Online Shopping for Gadgets

Domain Description

In today's tech-driven world, gadgets have become an integral part of daily life. Gadgets could encompass various items like smartphones, laptops, tablets, smartwatches, etc. The system will include functionalities for users to browse, search, purchase, and manage orders of these gadgets.

Relevance

- Convenience: Consumers appreciate the convenience of browsing and purchasing gadgets online from the comfort of their homes or on-the-go.

- Wider Selection: Online platforms offer a broader range of gadgets from various brands, often presenting a more extensive selection than brick-and-mortar stores.

- Accessibility to Information: Detailed product information, specifications, user reviews, and expert opinions are readily available online.

- Competitive Pricing and Discounts: Online shopping platforms frequently offer competitive pricing due to reduced overhead costs compared to traditional retail stores.

- Convenient Payment and Delivery Options: Flexible payment methods and various delivery options, including fast shipping or store pickups, enhance the shopping experience.

- Tech Enthusiasts and Early Adopters: Gadgets attract a particular group of consumers, including tech enthusiasts and early adopters.

- Customer Reviews and Recommendations: Online shopping allows customers to access user-generated reviews and recommendations, influencing their purchasing decisions.

- Global Accessibility: Online gadget stores often provide global accessibility, enabling customers from various locations to access and purchase gadgets from anywhere in the world.

- Personalization and Recommendations: Many online platforms use algorithms and user behavior data to personalize product recommendations.

Table Description

Role Table

Attribute	Data Type	Constraint
RoleID	Integer	Primary Key
Role	String	Not Null, Unique

User Table

Attribute	Data Type	Constraint
UserID	Integer	Primary Key
Username	String	Not Null, Unique
Email	String	Not Null, Unique
Password	String	Not Null

UserRole Table

Attribute	Data Type	Constraint
UserID	Integer	Foreign Key (User(UserID))
RoleID	Integer	Foreign Key (Role(RoleID))

Login Table

Attribute	Data Type	Constraint
LoginID	Integer	Primary Key
UserID	Integer	Foreign Key (User(UserID))
LoginTime	DateTime	Not Null
LogoutTime	DateTime	

Seller Table

Attribute	Data Type	Constraint
SellerID	Integer	Primary Key
CompanyName	String	Not Null
CompanyAddress	String	Not Null
Phone	String	Not Null

Customer Table

Attribute	Data Type	Constraint
UserID	Integer	Foreign Key (User(UserID))
CustomerID	Integer	Primary Key
Address	String	Not Null
Phone	String	Not Null

Category Table

Attribute	Data Type	Constraint
CategoryID	Integer	Primary Key
CategoryName	String	Not Null, Unique
Description	String	Not Null

Products Table

Attribute	Data Type	Constraint
ProductID	Integer	Primary Key
ProductName	String	Not Null, Unique
CategoryID	Integer	Foreign Key (Category(CategoryID))
SellerID	Integer	Foreign Key (Seller(SellerID))
UnitPrice	Decimal	Not Null

Stock Table

Attribute	Data Type	Constraint
StockID	Integer	Primary Key
ProductID	Integer	Foreign Key (Products(ProductID))
Quantity	Integer	Not Null

