Database Lab Endterm Project:
Team - CS301P-2023-21

IMT2021019 - Siddharth Kothari
IMT2021028 - Sankalp Kothari
IMT2021058 - M Srinivasan

Project Domain - Logistics (Merchandise Delivery)
Project Name - MerchLounge

Tables for the project: -

1) **Product**
2) **Seller**
3) **Order**
4) **OrderDesc**
5) **Shipment (delivery agent keeps track of delivery)**
6) **Delivery_Agent**
7) **Customer**
8) **Inventory (sellers keep track of items)**

IMT2021019 - Siddharth Kothari

1. **Placing the order-**

    a. **Description-** this is the event when the customer orders a certain amount of products. The order would be registered, and the sellers will update their inventories.

    b. **Actors -**
        i. Customer- the customer provides the details of the products that they want to order - the product id, the quantity.
        ii. Seller - the sellers register the products which they can supply from the given products, and update inventory accordingly.

    c. **Normal Flow-**
        i. The customer provides their id and the products they want to order, and the quantities. An order is created.

ii. We then find the seller who sells the product from the Inventory Table. If all the product ids and quantities are valid, corresponding entries for the products are created in OrderDesc (one order may have multiple products from different sellers).
iii. The seller updates the entry of the product in the Inventory Table.

d. Error Flow -

i. Invalid customer id causes the order to be dropped completely, and the message is displayed.
ii. If one of the product ids does not exist, the other products are unaffected and the user is told about which product ids are invalid. An order is created with the remaining products.
iii. If a customer orders more quantity than available with the seller, the customer is notified accordingly and that product is not included in the order. The rest of the order is unaffected.

2. Customer Tracks all his orders -

a. **Description -** this will use the join clause and give off the details of all the orders placed by a customer.

b. Actors -

i. The customer provides their customer id, and gets back the details of their orders. (The details of all the products in the order). Uses join query.

c. Normal Flow -

i. Customers provide their customer id.
ii. A join query is executed consisting of the customer id, and the orders for that id, the products for the order id, and the status of the order if not delivered.

d. Error Flow

i. If the wrong customer id is provided, a message is displayed telling about invalid customer id.

IMT2021028 - Sankalp Kothari

1. Updation of order status by seller -

a. **Description -** Seller has prepared the order for shipment and will have it delivered by delivery agent (updates the shipment table)

b. **Actors -**
    i. Seller (prepares shipment and updates status)
    ii. Delivery agent (order is assigned to agent who then takes up the task of further updating the status)

c. **Normal Flow -**
    i. Seller updates shipment status, and corresponding to it an order comes up in the shipment table
    ii. The first seller assigns delivery agent to that order and makes an entry in the shipment table and the following sellers use the same delivery agent to deliver their products.

d. **Exception -**
    i. In case wrong order id or delivery agent id is entered, print invalid in those cases

2. **Delivery agent updates status -**

    a. **Description -**
        i. Delivery agent in the middle handles the delays and updates the shipment. (out for delivery, delivered etc)

    b. **Actors -**
        i. Delivery Agent (updates the shipment status in case of delays etc or when product has been delivered)

    c. **Normal Flow -**
        i. The agent specifies whether the order is out for delivery / delivered etc and updates the shipment status accordingly for the customer to view
        ii. In case of delay, the expected delivery date of order is updated
        iii. In case of change in mode of shipment, the update is made in shipment table

    d. **Exception -**
        i. In case wrong order id is entered, print invalid id entered

3. **Delivery Agent sees his orders -**
    a. **Description -**
        i. Delivery agent views all the orders that he has to deliver to various people.
    b. **Actors -**
        i. Delivery Agent (views the orders that he has to deliver, where and to whom he has to deliver). Uses a join query.
    c. **Normal Flow -**

i. The delivery agent specifies his ID.
ii. The list of orders is returned to him, with details of whom and where to deliver the order to.

d. Exception Flow -

i. If a wrong deliveryID is provided, the message of invalid delivery agent is returned.

IMT2021058 - M Srinivasan

1) When the order is delivered.

a) Description -

i) This is when the delivery agent has delivered the product to the respective customer and he needs to update the necessary details in the database

b) **Actors -**

i) Delivery agent as he delivers the order to the customer and the Customer who receives the order he ordered.

ii)

c) Normal Flow -

i) Delivery agent updates the status of the order to “delivered” for that particular orderID.

ii) He removes entry of that order from the shipments table.

d) Exception -

i) In case the wrong OrderId is getting delivered then we should print that wrong orderId is delivered.

