**1. Customers**

This table holds information about the customers in the eCommerce system.

* **Entity Name:** customers
* **Columns:**
  + customer\_id: Unique identifier for each customer (Primary Key).
  + customer\_unique\_id: Another unique identifier for the customer, could be used for external reference.
  + customer\_zip\_code\_prefix: The customer's zip code (could be used to join with geolocation table).
  + customer\_city: The city where the customer resides.
  + customer\_state: The state where the customer resides.

**Use Case:** This entity stores information about users making purchases.

**2. Geolocation**

This table stores geographical data, mainly zip codes and their corresponding latitude, longitude, city, and state.

* **Entity Name:** geolocation
* **Columns:**
  + geolocation\_zip\_code\_prefix: The zip code, acts as a reference to identify locations.
  + geolocation\_lat: Latitude of the location.
  + geolocation\_lng: Longitude of the location.
  + geolocation\_city: City name.
  + geolocation\_state: State name.

**Use Case:** It is used for mapping zip codes to their geographical locations, aiding in delivery and logistics.

**3. Orders**

This table holds information related to the orders placed by customers.

* **Entity Name:** orders
* **Columns:**
  + order\_id: Unique identifier for each order (Primary Key).
  + customer\_id: Foreign Key referencing the customers table.
  + order\_status: Status of the order (e.g., pending, shipped, delivered).
  + order\_purchase\_timestamp: When the order was placed.
  + order\_approved\_at: When the order was approved.
  + order\_delivered\_carrier\_date: When the order was delivered to the carrier.
  + order\_delivered\_customer\_date: When the order was delivered to the customer.
  + order\_estimated\_delivery\_date: The estimated delivery date for the order.

**Use Case:** This entity tracks the details and status of orders placed by customers.

**4. Payments**

This table stores payment information for each order.

* **Entity Name:** payments
* **Columns:**
  + order\_id: Foreign Key referencing the orders table.
  + payment\_sequential: Sequence of the payment (if multiple payments are made for the same order).
  + payment\_type: The type of payment (e.g., credit card, debit card, voucher).
  + payment\_installments: Number of installments for the payment (if applicable).
  + payment\_value: Total value of the payment.

**Use Case:** This entity records the payment information for orders and supports multiple payment installments.

**5. Products**

This table contains information about the products sold in the eCommerce system.

* **Entity Name:** products
* **Columns:**
  + product\_id: Unique identifier for each product (Primary Key).
  + product\_category: The category the product belongs to (e.g., electronics, clothing).
  + product\_name\_length: Length of the product name (in characters).
  + product\_description\_length: Length of the product description (in characters).
  + product\_photos\_qty: Number of photos available for the product.
  + product\_weight\_g: Weight of the product (in grams).
  + product\_length\_cm: Length of the product (in cm).
  + product\_height\_cm: Height of the product (in cm).
  + product\_width\_cm: Width of the product (in cm).

**Use Case:** This entity stores information about the products available for sale.

**6. Sellers**

This table stores information about the sellers who provide products for sale on the platform.

* **Entity Name:** sellers
* **Columns:**
  + seller\_id: Unique identifier for each seller (Primary Key).
  + seller\_zip\_code\_prefix: The seller’s zip code.
  + seller\_city: The city where the seller is located.
  + seller\_state: The state where the seller is located.

**Use Case:** This entity tracks the sellers providing products for the platform.

**7. Order Items**

This table keeps track of each item in an order, acting as a bridge between orders, products, and sellers.

* **Entity Name:** order\_items
* **Columns:**
  + order\_id: Foreign Key referencing the orders table.
  + order\_item\_id: The unique ID for each item in the order.
  + product\_id: Foreign Key referencing the products table.
  + seller\_id: Foreign Key referencing the sellers table.
  + shipping\_limit\_date: Deadline for the seller to ship the product.
  + price: The price of the product.
  + freight\_value: The shipping cost for the product.

**Use Case:** This entity helps track individual items in orders and their respective details like price, shipping costs, and seller.

**Entity Relationships:**

* **Customers** place multiple **Orders**.
* **Orders** are paid for via **Payments** and contain multiple **Order Items**.
* **Order Items** are linked to **Products** and are sold by **Sellers**.
* **Geolocation** can be used to match customers and sellers to their respective locations using zip\_code\_prefix.

**Entity Diagram Overview:**

* **Customers** (1

) **Orders**

* **Orders** (1

) **Order Items**

* **Order Items** (N:1) **Products**
* **Order Items** (N:1) **Sellers**
* **Payments** (1:1) **Orders**
* **Geolocation** (N:1) **Customers/Sellers** (via zip code)

**Table Format:**

CREATE TABLE customers (

customer\_id VARCHAR(100) PRIMARY KEY,

customer\_unique\_id VARCHAR(100),

customer\_zip\_code\_prefix INT,

customer\_city VARCHAR(100),

customer\_state VARCHAR(100)

);

CREATE TABLE geolocation (

geolocation\_zip\_code\_prefix INT,

geolocation\_lat FLOAT,

geolocation\_lng FLOAT,

geolocation\_city VARCHAR(100),

geolocation\_state VARCHAR(100)

);

CREATE TABLE orders (

order\_id VARCHAR(100) PRIMARY KEY,

customer\_id VARCHAR(100),

order\_status VARCHAR(100),

order\_purchase\_timestamp VARCHAR(100),

order\_approved\_at VARCHAR(100),

order\_delivered\_carrier\_date VARCHAR(100),

order\_delivered\_customer\_date VARCHAR(100),

order\_estimated\_delivery\_date VARCHAR(100),

FOREIGN KEY (customer\_id) REFERENCES customers(customer\_id)

);

CREATE TABLE payments (

order\_id VARCHAR(100),

payment\_sequential INT,

payment\_type VARCHAR(100),

payment\_installments INT,

payment\_value FLOAT,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id)

);

CREATE TABLE products (

product\_id VARCHAR(100) PRIMARY KEY,

product\_category VARCHAR(100),

product\_name\_length FLOAT,

product\_description\_length FLOAT,

product\_photos\_qty FLOAT,

product\_weight\_g FLOAT,

product\_length\_cm FLOAT,

product\_height\_cm FLOAT,

product\_width\_cm FLOAT

);

CREATE TABLE sellers (

seller\_id VARCHAR(100) PRIMARY KEY,

seller\_zip\_code\_prefix INT,

seller\_city VARCHAR(100),

seller\_state VARCHAR(100)

);

CREATE TABLE order\_items (

order\_id VARCHAR(100),

order\_item\_id INT,

product\_id VARCHAR(100),

seller\_id VARCHAR(100),

shipping\_limit\_date VARCHAR(100),

price FLOAT,

freight\_value FLOAT,

FOREIGN KEY (order\_id) REFERENCES orders(order\_id),

FOREIGN KEY (product\_id) REFERENCES products(product\_id),

FOREIGN KEY (seller\_id) REFERENCES sellers(seller\_id)

);