Deadline 1

DBMS Project :
Defining Project Scope and Requirements

Vansh Yadav (2022559)

Jan 22, 2023

Project Scope

Objective is to establish an easily navigable online retail store, providing customers with a seamless browsing and purchasing experience. The system will employ a relational database powered by MySQL to store and oversee data related to customers, products, and transactions.

The primary goal is to enhance the convenience of shopping for customers and to efficiently handle product management, transactions, and customer interactions, allowing consumers to make purchases from the comfort of their homes. The project encompasses a broad range of functionalities, data management tasks, and technical specifications to create a resilient and secure online retail platform.

This platform will offer a diverse array of grocery items spanning various categories and sub-categories, while also incorporating multiple payment options for user flexibility.

The execution will cover essential backend processes and an intuitive user interface, ensuring a seamless and gratifying shopping experience for users. Ideally, the system should support effective catalog searches and be capable of handling a substantial volume of transactions.

Features and Project Objectives

1. *Customer Authentication and Account Management:*
    - Customers are required to log in using existing credentials or create a new account by providing necessary information and setting a secure password.

2. *Browsing and Product Selection:*
    - Customers should have the flexibility to explore the website's catalog, select a product, and review its details.
    - Options to add products to the cart, wish-list, or complete the purchase by providing personal and payment information should be available.

3. *Digital Transaction Options:*
    - For digital transactions, customers should be able to choose from options such as card or UPI.

4. *Post-Transaction Access and Control:*
    - Following a successful transaction, customers gain access to features like order tracking, product reviews, and a comprehensive order history overview.
    - Customers retain control over returns, order cancellations, and financial management through their wallets.

5. *Shop Owners' Dashboard:*

- Shop owners, the architects of the marketplace, can register and gain insights into their business performance, including items sold and revenue generated.
- They can actively manage product listings, request additions or removals, and propose new items or price adjustments to the website administrator.

6. *Delivery Agent Functionality:*
    - Delivery agents can view their assignments, including pick-up and delivery details.
    - Compensations for successfully completed deliveries are neatly organized in their wallets.

7. *Administrator Authority:*
    - The admin oversees all aspects of the product lineup.
    - They hold the authority to add or remove items, introduce discounts, and establish special offers for customers.

Identifying Entity Sets and Relationship Sets

Entity Sets:

1. **Customer**:
    - Attributes: CustomerID (or Login ID), Password, Email Address, Phone Number, Delivery Address.

2. **Product:**
    - Attributes: ProductID, Product Name, Description, Stock Availability.

3. **Order:**
    - Attributes: OrderID, CustomerID, ProductID, Order Status, Payment Information.

4. **Wallet:**
    - Attributes: WalletID, CustomerID, Compensation Amount.

5. **Shop Owner:**
    - Attributes: OwnerID, Shop Name, Revenue.

6. **Item Listing:**
    - Attributes: ListingID, ShopID, ProductID, Price.

7. **Delivery Agent:**
    - Attributes: AgentID, Delivery Status, Compensation Amount.

8. **Administrator:**
    - Attributes: AdminID, Special Offers, Discounts.

9. **Review:**
    - Attributes: ReviewID, CustomerID, ProductID, Review Text.

Relationship Sets and Cardinality:

1. **Customer-Order Relationship:**
    - Cardinality: One-to-Many (One customer can place many orders; each order is placed by exactly one customer.)
    - Participation: Mandatory on the side of the customer (Each customer must place at least one order).

2. **Product-Order Relationship:**
    - Cardinality: Many-to-Many (Each order can include multiple products, and each product can be part of multiple orders.)
    - Participation: Optional on both sides (An order can exist without products, and a product can exist without being part of any order).

3. **Customer-Review Relationship:**
    - Cardinality: One-to-Many (One customer can leave many reviews; each review is associated with exactly one customer.)
    - Participation: Optional on the side of the customer (A customer may not leave any reviews).

4. **Customer-Wallet Relationship:**
    - Cardinality: One-to-One (Each customer has exactly one E-wallet for financial transactions; each E-wallet is associated with exactly one customer.)
    - Participation: Mandatory on the side of the customer (Each customer must have an E-wallet).

5. **Shop Owner-Item Listing Relationship:**

- Cardinality: One-to-Many (One shop owner can list many items for sale; each item listing is associated with exactly one shop owner.)
- Participation: Mandatory on the side of the shop owner (Each shop owner must list at least one item).

6. **Product-Item Listing Relationship:**
   - Cardinality: One-to-Many (Each product can be listed in multiple item listings; each item listing is associated with exactly one product.)
   - Participation: Optional on the side of the product (A product may not be listed in any item listing).

7. **Delivery Agent-Delivery Relationship:**
   - Cardinality: One-to-Many (One delivery agent can be assigned to many deliveries; each delivery is assigned to exactly one delivery agent.)
   - Participation: Optional on the side of the delivery agent (A delivery agent may not be assigned to any deliveries).

