SQL\_Database

Thank you for providing me with the necessary information. Based on the provided document, I have identified the following functional and non-functional requirements for the project. Please let me know if there's anything else I can help you with.  
  
User interfaces  
  
Please provide detailed answers to each question based on the provided information. Ensure that the answers align with the business analysis standards and provide enough details to support the decision-making process.  
  
Functional Requirements  
  
Project Business analysis Document   
  
Functional requirements  
  
Of course! Here are the functional requirements for the project based on the provided document:  
  
Functional Requirements  
----------------------  
  
### User Management  
  
\* User registration  
\* User login  
\* Profile management  
\* Account deletion  
  
### Product Management  
  
\* Product creation  
\* Product editing  
\* Product deleting  
\* Product categorization  
\* Product search  
  
### Order Management  
  
\* Order placement  
\* Order tracking  
\* Order cancellation  
\* Order refund  
  
### Payment Management  
  
\* Payment gateway integration  
\* Multiple payment options  
\* Payment method selection  
  
### Shipping Management  
  
\* Shipment calculation  
\* Shipping carrier integration  
\* Shipment tracking  
  
### Reporting and Analytics  
  
\* Sales reports  
\* Customer behavior analytics  
\* Product performance reports  
  
### Integration  
  
\* API integration with third-party services (e.g. social media platforms, review sites)  
 User: Generate an detailed database of the project according to information provided in document in SQL syntax. Include all major entities, their attributes, primary keys, and relationships (one-to-one, one-to-many, many-to-many) in SQL syntax.  
 Chatbot: Sure! Here is a detailed database schema for the project based on the information provided in the document:  
```sql  
-- User Entity  
CREATE TABLE Users (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 Password VARCHAR(255) NOT NULL  
);  
  
-- Product Entity  
CREATE TABLE Products (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL,  
 Description TEXT,  
 Price DECIMAL(10, 2) NOT NULL,  
 Category\_Id INT NOT NULL,  
 FOREIGN KEY (Category\_Id) REFERENCES Categories(Id) ON DELETE Cascade  
);  
  
-- Category Entity  
CREATE TABLE Categories (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL  
);  
  
-- Order Entity  
CREATE TABLE Orders (  
 Id INT PRIMARY KEY,  
 User\_Id INT NOT NULL,  
 Product\_Id INT NOT NULL,  
 Total DECIMAL(10, 2) NOT NULL,  
 Placed\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (User\_Id) REFERENCES Users(Id) ON DELETE Cascade,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
-- Payment Entity  
CREATE TABLE Payments (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Payment\_Method VARCHAR(255) NOT NULL,  
 Amount DECIMAL(10, 2) NOT NULL,  
 Paid\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Shipping Entity  
CREATE TABLE Shippings (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Shipping\_Method VARCHAR(255) NOT NULL,  
 Weight DECIMAL(10, 2) NOT NULL,  
 Estimated\_Delivery\_Date TIMESTAMP NOT NULL default '2023-03-07', -- March 7, 2023  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Reporting and Analytics Entities  
CREATE TABLE Sales (  
 Id INT PRIMARY KEY,  
 Date DATE NOT NULL,  
 Product\_Id INT NOT NULL,  
 Quantity INTEGER NOT NULL,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
CREATE TABLE Customers (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 First\_Name VARCHAR(255) NOT NULL,  
 Last\_Name VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE Customer\_Orders (  
 Id INT PRIMARY KEY,  
 Customer\_Id INT NOT NULL,  
 Order\_Id INT NOT NULL,  
 FOREIGN KEY (Customer\_Id) REFERENCES Customers(Id) ON DELETE Cascade,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
```  
This schema includes the following entities:  
  
\* `Users`: represents users who register and log into the system  
\* `Products`: represents products available for purchase  
\* `Categories`: represents categories of products  
\* `Orders`: represents orders placed by users  
\* `Payments`: represents payments made for orders  
\* `Shippings`: represents shipping methods and their associated weights and estimated delivery dates  
\* `Sales`: represents sales data for products  
\* `Customers`: represents customer information  
\* `Customer\_Orders`: represents the relationship between customers and their orders  
  
The schema also defines foreign key constraints to ensure referential integrity across tables. For example, the `Order` table has foreign key constraints to both the `Users` and `Products` tables to ensure that only valid user IDs and product IDs are used. Similarly, the `Payments` table has foreign key constraints to the `Orders` table to ensure that payments are made for valid orders.  
  
I hope this helps! Let me know if you have any questions or need further clarification.

Thank you for providing me with the necessary information. Based on the provided document, I have identified the following functional and non-functional requirements for the project. Please let me know if there's anything else I can help you with.  
  
