**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 transaction\_cursor IS

SELECT c.CustomerID, c.Name, t.TransactionDate, t.Amount, t.TransactionType

FROM Customers c

JOIN Accounts a ON c.CustomerID = a.CustomerID

JOIN Transactions t ON a.AccountID = t.AccountID

WHERE TRUNC(t.TransactionDate, 'MM') = TRUNC(SYSDATE, 'MM');

BEGIN

FOR rec IN transaction\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || rec.Name || ' | Date: ' || rec.TransactionDate ||

' | Amount: ' || rec.Amount || ' | Type: ' || rec.TransactionType);

END LOOP;

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 account\_cursor IS

SELECT AccountID, Balance

FROM Accounts;

v\_annual\_fee CONSTANT NUMBER := 100; -- Define the annual fee

BEGIN

FOR rec IN account\_cursor LOOP

UPDATE Accounts

SET Balance = Balance - v\_annual\_fee

WHERE AccountID = rec.AccountID;

END LOOP;

COMMIT;

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 loan\_cursor IS

SELECT LoanID, InterestRate

FROM Loans;

v\_new\_interest\_rate CONSTANT NUMBER := 4.5; -- New policy interest rate

BEGIN

FOR rec IN loan\_cursor LOOP

UPDATE Loans

SET InterestRate = v\_new\_interest\_rate

WHERE LoanID = rec.LoanID;

END LOOP;

COMMIT;

END;

/