E-COMMERCE DATABASE MANAGEMENT SYSTEM

OBJECTIVES

The Prime Objective of our database project is to design a robust E-commerce database by performing operations such as

- ❖ Viewing orders
- ❖ Placing orders
- ❖ Updating database
- ❖ Reviewing products
- ❖ Maintaining data consistency across tables

Using features such as :

- Triggers
- Stored procedures
- Functions
- Transactions

PROPOSED BY

19CS1076 19CS1113 19CS1100

N.L.S HARSHA UTKARSH SAURABH KISHOR

FUNCTIONAL REQUIREMENTS

- ★ A Customer can see the account details and can update if required.
- ★ Customer can search the products according to the category.
- ★ Customer can add his wishlist to the cart and can see the total amount.
- ★ Customer can update the cart whenever required.
- ★ Customer can choose the mode of payment.
- ★ Customer can keep track of the order by seeing order status.
- ★ Customer can review the products which have been purchased.
- ★ A Seller can see the account details and can update if required.
- ★ Seller can add or delete the products.
- ★ Seller can update the stock of a particular product whether it is available or not.
- ★ Seller can keep track of total sales of his products.
- ★ Seller can know the sales on a particular day or month or year.
- ★ Seller can see the customer reviews and can improve the quality of products based on that.

NON FUNCTIONAL REQUIREMENTS

- ★ A Customer cannot access the Seller details and vice-versa.
- ★ There should not be any inconsistency in the data.
- ★ There should not be any loss of data.

Entities and their Attributes

ENTITIES	ATTRIBUTES	ATTRIBUTE TYPE	Entity Type
Customer	Customer_CustomerId Name Email DateOfBirth Phone Age	Simple Composite Simple Simple Multivalued Derived	Strong
Order	OrderId ShippingDate OrderDate OrderAmount Cart_CartID	Simple Simple Simple Simple Simple	Strong
OrderItem	Order_OrderId (PK) Product_ProductId(FK) MRP Quantity	Simple Simple Simple Simple	Weak

Product	productId (PK) ProductName(FK) sellerId MRP CategoryID Stock Brand	Simple Simple Simple Simple Simple Simple Simple	Strong

Review	ReviewId(PK)
	Description Ratings
	Product_ProductId
	Customer_CustomerID(FK)
		Simple
		Simple
		Simple
		Simple
			Strong

Cart	cartId (PK) Customer_customerId (FK) GrandTotal ItemsTotal	Simple Simple Derived Derived	Strong
Category	CategoryID(PK) CategoryName DESCRIPTION	Simple Simple Simple	Strong
seller	sellerId (PK) Name Phone Total_Sales	Simple Simple Multivalued Derived	Strong

Payment	payment_id (PK) Order_OrderId (FK) PaymentMode Customer_CustomerId Date_of_payment	Simple Simple Simple Simple Simple	Strong

Entities and Relations

Entities	Relation	Cardinality	Type of participation
Customer Address	Stays At	One To One	Total Partial
Customer Cart	Shops	One To One	Partial Total
Customer Order	Places	One To Many	Partial Total
Customer Payment	Makes	One To Many	Partial Total
Customer Review	Write	One To Many	Partial Total

Seller Product	Sells	Many To Many	Partial Total
Category Product	Categorizes	One To Many	Partial Total
Cart Product	Contains	Many To Many	Partial Partial
Product OrderItem	Includes	One To Many	Partial Total
Order OrderItem	Includes	One To One	Partial total
Payment Order	For	One To One	Total Total

QUERIES ON THE ABOVE RELATIONAL SCHEMA

1.Stored procedure for the details of the customer.

2.View for getting sales by category of products.

3.Using triggers to update the no.of products as soon as the payment is made.

4.Stored procedure for getting order history.

5.Processing an order

- To process an order, one should check whether those items are in stock.
- If items are in stock, they need to be reserved so that they go in hands of those who have expressed them in wishlist/order.
- Once ordered the available quantity must be reduced to reflect the correct value in the stock.
- Any items not in stock cannot be sanctioned; this requires confirmation from the seller.

- The customer needs to be informed as to which items are in stock (and can be shipped immediately) and which are cancelled.

6.

- Check whether the specified customer exists
- IF NOT EXISTS add him/her
- COMMIT the info
- Fetch the customer id
- INSERT a row to Order tables
- If unable to do so,ROLLBACK;
- Fetch the new orderid in orders table
- INSERT row to the order table for every product ordered
- If adding tuples to orderitems fails ROLL BACK all tuples of products added for and the tuple in order row

QUERY 1:Customers to find products with highest ratings for a given category.

QUERY 2:Customers to filter out the products according to their brand and price.

QUERY 3:Customers to compare the products based on their ratings and reviews.

QUERY 4:Customers to find the best seller of a particular product.