User interfaces  
  
Please provide detailed answers to each question based on the provided information. Ensure that the answers align with the business analysis standards and provide enough details to support the decision-making process.  
  
Functional Requirements  
  
Project Business analysis Document   
  
Functional requirements  
  
Of course! Here are the functional requirements for the project based on the provided document:  
  
Functional Requirements  
----------------------  
  
### User Management  
  
\* User registration  
\* User login  
\* Profile management  
\* Account deletion  
  
### Product Management  
  
\* Product creation  
\* Product editing  
\* Product deleting  
\* Product categorization  
\* Product search  
  
### Order Management  
  
\* Order placement  
\* Order tracking  
\* Order cancellation  
\* Order refund  
  
### Payment Management  
  
\* Payment gateway integration  
\* Multiple payment options  
\* Payment method selection  
  
### Shipping Management  
  
\* Shipment calculation  
\* Shipping carrier integration  
\* Shipment tracking  
  
### Reporting and Analytics  
  
\* Sales reports  
\* Customer behavior analytics  
\* Product performance reports  
  
### Integration  
  
\* API integration with third-party services (e.g. social media platforms, review sites)  
 User: Generate an detailed database of the project according to information provided in document in SQL syntax. Include all major entities, their attributes, primary keys, and relationships (one-to-one, one-to-many, many-to-many) in SQL syntax.  
 Chatbot: Sure! Here is a detailed database schema for the project based on the information provided in the document:  
```sql  
-- User Entity  
CREATE TABLE Users (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 Password VARCHAR(255) NOT NULL  
);  
  
-- Product Entity  
CREATE TABLE Products (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL,  
 Description TEXT,  
 Price DECIMAL(10, 2) NOT NULL,  
 Category\_Id INT NOT NULL,  
 FOREIGN KEY (Category\_Id) REFERENCES Categories(Id) ON DELETE Cascade  
);  
  
-- Category Entity  
CREATE TABLE Categories (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL  
);  
  
-- Order Entity  
CREATE TABLE Orders (  
 Id INT PRIMARY KEY,  
 User\_Id INT NOT NULL,  
 Product\_Id INT NOT NULL,  
 Total DECIMAL(10, 2) NOT NULL,  
 Placed\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (User\_Id) REFERENCES Users(Id) ON DELETE Cascade,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
-- Payment Entity  
CREATE TABLE Payments (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Payment\_Method VARCHAR(255) NOT NULL,  
 Amount DECIMAL(10, 2) NOT NULL,  
 Paid\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Shipping Entity  
CREATE TABLE Shippings (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Shipping\_Method VARCHAR(255) NOT NULL,  
 Weight DECIMAL(10, 2) NOT NULL,  
 Estimated\_Delivery\_Date TIMESTAMP NOT NULL default '2023-03-07', -- March 7, 2023  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Reporting and Analytics Entities  
CREATE TABLE Sales (  
 Id INT PRIMARY KEY,  
 Date DATE NOT NULL,  
 Product\_Id INT NOT NULL,  
 Quantity INTEGER NOT NULL,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
CREATE TABLE Customers (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 First\_Name VARCHAR(255) NOT NULL,  
 Last\_Name VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE Customer\_Orders (  
 Id INT PRIMARY KEY,  
 Customer\_Id INT NOT NULL,  
 Order\_Id INT NOT NULL,  
 FOREIGN KEY (Customer\_Id) REFERENCES Customers(Id) ON DELETE Cascade,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
```  
This schema includes the following entities:  
  
\* `Users`: represents users who register and log into the system  
\* `Products`: represents products available for purchase  
\* `Categories`: represents categories of products  
\* `Orders`: represents orders placed by users  
\* `Payments`: represents payments made for orders  
\* `Shippings`: represents shipping methods and their associated weights and estimated delivery dates  
\* `Sales`: represents sales data for products  
\* `Customers`: represents customer information  
\* `Customer\_Orders`: represents the relationship between customers and their orders  
  
The schema also defines foreign key constraints to ensure referential integrity across tables. For example, the `Order` table has foreign key constraints to both the `Users` and `Products` tables to ensure that only valid user IDs and product IDs are used. Similarly, the `Payments` table has foreign key constraints to the `Orders` table to ensure that payments are made for valid orders.  
  
I hope this helps! Let me know if you have any questions or need further clarification.

Thank you for providing me with the necessary information. Based on the provided document, I have identified the following functional and non-functional requirements for the project. Please let me know if there's anything else I can help you with.  
  
User interfaces  
  
Please provide detailed answers to each question based on the provided information. Ensure that the answers align with the business analysis standards and provide enough details to support the decision-making process.  
  
