**Exercise 6: Cursors**

**Scenario 1:** Generate monthly statements for all customers.

* + **Question:** Write a PL/SQL block using an explicit cursor **GenerateMonthlyStatements** that retrieves all transactions for the current month and prints a statement for each customer.

DECLARE

    CURSOR cur\_customers IS

        SELECT CustomerID, Name

        FROM Customers;

    CURSOR cur\_transactions (p\_customerID NUMBER) IS

        SELECT TransactionID, AccountID, TransactionDate, Amount, TransactionType

        FROM Transactions

        WHERE AccountID IN (SELECT AccountID FROM Accounts WHERE CustomerID = p\_customerID)

          AND TRUNC(TransactionDate, 'MM') = TRUNC(SYSDATE, 'MM');

    v\_customerID Customers.CustomerID%TYPE;

    v\_customerName Customers.Name%TYPE;

    v\_transactionID Transactions.TransactionID%TYPE;

    v\_accountID Transactions.AccountID%TYPE;

    v\_transactionDate Transactions.TransactionDate%TYPE;

    v\_amount Transactions.Amount%TYPE;

    v\_transactionType Transactions.TransactionType%TYPE;

BEGIN

    OPEN cur\_customers;

    LOOP

        FETCH cur\_customers INTO v\_customerID, v\_customerName;

        EXIT WHEN cur\_customers%NOTFOUND;

        DBMS\_OUTPUT.PUT\_LINE('Monthly Statement for Customer: ' || v\_customerName);

        DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------');

        OPEN cur\_transactions(v\_customerID);

        LOOP

            FETCH cur\_transactions INTO v\_transactionID, v\_accountID, v\_transactionDate, v\_amount, v\_transactionType;

            EXIT WHEN cur\_transactions%NOTFOUND;

            DBMS\_OUTPUT.PUT\_LINE('Transaction ID: ' || v\_transactionID ||

                                 ', Account ID: ' || v\_accountID ||

                                 ', Date: ' || TO\_CHAR(v\_transactionDate, 'DD-MON-YYYY') ||

                                 ', Amount: ' || v\_amount ||

                                 ', Type: ' || v\_transactionType);

        END LOOP;

        CLOSE cur\_transactions;

        DBMS\_OUTPUT.PUT\_LINE('-------------------------------------------');

    END LOOP;

    CLOSE cur\_customers;

END;

/

**Scenario 2:** Apply annual fee to all accounts.

* + **Question:** Write a PL/SQL block using an explicit cursor **ApplyAnnualFee** that deducts an annual maintenance fee from the balance of all accounts.

DECLARE

    CURSOR cur\_accounts IS SELECT AccountID, Balance FROM Accounts;

    v\_accountID Accounts.AccountID%TYPE;

    v\_balance Accounts.Balance%TYPE;

    v\_annualFee CONSTANT NUMBER := 50; -- Annual maintenance fee

BEGIN

    OPEN cur\_accounts;

    LOOP

        FETCH cur\_accounts INTO v\_accountID, v\_balance;

        EXIT WHEN cur\_accounts%NOTFOUND;

        UPDATE Accounts

        SET Balance = v\_balance - v\_annualFee, LastModified = SYSDATE

        WHERE AccountID = v\_accountID;

    END LOOP;

    CLOSE cur\_accounts;

END;

/

**Scenario 3:** Update the interest rate for all loans based on a new policy.

* + **Question:** Write a PL/SQL block using an explicit cursor **UpdateLoanInterestRates** that fetches all loans and updates their interest rates based on the new policy.

DECLARE

    CURSOR cur\_loans IS SELECT LoanID, InterestRate FROM Loans;

    v\_loanID Loans.LoanID%TYPE;

    v\_interestRate Loans.InterestRate%TYPE;

    v\_newInterestRate CONSTANT NUMBER := 6.5;

BEGIN

    OPEN cur\_loans;

    LOOP

        FETCH cur\_loans INTO v\_loanID, v\_interestRate;

        EXIT WHEN cur\_loans%NOTFOUND;

        UPDATE Loans

        SET InterestRate = v\_newInterestRate WHERE LoanID = v\_loanID;

    END LOOP;

    CLOSE cur\_loans;

END;

/