**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 c06 IS

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

FROM Transactions t JOIN Accounts a ON t.AccountID = a.AccountID

JOIN Customers c ON a.CustomerID = c.CustomerID WHERE t.TransactionDate BETWEEN TRUNC(SYSDATE, 'MM') AND LAST\_DAY(SYSDATE) ORDER BY c.CustomerID, t.TransactionDate;

v\_CustomerID Customers.CustomerID%TYPE := NULL;

v\_Name Customers.Name%TYPE;

v\_TransactionDate Transactions.TransactionDate%TYPE;

v\_Amount Transactions.Amount%TYPE;

v\_TotalAmount NUMBER := 0;

BEGIN

OPEN c06;

LOOP

FETCH c06 INTO v\_CustomerID, v\_Name, v\_TransactionDate, v\_Amount;

EXIT WHEN c06%NOTFOUND;

IF v\_CustomerID IS NOT NULL THEN

IF v\_CustomerID != NVL(v\_CustomerID, v\_CustomerID) THEN

DBMS\_OUTPUT.PUT\_LINE('Monthly Statement for Customer ID: ' || v\_CustomerID || ', Name: ' || v\_Name);

v\_TotalAmount := 0;

END IF;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Date: ' || TO\_CHAR(v\_TransactionDate, 'YYYY-MM-DD') || ', Amount: ' || v\_Amount);

v\_TotalAmount := v\_TotalAmount + v\_Amount;

IF c06%NOTFOUND OR v\_CustomerID != NVL(v\_CustomerID, v\_CustomerID) THEN

DBMS\_OUTPUT.PUT\_LINE('Total Amount: ' || v\_TotalAmount);

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

END IF;

END LOOP;

CLOSE c06;

END;

/

PL/SQL procedure successfully completed.

**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 c06b IS SELECT AccountID, Balance FROM Accounts;

v\_AccountID Accounts.AccountID%TYPE;

v\_Balance Accounts.Balance%TYPE;

v\_AnnualFee NUMBER := 100;

BEGIN

OPEN c06b;

LOOP

FETCH c06b INTO v\_AccountID, v\_Balance;

EXIT WHEN c06b%NOTFOUND;

UPDATE Accounts

SET Balance = v\_Balance - v\_AnnualFee

WHERE AccountID = v\_AccountID;

END LOOP;

CLOSE c06b;

COMMIT;

END;

/

PL/SQL procedure successfully completed.

**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 c\_loans IS SELECT LoanID, InterestRate FROM Loans;

v\_LoanID Loans.LoanID%TYPE;

v\_OldInterestRate Loans.InterestRate%TYPE;

v\_NewInterestRate Loans.InterestRate%TYPE;

v\_NewInterestRatePolicy NUMBER := 10;

BEGIN

OPEN c\_loans;

LOOP

FETCH c\_loans INTO v\_LoanID, v\_OldInterestRate;

EXIT WHEN c\_loans%NOTFOUND;

UPDATE Loans

SET InterestRate = v\_NewInterestRatePolicy

WHERE LoanID = v\_LoanID;

END LOOP;

CLOSE c\_loans;

COMMIT;

END;

/

PL/SQL procedure successfully completed.