



OCTOBER 12, 2015

E-COMMERCE STORE TO SHOP SHIRTS

SHOPONLINE.HEROBO.COM

BY - MANJUSHA BOLISHETTY
UIN: 922008991



Contents

Introduction	2
Details of the Project Implementation.....	2
Platform and Technologies used.....	2
Details of the Design Phase	2
1. Schema Design and its implementation.....	2
2. Data Collection and Storing in the tables.....	4
3. Details on the application and its functionality.....	4
Assumptions, Constraints and Future scope	5
Conclusion – Learnings	6
Appendixes	7

Introduction

The project is a simple implementation of a database application which is almost very frequently used in our day to day life. The name of the project is Shop Online which is a simple E-commerce site that enables a user to shop shirts online. The site has a variety of shirts, in variety of color and combinations displayed on the website. The user can choose a shirt of his choice and the required size and place an order on the site online. While the user places a request for the order, these details of the order are captured in different tables at the backend. This is a simple user friendly application.

Details of the Project Implementation

Platform and Technologies used

- PHP for building the User Interface
- MySQL server to store the data in the table

Details of the Design Phase

1. Schema Design and its implementation

Once I decided to build an E-commerce site, my next job was to figure out what are the details or information I need to capture in my database, which lead to the motivation of defining my tables and columns in each of my tables. In general for any ecommerce retail store, we need a products table capturing the details of the product, a table which captures more detailed information about each products like sizes of shirts available in this project, an inventory table that tracks the available inventory giving stock number at real time, and a table which contains history or details of the transaction and orders.

The application has 5 table as mentioned below.

- Products
- Sizes
- Products_Sizes
- Inventory
- Orders_Details

At any point of time the products, sizes and the products_sizes table cannot be manipulated by DML operations by a user. Only the owner or the administrator can modify these tables. The snapshot of the inventory and order_details tables changes with activity of each user. Attached is the SQL scripts for the creation of the tables in Appendixes. [1]

Below is a relationship diagram showing the relationship between these 5 tables.

