ER Diagram Deliverables

Description of Problem

The database stores details about the products that are listed on the website as well as the details of sellers and users who are buying these products to facilitate smooth working and closely match a modern online retail store

Scope of Project

The database stores details of sellers, users, products, delivery persons, customer care queries, orders, returns, reviews and payments.
It allows the users to filter products according to their preferences helping them narrow their search down to the best fit for the best price.
To allow the buyers to make better decisions each product will have reviews and a star rating attached to them given by previous buyers.
The purchases made will also be stored in the user's history.
The database also maintains a record of the deliveries made and the delivery persons assigned to the orders.
If unsatisfied, a user can return a product back and a delivery person will be assigned for the same. Retail Store points equivalent to the price of the product will be instantly credited to the user's account which can be redeemed for future purchases.
The users can also contact the helpline in case of any problems and the database will maintain a record of the problems registered.
The database will be updated at the end of each transaction to maintain consistency.

Stakeholders

Buyers - The users who browse the retail store to find and compare products offered by sellers and purchase them.
Sellers - Those who offer their products up for sale to the users of the retail store.
Employees - The people who represent the retail store to the users. In our case, these are of 2 types - Delivery Persons and Customer Care Employees

Defining Entities

USER - A user is any person who makes an account in the retail store. He/She can log into the account using the username and password that they set at the time of sign-up. Each user is identified by a unique User ID. The address that the user will enter at the time of login or at the time of purchase will be stored for future convenience. User is the entity who will navigate the front-end to browse, filter, review, purchase and return products. The user entity also keeps a record of the points that are credited to the user in case of a return.

A user will be defined by the following attributes
User ID
Username
Password
Name
E-mail
Contact No.
Registration Date
Address
- Apartment No.
- Street Name
- City
- State
- Pincode

Points

SELLER - A seller is an entity that offers various products on the retail store for the users to browse and buy. Each seller is identified by a unique Seller ID. The database keeps a record of each seller's personal details

A seller will be defined by the following attributes
Seller ID
Name
Location
Contact No
Email ID

CATEGORY - The category entity contains details about the various categories into which the available products can be sorted. Each category is identified by a unique Category ID.

A category will be defined by the following attributes

Category ID
Category Name

PRODUCT - Products is an entity that holds details about all the products offered by sellers and bought by users in this virtual retail store. Every product is identified by a unique Product ID. Each product is divided into categories and is linked to the appropriate category in the category entity via the Category_ID foreign key. The details of the product that the user might like to know like the seller, price, days of arrival are all stored.

A product will be defined by the following attributes
Product ID
Category ID
Name
Price
Quantity Available
Seller ID
Days to arrive
Description

CART - A cart is an entity that represents the products that a user wants to buy. A user can keep adding products to a cart and then purchase all of them together. Each cart is uniquely identified by a combination of the Cart ID, User ID and Product ID. The Cart ID gets incremented by 1 every time the user checks out his cart. A user can have many carts and one cart can hold many products.

A cart will be defined by the following attributes
Cart ID
User ID
Product ID
Quantity Ordered

PAYMENT - The payment entity keeps track of all the transactions made from the user to purchase products available in the retail store. Each payment transaction is uniquely identified by the Payment ID

A payment will be defined by the following attributes
Payment ID

User ID
Type Status
Date of translation
Amount

ORDERS - The orders entity contains a record of all the orders that were placed, i.e., all the carts that have been checked out. It is capable of tracing the products that were ordered in those carts by using a combination of User ID and Cart ID. Each order can be uniquely identified by its Order ID. The date of delivery (Calculated by adding days that a product needs to arrive to the current date) and a reference to the Payment Entity using the Payment ID is also stored.

An order will be defined by the following attributes
Order ID
User ID
Date of Delivery
Payment ID
Cart ID

REVIEW - A user can review a product in terms of stars or comments. These reviews are an entity. The reviews given to one product by one user can be viewed by other users which can help them make informed choices. Each review can be uniquely identified by a Review ID

A review will be defined by the following attributes
Review_ID
Text
Stars
User_ID
Product_ID
Timestamp

RETURN - A user applies for the return of an order. Thus, all the returns filed form a separate entity. A delivery person is assigned to pick up that return. The return can be uniquely identified by the order ID.

