Online Shopping Database - Requirements Documentation

Overview

The Online Shopping Database is designed to facilitate the management of an online shopping platform. It encompasses various entities, such as customers, products, reviews, orders, payments, and shipments. The goal is to provide a robust and scalable database system that supports core e-commerce functionalities.

Entities

1. Customer:
   - Attributes: Customer_id, First_name, Last_name, Age, Email, Customer_password, Gender, Registration_date
   - Responsibilities: Store customer information for account management.

2. **Product:**
   - Attributes: Product_id, Product_name, Product_description, Price, Image_url, Quantity_in_stock, Category_id, Created_date
   - Responsibilities: Manage product details, pricing, and inventory.

3. **Review:**
   - Attributes: Review_id, Cust_id, Product_id, Rating, Review_description, Review_date
   - Responsibilities: Capture customer reviews and ratings for products.

4. **Cart:**
   - Attributes: Cart_id, Cust_id, Product_id, Quantity, Date_added
   - Responsibilities: Record products added to the customer's shopping cart.

5. Orders:
   - Attributes: Order_id, Cust_id, Shipment_id, Order_date, Order_total
   - Responsibilities: Track customer orders and associated details.

6. Order_item
   - Attributes: Order_item_id, Order_id, Product_id, Quantity, Item_total
   - Responsibilities: Store individual items within an order.

7. Payment:
   - Attributes: Payment_id, Order_id, Payment_method, Amount, Payment_date
   - Responsibilities: Record payment transactions related to orders.

8. Shipment_address:
   - Attributes: Shipment_Address_id, Shipment_country, Shipment_city, Postal_code
   - Responsibilities: Store shipment addresses for orders.

9. Shipment:
   - Attributes: Shipment_id, Shipment_method, Shipment_cost, Shipment_Address_id
   - Responsibilities: Manage shipment methods and associated costs.

10. Category:
    - Attributes: Category_id, Category_name
    - Responsibilities: Categorize products for better organization.

Key Features

1.User Management:
   - Register new customers.
   - Store customer information securely.
   - Enable customer login/logout.

2. Product Management:
   - Add new products with detailed information.
   - Update product details, including quantity in stock.
   - Categorize products into different categories.

3. Order Processing:
   - Allow customers to add products to their carts.
   - Generate and manage customer orders.
   - Facilitate payment transactions.

4. Review and Rating:
   - Enable customers to submit reviews and ratings for products.
   - Store and display product reviews.

5. Shipment:
   - Manage shipment methods and associated costs.
   - Track shipment status and delivery information.

6. Reportin:
   - Provide various reports, including order details, product categories, and customer information.
   - Support customizable and parameterized reports.

Security and Access Control

1. Implement role-based access control:
   - SalesTeam: Read, insert, update, delete permissions on the Customer table.
   - InventoryTeam: Read, update permissions on the Product table.

- CustomerSupportTeam: Read permissions on the Review table.
- Admin: Various permissions for managing reviews, logs, and system components.

2. Utilize views and procedures to simplify data retrieval and operations.

3. Implement triggers for auditing changes to the Review table.

Normalization

1. Ensure the database schema adheres to normalization principles:
- 1NF, 2NF, 3NF, etc.
- Minimize data redundancy and maintain data integrity.

Flexibility and Ease of Use

1. Provide parameterized procedures for report generation, ensuring flexibility in data retrieval.

2. Utilize views to simplify complex queries and present information in a user-friendly manner.

Future Enhancements

1. Implement a recommendation system based on customer reviews and preferences.

2. Enhance security measures, including encryption for sensitive data.

3. Incorporate additional features such as user feedback, wishlist, and promotions.

Conclusion

The Online Shopping Database is designed to provide a comprehensive and efficient platform for managing various aspects of an online shopping experience. It is scalable, secure, and flexible to accommodate future enhancements and evolving business needs.
Database Schema Documentation

Table: `tbladmin`

- **Columns:**
  - `id` (bigint): Unique identifier for each admin.
  - `added_date` (datetime): Date and time when the admin was added.
  - `email` (varchar(100)): Email address of the admin.
  - `password` (varchar(100)): Password for admin login.
  - `name` (varchar(100)): Name of the admin.

- **Primary Key:**

  - `id`

Table: `tblcart`

- **Columns:**
  - `id` (bigint): Unique identifier for each cart entry.
  - `discount_price` (varchar(200)): Discounted price (optional).
  - `quantity` (int): Quantity of the product in the cart.
  - `total_price` (varchar(200)): Total price for the items in the cart.
  - `customer_id` (bigint): ID of the customer associated with the cart.
  - `product_id` (bigint): ID of the product in the cart.
  - `mrp_price` (varchar(200)): Maximum Retail Price of the product.

- **Primary Key:**
  - `id`

Table: `tblcustomer`

- **Columns:**
  - `id` (int): Unique identifier for each customer.
  - `address` (varchar(255)): Address of the customer.
  - `added_date` (timestamp): Date and time when the customer was added.
  - `email` (varchar(100)): Email address of the customer.
  - `gender` (varchar(6)): Gender of the customer.
  - `name` (varchar(50)): Name of the customer.
  - `password` (varchar(60)): Password for customer login.
  - `phone` (varchar(200)): Phone number of the customer.
  - `pin_code` (varchar(255)): Postal code of the customer.

- **Primary Key:**
  - `id`

Table: `tblorders`

- **Columns:**
  - `id` (int): Unique identifier for each order.
  - `order_no` (int): Order number.
  - `customer_name` (varchar(200)): Name of the customer placing the order.

  - `mobile_number` (varchar(100)): Mobile number of the customer.
  - `email_id` (varchar(100)): Email address of the customer.
  - `address` (varchar(400)): Shipping address for the order.
  - `address_type` (varchar(100)): Type of address (e.g., home, work).
  - `pincode` (varchar(100)): Postal code of the shipping address.
  - `image` (varchar(200)): Image associated with the order.
  - `product_name` (varchar(400)): Name of the product in the order.
  - `quantity` (int): Quantity of the product in the order.
  - `product_price` (varchar(100)): Price of the product.
  - `product_selling_price` (varchar(100)): Selling price of the product.
  - `product_total_price` (varchar(100)): Total price for the product in the order.
  - `order_status` (varchar(100)): Status of the order.
  - `order_date` (timestamp): Date and time when the order was placed.
  - `payment_mode` (varchar(100)): Payment mode for the order.
  - `payment_id` (int): ID of the payment associated with the order.

- **Primary Key:**
  - `id`

Table: `tblproduct`

- **Columns:**
  - `id` (bigint): Unique identifier for each product.
  - `active` (varchar(100)): Status of the product (active/inactive).
  - `code` (varchar(5)): Code associated with the product.
  - `create_date` (timestamp): Date and time when the product was created.
  - `description` (varchar(255)): Description of the product.
  - `image` (varchar(100)): Image associated with the product.
  - `image_name` (varchar(400)): Name of the product image.
  - `name` (varchar(30)): Name of the product.
  - `price` (varchar(200)): Price of the product.
  - `mrp_price` (varchar(200)): Maximum Retail Price of the product.

- **Primary Key:**
  - `id`

Initial Data Insertion:

- An admin account is inserted into `tbladmin` with the following credentials:
  - Email: admin@gmail.com

- Password: admin
- Name: DataFlair

Note: The documentation provides an overview of the tables, their columns, primary keys, and initial data insertion for the specified database schema.