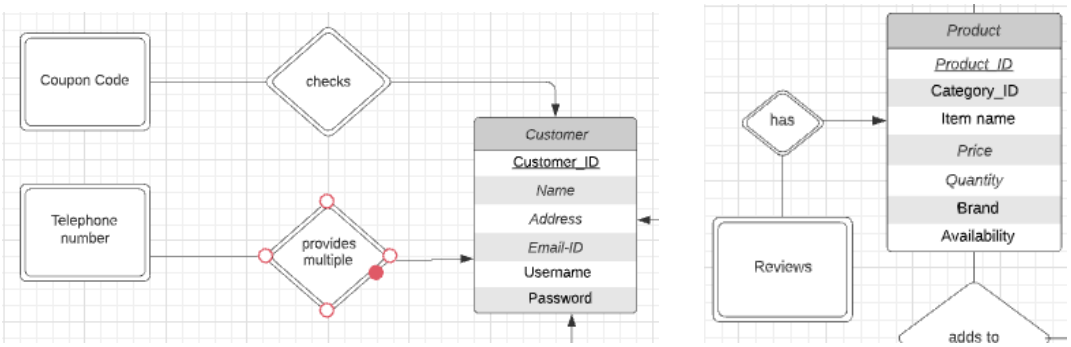
ER Diagram

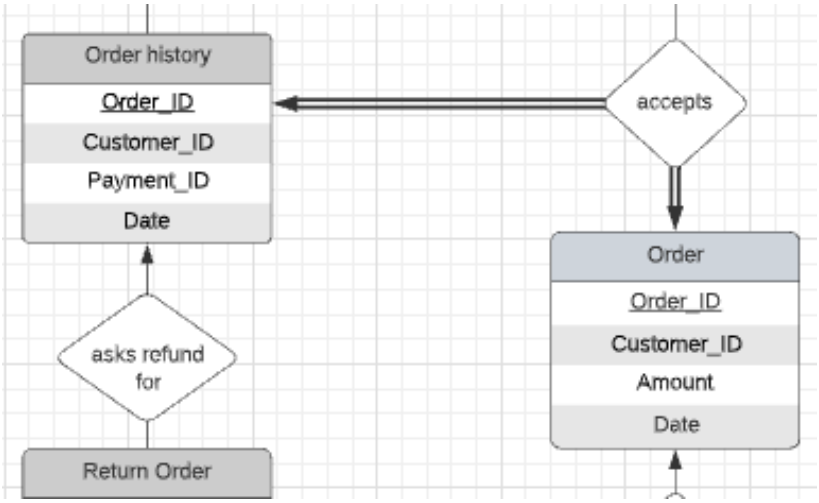
Identification of weak entity :-

- **Coupon Code** - The coupon codes can vary from customer to customer , one customer can have zero coupons another can have many . Hence coupon code is made a weak entity with “checks” as a many to one identifying relationship.
- **Telephone number** - Since we can provide multiple phone numbers for a certain customer we have taken it as a weak entity with “provides multiple” as a many to one identifying relationship.
- **Reviews** - We know that a product can have many reviews , Hence the relationship between the entities is a many to one relationship named “has” .Reviews is a weak entity since it is related to product and one product can have many reviews .

