

DBMS Project

**Online Retail Store Database
Design**

Mid Project Evaluation
By Group 46

Problem Statement and Relationship Roles

Customers can create an account specifying their first name, last name, shipping info(containing, address and the phone number). Each customer can save multiple shipping infos and can choose any one of them while placing an order.

Customers can browse for products by searching product name or look for different categories to browse further sub-categories and subsequently products belonging to different subcategories.

Customers can order products from sellers, each order can contain one or more pieces of a product as specified by the customer, a payment method, discount on the order, current status of the order(whether placed or delivered, or returned) and a final cost of the order after applying the discount. Customers can also write reviews for their orders and give a 1-5 rating based on their experience with the product.

Sellers can create an account with their name, phone, and email. Each seller can add multiple products with the available quantity for sale for the price specified by them, sellers can update the listing by adding or removing products from the listing or by increasing or decreasing the quantity or price of a particular product for sale, sellers can also specify and update discounts on the product in their listing.

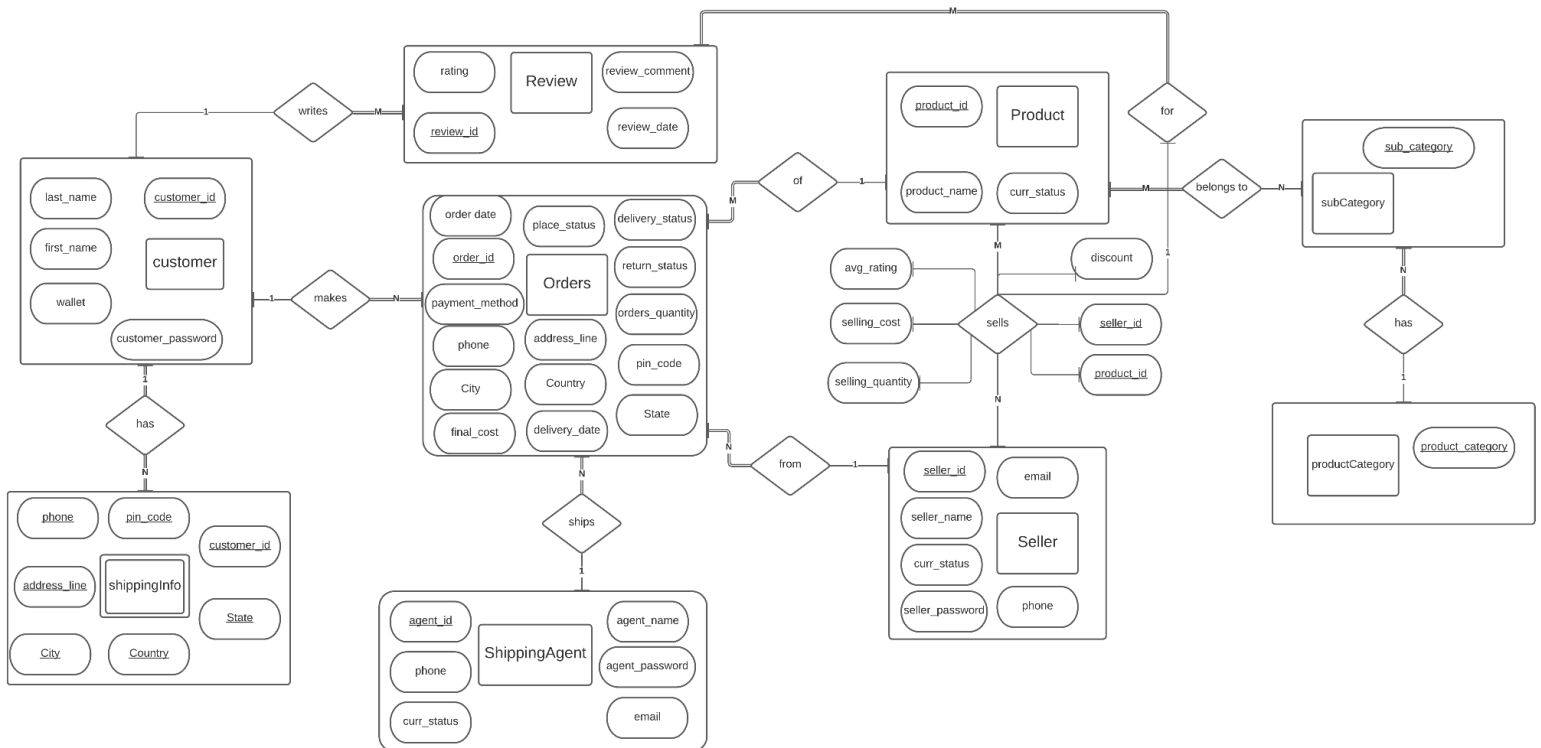
Shipping agents can create an account with their company name, email and phone number. Shipping agents ship orders from seller to customer and hence can update the delivery_status, and return_status of orders assigned to them.