QUERY 5:List the products which are delivered at a particular address.

QUERY 6:List the product whose sale is the highest on a particular day.

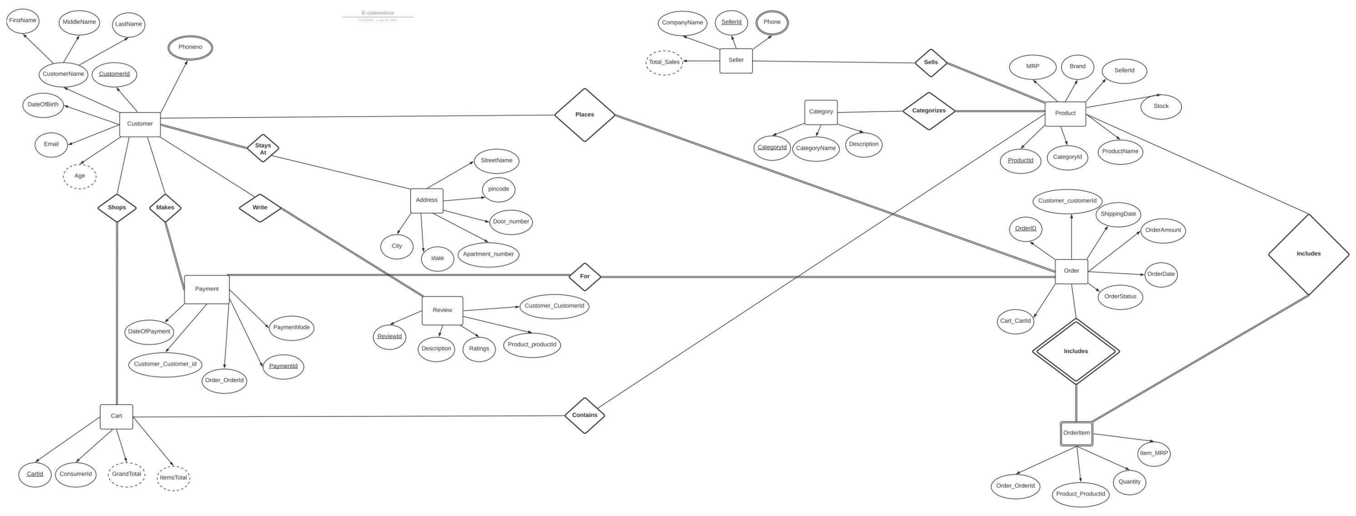
QUERY 7:List the category of product which has been sold the highest on a particular day.

QUERY 8:List the customers who bought products from a particular seller the most.

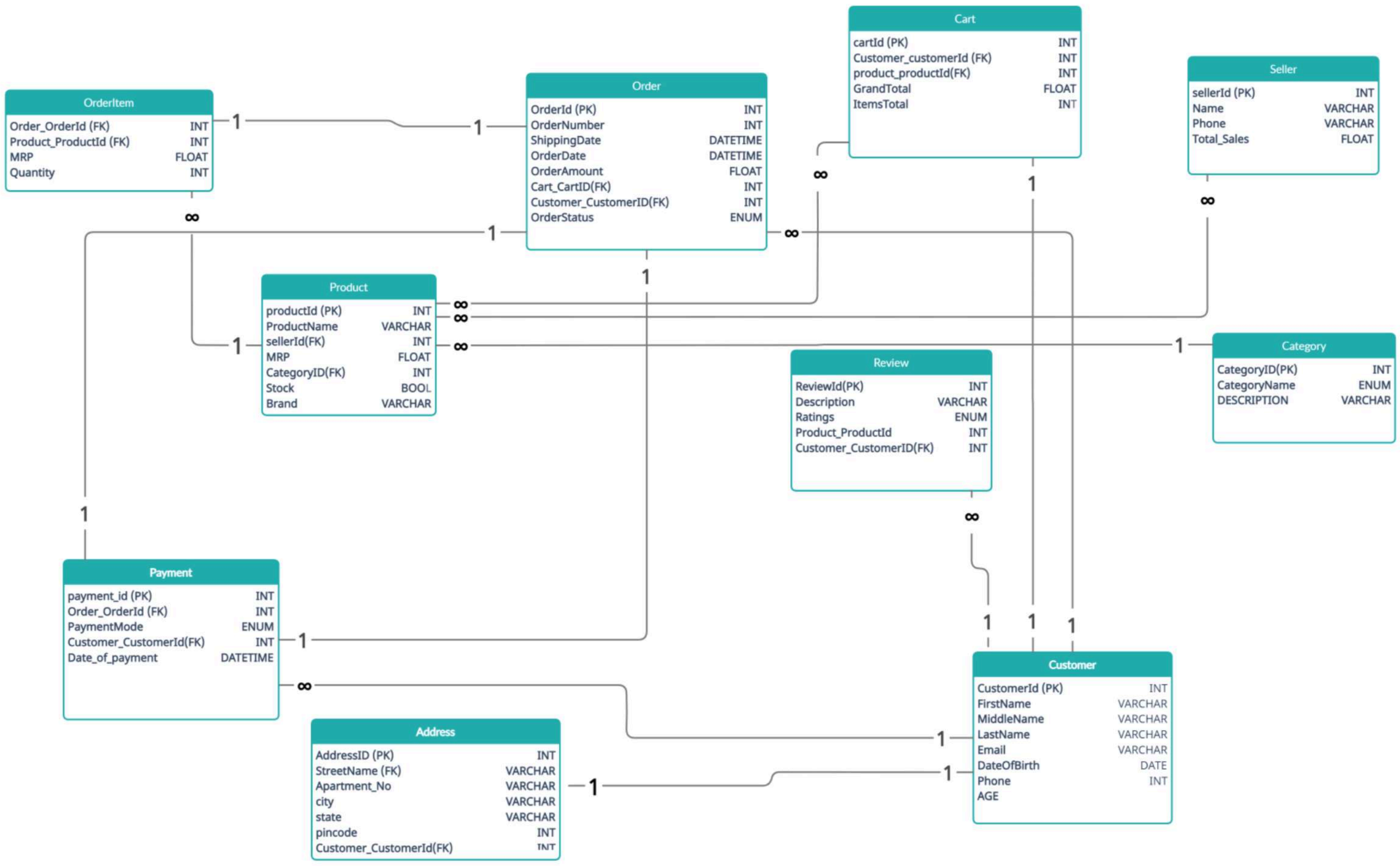
QUERY 9:List the most used payment mode on a particular day.

