ERD PAGE FROM: PROJECT IDEATION (1) and USER STORIES (2)

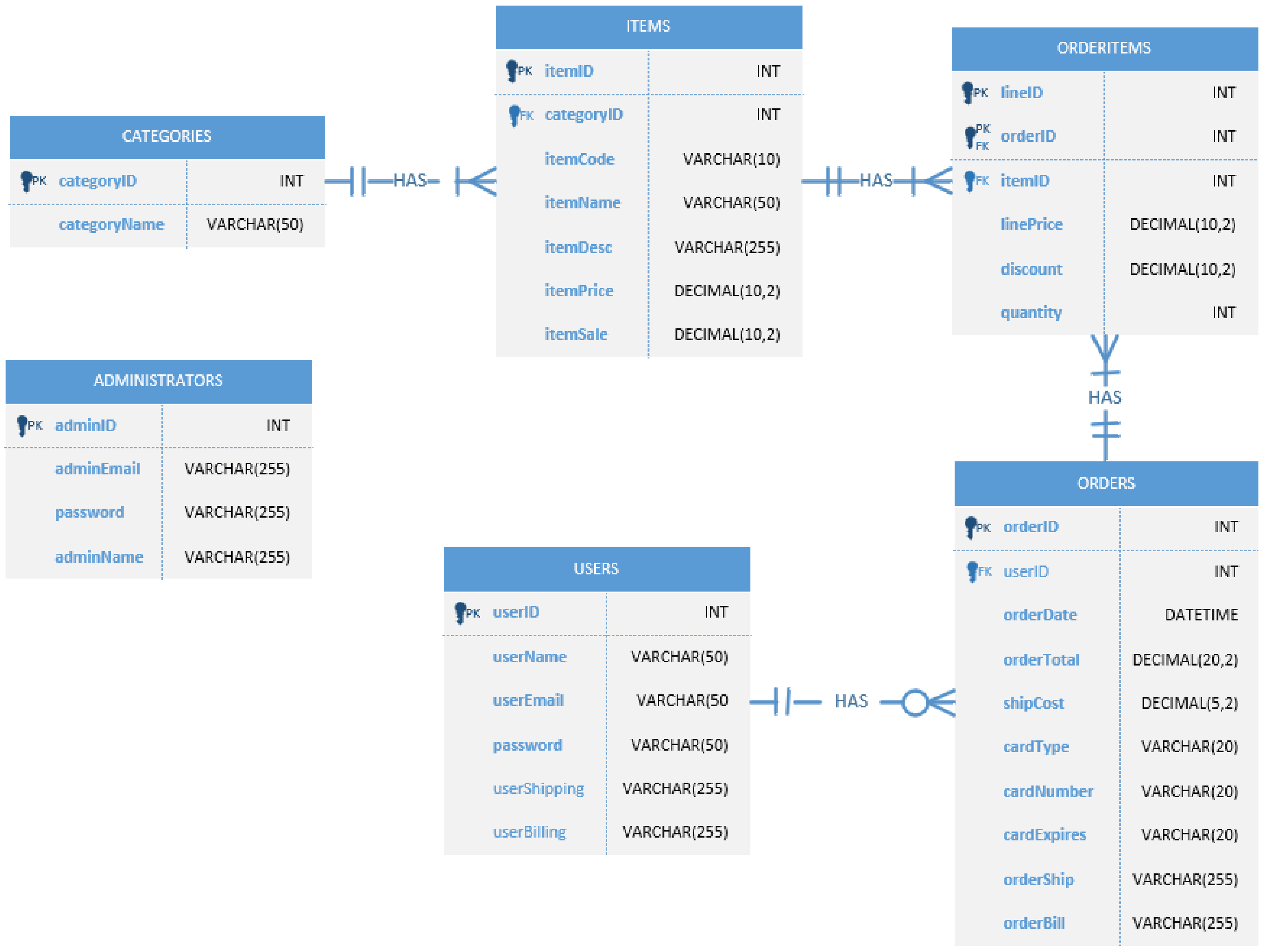
The many-to-many (M:N) relationship in my project is between ORDERS and ITEMS: each order can contain many items, and each item can be found in many orders. I will use an association table named ORDERITEMS to resolve this.

