CSCI 6560: Database Projects

A new dot-com company has decided to launch a new e-commerce system. The company has hired you as a database specialist on contract. Your job is to design and develop a database system to support the online shopping business. After several interviews with all stakeholders, you found out the following tables should be created to store customer, product and order information.

Customer (UserID, Email, Password, Firstname, Lastname, Address, Phone)

CreditCard (Credit_Card_ID, Credit_Card_Number, Holder_Name, Expire_Date, CVC_Code, Billing_Address, OwnerID) where OwnerID refers to Customer.UserID.

Product(Product id, Name, Quantity, Description, Cost_Price, Sales_Price, Discount)
Where discount on the product is like 5%, 10% or 20% off. User needs to pay (1-Discount)*Sales_Price for the product.

Order(Order id, UserID, Order_Date, Total_Amount, Credit_Card_ID, Shipping_address, Status)
(UserID, Credit_Card_ID) is one foreign key, referenceing CreditCart(Credit_Card_ID, OwnerID)
Status must be a value from {placed, in preparation, ready to ship, shipped}
Total_Amount of money is the money the user needs to pay, excluding tax and shipping, for the order. It is a derived value by summing Quantity*PaidPrice of all items in the order.

OrderItem(Order id, Product id, PaidPrice, Quantity)
PaidPrice is calculated from Sales price and Discount of the product when the order is placed.

The database should have different types of users with different sets of permissions. If a permission is not specified explicitly below, then the users shouldn't be granted such permission.

- *Customer* with the following permissions:
  - can view information of all products excluding Cost_Price
  - can view their own information and last 4 digits of credit cards
  - can update their own information
  - can insert/remove a credit card
  - can only modify Holder_Name and Billing_Address of existing credit card
- *Customer service representative* with the following permissions:
  - can view information of all products excluding Cost_Price;
  - can view customer information and orders;
  - can remove an order item from a placed order only if the order status is "in preparation". If an order doesn't contain order items, the order should also be removed;

    - can update the quantity of an order item from a placed order only if the order status is "in preparation".
    - can insert a new order item to a placed order only if the order status is "in preparation".
- *Sales* with the following permissions:
    - can insert/update product table,
    - cannot modify Cost_Price, Sales_Price, and Discount attributes.
- *Sales* Manager with the following permissions
    - can insert/update product table,
    - can update Cost_Price, Sales_Price, and Discount attributes of product.
    - can remove a product from database if its quantity is 0.
    - no permission on all other tables
- *Order Processors* with the following permissions
    - can view Order excluding Total_Amount, Credit_Card_ID attributes;
    - can view OrderItem excluding PaidPrice;
    - only modify Status attribute of Order table.

You also figure out the following constraints and requirements.

- OrderItem.PaidPrice should always be greater or equal to the cost price of the product. The company will never lose money by selling a product.
- OrderItem.PaidPrice and Order.Total_Amount should always be calculated automatically and consistent.
- Start charging the credit card whenever the order status is changed to [shipped]. Charge can be completed by printing a message of the following format:
  Credit Card ending with 1234 is charged $111.11 for the order with order id 1111111.
- When an order is placed, deduct OrderItem.Quantity from Product.Quantity for each order item.
- When an order item is removed, add OrderItem.Quantity back to Product.Quantity.
- Password, credit card number, and Product.Cost_Price must be encrypted.
- No one can modify user id, credit card id, order id, product id.

The audit requirement is summarized below:

- Track changes made to the product table, including information of the user who makes the change and data before and after the change.
- Track changes made to Order and OrderItem tables, including information of the user who makes the change and data before and after the change.
- Track any permission changes by GRANT/REVOKE/DENY statements.
- Audit successful/failed login and logout events.
    - Provide SQL statements to retrieve all failed logins for a given user
    - Provide SQL statements to retrieve all session information for a given user. For each session, list begin timestamp (from login event) and end timestamp (from logout event).

Your assignment is to design and implement the database satisfying all security and functionality requirements. You may add new table or attributes if necessary. Please submit the following:

1. File Schema.sql that contains SQL statements to create all tables, and/or views, and insert test data. (50 points)
2. File Objects.sql that contains SQL stored procedures, functions, triggers and other statements to implement all constraints. (150 points)
3. File Auditing.sql that contains SQL Statements to track all changes and audit sessions. (100 points)
4. File Encrypt.sql that contains SQL statements to encrypt the data and statements to retrieve Product table with clear text on Cost_Price. (50 points)
5. File Permission.sql that contains SQL statements to create users/logins/roles and statements to grant/deny/revoke permissions. (50 points)
6. File Testing.sql that contains test cases to demonstrate the satisfaction of all constraints and requirement as well as permission requirements. For each test case, please specify expected result and explain why the expected result will occur. (150 points)

How to submit: Zip all your files and email it to Zhijiang.Dong@mtsu.edu with subject: Database Project submission.