2) Order is to be returned

a) Description -

i) This is when the customer is unhappy with the product delivered and he wants to return any item from the order to that particular seller. This is very similar to the first case but just that the customer is the new seller and the seller is the new customer.

b) Actors -

i) Customer returns the product back to the seller and the delivery agent who will pick up the order from the customer and the seller who will update his inventory.

c) Normal Flow -

i) Customer sets the flag of that product as return and updates the table for initiating a return.

ii) Delivery agent picks up the order from the customer and ships it to the seller.

iii) The inventory will get updated for that particular seller.

d) **Exception -**
    i) If the customer tries to return some product that is not part of his order then we should tell him to enter correct product details. Customer returns the order.(same flow as ordering)

3) **Seller views his inventory**
    a) **Description -**
        i) This is when the seller at any point of time wants to view his inventory, and the items in it.
    b) **Actors -**
        i) Seller who wants to view his inventory.
    c) **Normal Flow**
        i) Seller provides his sellerID.
        ii) He is able to see all the products that he sells.
    d) **Exception Flow**
        i) If the seller provides a wrong id, we print a message conveying the same.

Tables for the database -

- ## Customer -
    - **CustomerId(primary key)**
    - **CustomerName**
    - **Address**
    - **PhoneNo**
- ## Seller -
    - **sellerID(primary key)**
    - **sellerName**
    - **Address**
    - **PhoneNo**
- ## Product -
    - **productID(primary key)**
    - **productName**
    - **price**
- ## OrderDesc -

    - **Slno(dummy primary key)**
    - **productID(foreign key)**
    - **orderID(foreign key)**
    - **Quantity**
    - **Type(D/R)**
    - **Status**

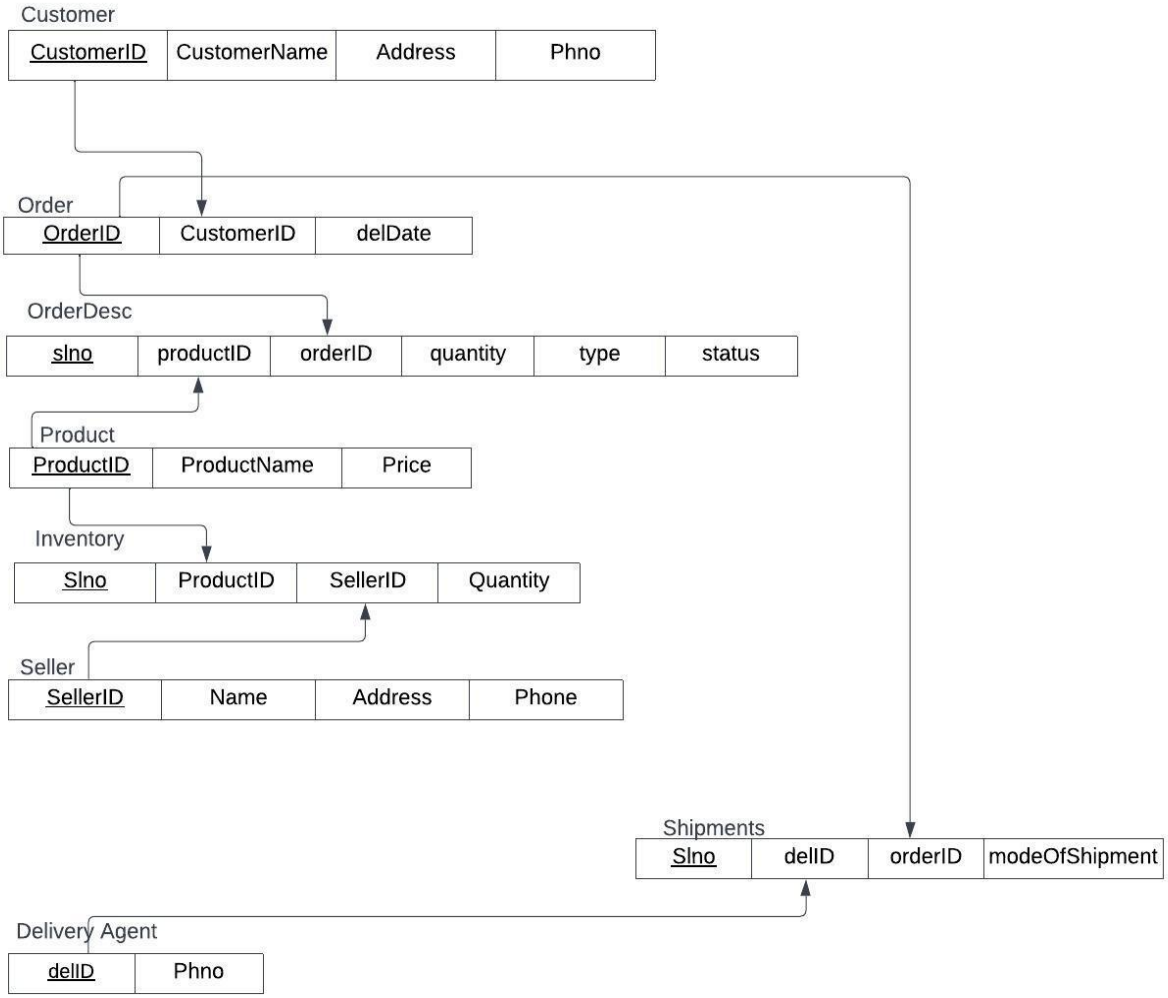
- **Order -**
    - **orderID(primary key)**
    - **customerID(foreign key)**
    - **delDate**