Orders Table

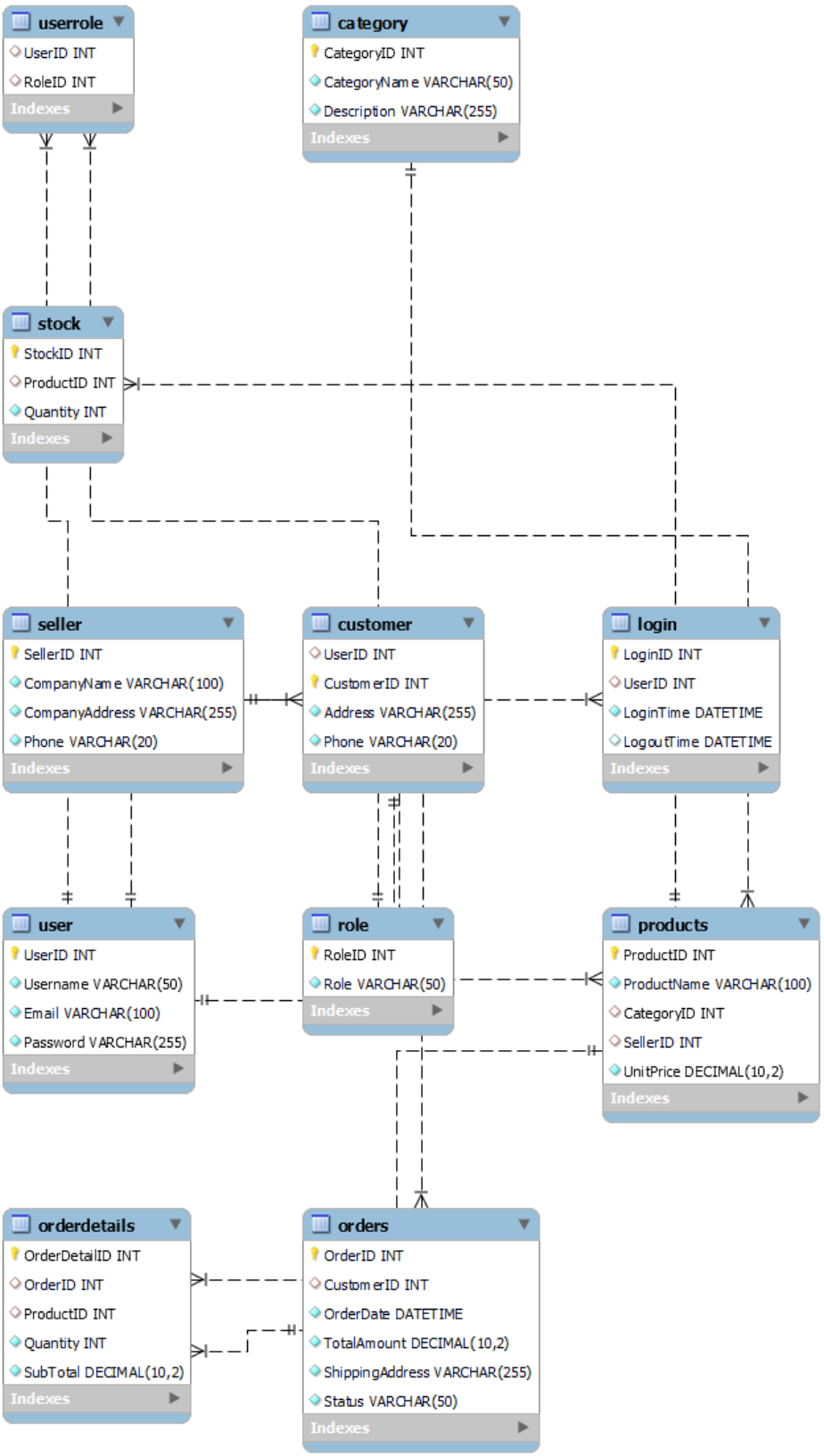
Attribute	Data Type	Constraint
OrderID	Integer	Primary Key

CustomerID	Integer	Foreign Key (Customer(CustomerID))
OrderDate	DateTime	Not Null
TotalAmount	Decimal	Not Null
ShippingAddress	String	Not Null
Status	String	Not Null

OrderDetails Table

Attribute	Data Type	Constraint
OrderDetailID	Integer	Primary Key
OrderID	Integer	Foreign Key (Orders(OrderID))
ProductID	Integer	Foreign Key (Products(ProductID))
Quantity	Integer	Not Null
SubTotal	Decimal	Not Null

ER Diagram



Relationship

- User - Role (One-to-Many):

  One user can have multiple roles, but each role is associated with only one user.

- User - Login (One-to-Many):

  One user can have multiple login instances, but each login is associated with only one user.

- Seller - Products (One-to-Many):

  One seller can offer multiple products, but each product belongs to only one seller.

- Customer - Orders (One-to-Many):

  One customer can place multiple orders, but each order is associated with only one customer.

- Category - Products (One-to-Many):

  One category can contain multiple products, but each product belongs to only one category.

- Products - Stock (One-to-One/One-to-Many):

  Each product may have stock information associated with it.

- Orders - OrderDetails (One-to-Many):

  One order can contain multiple order details, specifying different products within that order.