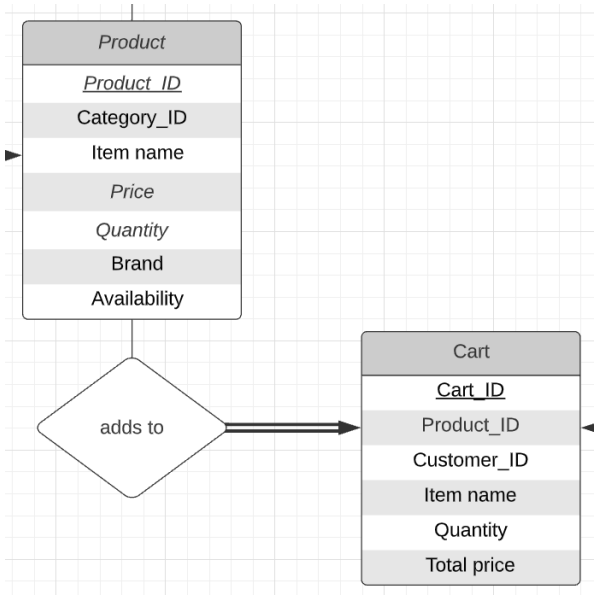


Entities Participation Type :-

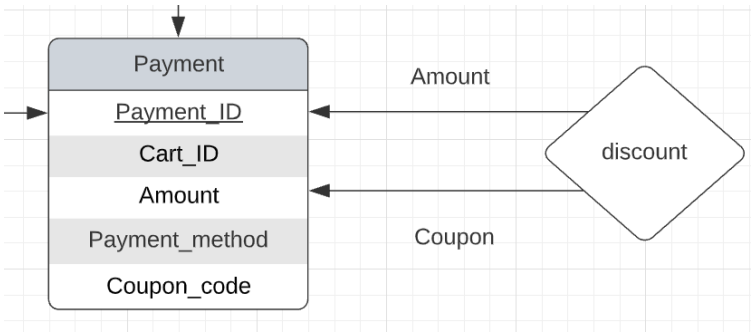
- **Total Participation :-** Here each entity is involved in at least one relationship as whenever a payment is completed an Order is placed which is accepted by that customer's Order history entity , therefore the acceptance of Order in Order history shows total participation.



- **Partial Participation :-** Here every product viewed by the customer does not always end up in the cart as it is the choice of the customer, therefore we observe a partial relationship here since only a particular set of products will be added to the cart.



Relationship Roles :- There is only one case of relationship role in the ER diagram . We apply one coupon to the relationship "Discount" Which gives us the discounted price "Amount".



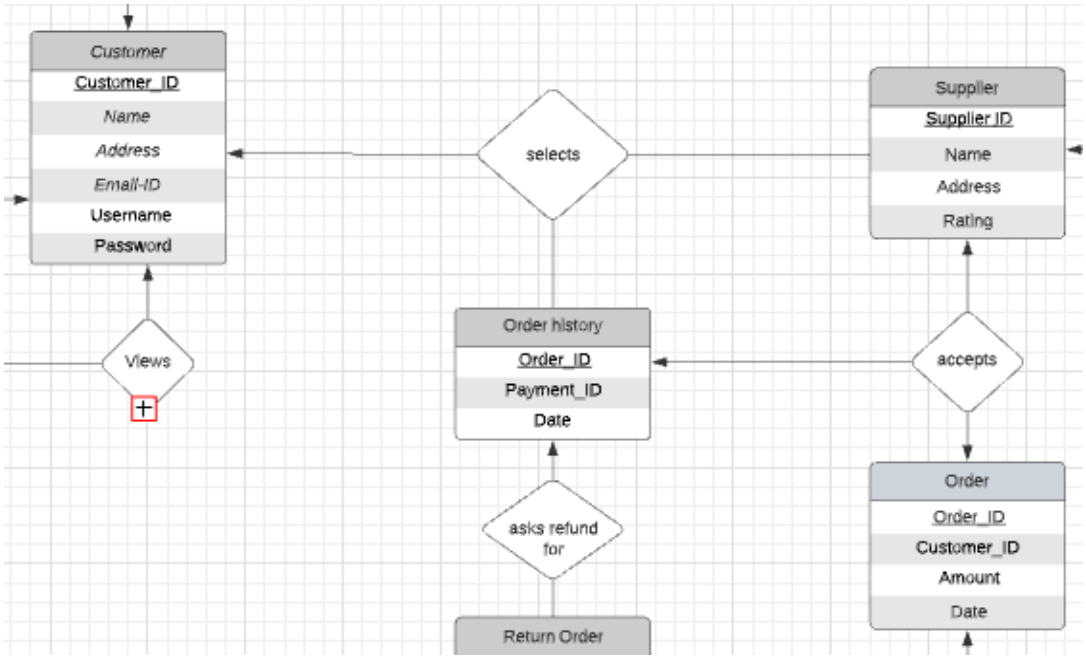
Constraints :-

- NOT NULL - This ensure that a column cannot have a NULL VALUE
- UNIQUE - Ensures that all values in a column are different
- PRIMARY KEY - Uniquely identifies each row in a table
- FOREIGN KEY - Prevents actions that would destroy links between tables.

Ternary Relationships :-

- **Selects** :- Views is a ternary relationship having Customer , Supplier and order history as entities . Customers will have to select the supplier . Inorder for customer to see previous orders order history will have to selected. To return the order the supplier will have to be selected.

- **Accepts** :-Accepts is also a ternary relationship having order,order_history,Supplier as supply accepts the order . Then order history accepts the order history as an entry when the order is accepted . Suppliers can access the order history of the particular orders he has delivered to the customer.



