

Password

****

Remove from Cart

Forms

EDIT PRODUCT INFO

Product ID :

Product Name :

Price :

Category :

Description :

Quantity :

Photo : Girl in a jacket

Update

CRUD Matrix: Forms versus Triggers

CSCI 4370 Group 4 Project Trigger CRUD Matrix			
Forms x Trigger	Create Cart for User	Decrement Product Quantity	Increment Product Quantity
Add Products			
Add Users	C		
Add Payment Cards			
Add to Cart		C	
View Cart			
Remove Item from Cart			C
Checkout			
Edit Product Info			

Triggers

Trigger 1 - Create Cart for User

This trigger creates a cart for the new user when a new user is added to the database

```
CREATE TRIGGER `create cart` AFTER INSERT ON `users`
 FOR EACH ROW INSERT INTO cart (cart_ID, user_ID)VALUES(new.user_ID,new.user_ID)
```

Sequence/Triggers/Stored Procedures

Trigger 2 - Decrement Product Quantity

This trigger decrements the qty amount when a product is added to a users cart

```
CREATE TRIGGER `subtract qty` AFTER INSERT ON `cartitems`
 FOR EACH ROW UPDATE products SET qty = qty - 1 WHERE prod_ID = new.prod_ID
```

Sequence/Triggers/Stored Procedures

Trigger 3 - Increment Product Quantity

This trigger increments the qty amount when a product is removed from their cart

```
CREATE TRIGGER `subtract qty` AFTER INSERT ON `cartitems`
 FOR EACH ROW UPDATE products SET qty = qty + 1 WHERE prod_ID = new.prod_ID
```

View

Users can view their own personal order history

CREATE THE VIEW

```
CREATE VIEW userOrders AS

SELECT * FROM orders WHERE user_ID = user_ID;
```

INVOKE THE VIEW

```
SELECT * FROM userOrders WHERE user_ID = ‘$user_ID’;
```

Problems Encountered and Solutions

We had a few problems with the design of the website:

Problem 1: We couldn't search for a product and add it.

Solution: We made a simple form where you can add a product to the cart using the product ID.

Problem 2: We had no login feature for the forms to edit user specific data.

Solution: Our forms can specify the user using email.

Problem 3: Implementation of triggers into the database

Solution: Because we had inexperience adding triggers into a database, we used trial and error, as well as online tutorials to add the triggers that we wanted into the database.

Demo

Questions?