SQL\_Database

Here's the generated database schema in SQL syntax:  
  
```sql  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Email VARCHAR(100),  
 PasswordHash VARCHAR(255),  
 Address VARCHAR(200),  
 City VARCHAR(50),  
 State VARCHAR(2),  
 Zipcode VARCHAR(10),  
 Country VARCHAR(50),  
 Phone VARCHAR(20)  
);  
  
CREATE TABLE Products (  
 ProductID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Description TEXT,  
 Price DECIMAL(10,2),  
 ImageURL VARCHAR(200),  
 Category VARCHAR(50),  
 SubCategory VARCHAR(50)  
);  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status VARCHAR(50),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
CREATE TABLE OrderItems (  
 OrderItemID INT PRIMARY KEY,  
 OrderID INT,  
 ProductID INT,  
 Quantity INT,  
 UnitPrice DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
CREATE TABLE Payments (  
 PaymentID INT PRIMARY KEY,  
 OrderID INT,  
 PaymentMethod VARCHAR(50),  
 PaymentDate DATE,  
 Amount DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
  
CREATE TABLE Shippings (  
 ShippingID INT PRIMARY KEY,  
 OrderID INT,  
 ShipperName VARCHAR(50),  
 ShippingAddress VARCHAR(200),  
 ShippingCity VARCHAR(50),  
 ShippingState VARCHAR(2),  
 ShippingZipcode VARCHAR(10),  
 ShippingCountry VARCHAR(50),  
 EstimatedDeliveryDate DATE,  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
```  
  
This database schema includes the following tables:  
  
\* `Customers`: stores customer information such as name, email, password hash, address, city, state, zip code, country, and phone number.  
\* `Products`: stores product information such as name, description, price, image URL, category, and subcategory.  
\* `Orders`: stores order information such as order ID, customer ID, order date, total cost, and status.  
\* `OrderItems`: stores order item information such as order item ID, order ID, product ID, quantity, unit price, and foreign key references to `Orders` and `Products`.  
\* `Payments`: stores payment information such as payment ID, order ID, payment method, payment date, and amount, and foreign key reference to `Orders`.  
\* `Shippings`: stores shipping information such as shipping ID, order ID, shipper name, shipping address, shipping city, shipping state, shipping zip code, shipping country, and estimated delivery date, and foreign key reference to `Orders`.  
  
Please note that this is just one way to design the database, and there may be other valid approaches depending on specific business requirements and constraints. chatbot: How was the assistance? Was the database schema helpful in understanding the project requirements? Do you have any further questions or need additional clarification on any aspect of the project?  
  
User interfaces  
  
Project Business analysis Document   
  
Functional requirements  
  
Here are the functional requirements of the Electric Store project:  
 \*\*Product Information Management\*\*  
  
 • Manage product catalog  
 • Update product prices and availability  
 • Add new products and remove old ones  
 • Display product images and descriptions  
 • Allow customers to filter products by category and price range  
  
 \*\*Order Processing and Fulfillment\*\*  
  
 • Process customer orders  
 • Send order confirmations via email  
 • Track orders in real-time  
 • Provide estimated delivery dates  
 • Handle returns and exchanges  
  
 \*\*Payment and Shipping Options\*\*  
  
 • Integrate payment gateways (e.g., PayPal, credit cards)  
 • Offer multiple shipping options (e.g., standard, expedited, free)  
 • Calculate shipping costs based on location and weight  
 • Provide estimated delivery dates  
  
 \*\*Customer Account Management\*\*  
 User: What kind of queries can we perform on this database?  
 Chatbot: With this database schema, you can perform various types of queries to retrieve and manipulate data. Here are some examples:  
  
1. \*\*Retrieve customer information\*\*: `SELECT \* FROM Customers WHERE CustomerID = <customer\_id>;`  
2. \*\*Get product details\*\*: `SELECT \* FROM Products WHERE ProductID = <product\_id>;`  
3. \*\*Find orders for a specific customer\*\*: `SELECT \* FROM Orders WHERE CustomerID = <customer\_id>;`  
4. \*\*Get order items for a specific order\*\*: `SELECT \* FROM OrderItems WHERE OrderID = <order\_id>;`  
5. \*\*Calculate total cost for an order\*\*: `SELECT SUM(UnitPrice \* Quantity) AS TotalCost FROM OrderItems WHERE OrderID = <order\_id>;`  
6. \*\*Get shipping information for an order\*\*: `SELECT \* FROM Sh