A return will be defined by the following attributes
Order ID
DP ID
Return Status

DELIVERY - Whenever a delivery is to be made, i.e., any transport of products is to take place, either purchase from seller to buyer or a return from buyer to seller, it is considered to be a delivery. A track of all such deliveries and the delivery person assigned to each delivery is kept as a separate entity. Each delivery is uniquely identifiable by a combination of the Order ID and the Deliver Person ID assigned to it.

A delivery will be defined by the following attributes
Order ID
DP ID
Delivery Status

DELIVERY PERSON - Delivery persons are responsible for delivering orders and picking up returns. The data of all delivery persons are stored as a separate entity. Each Delivery Person can be uniquely identified by Delivery Person ID.

A deliver person will be defined by the following attributes
DP_ID
Name
Date of Joining
Contact No.

CUSTOMER CARE - A user can talk to Customer Care in case of any query regarding the functionality of the portal. The details of each Customer Care individual are stored as a separate entity. Each customer care individual is uniquely identified by a Customer Care ID.

A customer Care individual will be defined by the following attributes
CC_ID
Name
Contact_No.

CUSTOMER QUERIES - Customer Queries asked by users are answered by Customer Care. The data regarding each query is stored as a separate entity. Each query can be uniquely identified by Query_ID.

A Query will be defined by the following attributes
Query_ID
User_ID
CC_ID

Time
Query

Relationships Established

- User **buys** product.
- User **writes** review about the product.
- User **adds** the products to the cart.
- User **places** the order.
- User **makes** the payment.
- User **initiates** the return for the order.
- Seller **sells** the products.
- Cart **contains** the products.
- A Product **belongs to** a category.
- Payment **is done for** the order.
- Order **is sent for** delivery.
- Delivery Person **picks up** the returned order.
- Delivery is **delivered by** the Delivery Person.
- Cart **checkout** with the payment.
- User **talks to** Customer Care.
- User **asks** Customer Queries.
- Customer Queries **are answered by** Customer Care.

Weak Entities

- Return
  Return is a weak entity because returns can not be initiated until an order is placed by the User.
- Delivery
  Delivery is a weak entity because a delivery person cannot make a delivery until an order is placed.

Entities Participation Type

Total Participation

- All products belong to a category
- All products are sold by a seller

- All orders are sent for delivery
- All returns are picked up by delivery persons
- All deliveries are delivered by delivery persons
- All queries are answered by Customer Care Employees

Partial Participation

- Some users may write reviews for products
- Some users might buy a product, i.e., Some users might add products to a cart and check them out and pay for the order
- Some users might initiate a return
- Some products are contained in a cart
- Some users ask queries

Relationship Roles

Relationship	Roles
User writes review	FEEDBACK
User initiates return	APPLIES
Delivery Person picks up the return order.	TAKES
Order is sent for delivery	DISPATCHED
User makes payment	PAYS
Seller sells the product.	RETAILS
Customer Query is answered By Customer Care.	SOLVED

Relationship Constraints

- 1 User can write many reviews about a product.
- 1 User can add many products to the cart (At least 1)
- 1 User can place many orders
- 1 User can make many payments
- 1 User can initiate many orders
- 1 Seller can sell many products and 1 product can be sold by many sellers
- 1 Cart can contain many products and 1 product can be present in the multiple carts
- Many Products belong to 1 category.
- 1 Payment is done for 1 order.

- 1 Order is sent for delivery once. In case of a return, the order will be delivered twice. A delivery may contain many orders.
- 1 Delivery Person can pick up many returned orders
- Many Deliveries are delivered by 1 Delivery Person (At least 1)
- Many carts can be checked out for many payments. (At least 1)
- 1 Customer Care individual can solve any number of queries.
- 1 user can talk to many Customer Care Individuals.
- 1 user can ask many queries.

Ternary Relationship

1. The USER **adds** PRODUCT to the CART.
   Here all the three entities user, product, cart take part in the relationship adds. A product is added to the cart by the user.
2. The USER **writes** REVIEW about the PRODUCTS.
   Here all the three entities user, product, review take part in the relationship writes.
3. The USER **initiates** RETURN for the ORDER.
   Here all the three entities user, return, order take part in the relationship initiates.

