**Case Study: Healthcare Management System**

**Problem Statement:**

Design and implement a Healthcare Management System using Oracle SQL and PL/SQL. The system will be used to manage patient records, doctor information, appointments, and medical histories for a healthcare facility. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle patient management, appointment scheduling, and medical record keeping.

**Requirements:**

1. **Patient Management**:
   * Implement the functionality to add, update, delete, and search for patient records.
   * Ensure that each patient record has attributes such as PATIENT\_ID, FIRST\_NAME, LAST\_NAME, DATE\_OF\_BIRTH, GENDER, PHONE\_NUMBER, and EMAIL.
2. **Appointment Scheduling**:
   * Implement the functionality to schedule, update, cancel, and search for appointments.
   * Ensure that each appointment has attributes such as APPOINTMENT\_ID, PATIENT\_ID, DOCTOR\_ID, APPOINTMENT\_DATE, APPOINTMENT\_TIME, and STATUS.
3. **Medical Record Keeping**:
   * Implement the functionality to record and update medical histories for patients.
   * Ensure that each medical record has attributes such as RECORD\_ID, PATIENT\_ID, DOCTOR\_ID, VISIT\_DATE, DIAGNOSIS, PRESCRIPTION, and NOTES.
4. **Doctor Information**:
   * Implement the functionality to add, update, delete, and search for doctor information.
   * Ensure that each doctor record has attributes such as DOCTOR\_ID, FIRST\_NAME, LAST\_NAME, SPECIALTY, PHONE\_NUMBER, and EMAIL.

**Tasks:**

1. **Design the Database Schema**:
   * Create the Patients, Appointments, MedicalRecords, and Doctors 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, and Appointments tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to handle patient management. The procedure should insert, update, and delete patient records.
   * Create a procedure to manage appointment scheduling. The procedure should insert, update, cancel, and search for appointments.
   * Create a procedure to manage medical record keeping. The procedure should insert, update, and search for medical records.

**Expected Outcomes:**

1. **Patients Table**:
   * Contains all information about the patients registered with the healthcare facility.
2. **Appointments Table**:
   * Tracks the appointments scheduled for patients with doctors.
3. **MedicalRecords Table**:
   * Stores the medical histories and records for patients.
4. **Doctors Table**:
   * Contains information about the doctors working at the healthcare facility.
5. **PL/SQL Procedures**:
   * Efficiently manage patient records, appointment scheduling, and medical record keeping, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the Patients, Appointments, MedicalRecords, and Doctors tables.
2. SQL scripts to insert sample data into the tables.
3. PL/SQL scripts for the procedures to handle patient management, appointment scheduling, and medical record keeping.
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)
   * **DATE\_OF\_BIRTH**: Date
   * **GENDER**: Varchar2(10)
   * **PHONE\_NUMBER**: Varchar2(15)
   * **EMAIL**: Varchar2(100)
2. **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)
3. **MedicalRecords Table**:
   * **RECORD\_ID**: Number, Primary Key
   * **PATIENT\_ID**: Number, Foreign Key References Patients(PATIENT\_ID)
   * **DOCTOR\_ID**: Number, Foreign Key References Doctors(DOCTOR\_ID)
   * **VISIT\_DATE**: Date
   * **DIAGNOSIS**: Clob
   * **PRESCRIPTION**: Clob
   * **NOTES**: Clob
4. **Doctors Table**:
   * **DOCTOR\_ID**: Number, Primary Key
   * **FIRST\_NAME**: Varchar2(50)
   * **LAST\_NAME**: Varchar2(50)
   * **SPECIALTY**: Varchar2(50)
   * **PHONE\_NUMBER**: Varchar2(15)
   * **EMAIL**: Varchar2(100)

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the Patients, Doctors, and Appointments tables.
* **Develop**: Write PL/SQL procedures for handling patient management, appointment scheduling, and medical record keeping.
* **Test**: Test the procedures with various scenarios (e.g., managing patients, scheduling appointments, recording medical histories, ensuring proper updates).