Here's the generated database schema in SQL syntax:  
  
```sql  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Email VARCHAR(100),  
 PasswordHash VARCHAR(255),  
 Address VARCHAR(200),  
 City VARCHAR(50),  
 State VARCHAR(2),  
 Zipcode VARCHAR(10),  
 Country VARCHAR(50),  
 Phone VARCHAR(20)  
);  
  
CREATE TABLE Products (  
 ProductID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Description TEXT,  
 Price DECIMAL(10,2),  
 ImageURL VARCHAR(200),  
 Category VARCHAR(50),  
 SubCategory VARCHAR(50)  
);  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status VARCHAR(50),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
CREATE TABLE OrderItems (  
 OrderItemID INT PRIMARY KEY,  
 OrderID INT,  
 ProductID INT,  
 Quantity INT,  
 UnitPrice DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
CREATE TABLE Payments (  
 PaymentID INT PRIMARY KEY,  
 OrderID INT,  
 PaymentMethod VARCHAR(50),  
 PaymentDate DATE,  
 Amount DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
  
CREATE TABLE Shippings (  
 ShippingID INT PRIMARY KEY,  
 OrderID INT,  
 ShipperName VARCHAR(50),  
 ShippingAddress VARCHAR(200),  
 ShippingCity VARCHAR(50),  
 ShippingState VARCHAR(2),  
 ShippingZipcode VARCHAR(10),  
 ShippingCountry VARCHAR(50),  
 EstimatedDeliveryDate DATE,  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
```  
  
This database schema includes the following tables:  
  
\* `Customers`: stores customer information such as name, email, password hash, address, city, state, zip code, country, and phone number.  
\* `Products`: stores product information such as name, description, price, image URL, category, and subcategory.  
\* `Orders`: stores order information such as order ID, customer ID, order date, total cost, and status.  
\* `OrderItems`: stores order item information such as order item ID, order ID, product ID, quantity, unit price, and foreign key references to `Orders` and `Products`.  
\* `Payments`: stores payment information such as payment ID, order ID, payment method, payment date, and amount, and foreign key reference to `Orders`.  
\* `Shippings`: stores shipping information such as shipping ID, order ID, shipper name, shipping address, shipping city, shipping state, shipping zip code, shipping country, and estimated delivery date, and foreign key reference to `Orders`.  
  
Please note that this is just one way to design the database, and there may be other valid approaches depending on specific business requirements and constraints. chatbot: How was the assistance? Was the database schema helpful in understanding the project requirements? Do you have any further questions or need additional clarification on any aspect of the project?  
  
User interfaces  
  
Project Business analysis Document   
  
Functional requirements  
  
Here are the functional requirements of the Electric Store project:  
 \*\*Product Information Management\*\*  
  
 • Manage product catalog  
 • Update product prices and availability  
 • Add new products and remove old ones  
 • Display product images and descriptions  
 • Allow customers to filter products by category and price range  
  
 \*\*Order Processing and Fulfillment\*\*  
  
 • Process customer orders  
 • Send order confirmations via email  
 • Track orders in real-time  
 • Provide estimated delivery dates  
 • Handle returns and exchanges  
  
 \*\*Payment and Shipping Options\*\*  
  
 • Integrate payment gateways (e.g., PayPal, credit cards)  
 • Offer multiple shipping options (e.g., standard, expedited, free)  
 • Calculate shipping costs based on location and weight  
 • Provide estimated delivery dates  
  
 \*\*Customer Account Management\*\*  
 User: What kind of queries can we perform on this database?  
 Chatbot: With this database schema, you can perform various types of queries to retrieve and manipulate data. Here are some examples:  
  
1. \*\*Retrieve customer information\*\*: `SELECT \* FROM Customers WHERE CustomerID = <customer\_id>;`  
2. \*\*Get product details\*\*: `SELECT \* FROM Products WHERE ProductID = <product\_id>;`  
3. \*\*Find orders for a specific customer\*\*: `SELECT \* FROM Orders WHERE CustomerID = <customer\_id>;`  
4. \*\*Get order items for a specific order\*\*: `SELECT \* FROM OrderItems WHERE OrderID = <order\_id>;`  
5. \*\*Calculate total cost for an order\*\*: `SELECT SUM(UnitPrice \* Quantity) AS TotalCost FROM OrderItems WHERE OrderID = <order\_id>;`  
6. \*\*Get shipping information for an order\*\*: `SELECT \* FROM Sh