DDL

```
CREATE TABLE `User` (
  `User_ID` Integer NOT NULL,
  `Username` VARCHAR(50) NOT NULL UNIQUE,
  `Password` VARCHAR(50) NOT NULL,
  `Name` VARCHAR(50) NOT NULL,
  `Email` VARCHAR(500) NOT NULL UNIQUE,
  `Contact_No` VARCHAR(15) NOT NULL,
  `Reg_Date` Date,
  `Address` VARCHAR(50) NOT NULL,
  `Points` Integer DEFAULT 0,
  PRIMARY KEY (`User_ID`)
);

CREATE TABLE `CATEGORIES` (
```

```
  `Category_ID` Integer NOT NULL,
  `Category_Name` VARCHAR(15) NOT NULL UNIQUE,
  PRIMARY KEY (`Category_ID`)
);

CREATE TABLE `SELLER` (
  `Seller_ID` Integer NOT NULL,
  `Name` VARCHAR(50) NOT NULL,
  `Location` VARCHAR(50) NOT NULL,
  `Contact_No` VARCHAR(15) NOT NULL,
  `Email_Id` VARCHAR(100) NOT NULL UNIQUE,
  PRIMARY KEY (`Seller_ID`)
);

CREATE TABLE `PRODUCTS` (
  `Product_ID` Integer,
  `Category_ID` Integer NOT NULL,
  `Name` VARCHAR (50),
  `Price` Float NOT NULL,
  `Quantity_Available` Integer DEFAULT 0,
  `Seller_ID` Integer NOT NULL,
  `Days_to_Arrive` Integer NOT NULL,
  `Description` VARCHAR(500) NOT NULL,
  PRIMARY KEY (`Product_ID`),
  FOREIGN KEY(Category_ID) REFERENCES
CATEGORIES(Category_ID),
FOREIGN KEY(Seller_ID) REFERENCES SELLER(Seller_ID)
);

CREATE TABLE `Reviews` (
    `Review_ID` INTEGER NOT NULL,
    `Text` VARCHAR(500),
    `Stars` INTEGER NOT NULL,
    `User_ID` INTEGER NOT NULL,
    `Product_ID` INTEGER NOT NULL,
    `TimeStamp` TIMESTAMP NOT NULL,
```

```
    PRIMARY KEY (`Review_ID`),
    FOREIGN KEY (User_ID)
        REFERENCES USER (User_ID),
    FOREIGN KEY (Product_ID)
        REFERENCES PRODUCTS (Product_ID)
);

CREATE TABLE `Cart` (
  `Cart_ID` Integer,
  `User_ID` Integer NOT NULL,
  `Product_ID` Integer NOT NULL,
  `Quantity_Ordered` Integer DEFAULT 1,
  PRIMARY KEY (`Cart_ID`,`User_ID`,`Product_ID`),
FOREIGN KEY(User_ID) REFERENCES USER(User_ID),
FOREIGN KEY(Product_ID) REFERENCES PRODUCTS(Product_ID)
);

CREATE TABLE `Payment` (
  `Payment_ID` Integer NOT NULL,
  `User_ID` Integer NOT NULL,
  `Type` VARCHAR(50) NOT NULL,
  `Status` VARCHAR(15) NOT NULL,
  `Date` Date NOT NULL,
  `Amount` Float NOT NULL,
  PRIMARY KEY (`Payment_ID`),
FOREIGN KEY(User_ID) REFERENCES USER(User_ID)
);

CREATE TABLE `Orders` (
  `Order_ID` Integer,
  `User_ID` Integer NOT NULL,
  `Date_Of_Delivery` Date NOT NULL,
  `Payment_ID` Integer NOT NULL,
  `Cart_ID` Integer NOT NULL,
  PRIMARY KEY (`Order_ID`),
FOREIGN KEY(User_ID) REFERENCES Cart(User_ID),
```