Functional Requirements  
  
Project Business analysis Document   
  
Functional requirements  
  
Of course! Here are the functional requirements for the project based on the provided document:  
  
Functional Requirements  
----------------------  
  
### User Management  
  
\* User registration  
\* User login  
\* Profile management  
\* Account deletion  
  
### Product Management  
  
\* Product creation  
\* Product editing  
\* Product deleting  
\* Product categorization  
\* Product search  
  
### Order Management  
  
\* Order placement  
\* Order tracking  
\* Order cancellation  
\* Order refund  
  
### Payment Management  
  
\* Payment gateway integration  
\* Multiple payment options  
\* Payment method selection  
  
### Shipping Management  
  
\* Shipment calculation  
\* Shipping carrier integration  
\* Shipment tracking  
  
### Reporting and Analytics  
  
\* Sales reports  
\* Customer behavior analytics  
\* Product performance reports  
  
### Integration  
  
\* API integration with third-party services (e.g. social media platforms, review sites)  
 User: Generate an detailed database of the project according to information provided in document in SQL syntax. Include all major entities, their attributes, primary keys, and relationships (one-to-one, one-to-many, many-to-many) in SQL syntax.  
 Chatbot: Sure! Here is a detailed database schema for the project based on the information provided in the document:  
```sql  
-- User Entity  
CREATE TABLE Users (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 Password VARCHAR(255) NOT NULL  
);  
  
-- Product Entity  
CREATE TABLE Products (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL,  
 Description TEXT,  
 Price DECIMAL(10, 2) NOT NULL,  
 Category\_Id INT NOT NULL,  
 FOREIGN KEY (Category\_Id) REFERENCES Categories(Id) ON DELETE Cascade  
);  
  
-- Category Entity  
CREATE TABLE Categories (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL  
);  
  
-- Order Entity  
CREATE TABLE Orders (  
 Id INT PRIMARY KEY,  
 User\_Id INT NOT NULL,  
 Product\_Id INT NOT NULL,  
 Total DECIMAL(10, 2) NOT NULL,  
 Placed\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (User\_Id) REFERENCES Users(Id) ON DELETE Cascade,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
-- Payment Entity  
CREATE TABLE Payments (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Payment\_Method VARCHAR(255) NOT NULL,  
 Amount DECIMAL(10, 2) NOT NULL,  
 Paid\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Shipping Entity  
CREATE TABLE Shippings (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Shipping\_Method VARCHAR(255) NOT NULL,  
 Weight DECIMAL(10, 2) NOT NULL,  
 Estimated\_Delivery\_Date TIMESTAMP NOT NULL default '2023-03-07', -- March 7, 2023  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Reporting and Analytics Entities  
CREATE TABLE Sales (  
 Id INT PRIMARY KEY,  
 Date DATE NOT NULL,  
 Product\_Id INT NOT NULL,  
 Quantity INTEGER NOT NULL,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
CREATE TABLE Customers (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 First\_Name VARCHAR(255) NOT NULL,  
 Last\_Name VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE Customer\_Orders (  
 Id INT PRIMARY KEY,  
 Customer\_Id INT NOT NULL,  
 Order\_Id INT NOT NULL,  
 FOREIGN KEY (Customer\_Id) REFERENCES Customers(Id) ON DELETE Cascade,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
```  
This schema includes the following entities:  
  
\* `Users`: represents users who register and log into the system  
\* `Products`: represents products available for purchase  
\* `Categories`: represents categories of products  
\* `Orders`: represents orders placed by users  
\* `Payments`: represents payments made for orders  
\* `Shippings`: represents shipping methods and their associated weights and estimated delivery dates  
\* `Sales`: represents sales data for products  
\* `Customers`: represents customer information  
\* `Customer\_Orders`: represents the relationship between customers and their orders  
  
The schema also defines foreign key constraints to ensure referential integrity across tables. For example, the `Order` table has foreign key constraints to both the `Users` and `Products` tables to ensure that only valid user IDs and product IDs are used. Similarly, the `Payments` table has foreign key constraints to the `Orders` table to ensure that payments are made for valid orders.  
  
I hope this helps! Let me know if you have any questions or need further clarification.

Thank you for providing me with the necessary information. Based on the provided document, I have identified the following functional and non-functional requirements for the project. Please let me know if there's anything else I can help you with.  
  
User interfaces  
  
Please provide detailed answers to each question based on the provided information. Ensure that the answers align with the business analysis standards and provide enough details to support the decision-making process.  
  