Here's the generated database schema in SQL syntax:  
  
```sql  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Email VARCHAR(100),  
 PasswordHash VARCHAR(255),  
 Address VARCHAR(200),  
 City VARCHAR(50),  
 State VARCHAR(2),  
 Zipcode VARCHAR(10),  
 Country VARCHAR(50),  
 Phone VARCHAR(20)  
);  
  
CREATE TABLE Products (  
 ProductID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Description TEXT,  
 Price DECIMAL(10,2),  
 ImageURL VARCHAR(200),  
 Category VARCHAR(50),  
 SubCategory VARCHAR(50)  
);  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status VARCHAR(50),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
CREATE TABLE OrderItems (  
 OrderItemID INT PRIMARY KEY,  
 OrderID INT,  
 ProductID INT,  
 Quantity INT,  
 UnitPrice DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
CREATE TABLE Payments (  
 PaymentID INT PRIMARY KEY,  
 OrderID INT,  
 PaymentMethod VARCHAR(50),  
 PaymentDate DATE,  
 Amount DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
  
CREATE TABLE Shippings (  
 ShippingID INT PRIMARY KEY,  
 OrderID INT,  
 ShipperName VARCHAR(50),  
 ShippingAddress VARCHAR(200),  
 ShippingCity VARCHAR(50),  
 ShippingState VARCHAR(2),  
 ShippingZipcode VARCHAR(10),  
 ShippingCountry VARCHAR(50),  
 EstimatedDeliveryDate DATE,  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
```  
  
This database schema includes the following tables:  
  
\* `Customers`: stores customer information such as name, email, password hash, address, city, state, zip code, country, and phone number.  
\* `Products`: stores product information such as name, description, price, image URL, category, and subcategory.  
\* `Orders`: stores order information such as order ID, customer ID, order date, total cost, and status.  
\* `OrderItems`: stores order item information such as order item ID, order ID, product ID, quantity, unit price, and foreign key references to `Orders` and `Products`.  
\* `Payments`: stores payment information such as payment ID, order ID, payment method, payment date, and amount, and foreign key reference to `Orders`.  
\* `Shippings`: stores shipping information such as shipping ID, order ID, shipper name, shipping address, shipping city, shipping state, shipping zip code, shipping country, and estimated delivery date, and foreign key reference to `Orders`.  
  
Please note that this is just one way to design the database, and there may be other valid approaches depending on specific business requirements and constraints. chatbot: How was the assistance? Was the database schema helpful in understanding the project requirements? Do you have any further questions or need additional clarification on any aspect of the project?  
  
User interfaces  
  
Project Business analysis Document   
  
Functional requirements  
  
Here are the functional requirements of the Electric Store project:  
 \*\*Product Information Management\*\*  
  
 • Manage product catalog  
 • Update product prices and availability  
 • Add new products and remove old ones  
 • Display product images and descriptions  
 • Allow customers to filter products by category and price range  
  
 \*\*Order Processing and Fulfillment\*\*  
  
 • Process customer orders  
 • Send order confirmations via email  
 • Track orders in real-time  
 • Provide estimated delivery dates  
 • Handle returns and exchanges  
  
 \*\*Payment and Shipping Options\*\*  
  
 • Integrate payment gateways (e.g., PayPal, credit cards)  
 • Offer multiple shipping options (e.g., standard, expedited, free)  
 • Calculate shipping costs based on location and weight  
 • Provide estimated delivery dates  
  
 \*\*Customer Account Management\*\*  
 User: What kind of queries can we perform on this database?  
 Chatbot: With this database schema, you can perform various types of queries to retrieve and manipulate data. Here are some examples:  
  
1. \*\*Retrieve customer information\*\*: `SELECT \* FROM Customers WHERE CustomerID = <customer\_id>;`  
2. \*\*Get product details\*\*: `SELECT \* FROM Products WHERE ProductID = <product\_id>;`  
3. \*\*Find orders for a specific customer\*\*: `SELECT \* FROM Orders WHERE CustomerID = <customer\_id>;`  
4. \*\*Get order items for a specific order\*\*: `SELECT \* FROM OrderItems WHERE OrderID = <order\_id>;`  
5. \*\*Calculate total cost for an order\*\*: `SELECT SUM(UnitPrice \* Quantity) AS TotalCost FROM OrderItems WHERE OrderID = <order\_id>;`  
6. \*\*Get shipping information for an order\*\*: `SELECT \* FROM Sh

