ANP -C7971

StudentID:AF0401622

Name:Mallepelly vaishnavi

Database Design For

Super Market Billing System

SUPER MARKET BILLING SYSTEM

The **Supermarket Billing System** is a software application designed to facilitate the sales and management processes in a supermarket. This system helps manage various operations, including user roles such as administrators, sellers, and customers, as well as tracking products and orders. By utilizing this system, supermarkets can streamline their billing process, improve customer service, and efficiently manage their inventory.

Main Objective

The main objectives of the Supermarket Billing System include:

1. **Transaction Processing**: To enable quick and accurate billing of customer purchases, ensuring a smooth checkout experience.
2. **User Management**: To provide a secure and organized way to manage users, allowing admins to oversee sellers and maintain their accounts easily.
3. **Inventory Management**: To track product availability and stock levels automatically as sales occur, helping prevent stockouts and overstock situations.
4. **Sales Tracking and Reporting**: To capture sales data for analysis, enabling the supermarket to assess performance, manage inventory effectively, and identify popular products.
5. **Enhanced Customer Experience**: To improve customer satisfaction by streamlining the purchasing process and allowing for easy management of customer interactions.

System comprises several key components:

- **Admin**: The admin oversees the system, managing seller accounts and product listings.
- **Seller**: Sellers are responsible for processing customer transactions at the checkout.
- **Customer**: Customers interact with the system to make purchases, providing their details for billing.
- **Order**: The order component captures all details related to a customer's purchase, linking it to the seller and customer.
- **Product**: The product table stores information about the items available for sale, including their prices and stock levels.

1. Admin

- **Columns**:
  - id
  - username
  - password
  - email

2. Seller

- **Columns**:
  - id
  - admin_id
  - username
  - password

3. Customer

- : **Columns**
    - id
    - name
    - phone
    - email

4. Product

- **Columns**:
    - id
    - name
    - price
    - stock_quantity

5. Order

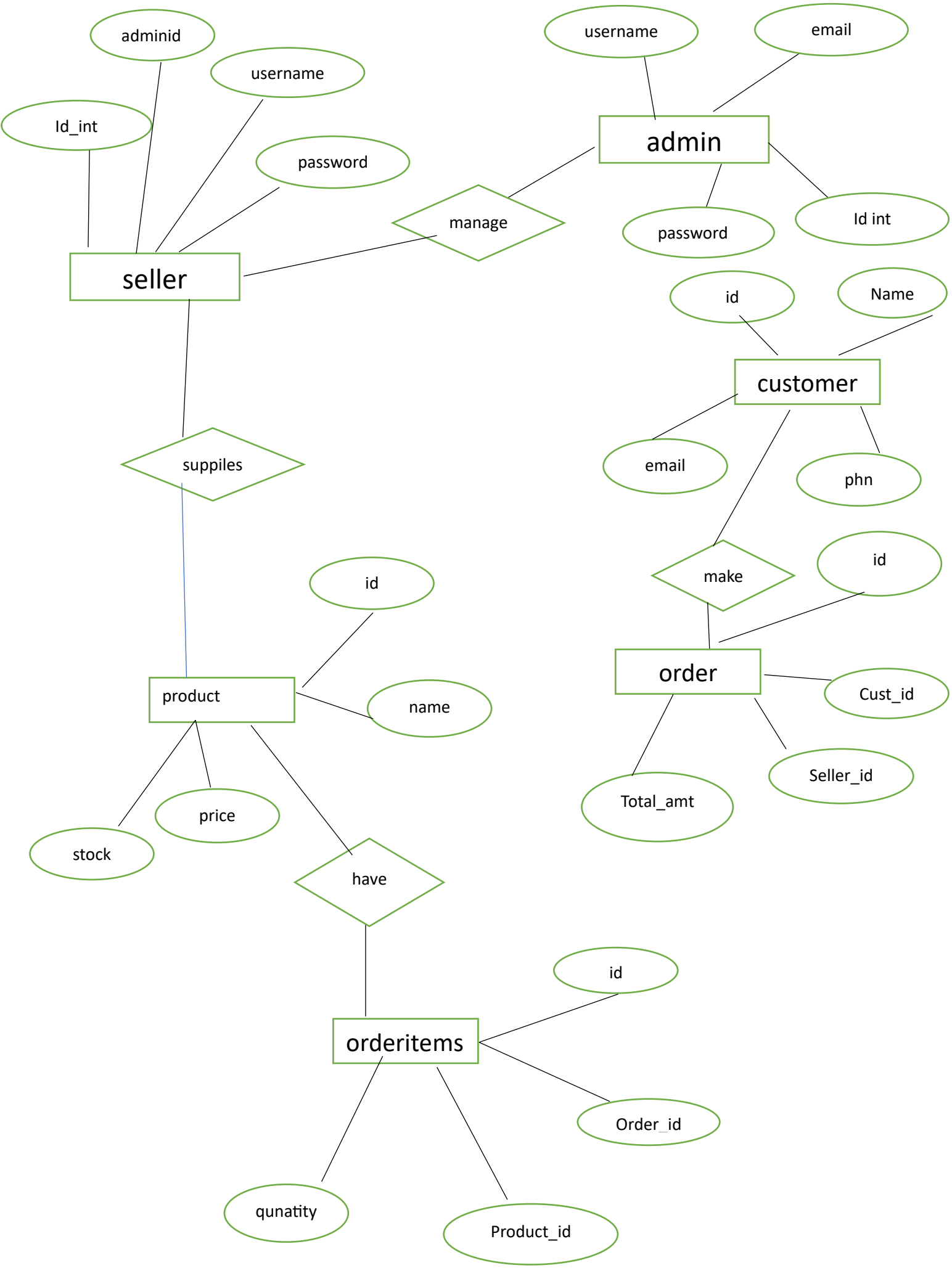
- **Columns**:
    - id
    - customer_id
    - seller_id
    - total_amount

