**Case Study: Airline Reservation Management System**

**Section 1: Python Standalone Console Application**

Design and implement a standalone console application for an Airline Reservation Management System using Python. The application should utilize collections, object-oriented programming (OOP), and exception handling to manage flight schedules and passenger reservations.

**Requirements:**

1. **Flight Schedule Management**:
   * Implement the functionality to add, update, and delete flight records.
   * Each flight should have attributes such as flight\_id, airline, destination, departure\_time, and available\_seats.
2. **Passenger Reservation**:
   * Implement the functionality to manage passenger reservations.
   * Each reservation should have attributes such as reservation\_id, passenger\_name, flight\_id, seat\_number, and reservation\_date.
3. **Reporting**:
   * Implement the functionality to generate a report of all passengers booked on a specific flight.

**Business Functionalities:**

1. **Add/Update/Delete Flights**:
   * Create a class Flight with attributes flight\_id, airline, destination, departure\_time, and available\_seats.
   * Implement methods to add a new flight, update existing flight details, and delete a flight from the system.
2. **Manage Reservations**:
   * Create a class Reservation with attributes reservation\_id, passenger\_name, flight\_id, seat\_number, and reservation\_date.
   * Implement methods to add a new reservation, update reservation details, and cancel a reservation.
3. **Passengers on Flight Report**:
   * Implement a method to generate a list of passengers booked on a specific flight using flight\_id.

**Section 2: MySQL Database Management**

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

**Table Structures:**

1. **Flights Table**:
   * flight\_id: INT, Primary Key
   * airline: VARCHAR(100)
   * destination: VARCHAR(100)
   * departure\_time: DATETIME
   * available\_seats: INT
2. **Passengers Table**:
   * passenger\_id: INT, Primary Key
   * name: VARCHAR(100)
   * contact\_info: VARCHAR(100)
3. **Reservations Table**:
   * reservation\_id: INT, Primary Key
   * passenger\_id: INT, Foreign Key References Passengers(passenger\_id)
   * flight\_id: INT, Foreign Key References Flights(flight\_id)
   * seat\_number: VARCHAR(10)
   * reservation\_date: DATE

**Problem Statements:**

1. Write a query to find the total number of seats booked on each flight.
2. Write a query to find the names of passengers and the flights they are booked on.
3. Write a query to find the flights that have never been booked by any passenger.
4. Write a query to find the passengers who have made reservations for more than 3 flights.
5. Write a query to find the details of flights with less than 5 available seats.