SQL Queries

```
create database gadgetshopping;
use gadgetshopping;
CREATE TABLE Role (
   RoleID INT PRIMARY KEY,
   Role VARCHAR(50) NOT NULL UNIQUE
);
CREATE TABLE User (
   UserID INT PRIMARY KEY,
   Username VARCHAR(50) NOT NULL UNIQUE,
   Email VARCHAR(100) NOT NULL UNIQUE,
   Password VARCHAR(255) NOT NULL
);
CREATE TABLE UserRole (
   UserID INT,
   RoleID INT,
   FOREIGN KEY (UserID) REFERENCES User(UserID),
   FOREIGN KEY (RoleID) REFERENCES Role(RoleID)
);
CREATE TABLE Login (
   LoginID INT PRIMARY KEY,
   UserID INT,
   LoginTime DATETIME NOT NULL,
   LogoutTime DATETIME,
   FOREIGN KEY (UserID) REFERENCES User(UserID)
);
CREATE TABLE Seller (
   SellerID INT PRIMARY KEY,
   CompanyName VARCHAR(100) NOT NULL,
```

```
    CompanyAddress VARCHAR(255) NOT NULL,
    Phone VARCHAR(20) NOT NULL
);
CREATE TABLE Customer (
    UserID INT,
    CustomerID INT PRIMARY KEY,
    Address VARCHAR(255) NOT NULL,
    Phone VARCHAR(20) NOT NULL,
    FOREIGN KEY (UserID) REFERENCES User(UserID)
);
CREATE TABLE Category (
    CategoryID INT PRIMARY KEY,
    CategoryName VARCHAR(50) NOT NULL UNIQUE,
    Description VARCHAR(255) NOT NULL
);
CREATE TABLE Products (
    ProductID INT PRIMARY KEY,
    ProductName VARCHAR(100) NOT NULL UNIQUE,
    CategoryID INT,
    SellerID INT,
    UnitPrice DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (CategoryID) REFERENCES Category(CategoryID),
    FOREIGN KEY (SellerID) REFERENCES Seller(SellerID)
);
CREATE TABLE Stock (
    StockID INT PRIMARY KEY,
    ProductID INT,
    Quantity INT NOT NULL,
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
```

```
);
CREATE TABLE Orders (
    OrderID INT PRIMARY KEY,
    CustomerID INT,
    OrderDate DATETIME NOT NULL,
    TotalAmount DECIMAL(10, 2) NOT NULL,
    ShippingAddress VARCHAR(255) NOT NULL,
    Status VARCHAR(50) NOT NULL,
    FOREIGN KEY (CustomerID) REFERENCES Customer(CustomerID)
);
CREATE TABLE OrderDetails (
    OrderDetailID INT PRIMARY KEY,
    OrderID INT,
    ProductID INT,
    Quantity INT NOT NULL,
    SubTotal DECIMAL(10, 2) NOT NULL,
    FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),
    FOREIGN KEY (ProductID) REFERENCES Products(ProductID)
);
```

Normalized Tables

Role Table

RoleID	Role
1	Admin
2	Seller
3	Customer

User Table

UserID	Username	Email	Password
1	admin1	admin@example.com	adminpass
2	seller1	seller@example.com	sellerpass
3	customer1	customer@example.com	customerpass

User Role Table

UserID	RoleID
1	1
2	2
3	3

Login Table

LoginID	UserID	LoginTime	LogoutTime
1	1	2023-12-20 09:00:00	2023-12-20 17:00:00
2	2	2023-12-20 10:00:00	2023-12-20 16:30:00
3	3	2023-12-20 12:00:00	2023-12-20 15:45:00

Seller Table

SellerID	CompanyName	CompanyAddress	Phone
1	SellerCo	Seller Address	123-456-7890
2	TechBiz	Tech Address	987-654-3210

Customer Table

UserID	CustomerID	Address	Phone
3	1	Customer Address	555-123-4567

Category Table

CategoryID	CategoryName	Description
1	Electronics	Electronics Category
2	Gadgets	Gadgets Category

Products Table

ProductID	ProductName	CategoryID	SellerID	UnitPrice
1	Laptop Pro	1	1	999.99
2	Smartphone X	1	2	599.99
3	Tablet Mini	1	1	399.99

Stock Table

StockID	ProductID	Quantity
1	1	50
2	2	100
3	3	75

Orders Table

OrderID	CustomerID	OrderDate	TotalAmount	ShippingAddress	Status
1	1	2023-12-20 13:00:00	999.99	Shipping Address	Shipped
2	1	2023-12-20 14:30:00	1199.97	Shipping Address	Delivered

OrderDetails Table

OrderDetailID	OrderID	ProductID	Quantity	SubTotal
1	1	1	1	999.99
2	2	2	2	1199.97