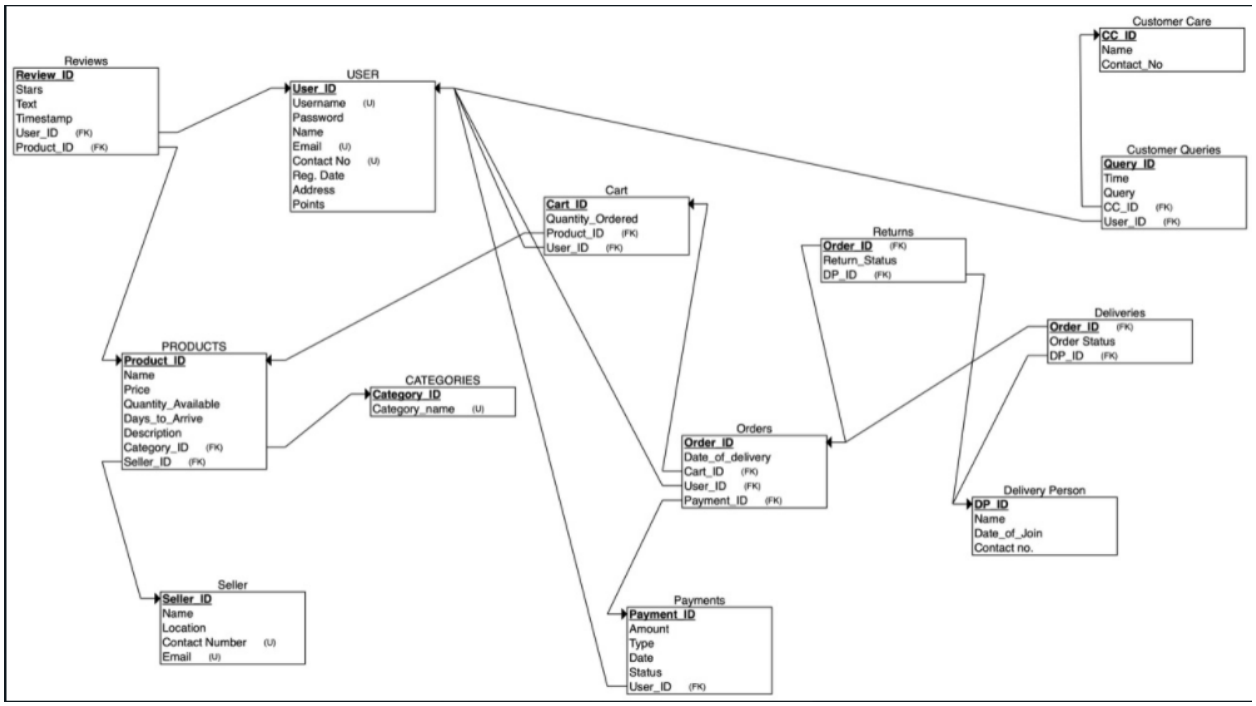
```
FOREIGN KEY(Payment_ID) REFERENCES Payment(Payment_ID),
FOREIGN KEY(Cart_ID) REFERENCES Cart(Cart_ID)

);

CREATE TABLE `Delivery_Person` (
  `DP_ID` Integer,
  `Name` VARCHAR(100) NOT NULL,
  `DOJ` Date NOT NULL,
  `Contact` VARCHAR(100) NOT NULL,
  PRIMARY KEY (`DP_ID`)
);

CREATE TABLE `Deliveries` (
  `Order_ID` Integer NOT NULL,
  `DP_ID` Integer NOT NULL,
  `Order_Status` VARCHAR(50) NOT NULL,
FOREIGN KEY(Order_ID) REFERENCES ORDERS(ORDER_ID),
FOREIGN KEY(DP_ID) REFERENCES Delivery_Person(DP_ID)
);

CREATE TABLE `Returns` (
  `Order_Id` Integer NOT NULL,
  `DP_ID` Integer NOT NULL,
  `Return_Status` VARCHAR(50) NOT NULL,
FOREIGN KEY(Order_ID) REFERENCES ORDERS(ORDER_ID),
FOREIGN KEY(DP_ID) REFERENCES Delivery_Person(DP_ID)
);

CREATE TABLE `Customer_Care` (
  `CC_ID` Integer,
  `Name` VARCHAR(50),
  `Contact_No` VARCHAR(15),
  PRIMARY KEY (`CC_ID`)
);
```

```
CREATE TABLE `Customer_Queries` (
  `Query_ID` Integer,
  `User_ID` Integer,
  `CC_ID` Integer,
  `Time` Timestamp,
  `Query` VARCHAR(200),
  PRIMARY KEY (`Query_ID`),
FOREIGN KEY(User_ID) REFERENCES USER(User_ID),
FOREIGN KEY(CC_ID) REFERENCES Customer_Care(CC_ID)
);
```

RELATIONAL SCHEMA

Primary Key
Foreign Key
Primary Key+Foreign Key
Unique Attribute
*->Discriminator of a weak entity set.