Stakeholders, Entities and Attributes

Entities	Stakeholder	Attributes
Customer	Yes	<ul style="list-style-type: none">• First name• Last name• Customer id• Customer password• Wallet
Shipping Info	No	<ul style="list-style-type: none">• Phone• Pin code• Customer id• Address line• City• State• Country
Seller	Yes	<ul style="list-style-type: none">• Email• Phone• Seller id• Seller name• Curr Status• Seller Password
Product	No	<ul style="list-style-type: none">• Product id• Product Status• Curr status
Sub Category	No	<ul style="list-style-type: none">• Sub Category
Product Category	No	<ul style="list-style-type: none">• Product Category
Orders	No	<ul style="list-style-type: none">• Order date• Order id• Payment method• Phone• Address line• City• State• Country• Pin code• Final cost• Order Quantity• Place status• Delivery status• Return Status• Delivery date

Shipping Agent	Yes	<ul style="list-style-type: none">• Agent id• Email• Phone• Agent name• Agent password• Curr status
Review	No	<ul style="list-style-type: none">• Review id• Rating• Review Comment• Review date

ER DIAGRAM:



Relational Schema:

customer (customer_id, first_name, last_name, password, wallet)

shippingInfo (customer_id, pin_code, phone, address_line, city, country, state)

shippingAgent (agent_id, agent_name, phone, password, email, curr_status)

Seller (seller_id, seller_name, phone, email, password, curr_status)

Product (product_id, product_name, curr_status)

Review (review_id, rating, review_comment, review_date, *customer_id*, *product_id*, *seller_id*)

Orders (orders_id, place_status, delivery_status, return_status, orders_quantity,
payment_method,
final_cost, orders_date, phone, city, country, address_line, state, pin_code, *customer_id*,
agent_id, *seller_id*, *product_id*)

sells (seller_id, product_id, selling_cost, avg_rating, selling_quantity, discount)

subCategory (sub_category, *product_category*)

productCategory (product_category)

belongsTo (product_id, sub_category)

The Weak Entity Present in the Relational Schema is :shippingInfo

Because the existence of shippingInfo is dependent on Customer. Like two different customers can have the same shipping info saved.

No Ternary Relationship found.

SQL QUERIES :

1. First name of customers who have placed orders with pincode 281002

```
SELECT first_name FROM  
customer JOIN orders  
WHERE customer.customer_id = orders.customer_id AND orders.pin_code = 281002  
AND orders.place_status = 'true';
```

2. Last name of customers who have state as gujarat in their saved shipping info

```
Select DISTINCT C.last_name  
from customer C, shippingInfo S  
where C.customer_id = S.customer_id and state = 'Gujarat';
```

3. Customer names with their delivered product_names

```
SELECT customer.first_name, product.product_name  
FROM ((customer JOIN orders) JOIN product)  
WHERE orders.delivery_status = 'true' AND  
orders.customer_id = customer.customer_id AND  
orders.product_id = product.product_id;
```

4. Seller, product category, product name for products on sale

```
SELECT seller.seller_name, product.product_name, productCategory.product_category  
FROM seller JOIN sells JOIN product JOIN belongTo JOIN subCategory JOIN  
productCategory  
WHERE sells.product_id = product.product_id AND  
sells.seller_id = seller.seller_id AND  
product.product_id = belongTo.product_id AND  
subCategory.sub_category = belongTo.sub_category AND  
subCategory.product_category = productCategory.product_category;
```

5. Shipping agents who have WORKING current_status

```
SELECT agent_name FROM shippingAgent  
WHERE curr_status = 'WORKING';
```

6. Show distinct products that have atleast one seller for less than 300 price

```
SELECT DISTINCT product_name FROM product
WHERE product_id IN (SELECT product_id FROM sells WHERE selling_cost < 300);
```

7. Reviews and product name after month October

```
SELECT rating, review_comment, review_date, product_name FROM
review JOIN product
WHERE review.review_date > '2021-10-31' AND
review.product_id = product.product_id;
```

8. Products in Kitchen product category

```
SELECT product_name FROM
product JOIN belongTo JOIN subCategory JOIN productCategory
WHERE productCategory.product_category = 'Kitchen' AND
product.product_id = belongTo.product_id AND
belongTo.sub_category = subCategory.sub_category AND
subCategory.product_category = productCategory.product_category;
```

9. Customers who have any order with return status accepted

```
SELECT customer.first_name, customer.last_name FROM customer, orders
WHERE customer.customer_id = orders.customer_id AND
orders.return_status = 'ACCEPTED';
```

10. total sales by each seller

```
SELECT SUM(final_cost), seller_name
FROM orders JOIN seller
WHERE orders.seller_id = seller.seller_id
GROUP BY seller.seller_name;
```


Member contribution

ER Diagram - We all contributed different entities and relationships between them and brainstormed simultaneously.

Relational schema - Shahzan translated the ER into relational schema and everyone verified the correctness of the schema.

SQL:

1. Tables structure:

Apoorv and Anas and Divyansh

2. Integrity constraints & Data population:

Anas, Shahzan, Divyansh and Apoorv

3. SQL queries:

Every member wrote 3-4 queries and most appropriate ones were Selected from them.

Link To Figma:

<https://www.figma.com/file/Z2ypyl7X8KdEVbts77qhls/DBMS-Data-Population?node-id=0%3A1>