**Case Study: Restaurant Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for a Restaurant Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage menu items, customers, and orders.

**Requirements:**

1. **Menu Management**:
   * Implement the functionality to add, update, and delete menu item records.
   * Each menu item should have attributes such as item\_id, name, category, price, and availability.
2. **Customer Management**:
   * Implement the functionality to manage customer information.
   * Each customer should have attributes such as customer\_id, name, contact\_number, and email.
3. **Order Management**:
   * Implement the functionality to handle customer orders.
   * Each order should have attributes such as order\_id, customer\_id, item\_id, order\_date, and quantity.

**Business Functionalities:**

1. **Add/Update/Delete Menu Items**:
   * Create a class MenuItem with attributes item\_id, name, category, price, and availability.
   * Implement methods to add a new menu item, update existing menu item details, and delete a menu item from the system.
2. **Manage Customers**:
   * Create a class Customer with attributes customer\_id, name, contact\_number, and email.
   * 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, item\_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 Restaurant Management System and provide problem statements for querying the database.

**Table Structures:**

1. **MenuItems Table**:
   * item\_id: INT, Primary Key
   * name: VARCHAR(100)
   * category: VARCHAR(50)
   * price: DECIMAL(10, 2)
   * availability: BOOLEAN
2. **Customers Table**:
   * customer\_id: INT, Primary Key
   * name: VARCHAR(100)
   * contact\_number: VARCHAR(15)
   * email: VARCHAR(100)
3. **Orders Table**:
   * order\_id: INT, Primary Key
   * customer\_id: INT, Foreign Key References Customers(customer\_id)
   * item\_id: INT, Foreign Key References MenuItems(item\_id)
   * order\_date: DATE
   * quantity: INT

**Problem Statements:**

1. Write a query to find the total revenue generated from all orders.
2. Write a query to find the names and contact numbers of customers who have placed more than five orders.
3. Write a query to find the menu items that are currently unavailable.
4. Write a query to find the customers who have ordered items from more than three different categories.
5. Write a query to find the details of orders placed in the last 7 days.