

Project Title:

Fantasy Shop Inventory

Group members:

James Meyer, Jordan Odenthal, Jason Shea, Zoe Zellner, Tony Shelton-McGaha

Description:

The purpose of this database is to keep track of a store's products. This would include the quantity of each product, the pricing of each product, what type of product it is, and where the product came from.

Our product would be used by store managers in order to keep track of stock, know when to add more inventory, and manage loss prevention. They would also be able to add products, or adjust prices.

Our potential customers would be individuals looking online to see prices and stock of what we have in our store, as well as managers who need an easy way to manage inventory. All our items will be shown to them to give them both an understanding of what we are selling and for how much. Users will be able to look through our stock and sort by various categories and prices. The customer will then be able to decide what they want from our store and have the foreknowledge of the website to be able to shop confidently within our store.

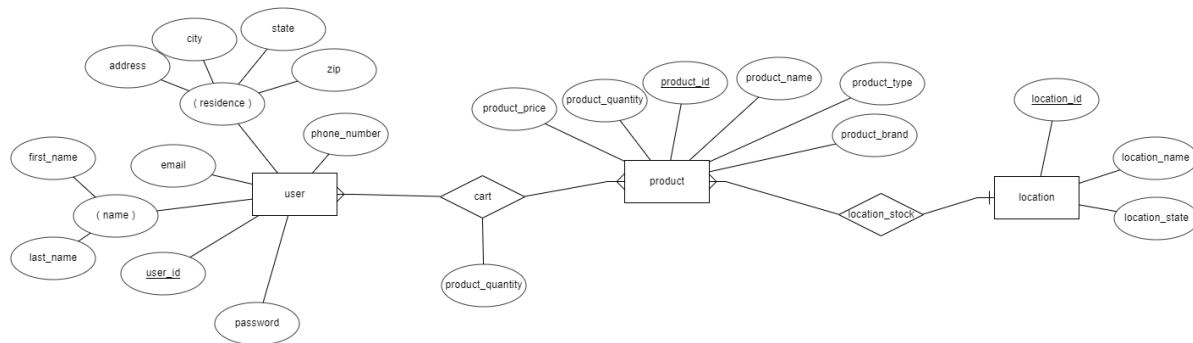
Data requirements of users:

The primary data requirement is the store inventory itself, which will represent all the products, their prices, their quantity, etc. It may also require a table of users. The primary requirements for standard users would be having a decent internet connection, access to a desktop or laptop, and a browser of choice. The primary requirements for an administrator or a manager would be almost identical to the standard user requirements except that they would also require a special URL to make changes to the store inventory.

Functional requirements of users:

The information from the database will be displayed in rows and formatted with HTML/CSS. Along with this we will use PHP to design and create the website. On the side of the screen, we will have a menu that will allow users to sort by item type, quantity, or price. Users will be able to log items that they wish to purchase within the store. This will then be cataloged and then saved, you will then be able to pick up and pay for those items that you saved on the website in our stores location.

2. ER Diagram



3. Explanation of Notation for ER Diagram

We have chosen to use horizontal/vertical lines through our connecting relationships to denote the “one” side of relationships, with the opposite relationship (“many”) being denoted by one line split into 3 (like a crow's foot). For example, the relationship from cart to user is one to many, therefore the line by user would have a perpendicular slash, and the line going to cart would split into three lines. Entity sets are rectangles, and relationship sets are diamonds. Attributes are ovals.

4. Description of 2 relationships of ER Diagram

For the relationship of user to product, each user can buy many products, and each product can be bought by many different users. Thus, this is a many-to-many relationship, which we resolved with a cart table. The cart table keeps track of which user is buying which product, as well as the quantity of that product.

Product to location is a one-to-many relationship. For the purposes of this product, we're assuming that each product is only sold at one location -- our website would serve as a hub for many different fictional retail companies. However, each location can sell many different products. Thus, the relationship between product and location is a one-to-many relationship.

5. Convert ER Diagram to Relational Schemas:

[user]

user_id (**primary**) [int]
first_name [varchar(10)]
last_name [varchar(10)]
email [varchar(20)]
password [varchar(20)]
address [varchar(25)]

city [varchar(20)]
state [varchar(2)]
zip [int(5)]
phone_number[int]

[product]

product_id(**primary**) [int]
product_name [varchar(10)]
product_type [varchar(10)]
product_brand [varchar(10)]
product_quantity [int]
product_price [float(2 decimal places)]
location_id (**foreign key referring to location::location_id**)

[location]

location_id(**primary**) [int]
location_name [varchar(10)]
location_state [varchar(2)]

[cart]

product_quantity
user_id (**primary**)(**foreign key referring to product::product_id**) [int]
product_id (**primary**)(**foreign key referring to product::product_id**)[int]

6. Relational Schema Diagram