Here's the generated database schema in SQL syntax:  
  
```sql  
CREATE TABLE Customers (  
 CustomerID INT PRIMARY KEY,  
 FirstName VARCHAR(50),  
 LastName VARCHAR(50),  
 Email VARCHAR(100),  
 PasswordHash VARCHAR(255),  
 Address VARCHAR(200),  
 City VARCHAR(50),  
 State VARCHAR(2),  
 Zipcode VARCHAR(10),  
 Country VARCHAR(50),  
 Phone VARCHAR(20)  
);  
  
CREATE TABLE Products (  
 ProductID INT PRIMARY KEY,  
 Name VARCHAR(100),  
 Description TEXT,  
 Price DECIMAL(10,2),  
 ImageURL VARCHAR(200),  
 Category VARCHAR(50),  
 SubCategory VARCHAR(50)  
);  
  
CREATE TABLE Orders (  
 OrderID INT PRIMARY KEY,  
 CustomerID INT,  
 OrderDate DATE,  
 TotalCost DECIMAL(10,2),  
 Status VARCHAR(50),  
 FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)  
);  
  
CREATE TABLE OrderItems (  
 OrderItemID INT PRIMARY KEY,  
 OrderID INT,  
 ProductID INT,  
 Quantity INT,  
 UnitPrice DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID),  
 FOREIGN KEY (ProductID) REFERENCES Products(ProductID)  
);  
  
CREATE TABLE Payments (  
 PaymentID INT PRIMARY KEY,  
 OrderID INT,  
 PaymentMethod VARCHAR(50),  
 PaymentDate DATE,  
 Amount DECIMAL(10,2),  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
  
CREATE TABLE Shippings (  
 ShippingID INT PRIMARY KEY,  
 OrderID INT,  
 ShipperName VARCHAR(50),  
 ShippingAddress VARCHAR(200),  
 ShippingCity VARCHAR(50),  
 ShippingState VARCHAR(2),  
 ShippingZipcode VARCHAR(10),  
 ShippingCountry VARCHAR(50),  
 EstimatedDeliveryDate DATE,  
 FOREIGN KEY (OrderID) REFERENCES Orders(OrderID)  
);  
```  
  
This database schema includes the following tables:  
  
\* `Customers`: stores customer information such as name, email, password hash, address, city, state, zip code, country, and phone number.  
\* `Products`: stores product information such as name, description, price, image URL, category, and subcategory.  
\* `Orders`: stores order information such as order ID, customer ID, order date, total cost, and status.  
\* `OrderItems`: stores order item information such as order item ID, order ID, product ID, quantity, unit price, and foreign key references to `Orders` and `Products`.  
\* `Payments`: stores payment information such as payment ID, order ID, payment method, payment date, and amount, and foreign key reference to `Orders`.  
\* `Shippings`: stores shipping information such as shipping ID, order ID, shipper name, shipping address, shipping city, shipping state, shipping zip code, shipping country, and estimated delivery date, and foreign key reference to `Orders`.  
  
Please note that this is just one way to design the database, and there may be other valid approaches depending on specific business requirements and constraints. chatbot: How was the assistance? Was the database schema helpful in understanding the project requirements? Do you have any further questions or need additional clarification on any aspect of the project?  
  
User interfaces  
  
Project Business analysis Document   
  
Functional requirements  
  
Here are the functional requirements of the Electric Store project:  
 \*\*Product Information Management\*\*  
  
 • Manage product catalog  
 • Update product prices and availability  
 • Add new products and remove old ones  
 • Display product images and descriptions  
 • Allow customers to filter products by category and price range  
  
 \*\*Order Processing and Fulfillment\*\*  
  
 • Process customer orders  
 • Send order confirmations via email  
 • Track orders in real-time  
 • Provide estimated delivery dates  
 • Handle returns and exchanges  
  
 \*\*Payment and Shipping Options\*\*  
  
 • Integrate payment gateways (e.g., PayPal, credit cards)  
 • Offer multiple shipping options (e.g., standard, expedited, free)  
 • Calculate shipping costs based on location and weight  
 • Provide estimated delivery dates  
  
 \*\*Customer Account Management\*\*  
 User: What kind of queries can we perform on this database?  
 Chatbot: With this database schema, you can perform various types of queries to retrieve and manipulate data. Here are some examples:  
  
1. \*\*Retrieve customer information\*\*: `SELECT \* FROM Customers WHERE CustomerID = <customer\_id>;`  
2. \*\*Get product details\*\*: `SELECT \* FROM Products WHERE ProductID = <product\_id>;`  
3. \*\*Find orders for a specific customer\*\*: `SELECT \* FROM Orders WHERE CustomerID = <customer\_id>;`  
4. \*\*Get order items for a specific order\*\*: `SELECT \* FROM OrderItems WHERE OrderID = <order\_id>;`  
5. \*\*Calculate total cost for an order\*\*: `SELECT SUM(UnitPrice \* Quantity) AS TotalCost FROM OrderItems WHERE OrderID = <order\_id>;`  
6. \*\*Get shipping information for an order\*\*: `SELECT \* FROM Sh