**Case Study: Retail Sales Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for a Retail Sales Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage sales records and customer information.

**Requirements:**

1. **Customer Management**:
   * Implement the functionality to add, update, and delete customer records.
   * Each customer should have attributes such as customer\_id, name, contact\_info, and loyalty\_points.
2. **Sales Transactions**:
   * Implement the functionality to record sales transactions.
   * Each transaction should have attributes such as transaction\_id, customer\_id, product\_name, quantity\_sold, and sale\_amount.
3. **Reporting**:
   * Implement the functionality to generate a report of top customers based on loyalty points (loyalty\_points > 100).

**Business Functionalities:**

1. **Add/Update/Delete Customers**:
   * Create a class Customer with attributes customer\_id, name, contact\_info, and loyalty\_points.
   * Implement methods to add a new customer, update existing customer details, and delete a customer from the system.
2. **Record Sales Transactions**:
   * Create a class Transaction with attributes transaction\_id, customer\_id, product\_name, quantity\_sold, and sale\_amount.
   * Implement methods to record a new sales transaction.
3. **Top Customers Report**:
   * Implement a method to generate a list of customers with loyalty points greater than 100.

**Section 2: MySQL Database Management**

Design a MySQL database schema to support the Retail Sales Management System and provide solutions for the problem statements.

**Table Structures:**

1. **Customers Table**:
   * customer\_id: INT, Primary Key
   * name: VARCHAR(100)
   * contact\_info: VARCHAR(100)
   * loyalty\_points: INT
2. **Transactions Table**:
   * transaction\_id: INT, Primary Key
   * customer\_id: INT, Foreign Key References Customers(customer\_id)
   * product\_name: VARCHAR(100)
   * quantity\_sold: INT
   * sale\_amount: DECIMAL(10, 2)
   * transaction\_date: DATE

**Problem Statements:**

1. Write a query to find the total sales amount for each customer.
2. Write a query to find the names of customers and their total sales amounts.
3. Write a query to find the names of customers who have never made a purchase.
4. Write a query to find the products that have been sold more than 50 times.
5. Write a query to find the customer names and their loyalty points for those who have made more than 10 transactions.