User(**User_ID**,Username,Password,Name,Email,ContactNo.,Reg. Date,Address,Points)
Products(**Product_ID**,Name,Price,Quantity_Available,Days_to_Arrive,Description, Category_ID,Seller_ID)
Seller(**Seller_ID**,Name,Location,Contact Number,Email)
Category(**Category_ID**,Category_name)
Review(**Review_ID**,Stars,Text,TimeStamp,User_ID,Product_ID)
Cart(**Cart_ID**,Quantity_Ordered,Product_ID,User_ID)
Order(**Order_ID**,Date_of_delivery,Cart_ID ,User_ID,Payment_ID)
Payment(**Payment_ID**,Amount,Type,Date,Status,User_ID)
Return(**Order_ID***,Return_Status,DP_ID)
DeliveryPerson(**DP_ID**,Name,Date_of_Join,Contact no.)
Delivery(**Order_ID***,Order Status,DP_ID)
Customer Care(**CC_ID**,Name,Contact_No)
Customer Queries(**Query_ID**,User_ID,CC_ID,Time,Query)



SQL QUERIES:

1. **Find out the purchase history of the customer**
   select name from products p where p.product_id in (select product_id from cart c, orders o where o.cart_id= c.cart_id AND c.user_id = 1);

2. **Find the delivery status for any given user.**
   select order_status from deliveries d where d.order_id in (select o.order_id from orders o where o.user_id = 1);

3. **Filter products according to different price**
   select p.product_id, p.Name, p.Price from products p
   order by p.price;

4. **Give the contact no of delivery person who is delivering an order to the respective customer**
   select DP.Name, DP.Contact from DELIVERY_PERSON DP where DP.DP_ID in(select DP_ID from deliveries d where d.order_id in (select o.order_id from orders o where o.user_id = 1) AND d.order_id = 7);

5. **Provide reviews of a particular rating for a product to the customer.**
   Select * from reviews where product_id =13 and stars>=4;

6. **Provide the location of the seller of a product to the customer.**
   select location from seller where seller_id in(
   select seller_id from products where product_id = 1);

7. **See the sales of a particular product**
   select c.product_id, sum(c.quantity_ordered), p.price, sum(c.quantity_ordered) * p.price as revenue from cart c, products p where c.cart_id in
   (select cart_id from orders)
   and p.product_id = c.product_id
   group by c.product_id**;**

8. **Give the contact info of the customer care representative handling a particular query to the customer**
   select cc.name, cc.Contact_no from CUSTOMER_CARE cc where cc.CC_ID in
   (select CC_ID from CUSTOMER_QUERIES cq where cq.user_id = 13);

9. **List a particular user's most ordered "category"**
   (select ca.Category_Name, p.category_id, count(p.category_id) as no_of_products from products p, categories ca where p.product_id in
   (select product_id from cart c, orders o where o.cart_id= c.cart_id AND c.user_id = 1)
   and ca.category_id = p.category_id
   group by category_id
   order by no_of_products desc);

10. **List no. of deliveries done by all delivery people**
    select d.dp_id, dp.name, count(d.dp_id) as No_of_deliveries_made from deliveries d, delivery_person dp where d.dp_id = dp.dp_id group by d.dp_id;

Links

1) **ER-Diagram**
   https://lucid.app/lucidchart/6ba25276-60f8-43aa-a3f3-926e5be456ac/edit?invitationId=inv_1078bfbe-8dc1-4f6b-8578-da3cbb749114

2) **Drive folder for all SQL dump**
   https://drive.google.com/drive/folders/1LHdAMIb48hCVKAysR2fCSatx8gfe2C9x?usp=sharing

Contributions

Gautam Reddy
- ER Diagram
- Sufficient and valid Data Population in all tables
- Insertion of data and cardinality
- Identification and formulation of 10 distinct SQL Queries

Karan Baboota
- ER Diagram
- Identification and formulation of 10 distinct SQL Queries
- Sufficient and valid Data Population in all tables
- Insertion of data and cardinality

Khwaish Rupani
- ER Diagram
- Identifying the Scope, Stakeholders and Relationships
- Identification of Weak Entity, Ternary Relationship, Entity Relationship, Participation and Definition

Muskan Yadav
- ER Diagram
- Converting ER Diagram into Relationship Schema - Logical Database Design
- Mapping sufficient and valid Constraints in DDL