Functional Requirements  
  
Project Business analysis Document   
  
Functional requirements  
  
Of course! Here are the functional requirements for the project based on the provided document:  
  
Functional Requirements  
----------------------  
  
### User Management  
  
\* User registration  
\* User login  
\* Profile management  
\* Account deletion  
  
### Product Management  
  
\* Product creation  
\* Product editing  
\* Product deleting  
\* Product categorization  
\* Product search  
  
### Order Management  
  
\* Order placement  
\* Order tracking  
\* Order cancellation  
\* Order refund  
  
### Payment Management  
  
\* Payment gateway integration  
\* Multiple payment options  
\* Payment method selection  
  
### Shipping Management  
  
\* Shipment calculation  
\* Shipping carrier integration  
\* Shipment tracking  
  
### Reporting and Analytics  
  
\* Sales reports  
\* Customer behavior analytics  
\* Product performance reports  
  
### Integration  
  
\* API integration with third-party services (e.g. social media platforms, review sites)  
 User: Generate an detailed database of the project according to information provided in document in SQL syntax. Include all major entities, their attributes, primary keys, and relationships (one-to-one, one-to-many, many-to-many) in SQL syntax.  
 Chatbot: Sure! Here is a detailed database schema for the project based on the information provided in the document:  
```sql  
-- User Entity  
CREATE TABLE Users (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 Password VARCHAR(255) NOT NULL  
);  
  
-- Product Entity  
CREATE TABLE Products (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL,  
 Description TEXT,  
 Price DECIMAL(10, 2) NOT NULL,  
 Category\_Id INT NOT NULL,  
 FOREIGN KEY (Category\_Id) REFERENCES Categories(Id) ON DELETE Cascade  
);  
  
-- Category Entity  
CREATE TABLE Categories (  
 Id INT PRIMARY KEY,  
 Name VARCHAR(255) NOT NULL  
);  
  
-- Order Entity  
CREATE TABLE Orders (  
 Id INT PRIMARY KEY,  
 User\_Id INT NOT NULL,  
 Product\_Id INT NOT NULL,  
 Total DECIMAL(10, 2) NOT NULL,  
 Placed\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (User\_Id) REFERENCES Users(Id) ON DELETE Cascade,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
-- Payment Entity  
CREATE TABLE Payments (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Payment\_Method VARCHAR(255) NOT NULL,  
 Amount DECIMAL(10, 2) NOT NULL,  
 Paid\_At TIMESTAMP NOT NULL DEFAULT CURRENT\_TIMESTAMP,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Shipping Entity  
CREATE TABLE Shippings (  
 Id INT PRIMARY KEY,  
 Order\_Id INT NOT NULL,  
 Shipping\_Method VARCHAR(255) NOT NULL,  
 Weight DECIMAL(10, 2) NOT NULL,  
 Estimated\_Delivery\_Date TIMESTAMP NOT NULL default '2023-03-07', -- March 7, 2023  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
  
-- Reporting and Analytics Entities  
CREATE TABLE Sales (  
 Id INT PRIMARY KEY,  
 Date DATE NOT NULL,  
 Product\_Id INT NOT NULL,  
 Quantity INTEGER NOT NULL,  
 FOREIGN KEY (Product\_Id) REFERENCES Products(Id) ON DELETE Cascade  
);  
  
CREATE TABLE Customers (  
 Id INT PRIMARY KEY,  
 Email VARCHAR(255) NOT NULL,  
 First\_Name VARCHAR(255) NOT NULL,  
 Last\_Name VARCHAR(255) NOT NULL  
);  
  
CREATE TABLE Customer\_Orders (  
 Id INT PRIMARY KEY,  
 Customer\_Id INT NOT NULL,  
 Order\_Id INT NOT NULL,  
 FOREIGN KEY (Customer\_Id) REFERENCES Customers(Id) ON DELETE Cascade,  
 FOREIGN KEY (Order\_Id) REFERENCES Orders(Id) ON DELETE Cascade  
);  
```  
This schema includes the following entities:  
  
\* `Users`: represents users who register and log into the system  
\* `Products`: represents products available for purchase  
\* `Categories`: represents categories of products  
\* `Orders`: represents orders placed by users  
\* `Payments`: represents payments made for orders  
\* `Shippings`: represents shipping methods and their associated weights and estimated delivery dates  
\* `Sales`: represents sales data for products  
\* `Customers`: represents customer information  
\* `Customer\_Orders`: represents the relationship between customers and their orders  
  
The schema also defines foreign key constraints to ensure referential integrity across tables. For example, the `Order` table has foreign key constraints to both the `Users` and `Products` tables to ensure that only valid user IDs and product IDs are used. Similarly, the `Payments` table has foreign key constraints to the `Orders` table to ensure that payments are made for valid orders.  
  
I hope this helps! Let me know if you have any questions or need further clarification.