**Case Study: Vehicle Rental Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for a Vehicle Rental Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage vehicle inventory and rental transactions.

**Requirements:**

1. **Vehicle Management**:
   * Implement the functionality to add, update, and delete vehicle records.
   * Each vehicle should have attributes such as vehicle\_id, make, model, year, rental\_rate, and available.
2. **Rental Transactions**:
   * Implement the functionality to process rental transactions.
   * Each rental should have attributes such as rental\_id, customer\_name, vehicle\_id, rental\_start\_date, and rental\_end\_date.
3. **Reporting**:
   * Implement the functionality to generate a report of vehicles currently rented out.

**Business Functionalities:**

1. **Add/Update/Delete Vehicles**:
   * Create a class Vehicle with attributes vehicle\_id, make, model, year, rental\_rate, and available.
   * Implement methods to add a new vehicle, update existing vehicle details, and delete a vehicle from the system.
2. **Process Rental Transactions**:
   * Create a class Rental with attributes rental\_id, customer\_name, vehicle\_id, rental\_start\_date, and rental\_end\_date.
   * Implement methods to process a new rental transaction and update the availability of the vehicle.
3. **Rented Vehicles Report**:
   * Implement a method to generate a list of vehicles that are currently rented out.

**Section 2: MySQL Database Management**

Design a MySQL database schema to support the Vehicle Rental Management System and provide problem statements for querying the database.

**Table Structures:**

1. **Vehicles Table**:
   * vehicle\_id: INT, Primary Key
   * make: VARCHAR(50)
   * model: VARCHAR(50)
   * year: INT
   * rental\_rate: DECIMAL(10, 2)
   * available: BOOLEAN
2. **Customers Table**:
   * customer\_id: INT, Primary Key
   * name: VARCHAR(100)
   * contact\_info: VARCHAR(100)
3. **Rentals Table**:
   * rental\_id: INT, Primary Key
   * customer\_id: INT, Foreign Key References Customers(customer\_id)
   * vehicle\_id: INT, Foreign Key References Vehicles(vehicle\_id)
   * rental\_start\_date: DATE
   * rental\_end\_date: DATE

**Problem Statements:**

1. Write a query to find the total rental income for each vehicle.
2. Write a query to find the names of customers and the vehicles they have rented.
3. Write a query to find the vehicles that have never been rented.
4. Write a query to find the customers who have rented more than 3 different vehicles.
5. Write a query to find the vehicle makes and models with less than 5 available units for rent.