8. **Administrator-Product Relationship:**
   - Cardinality: One-to-Many (The administrator oversees many products; each product is overseen by exactly one administrator.)
   - Participation: Optional on the side of the administrator (An administrator may not oversee any products).

Technical Requirements

Following are the tools and technologies we plan on using in this project. Please note since this is only the first draft/deadline of the project, we are open to changes and may modify the tools we use in the final project.

- **JavaScript**
- **Python3**
- **CSS**
- **HTML**
- **MySQL (Relational Database Management System)**
- **Django (Python-based Backend Framework)**
- **React (JavaScript-based Frontend Library)**

Deadline 2

DBMS Project :

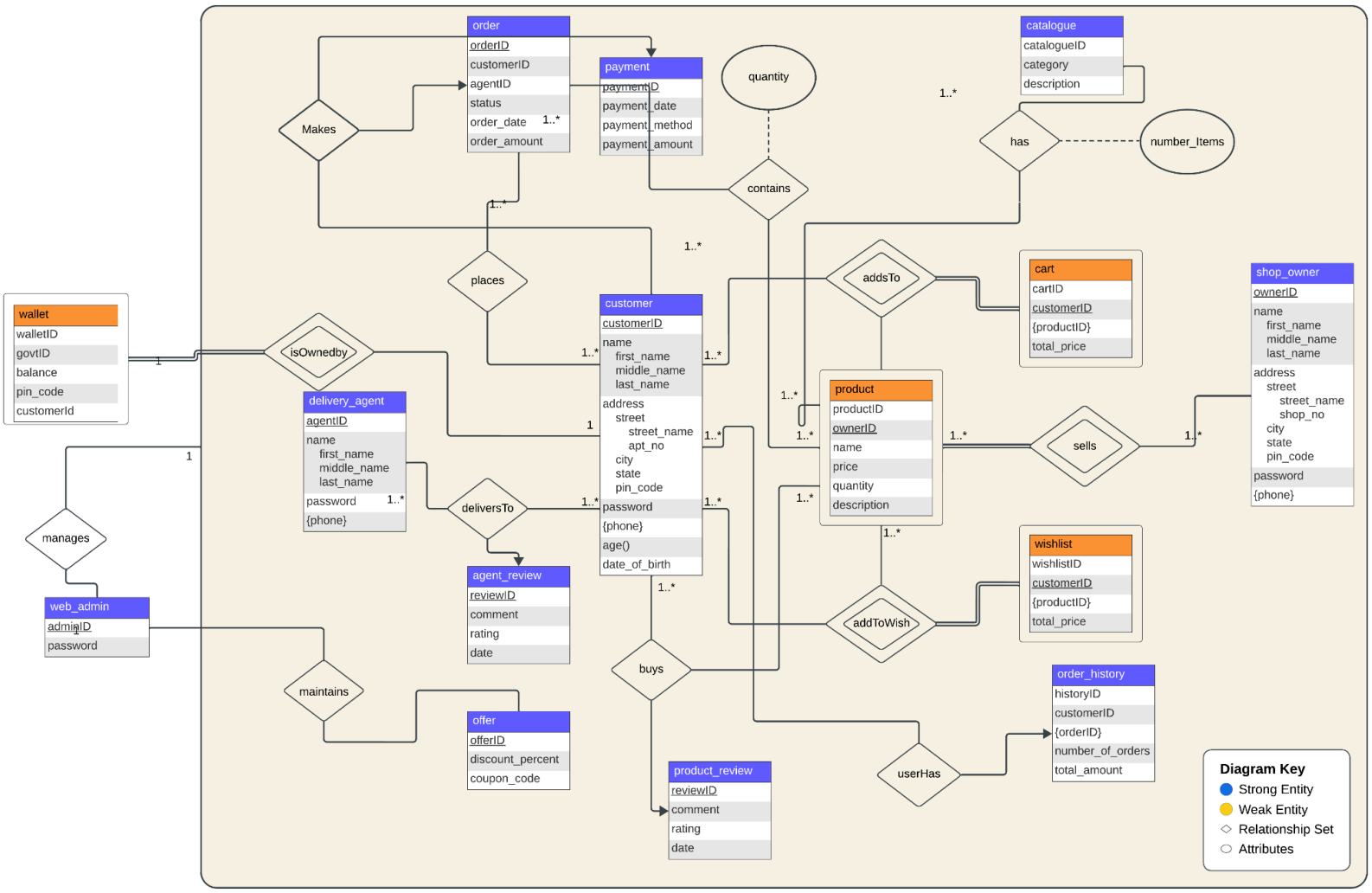
Describing the ER model and converting it into Relational model

Vansh Yadav (2022559)

Jan 29, 2023

Entity-Relationship Model

The Entity-Relationship (ER) model is a conceptual data model used in database design to represent relationships between entities. It defines entities as objects or concepts in the real world and their relationships, providing a visual representation that aids in understanding the structure and organization of a database.



Relational Model

Relationship Models serve as a means to depict the storage structure of data in a database, encompassing the characteristics of each entity and relationship.