- **Delivery-Agent -**
    - **delID(primary key)**
    - **PhoneNo**

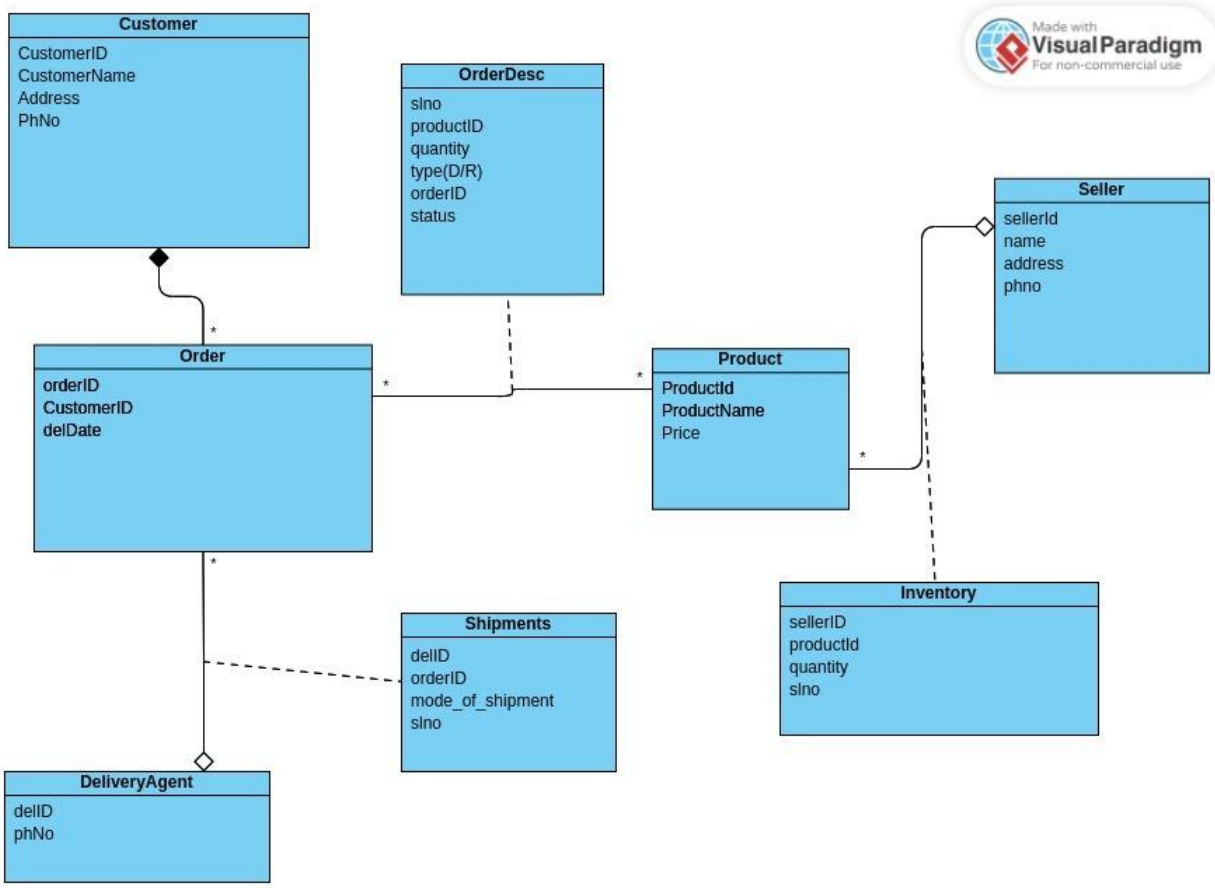
- **Shipments -**
    - **delID(foreign key)**
    - **orderID(foreign key)**
    - **Mode of shipment**
    - **Slno(primary key)**

- **Inventory -**
    - **sellerID(foreign key)**
    - **productID(foreign key)**
    - **Quantity**
    - **Slno(primary key)**

MerchLounge Database



Schema Diagram



UML Class Diagram