Case Study 1: Shop Here

The entities for the Shop Here database management system are:

- Employee
- ItemDetails
- OrderDetails
- SupplierDetails
- ProductCategory

The attributes of the entities listed above are as follows:

- **Employees**: Employee ID, First Name, Last Name, City, Phone
- **ItemDetails**: Item ID, Item Name, Item Description, Unit Price, Quantity In Hand, Reorder Level, Reorder Quantity, Category ID, Supplier ID
- **OrderDetails**: Purchase Order ID, Employee ID, Order Date, Receiving Date, Item ID, Quantity Ordered, Quantity Received, Unit Price, Ship Method, Order Status
- **SupplierDetails**: Supplier ID, First Name, Last Name, Address, Phone, Country
- **ProductCategory**: Category ID, Category Name, Category Description

To create a computerized transaction system for Shop Here using SQL Server 2005, the project team will have to perform the following tasks:

1. Create a database, **ShopHere**.
2. Create the tables as per the relationship diagram, ensuring minimum disk space utilization.
3. Perform validations on the tables as per the following guidelines:

   Table: Items.ItemDetails

   - ItemID must be auto generated.
   - ItemName should not be left blank.
   - ItemDescription should not be left blank.
   - QuantityInHand should be greater than 0. The record should not be inserted or modified manually if QuantityInHand is 0.
   - UnitPrice should be greater than 0.
   - ReorderQuantity should be greater than 0.
   - ReorderLevel should be greater than 0.
   - CategoryID should be the foreign key from the ProductCategory table.
   - SupplierID should be the foreign key from the SupplierDetails table.

Table: Items.ProductCategory

- CategoryID must be auto generated.
- CategoryName and CategoryDescription should not be left blank.
- CategoryName should be Household, Sports, Accessories, or Clothing.

Table: Supplier.SupplierDetails

- SupplierID must be auto generated.
- FirstName, LastName, Address, Phone, and Country should not be left blank.
- Phone must be in the following format:
  '[0-9][0-9]-[0-9] [0-9][0-9]-[0-9][0-9][0-9][0-9]-[0-9][0-9] [0-9]-[0-9] [0-9] [0-9]'
  Example: 11-111-1111-111-111

Table: HumanResources.Employee

- EmployeeID must be auto generated.
- Employee FirstName, LastName, City, and Phone should not be left blank.
- Phone must be in the following format:
  '[0-9][0-9]-[0-9] [0-9][0-9]-[0-9][0-9][0-9][0-9]-[0-9][0-9] [0-9]-[0-9] [0-9] [0-9]'
  Example: 11-111-1111-111-111

Table: Transactions.OrderDetails

- PurchaseOrderID must be auto generated.
- OrderDate should not be greater than the current date.
- If the order date is not entered, the current date should be taken as the default date.
- QuantityOrdered, QuantityReceived, and UnitPrice should be greater than 0.
- QuantityReceived should allow NULL.
- QuantityReceived cannot be greater than QuantityOrdered.
- QuantityReceived should be added to QuantityInHand in the Items table.
- When a record is inserted into the table, QuantityInHand in the Items table should be updated automatically.
- OrderStatus must be any of the following values: 'InTransit', 'Received', or 'Cancelled'.
- ReceivingDate should allow NULL and should be greater than OrderDate.

4. Create appropriate relationships between the tables.
5. Store the order details in a text file on a day-to-day basis. Make use of the required tools to perform the data transfer.
6. Create the appropriate indexes to speed up the execution of the following tasks:
   - Extract the order details for all the purchase orders in the current month.
   - Extract the details of all the orders placed more than two years back.
   - Extract the details of the top five suppliers to whom the maximum number of orders have been placed in the current month.
7. Simplify the following tasks:
   - Calculation of the total cost for a particular order
   - Calculation of the total of all the orders placed by a particular employee in a particular month
8. Implement an appropriate security policy on the database. For this, create logins named George, John, and Sara. George is the database administrator and John and Sara are database developers.
9. Back up the database daily and store it in the C drive.
10. Store crucial data in encrypted format.
11. Ensure that an alert is sent to George whenever the size of the temporary space in the database server falls below 20 MB.