Relational Schema

Logical Database Design :-

- **Customer**(Customer_ID, Name, Address, Email-ID, Username, Password, {Coupon Code}, {Telephone number})
- **Category**(Category_ID, Category Name)
- **Product**(Product_ID, Category_ID, Item name, Price, Quantity, Brand, Availability, {Reviews})
- **Cart**(Cart_ID, Product_ID, Item_name, Quantity, Total price)
- **Payment**(Payment_ID, Cart_ID, Amount, Payment_method, Coupon_code)
- **Order**(Order_ID, Customer_ID, Amount, Date)
- **Supplier**(Supplier_ID, Name, Address, Rating)
- **Delivery**(Delivery_ID, Customer_ID, Status, Helpline)
- **Order history**(Order_ID, Payment_ID, Date)
- **Return Order**(Return_ID, Order_ID)

Mapping Constraints :-

- **Customer to Supplier (I : M)**
- **Customer to Category (I : M)**
- **Customer to Order_history (I : M)**
- **Category to Product (I : M)**
- **Product to Cart (M : I)**
- **Cart to Payment (I : I)**
- **Payment to Order (I : I)**
- **Order to Supplier (I : I)**
- **Order to Order_history (I : I)**
- **Order_history to Return_Order (I : M)**
- **Supplier to Delivery (I : M)**

Integrity Constraints :-

- **Customer**
    - Customer_ID (integer(5), primary key)
    - Name (char(25), not null)
    - Address (varchar(255), not null)
    - Email_ID (varchar(255), default null)
    - Username (varchar(30), not null, unique)
    - Password (varchar(20), not null, unique

- **Telephone_number**
    - Phone_number (varchar(13), not null)
    - Customer_ID (integer(5), referential integrity constraint Customer(Customer_ID))

- **Coupon Code**
    - Coupon_name varchar(7) not null)
    - Customer_ID integer(5) not null)
    - Customer_ID (integer(5), referential integrity constraint Customer(Customer_ID))

- **Category**
    - Category_ID (integer(3) primary key)
    - Category_name (char(25) ,not null ,unique)

- **Product**
    - Product_ID (integer(5), primary key)
    - Category_ID (integer(3), not null)
    - Item_name (char(25), not null)
    - Price (float(5), not null)
    - Quantity (integer(3), not null)
    - Brand (char(10), default null)
    - Availability (varchar(5), not null)

- **Reviews**
    - Review (varchar(255) ,not null)
    - Product_ID (integer(5) ,not null)
    - Product_ID(integer(5), referential integrity constraint Product(Product_ID))

- **Cart**
    - Retail_ID (integer (5), primary key)
    - Cart_ID (integer (5), not null)
    - Product_ID (integer(5) .not null .unique)
    - Customer_ID (integer(5), not null)
    - Item_name (char(25) ,not null)
    - Quantity (int(3),not null)
    - Total_price (float(7),not null)
    - Product_ID(integer(5), referential integrity constraint Product(Product_ID)

- **Payment**
    - Payment_ID (integer(10), primary key)
    - Cart_ID (integer(5), not null)
    - Amount (float(7), not null)
    - Payment_method (varchar(10), default null)
    - Coupon_code (varchar(7), default null)

- **_Order_**
    - Order_ID (int (5) ,primary key)
    - Payment_ID (integer(10), not null, unique, referential integrity constraint Payment (Payment_ID))
    - Customer_ID (int(5) ,not null, unique)
    - Amount (float(7) ,not null)
    - Order_Date (DATE)

- **Delivery**
    - Delivery_ID (integer(5), primary key)
    - Customer_ID (integer(5), not null)
    - Order_ID (integer(5), not null, unique, referential integrity constraint _Order_(Order_ID))
    - Status (char(20), not null)
    - Helpline (integer(13), not null)

- **Supplier**
    - Supplier_ID (integer(5), primary key)
    - Supplier_name (char(25), not null)
    - Address (varchar(255), not null)
    - Rating (integer(5), not null)

- **Order_history**
    - Order_ID (integer(5), primary key)
    - Customer_ID (integer(5), not null, unique, referential integrity constraint Customer(Customer_ID))
    - Payment_ID (integer(10), not null. unique, referential integrity constraint Payment (Payment_ID))
    - Order_Date (DATE, not null)

- **Return_order**
    - Return_ID (integer(5), primary key)
    - Order_ID (integer(5), not null, unique, referential integrity constraint _Order_(Order_ID))