The design of the Relational Model adheres to the specified assumptions and constraints outlined in the document above.

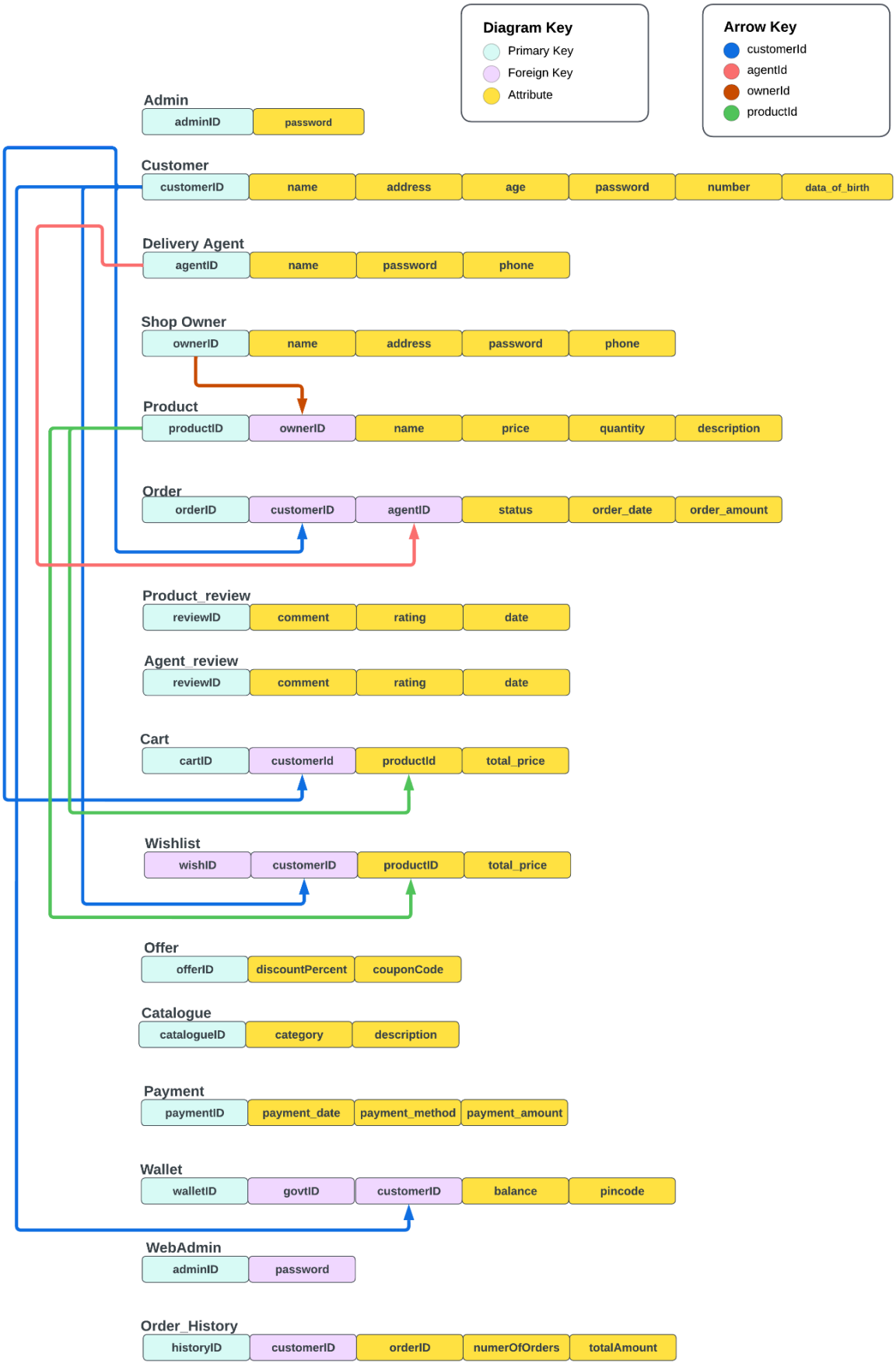


Diagram Key
Primary Key
Foreign Key
Attribute
Arrow Key
customerId
agentId
ownerId
productId
Admin
adminID
password
Customer
customerID
name
address
age
password
number
data_of_birth
Delivery Agent
agentID
name
password
phone
Shop Owner
ownerID
name
address
password
phone
Product
productID
ownerID
name
price
quantity
description
Order
orderID
customerID
agentID
status
order_date
order_amount
Product_review
reviewID
comment
rating
date
Agent_review
reviewID
comment
rating
date
Cart
cartID
customerId
productId
total_price
Wishlist
wishID
customerID
productID
total_price
Offer
offerID
discountPercent
couponCode
Catalogue
catalogueID
category
description
Payment
paymentID
payment_date
payment_method
payment_amount
Wallet
walletID
govtID
customerID
balance
pincode
WebAdmin
adminID
password
Order_History
historyID
customerID
orderID
numerOfOrders
totalAmount

Contributions:

Vansh Yadav
(2022559)