**Case Study: Insurance Policy Management System**

**Problem Statement:**

Design and implement an Insurance Policy Management System using Oracle SQL and PL/SQL. The system will be used to manage insurance policies, process policy requests, and generate policy reports. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle policy management, request processing, and report generation.

**Requirements:**

1. **Policy Management**:
   * Implement the functionality to add, update, delete, and search for insurance policies.
   * Ensure that each policy has attributes such as POLICY\_ID, CUSTOMER\_ID, POLICY\_TYPE, START\_DATE, END\_DATE, and STATUS.
2. **Policy Processing**:
   * Implement the functionality to process policy requests and update their status.
   * Ensure that each processing record has attributes such as PROCESSING\_ID, POLICY\_ID, PROCESSING\_DATE, DECISION, and REMARKS.
3. **Report Generation**:
   * Implement the functionality to generate policy reports.
   * Reports should include details such as total policies issued, active policies, expired policies, and pending requests.

**Tasks:**

1. **Design the Database Schema**:
   * Create the InsurancePolicies, PolicyProcessings, and PolicyReports 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 InsurancePolicies and PolicyProcessings tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to handle insurance policy management. The procedure should insert, update, and delete policy records.
   * Create a procedure to process policy requests. The procedure should insert processing records and update policy status.
   * Create a procedure to generate policy reports. The procedure should aggregate data to provide insights into policy status.

**Expected Outcomes:**

1. **InsurancePolicies Table**:
   * Contains all information about the insurance policies issued.
2. **PolicyProcessings Table**:
   * Tracks the processing of each policy request.
3. **PolicyReports Table**:
   * Stores the generated policy reports.
4. **PL/SQL Procedures**:
   * Efficiently manage insurance policies, process policy requests, and generate reports, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the InsurancePolicies, PolicyProcessings, and PolicyReports tables.
2. SQL scripts to insert sample data into the InsurancePolicies and PolicyProcessings tables.
3. PL/SQL scripts for the procedures to handle insurance policy management, policy processing, and generate policy reports.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **InsurancePolicies Table**:
   * **POLICY\_ID**: Number, Primary Key
   * **CUSTOMER\_ID**: Number
   * **POLICY\_TYPE**: Varchar2(50)
   * **START\_DATE**: Date
   * **END\_DATE**: Date
   * **STATUS**: Varchar2(50)
2. **PolicyProcessings Table**:
   * **PROCESSING\_ID**: Number, Primary Key
   * **POLICY\_ID**: Number, Foreign Key References InsurancePolicies(POLICY\_ID)
   * **PROCESSING\_DATE**: Date
   * **DECISION**: Varchar2(50)
   * **REMARKS**: Varchar2(255)
3. **PolicyReports Table**:
   * **REPORT\_ID**: Number, Primary Key
   * **REPORT\_DATE**: Date
   * **TOTAL\_POLICIES**: Number
   * **ACTIVE\_POLICIES**: Number
   * **EXPIRED\_POLICIES**: Number
   * **PENDING\_REQUESTS**: Number

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the InsurancePolicies and PolicyProcessings tables.
* **Develop**: Write PL/SQL procedures for handling insurance policy management, policy processing, and generating policy reports.
* **Test**: Test the procedures with various scenarios (e.g., managing policies, processing requests, generating reports, ensuring proper updates).