CIS 451/551 Final Project

Winter 2023

name: Aliya Ware

project title: Aliya's Clothing Store

Connection information

port number: 3185
Hostname: ix-dev.cs.uoregon.edu
guest account login/password: guest
database name: MCS2

project URL: https://ix.cs.uoregon.edu/~aliyaw/finalProject/frontPage.html

Highlights:

- Using the GET operation, the follow up page for browsing all brands to what types of clothes a brand carries takes information from one page and stores it for access on the next.
- If working properly, the create-user page can insert new customers into an already existing list of customers.

Contents

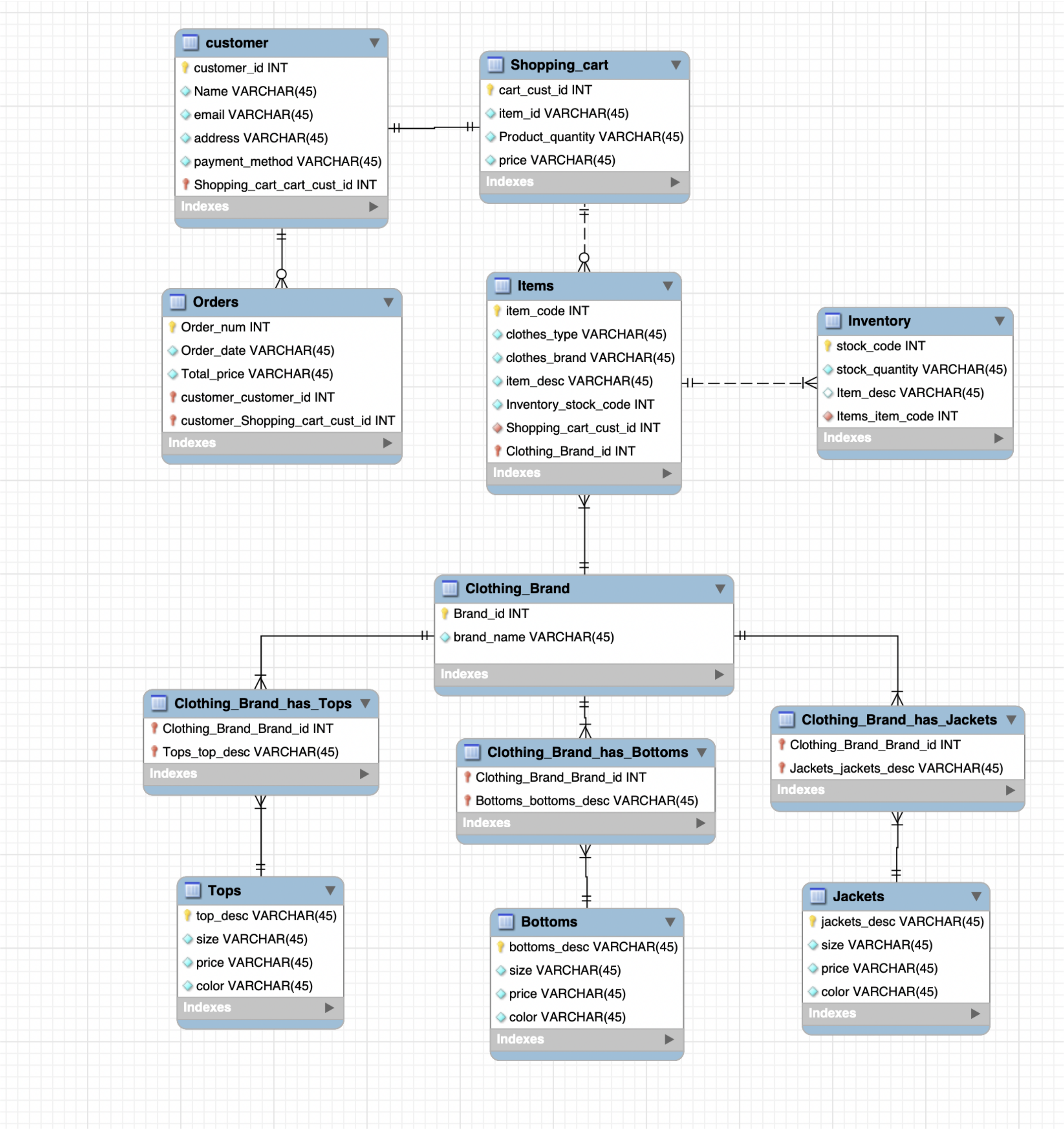
1 Summary

Aliya's Department store is a small clothing store database which I have created, modeling the key features that a department store would have. The main application of this program is to keep track of multiple clothing brands and their products that I sell , store information about my customers, and track orders made by my customers. The website is a standard user-based website with the ability for customers to browse different products which we carry and lets them decide if they want to order the items. They can see the sizes, colors, and prices available as well as specific types of clothing like the types of jackets, shirts, and bottoms we carry (independently of each other).

There are 8 main tables: items, inventory, clothing_brand, tops, bottoms, jackets, customers, orders, and shopping_cart. There are also 3 bridge tables which are Clothing_brand_has_Tops, Clothing_brand_has_Bottoms, and Clothing_brand_has_Jackets. Each table contains the appropriate key attributes which describe the entities. Most of the data being stored is textual content such as descriptions of clothing and customer profiles. Users can browse the site freely like they would an online clothing store.

2 Logical Design

A relational diagram of my database, made in Crow's Foot notation, can be seen on the next page.


customer
customer_id INT
Name VARCHAR(45)
email VARCHAR(45)
address VARCHAR(45)
payment_method VARCHAR(45)
Shopping_cart_cart_cust_id INT
Indexes
Shopping_cart
cart_cust_id INT
item_id VARCHAR(45)
Product_quantity VARCHAR(45)
price VARCHAR(45)
Indexes
Orders
Order_num INT
Order_date VARCHAR(45)
Total_price VARCHAR(45)
customer_customer_id INT
customer_Shopping_cart_cust_id INT
Indexes
Items
item_code INT
clothes_type VARCHAR(45)
clothes_brand VARCHAR(45)
item_desc VARCHAR(45)
Inventory_stock_code INT
Shopping_cart_cust_id INT
Clothing_Brand_id INT
Indexes
