**Case Study: Flight Reservation System**

**Problem Statement:**

Design and implement a Flight Reservation System using Oracle SQL and PL/SQL. The system will be used to manage flight reservations, customer information, and flight availability for an airline. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle reservation management, customer registration, and flight availability tracking.

**Requirements:**

1. **Reservation Management**:
   * Implement the functionality to add, update, delete, and search for reservations.
   * Ensure that each reservation has attributes such as RESERVATION\_ID, CUSTOMER\_ID, FLIGHT\_ID, DATE, SEAT\_NUMBER, and STATUS.
2. **Customer Registration**:
   * Implement the functionality to register new customers.
   * Ensure that each customer record has attributes such as CUSTOMER\_ID, FIRST\_NAME, LAST\_NAME, EMAIL, PHONE\_NUMBER, and ADDRESS.
3. **Flight Availability Tracking**:
   * Implement the functionality to track flight availability.
   * Ensure that each flight has attributes such as FLIGHT\_ID, DEPARTURE\_CITY, ARRIVAL\_CITY, DEPARTURE\_DATE, ARRIVAL\_DATE, SEATS\_AVAILABLE, and PRICE.

**Tasks:**

1. **Design the Database Schema**:
   * Create the Reservations, Customers, and Flights tables with the appropriate fields and constraints.
   * Define primary keys and foreign keys to maintain data integrity.
2. **Populate the Database with Sample Data**:
   * Insert sample records into the Reservations, Customers, and Flights tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to handle reservation management. The procedure should insert, update, and delete reservation records.
   * Create a procedure to manage customer registration. The procedure should insert new customer records.
   * Create a procedure to track flight availability. The procedure should update seat availability based on reservations.

**Expected Outcomes:**

1. **Reservations Table**:
   * Contains all information about the reservations made by customers.
2. **Customers Table**:
   * Stores customer information for the airline.
3. **Flights Table**:
   * Tracks the flight availability and details for the airline.
4. **PL/SQL Procedures**:
   * Efficiently manage reservations, customer registration, and flight availability, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the Reservations, Customers, and Flights tables.
2. SQL scripts to insert sample data into the tables.
3. PL/SQL scripts for the procedures to handle reservation management, customer registration, and flight availability tracking.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **Reservations Table**:
   * **RESERVATION\_ID**: Number, Primary Key
   * **CUSTOMER\_ID**: Number, Foreign Key References Customers(CUSTOMER\_ID)
   * **FLIGHT\_ID**: Number, Foreign Key References Flights(FLIGHT\_ID)
   * **DATE**: Date
   * **SEAT\_NUMBER**: Varchar2(10)
   * **STATUS**: Varchar2(50)
2. **Customers Table**:
   * **CUSTOMER\_ID**: Number, Primary Key
   * **FIRST\_NAME**: Varchar2(50)
   * **LAST\_NAME**: Varchar2(50)
   * **EMAIL**: Varchar2(100)
   * **PHONE\_NUMBER**: Varchar2(15)
   * **ADDRESS**: Varchar2(255)
3. **Flights Table**:
   * **FLIGHT\_ID**: Number, Primary Key
   * **DEPARTURE\_CITY**: Varchar2(50)
   * **ARRIVAL\_CITY**: Varchar2(50)
   * **DEPARTURE\_DATE**: Date
   * **ARRIVAL\_DATE**: Date
   * **SEATS\_AVAILABLE**: Number
   * **PRICE**: Number

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the Reservations, Customers, and Flights tables.
* **Develop**: Write PL/SQL procedures for handling reservation management, customer registration, and flight availability tracking.
* **Test**: Test the procedures with various scenarios (e.g., managing reservations, registering customers, tracking flight availability, ensuring proper updates).