**Case Study: Financial Account Management System**

**Problem Statement:**

Design and implement a Financial Account Management System using Oracle SQL and PL/SQL. The system will be used to manage financial accounts, track account transactions, and generate account reports. Your task is to create the necessary database schema, populate the database with sample data, and develop PL/SQL procedures to handle account management, transaction tracking, and report generation.

**Requirements:**

1. **Account Management**:
   * Implement the functionality to add, update, delete, and search for financial accounts.
   * Ensure that each account has attributes such as ACCOUNT\_ID, CUSTOMER\_ID, ACCOUNT\_TYPE, OPENING\_DATE, and BALANCE.
2. **Transaction Tracking**:
   * Implement the functionality to track transactions for each account.
   * Ensure that each transaction record has attributes such as TRANSACTION\_ID, ACCOUNT\_ID, TRANSACTION\_DATE, AMOUNT, and DESCRIPTION.
3. **Report Generation**:
   * Implement the functionality to generate account reports.
   * Reports should include details such as account balances, transaction summaries, and monthly statements.

**Tasks:**

1. **Design the Database Schema**:
   * Create the FinancialAccounts, AccountTransactions, and AccountReports 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 FinancialAccounts and AccountTransactions tables to facilitate testing of the system.
3. **Develop PL/SQL Procedures**:
   * Create a procedure to handle financial account management. The procedure should insert, update, and delete account records.
   * Create a procedure to track account transactions. The procedure should insert transaction records and update account balances.
   * Create a procedure to generate account reports. The procedure should aggregate data to provide insights into account balances and transaction summaries.

**Expected Outcomes:**

1. **FinancialAccounts Table**:
   * Contains all information about the financial accounts.
2. **AccountTransactions Table**:
   * Tracks the transactions for each financial account.
3. **AccountReports Table**:
   * Stores the generated account reports.
4. **PL/SQL Procedures**:
   * Efficiently manage financial accounts, track transactions, and generate reports, maintaining accurate records in the database.

**Deliverables:**

1. SQL scripts to create the FinancialAccounts, AccountTransactions, and AccountReports tables.
2. SQL scripts to insert sample data into the FinancialAccounts and AccountTransactions tables.
3. PL/SQL scripts for the procedures to handle financial account management, transaction tracking, and generate account reports.
4. Documentation explaining how to set up and use the system, including how to run the PL/SQL procedures.

**Database Schema:**

1. **FinancialAccounts Table**:
   * **ACCOUNT\_ID**: Number, Primary Key
   * **CUSTOMER\_ID**: Number
   * **ACCOUNT\_TYPE**: Varchar2(50)
   * **OPENING\_DATE**: Date
   * **BALANCE**: Number
2. **AccountTransactions Table**:
   * **TRANSACTION\_ID**: Number, Primary Key
   * **ACCOUNT\_ID**: Number, Foreign Key References FinancialAccounts(ACCOUNT\_ID)
   * **TRANSACTION\_DATE**: Date
   * **AMOUNT**: Number
   * **DESCRIPTION**: Varchar2(255)
3. **AccountReports Table**:
   * **REPORT\_ID**: Number, Primary Key
   * **REPORT\_DATE**: Date
   * **ACCOUNT\_BALANCES**: Clob
   * **TRANSACTION\_SUMMARY**: Clob
   * **MONTHLY\_STATEMENT**: Clob

**Case Study Task:**

* **Design**: Create the database schema as provided.
* **Implement**: Insert sample data into the FinancialAccounts and AccountTransactions tables.
* **Develop**: Write PL/SQL procedures for handling financial account management, transaction tracking, and generating account reports.
* **Test**: Test the procedures with various scenarios (e.g., managing accounts, tracking transactions, generating reports, ensuring proper updates).