Inventory
stock_code INT
stock_quantity VARCHAR(45)
Item_desc VARCHAR(45)
Items_item_code INT
Indexes
Clothing_Brand
Brand_id INT
brand_name VARCHAR(45)
Indexes
Clothing_Brand_has_Tops
Clothing_Brand_Brand_id INT
Tops_top_desc VARCHAR(45)
Indexes
Clothing_Brand_has_Bottoms
Clothing_Brand_Brand_id INT
Bottoms_bottoms_desc VARCHAR(45)
Indexes
Clothing_Brand_has_Jackets
Clothing_Brand_Brand_id INT
Jackets_jackets_desc VARCHAR(45)
Indexes
Tops
top_desc VARCHAR(45)
size VARCHAR(45)
price VARCHAR(45)
color VARCHAR(45)
Indexes
Bottoms
bottoms_desc VARCHAR(45)
size VARCHAR(45)
price VARCHAR(45)
color VARCHAR(45)
Indexes
Jackets
jackets_desc VARCHAR(45)
size VARCHAR(45)
price VARCHAR(45)
color VARCHAR(45)
Indexes

3 Physical database design

The following relationships exist in the system:

Customer Table:

A **customer** has a unique **customer id,** as well as a **name, email, shipping address,** and a **payment method.** They can place **Orders** and add to their **shopping cart.**

Order Table:

Each **Order** has a unique **order_num** It also has an **order_date** and **total_price.**

Shopping_Cart Table:

Each **shopping cart** has a **customer_cart_id,** unique to each customer. A shopping cart includes **item id, product quantity,** and a **total price.**

Items Table:

All **Items** have an **item_code,** unique to each item. All items are different types of clothing so they have the following attributes: **clothes_type, clothes_brand, item_description,** and an **inventory_stock_code.** Items can be added to a cart and they can range from shirts to jackets of any brand. **Items** have foreign keys **shopping_cart_cust_id** and **Clothing_brand_id.**

Inventory Table:

Inventory keeps track of how many items of each clothing type and brand are in stock. It has a primary key **stock_code** and attributes **stock_quantity** and **item_desc.** It also has a reference to **Items,** as they are related through the **item_code.**

Clothing_Brand Table:

The **Clothing_Brand** entity keeps track of all the brands we carry in store. It is identified by primary key **Brand_id** to uniquely identify the brand and has the attribute **brand_name** which customers should be able to recognize**.**

Tops Table:

The **Tops** table is where I store every type of shirt that is for sale. The primary key is **top_desc** because tops can be categorized into the unique value of T-shirts, Tank-tops, button-ups, etc. This entity has **size**, **price**, and **color** attributes just like an online clothing store would.