QUERY 10:List all addresses of customers whose total amount is greater than 5000.

QUERY 11:List the seller who has the highest stock of a particular product.



E-commerce
UTKARSH | July 28, 2021
FirstName
MiddleName
LastName
Phoneno
CustomerName
CustomerId
DateOfBirth
Customer
Email
Age
Shops
Makes
Write
Stays At
Address
StreetName
pincode
Door_number
City
state
Apartment_number
Payment
DateOfPayment
PaymenMode
Customer_Customer_id
Order_OrderId
PaymentId
Review
Customer_CustomerId
ReviewId
Description
Ratings
Product_productId
Cart
CartId
ConsumerId
GrandTotal
ItemsTotal
Places
For
Contains
CompanyName
SellerId
Phone
Total_Sales
Seller
Sells
Category
Categorizes
CategoryId
CategoryName
Description
MRP
Brand
SellerId
Product
Stock
ProductId
CategoryId
ProductName
Customer_customerId
OrderID
ShippingDate
OrderAmount
Order
OrderDate
OrderStatus
Cart_CartId
Includes
Includes
OrderItem
Item_MRP
Order_OrderId
Product_ProductId
Quantity


OrderItem
Order_OrderId (FK) INT
Product_ProductId (FK) INT
MRP FLOAT
Quantity INT
Order
OrderId (PK) INT
OrderNumber INT
ShippingDate DATETIME
OrderDate DATETIME
OrderAmount FLOAT
Cart_CartID(FK) INT
Customer_CustomerID(FK) INT
OrderStatus ENUM
Cart
cartId (PK) INT
Customer_customerId (FK) INT
product_productId(FK) INT
GrandTotal FLOAT
ItemsTotal INT
Seller
sellerId (PK) INT
Name VARCHAR
Phone VARCHAR
Total_Sales FLOAT
Product
productId (PK) INT
ProductName VARCHAR
sellerId(FK) INT
MRP FLOAT
CategoryID(FK) INT
Stock BOOL
Brand VARCHAR
Review
ReviewId(PK) INT
Description VARCHAR
Ratings ENUM
Product_ProductId INT
Customer_CustomerID(FK) INT
Category
CategoryID(PK) INT
CategoryName ENUM
DESCRIPTION VARCHAR
Payment
payment_id (PK) INT
Order_OrderId (FK) INT
PaymentMode ENUM
Customer_CustomerId(FK) INT
Date_of_payment DATETIME
Address
AddressID (PK) INT
StreetName (FK) VARCHAR
Apartment_No VARCHAR
city VARCHAR
state VARCHAR
pincode INT
Customer_CustomerId(FK) INT
Customer
CustomerId (PK) INT
FirstName VARCHAR
MiddleName VARCHAR
LastName VARCHAR
Email VARCHAR
DateOfBirth DATE
Phone INT
AGE
1
1
∞
1
∞
1
∞
∞
∞
1
∞
1
∞
1
1
∞
1
1
1
∞
1
1
1