DBMS Project - Group 99

Sarthak Kumar - 2020241
Arjun Mehra - 2020178

Scope of the Project

A similar to Big Bazaar online retail store will be implemented by us. Users will be able to look over and purchase items sold by Big Bazaar (Admin). Users will give us their data which the system will accommodate. Both user data (such as name, password, cart, etc.) and admin-uploaded data (such as categories, goods, pricing, stock, etc.) are stored by our system. The system will ensure that the product operates effectively, rapidly, and efficiently while adhering to protocols for data integrity, security, recovery, etc. This will guarantee that the product improves the store's performance and responsiveness. Through this, the entire system ought to function as intended.

For customers:

● Browsing products: Customers can browse products by category, view product details, and view product images.
● Searching products: Customers can search for products by keyword, filter search results by category, price, and other criteria.
● Shopping cart: Customers can add products to their shopping cart and view the contents of their cart at any time.
● Checkout: Customers can complete their purchase by providing shipping and billing information, selecting a shipping method, and submitting payment.
● Order tracking: Customers can view the status of their orders and track the delivery of their items.
● Customer account: Customers can create an account to save their shipping and billing information, view their order history, and access other account-specific features.
● Customer support: Customers can contact customer support for assistance with their orders or any other issues.

Stakeholders

1. Users: Users can choose the things they want to buy and anticipate receiving them according to the shipper's pace.

2. Shippers: The shippers can use the database to obtain information about the various orders placed by users and can then adjust their delivery procedures as necessary.

3. Employee: Employee resolves the queries of the customer.

Entity Relationship Diagram
Here is the link to the ER Diagram

Entities, Attributes & Schemas

1) Account Table

account_Id	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
Username	VARCHAR(50) NOT NULL UNIQUE
Password	VARCHAR(50) NOT NULL

2) Customer

Customer_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
customer_name	VARCHAR(50) NOT NULL
phone_no	VARCHAR(50) NOT NULL
customer_email	VARCHAR(50)

	NOT NULL
customer_address	VARCHAR(50) NOT NULL
DOB	VARCHAR(15) NOT NULL

3) Product

P_id	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
Pname	VARCHAR(50) NOT NULL
Brand	VARCHAR(50) NOT NULL
Stock	VARCHAR(13) NOT NULL
Price	VARCHAR(50) NOT NULL CHECK Price > 0
Offer	INT NOT NULL

4) Review

Rating	INT NOT NULL
Comments	VARCHAR(50) NOT NULL

5) Issue

Forum	VARCHAR(50) NOT NULL

Message	VARCHAR(50) NOT NULL
Query Status	VARCHAR(50) NOT NULL

6) Employee

Employee_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
Ename	VARCHAR(50) NOT NULL
Designation	VARCHAR(50) NOT NULL
Email	VARCHAR(50) NOT NULL
Phone_no	VARCHAR(50) NOT NULL

7) Supplier

S_ID	INT NOT NULL AUTO_INCREMENT PRIMARY KEY
Sname	VARCHAR(50) NOT NULL
Contact	VARCHAR(50) NOT NULL
Address	VARCHAR(50) NOT NULL

8) Category

Cid	INT NOT NULL AUTO_INCREMENT

	PRIMARY KEY
Category_Name	VARCHAR(11) NOT NULL
Category_info	VARCHAR(50) NOT NULL

9) Chooses

product_id	INT NOT NULL FOREIGN KEY REFERENCES PRODUCT
category_id	INT NOT NULL FOREIGN KEY REFERENCES CUSTOMER
quantity	INT NOT NULL CHECK Quantity > 0

10) resolve

Employee_ID	INT NOT NULL FOREIGN KEY REFERENCES EMPLOYEE
Customer_ID	INT NOT NULL FOREIGN KEY REFERENCES CUSTOMER
forum	VARCHAR(50) NOT NULL
f_status	VARCHAR(50) NOT NULL

Strong Entities

Customer(Name, Age, Address, EmailID, phone No., CustomerID)

AccountID(AccountID, Password)
Supplier(SID, S-contact, S_address, S_name, **OID**)
Order(OrderID, Payment Mode, Shipping address, Expected Delivery, Order total, **AccountID, PID, SID**)
Product(PID, Price, P_name, P_model, Offer, Stock, Process, **CID, AccountID, SID**)
Category(CID, C_name, C_info, **PID**)
Employee(EID, E_name, Designation, email, phone no.)

Multi-Valued Attribute

Order_Payment(OID)
Offer_Product(PID)

Weak Entities

Review(Rating, Comment)
Issues(Forum, Message, status)

Relational Schema

has(Aid, CustomerId)
has(Aid, Eid)
selects(customerId, categoryId)
chooses(customerId, Quantity, PID)
Ships(OID, SID)has(CustomerId, OrderId, PaymentId)
has(CustomerId, ProductId, rating)
has an(CustomerId, Forum, message, status)
resolve(EId, CustomerId, Forum, Status)

Ternary relationships

1. Has
2. chooses

Has is a ternary relationship because it connects customer, product and reviews

Since the customer does not need to know the product id, we have three views called userProductView that are used to display simply the product name, brand, and price to the customer. Similar to that, categoryUserView is used to shield the protected category from the customer's view. UserView is used to protect users' passwords by hiding it from the administrator's view while still displaying the users on our online retail store.

-- View products using customer privileges

```
Create VIEW userProductView AS
SELECT product_name, product_cost, brand_name From product;

 -- View categories from customer privileges
Create view categoryUserView AS
SELECT category_name, category_info From category;

 -- View users using admin privileges
Create VIEW protectedUserView AS
SELECT id, address, name, EmailID, PhoneNumber From user;
```