Bottoms Table:

The **Bottoms** table is where I store every type of pant/skirt that is for sale. The primary key is **bottom_desc** because each type of bottom is unique (ex: jeans, jean-shorts, skirts, leggings, etc). For the same reason as above, this entity has **size**, **price**, and **color** attributes.

Jackets Table:

The **Jackets** table is where I store every type of pant/skirt that is for sale. The primary key is **jackets_desc** because each type of jacket is unique (ex: sweatshirt, zip-up, puffer, etc). For the same reason as above, this entity has **size**, **price**, and **color** attributes.

Clothing_Brand_has_Tops Table:

The **Clothing_Brand_has_Tops** table is dependent on the **Clothing_Brand** and **Tops** Primary Keys. This table is intended to make a many to many relationship between a clothing brand and the tops it carries.

Clothing_Brand_has_Bottoms Table:

The **Clothing_Brand_has_Bottoms** table is dependent on the **Clothing_Brand** and **Bottoms** Primary Keys. This table is intended to make a many to many relationship between a clothing brand and the bottoms it carries.

Clothing_Brand_has_Jackets Table:

The **Clothing_Brand_has_Jackets** table is dependent on the **Clothing_Brand** and **Jackets** Primary Keys. This table is intended to make a many to many relationship between a clothing brand and the jackets it carries.

4 List of applications

Main Application Summaries:

See Brands and Items

This application allows users to see all the brands that the department store carries. From this page, users will be able to see more information on individual brands and go to checkout if they like an item that they see. From the Home page, a user can select to look at all of the items in the department store and return to the homepage whenever.

Browse Clothes From Brand Or Just Clothing Types

My application allows users to browse available clothing items per brand and in general. A user can specifically see what a particular brand carries in regards to shirts, bottoms, and jackets. The customer can also decide to just look at shirts, bottoms, or jackets independent of each other. They can also see every item that the store could potentially carry.

Checkout

** Did not have time to complete this ** When a user explores the different brands, they have the option to add that brand's product to their cart. Once they do this, they can "check out" which will add the item to the user's list of orders.

Create User

When you get to the home page, there is an option to create a new account. If a user navigates to this page, they will be prompted to enter in their name, email, address, and a payment method, just as a real online store would ask. The user will be added to the list and can continue shopping.

5 User's guide

Go to https://ix.cs.uoregon.edu/~aliyaw/finalProject/frontPage.html and begin navigating the page by clicking any blue highlighted hyperlinks.

6 Table of contents

https://github.com/aliyaaware/CS451-DatabasesFinalProject/blob/main/test2.sql

7 Implementation Code

Implementation code is found here: https://github.com/aliyaaware/CS451-DatabasesFinalProject or iIt is linked under every section that says "Code" on the website.

8 Conclusion

This project challenged me in many new ways and got me much more familiar with database design and I learned how to use PHP and MySQL. For the future, I know how to design a better data-base from the start. In this project, I struggled designing and implementing the database, as I didn't have a full grasp on the importance of getting your entities and information right. I had to redesign the database a couple of times because of missing attributes and not naming my attributes well. Also, having never used PHP before, I was learning as I was going so some of my code could have been implemented in a cleaner way.

I finished a lot of the basic design of my website and was able to make the front page look organized. For future work, I would like to have been able to add more features that would keep track of each user and their orders. I wanted to make it so if you go to a user's profile, you could see a list of what they had purchased and the total price. I would also have liked to make each page look more organized, however since I don't know HTML or CSS very well it was a challenge for me to do in the time we had. Finally, I would have liked more time to get comfortable with PHP, however this project challenged me in a way that forced me to get better very quickly which I liked.