**Case Study: Medical Record Management System**

**Problem Statement:**

Design and implement a Hospital Management System using Oracle SQL and PL/SQL. The system will be used to manage patients, doctors, appointments, and medical records. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle appointment scheduling, patient admissions, and generating medical reports.

**Requirements:**

1. **Patient Management**:
   * Implement the functionality to add, update, delete, and search for patients.
   * Ensure that each patient has attributes such as PATIENT\_ID, FIRST\_NAME, LAST\_NAME, DOB, GENDER, ADDRESS, and PHONE\_NUMBER.
2. **Doctor Management**:
   * Implement the functionality to add, update, delete, and search for doctors.
   * Ensure that each doctor has attributes such as DOCTOR\_ID, FIRST\_NAME, LAST\_NAME, SPECIALIZATION, PHONE\_NUMBER, and EMAIL.
3. **Appointment Management**:
   * Implement the functionality to schedule, reschedule, and cancel appointments.
   * Ensure that each appointment has attributes such as APPOINTMENT\_ID, PATIENT\_ID, DOCTOR\_ID, APPOINTMENT\_DATE, APPOINTMENT\_TIME, and STATUS.
4. **Medical Record Management**:
   * Implement the functionality to create, update, and view medical records for patients.
   * Ensure that each medical record has attributes such as RECORD\_ID, PATIENT\_ID, DOCTOR\_ID, DATE\_OF\_VISIT, SYMPTOMS, DIAGNOSIS, TREATMENT, and PRESCRIPTION.

**Tasks:**

1. **Design the Database Schema**:
   * Create the Patients, Doctors, Appointments, and Medical\_Records 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 Patients, Doctors, Appointments, and Medical\_Records tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to handle appointment scheduling. The procedure should check doctor availability, insert a new appointment record, and update the appointment status.
   * Create a procedure to handle patient admission. The procedure should insert a new patient record and update the patient's medical history.
   * Create a procedure to generate medical reports, including details such as patient information, doctor information, visit date, symptoms, diagnosis, treatment, and prescription.

**Expected Outcomes:**

1. **Patients Table**:
   * Contains all information about the patients admitted to the hospital.
2. **Doctors Table**:
   * Contains details of all doctors working at the hospital.
3. **Appointments Table**:
   * Tracks the appointment schedule, including appointment details and status.
4. **Medical\_Records Table**:
   * Contains the medical records of all patients, including visit details, symptoms, diagnosis, treatment, and prescription.
5. **PL/SQL Procedures**:
   * Efficiently manage appointment scheduling, patient admissions, and generating medical reports, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the Patients, Doctors, Appointments, and Medical\_Records tables.
2. SQL scripts to insert sample data into the tables.
3. PL/SQL scripts for the procedures to schedule appointments, admit patients, and generate medical reports.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **Patients Table**:
   * **PATIENT\_ID**: Number, Primary Key
   * **FIRST\_NAME**: Varchar2(50)
   * **LAST\_NAME**: Varchar2(50)
   * **DOB**: Date
   * **GENDER**: Varchar2(10)
   * **ADDRESS**: Varchar2(255)
   * **PHONE\_NUMBER**: Varchar2(15)
2. **Doctors Table**:
   * **DOCTOR\_ID**: Number, Primary Key
   * **FIRST\_NAME**: Varchar2(50)
   * **LAST\_NAME**: Varchar2(50)
   * **SPECIALIZATION**: Varchar2(100)
   * **PHONE\_NUMBER**: Varchar2(15)
   * **EMAIL**: Varchar2(100)
3. **Appointments Table**:
   * **APPOINTMENT\_ID**: Number, Primary Key
   * **PATIENT\_ID**: Number, Foreign Key References Patients(PATIENT\_ID)
   * **DOCTOR\_ID**: Number, Foreign Key References Doctors(DOCTOR\_ID)
   * **APPOINTMENT\_DATE**: Date
   * **APPOINTMENT\_TIME**: Varchar2(10)
   * **STATUS**: Varchar2(20)
4. **Medical\_Records Table**:
   * **RECORD\_ID**: Number, Primary Key
   * **PATIENT\_ID**: Number, Foreign Key References Patients(PATIENT\_ID)
   * **DOCTOR\_ID**: Number, Foreign Key References Doctors(DOCTOR\_ID)
   * **DATE\_OF\_VISIT**: Date
   * **SYMPTOMS**: Varchar2(255)
   * **DIAGNOSIS**: Varchar2(255)
   * **TREATMENT**: Varchar2(255)
   * **PRESCRIPTION**: Varchar2(255)

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the tables.
* **Develop**: Write PL/SQL procedures for scheduling appointments, admitting patients, and generating medical reports.
* **Test**: Test the procedures with various scenarios (e.g., scheduling an appointment, admitting a patient, generating reports, ensuring proper updates).