Gingerbread Standard Project ERD

Carts

cart_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each cart
user_id	INT NOT NULL	Foreign key referencing Accounts(user_id) (who owns the cart)

add the price to the cart_product so that the price can be updated daily

Cart_Products

cart_id	INT NOT NULL	Foreign key referencing Carts(cart_id), linking the cart to a user
product_id	INT NOT NULL	Foreign key referencing Products(product_id), identifying the product
quantity	INT NOT NULL DEFAULT 1	Number of units of the specific product in the cart
seller_id	INT NOT NULL	Foreign key referencing Products(product_id), identifying the purchased product.
latest_price	DECIMAL(10,2) NOT NULL	Latest price of product added to the cart

add a new primary key to the inventory table

Inventory

inventory_id	INT NOT NULL	Unique identifier for each inventory record
seller_id	INT NOT NULL	Foreign key referencing Accounts(user_id), identifying the seller
owner_id	INT NOT NULL	Foreign key referencing Accounts(user_id), identifying the owner of the product
product_id	INT NOT NULL	Foreign key referencing Products(product_id), identifying the product
quantity	INT NOT NULL DEFAULT 0	Number of available units in stock
unit_price	INT NOT NULL	Price per unit of the product

Products_Categories

category_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each product category (system-assigned)
category_name	VARCHAR(100) NOT NULL UNIQUE	Name of the product category (e.g., Electronics, Clothing) for users to search

Products

product_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each product (system-assigned)
category_id	INT NOT NULL	Foreign key referencing Categories(category_id), identifying the product category
product_name	VARCHAR(100) NOT NULL	Name of the product, must be unique to search
description	TEXT	Detailed description of the product
image	VARCHAR(255)	URL of the product image
owner_id	INT NOT NULL	Foreign key referencing Accounts(userid), indicating the owner of the product

also add a foreign key to the products table to distinguish between all the sellers and the owner of the product who can make the changes to the product details

Product Reviews

product_review_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each review for product
user_id	INT NOT NULL	Foreign key referencing Accounts(user_id), identifying the reviewer/user
comment	TEXT NOT NULL	Text comment for the product
review_date	DATETIME DEFAULT CURRENT_TIMESTAMP	Price per unit of the product
product_id	INT or NULL	Foreign key referencing Products(product_id), if reviewing a product
rating	INT NOT NULL	Rating score (1-5)

separate the reviews table from products and sellers

Seller Reviews

seller_review_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each review for seller
user_id	INT NOT NULL	Foreign key referencing Accounts(user_id), identifying the reviewer/user
comment	TEXT NOT NULL	Text comment for seller
review_date	DATETIME DEFAULT CURRENT_TIMESTAMP	Price per unit of the product
seller_id	INT or NULL	Foreign key referencing Sellers(seller_id), if reviewing a seller
rating	INT NOT NULL	Rating score (1-5)

Accounts

user_id	INT PRIMARY KEY	Unique identifier for each user (system assigned, cannot be updated)
email	VARCHAR(255) UNIQUE NOT NULL	User's email address (must be unique)
last_name	VARCHAR(100) NOT NULL	User's last name
first_name	VARCHAR(100) NOT NULL	User's first name
address	TEXT	User's address
password	VARCHAR(255) NOT NULL	Hashed password for authentication
current_balance	DECIMAL(10,2) DEFAULT 0.00	User's current account balance
is_seller	BOOLEAN DEFAULT FALSE	Indicates if the user is a seller (TRUE/FALSE)

separate the big order to different sellers, so that the seller can manage the smaller orders

Orders (separated by seller)

order_id	INT PRIMARY KEY AUTO_INCREMENT	Unique identifier for each order, auto-incremented
seller_id	INT NOT NULL	Foreign key referencing Products(product_id), identifying the purchased product. **Each order is split by seller_id**, meaning each seller has a separate order record
buyer_id	INT NOT NULL	Foreign key referencing Accounts(userid), indicating the buyer
total_amount	DECIMAL(10,2) NOT NULL	Total order amount, stored with two decimal places
order_date	DATETIME DEFAULT CURRENT_TIMESTAMP	Timestamp of when the order was placed, default is the current time
num_products	INT NOT NULL	Number of products in the order
order_status	ENUM('Unfulfilled', 'Fulfilled') NOT NULL DEFAULT 'Unfulfilled'	Order status: 'Unfulfilled' (not yet processed) or 'Fulfilled' (completed when all the status of order_item associated with the same order_id are completed)

Orders_Products

order_id	INT NOT NULL	Foreign key referencing Orders(orderid), linking items to an order
product_id	INT NOT NULL	Foreign key referencing Products(product_id), identifying the purchased product
quantity	INT NOT NULL	Number of units of the specific product in this order
price	DECIMAL(10,2) NOT NULL	Price per unit at the time of purchase
seller_id	INT NOT NULL	Foreign key referencing Accounts(userid), indicating the seller
status	ENUM('Unfulfilled', 'Fulfilled') NOT NULL DEFAULT 'Unfulfilled'	Status of the order item (whether it has been completed)
fulfillment_date	DATETIME	Timestamp of when the order was fulfilled

add fulfillment_date to record the date that the orders are fulfilled, default value should be none.