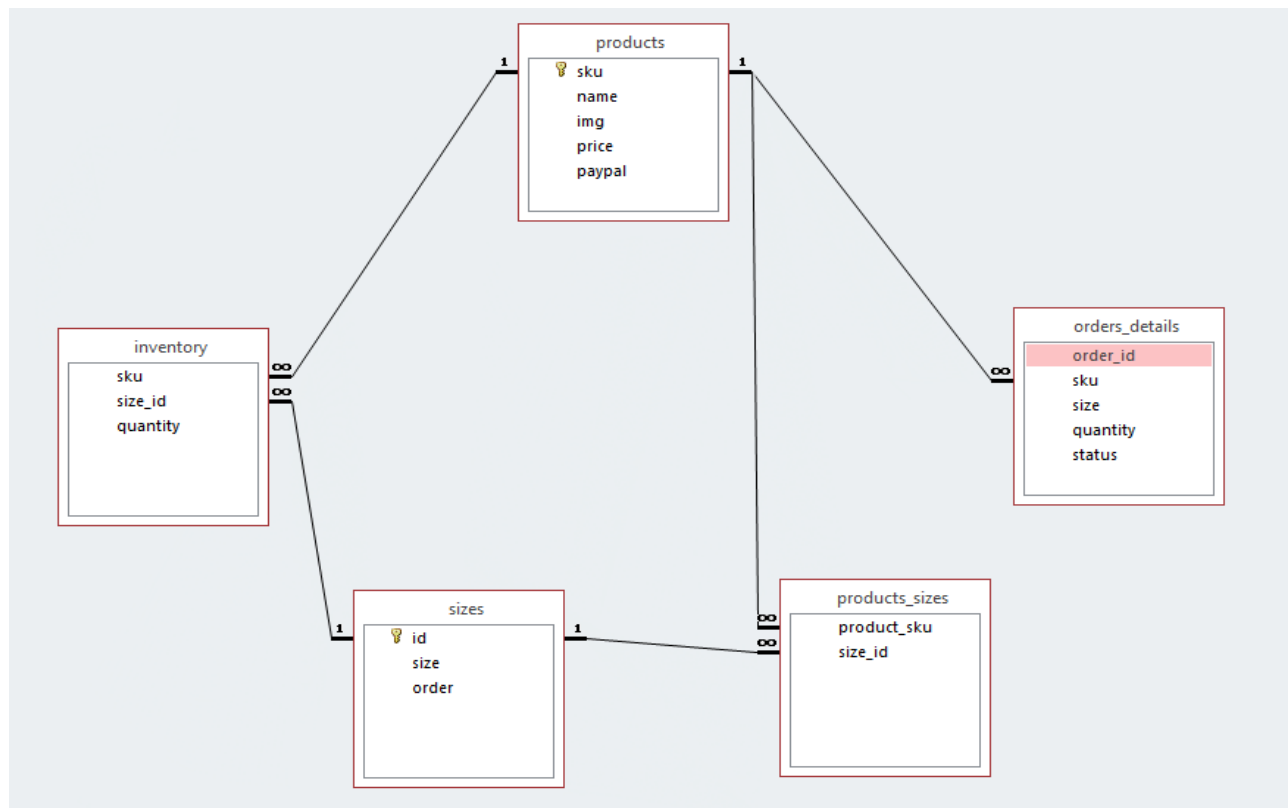


Fig 1. Relationship diagram

2. Data Collection and Storing in the tables

For the application after the design and implementation of the table's next step was the data collection process. For this application I simulated the data considering real world scenarios and day to day experiences of mine, asking myself questions what all are the features I look for when I shop a product online like size availability, pricing of the products. These are some things which a user would like to know even before placing an order. Accordingly I built scripts to insert data into the tables. Some of the data I was able to get from a reference site used to build this website. [3][4]

Attached are the insert scripts for inserting data into the tables. [2]

3. Details on the application and its functionality

Like any other E-commerce store Shop Online gives users the opportunity to view the products and its details online, search for a shirt based on key word, add to cart, update the cart and modify the cart as required, contact the customer service with a mail option. The home page of the application displays the latest shirts. Following are the functionalities of the site.

- *Shirts: View the entire catalogue of shirts.*
- *Add to Cart: Add a shirt of choice to the cart and place order*
- *Search a shirt based on keyword*
- *Contact*

1. *Shirts: View the entire catalogue of shirts.*

The user can click on the shirts link on the homepage to view the entire catalogue of shirts by page.

2. *Add to Cart: Add a shirt of choice to the cart and place order*

In the cart the user can update the quantity of shirts required. Remove the exiting item from the cart or confirm the order. When a user add a shirt to the cart, the details of the item id, quantity and status of shopping are captured in the Orders_Details table. When a user updates quantity, update is made on the Orders_Details table to reflect new quantity. If a user decides not to get the shirts when he removes the

shirt from the cart, a delete operation happens on the Orders_Details table on the record which was inserted when the shirt was added to the cart. In all these three operations the quantity in the Orders_Details might get updated on updating adding a shirt to cart, updating the quantity and deleting an item from cart. Once the user is sure of his choice he can place an order. Again here with this operation, the Orders_Details table undergoes an update on the status column which updates the status to “order placed successfully”. When a product is removed from the cart, the record from Orders_Details table gets deleted. Immediately when a user adds and Item to the cart, the details of this are inserted in Orders_Details table with status “Shopping in progress”.

3. Search a shirt based on keyword

The user can search for a shirt based on key word say may be the user wants to see all green shirts. The application selects all the shirts of green color from the database and displays it to the user.

4. Contact

The users can contact the web admin, in case of any shopping difficulties, suggestion, concerns by providing their name, email id and message.

Assumptions, Constraints and Future scope

1. At any point of time, only one user will be using the site, multiuser session is proposed for the next version of the application.
2. Currently a user can add and check out only one item (of multiple quantities) from the application. Scope to add multiple items in the cart exists and will be taken a future point of time.
3. Integrating Paypal with the existing application to enhance user’s shopping experience.
4. With modification to the shopping experience of the user, the quantity in the inventory table needs to get updated according. Due to time constraints this could not be achieved at this point of time and hence proposed for next phase.

Conclusion – Learnings

There certainly were a lot of hindrances when designing the front end and trying to connect it to the database. The project gave me a good hang of PHP and working with PHP, connecting PHP to MySQL.

Following are the learnings from the project

- One major aspect of best developing practices I learnt in the project was separation of concerns. Try as much to keep concerns of functionalities separate so that at any point of time we want to enhance the application, it is easier and we need not modify the entire code.
- Another point I learnt in the project was avoiding redundant data in tables. Try to normalize the tables even if it means to use joins to access data in them.
- Avoid duplication code in multiple places in code. Try to have one common files which might have common code, structure used across multiple pages and include the common files like header, footer in the other files.
- Avoid adding the credentials to connect to the database in each file. Best is to have them in a config.php file. Same applies to path as well. Set a Root URL as your root or starting path for the files. Anytime we need to update the root path, it is only in this file.
- Before implementing the database design, make sure your schema design looks good and is prefect to start implementing. If not the process of building the database and designing the schema go in phases of rotation and hence might be time consuming.

Appendixes

1. SQL Script – creation of tables.



create_table (1).sql

2. SQL Script – Insert statements.



Insert (2).sql

3. <http://teamtreehouse.com/library/build-a-simple-php-application>
4. <http://teamtreehouse.com/library/using-php-with-mysql>