**Case Study: Online Shopping System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for an Online Shopping System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage products, customers, and orders.

**Requirements:**

1. **Product Management**:
   * Implement the functionality to add, update, and delete product records.
   * Each product should have attributes such as product\_id, name, category, price, and stock\_quantity.
2. **Customer Management**:
   * Implement the functionality to manage customer information.
   * Each customer should have attributes such as customer\_id, name, email, and address.
3. **Order Management**:
   * Implement the functionality to handle customer orders.
   * Each order should have attributes such as order\_id, customer\_id, product\_id, order\_date, and quantity.

**Business Functionalities:**

1. **Add/Update/Delete Products**:
   * Create a class Product with attributes product\_id, name, category, price, and stock\_quantity.
   * Implement methods to add a new product, update existing product details, and delete a product from the system.
2. **Manage Customers**:
   * Create a class Customer with attributes customer\_id, name, email, and address.
   * Implement methods to add a new customer, update customer details, and delete a customer.
3. **Manage Orders**:
   * Create a class Order with attributes order\_id, customer\_id, product\_id, order\_date, and quantity.
   * Implement methods to add a new order, update order details, and cancel an order.

**Section 2: MySQL Database Management**

Design a MySQL database schema to support the Online Shopping System and provide problem statements for querying the database.

**Table Structures:**

1. **Products Table**:
   * product\_id: INT, Primary Key
   * name: VARCHAR(100)
   * category: VARCHAR(50)
   * price: DECIMAL(10, 2)
   * stock\_quantity: INT
2. **Customers Table**:
   * customer\_id: INT, Primary Key
   * name: VARCHAR(100)
   * email: VARCHAR(100)
   * address: VARCHAR(255)
3. **Orders Table**:
   * order\_id: INT, Primary Key
   * customer\_id: INT, Foreign Key References Customers(customer\_id)
   * product\_id: INT, Foreign Key References Products(product\_id)
   * order\_date: DATE
   * quantity: INT

**Problem Statements:**

1. Write a query to find the total sales revenue generated from all orders.
2. Write a query to find the names and email addresses of customers who have placed more than three orders.
3. Write a query to find the product categories that have sold out.
4. Write a query to find the customers who have ordered products from more than two different categories.
5. Write a query to find the details of orders placed in the last 30 days.