M:N visual (below)

1 ORDER, M ITEMS

M ORDERS, 1 ITEM

MY ENTITY RELATIONSHIP DIAGRAM:



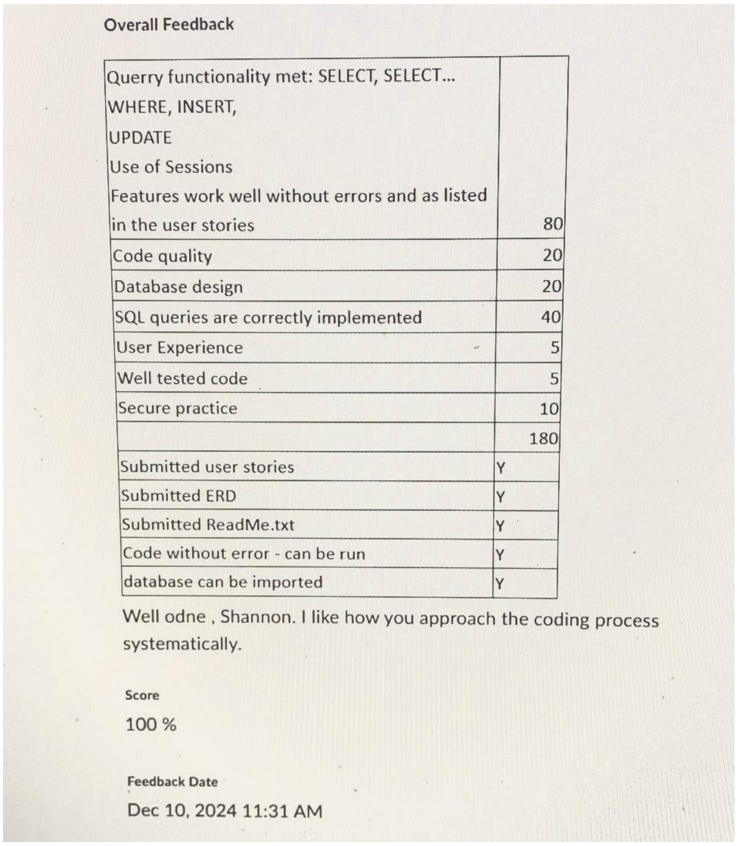
BUSINESS RULES:

a. Each CATEGORY must contain at least one ITEM.
   Each ITEM can only belong to one CATEGORY.
b. One ITEM can be in Many ORDERs.
   Many ITEMs are placed in one ORDER. Resolved.
c. A USER can place Many ORDERs.
   Each ORDER is placed by One USER only.
   Placing an ORDER is Optional.
d. One USER can checkout with Many ITEMs.
   Many USERs checkout with only One ITEM.

ERD PAGE FROM: PROJECT IDEATION (1) and USER STORIES (2)

Project Summary:

My project is an e-commerce platform that sells clothing items. Data associated with item inventory, users/customers, and order details are kept in a database to store, process, and access. It allows all casual users (registered and guest) to view items, add items to their cart, and place orders if at least one item is added to their cart and valid payment is provided. Their cart will be stored in a session. Items are sorted by category for convenient viewing. Registered users log in with an account using verifiable credentials, receive free membership shipping on all orders, and can view their personal order history. Registered users can also edit the information on their profile. Guest users only have permission to view items, add items to their cart, and place orders. Anyone who does not have a pre-existing account, guest or otherwise, can create one to become (and receive all the benefits of) a registered user. Administrators are given minimal restrictions in their ability to modify and maintain the database, having the full range of permissions for this project: select, update, insert. None of the users can delete anything from the database through this webapp, because it is outside the scope for this project's requirements.



Overall Feedback

Querry functionality met: SELECT, SELECT... WHERE, INSERT, UPDATE Use of Sessions Features work well without errors and as listed in the user stories	80
Code quality	20
Database design	20
SQL queries are correctly implemented	40
User Experience	5
Well tested code	5
Secure practice	10
	180
Submitted user stories	Y
Submitted ERD	Y
Submitted ReadMe.txt	Y
Code without error - can be run	Y
database can be imported	Y

Well odne , Shannon. I like how you approach the coding process systematically.

Score

100 %

Feedback Date

Dec 10, 2024 11:31 AM