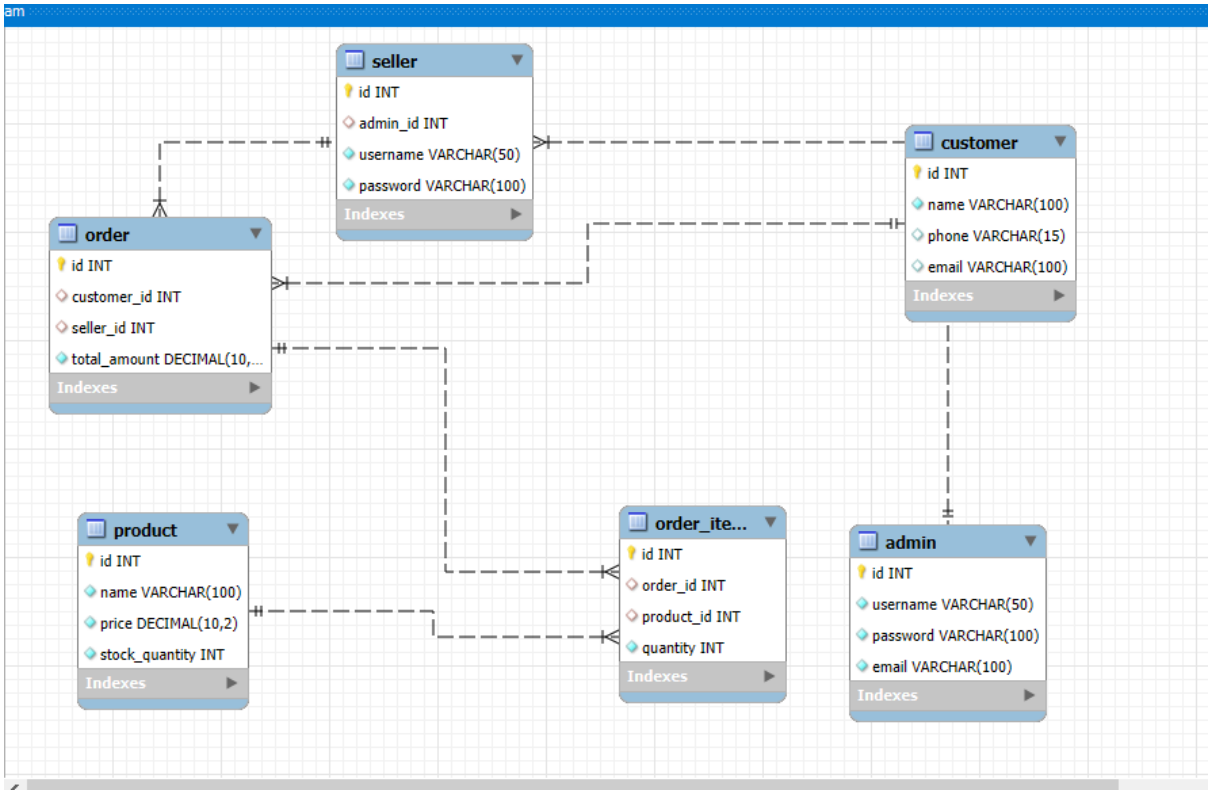
6. Order Items

- **Columns**:
    - id
    - order_id
    - product_id
    - quantity

Explanation of the Entity Relationships

- **Admin to Seller**:
    - An admin can manage multiple sellers, establishing a one-to-many relationship.
- **Seller to Order**:
    - A seller can handle multiple orders, creating a one-to-many relationship.
- **Customer to Order**:
    - A customer can place multiple orders, establishing a one-to-many relationship.
- **Order to Order Items**:
    - An order can contain multiple items, leading to a one-to-many relationship between orders and order items.
- **Product to Order Items**:
    - A product can be part of multiple order items, establishing a one-to-many relationship.


adminid
username
Id_int
password
seller
manage
username
email
admin
password
Id int
id
Name
customer
email
phn
suppiles
make
id
id
product
name
order
Cust_id
price
stock
have
Total_amt
Seller_id
id
orderitems
Order_id
qunatity
Product_id


am
seller
id INT
admin_id INT
username VARCHAR(50)
password VARCHAR(100)
Indexes
customer
id INT
name VARCHAR(100)
phone VARCHAR(15)
email VARCHAR(100)
Indexes
order
id INT
customer_id INT
seller_id INT
total_amount DECIMAL(10,...
Indexes
product
id INT
name VARCHAR(100)
price DECIMAL(10,2)
stock_quantity INT
Indexes
order_ite...
id INT
order_id INT
product_id INT
quantity INT
Indexes
admin
id INT
username VARCHAR(50)
password VARCHAR(100)
email VARCHAR(100)
Indexes