**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 t.TransactionID, t.AccountID, t.TransactionDate, t.Amount, t.TransactionType, a.CustomerID, c.Name

FROM Transactions t

JOIN Accounts a ON t.AccountID = a.AccountID

JOIN Customers c ON a.CustomerID = c.CustomerID

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

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;

v\_CustomerID Accounts.CustomerID%TYPE;

v\_CustomerName Customers.Name%TYPE;

BEGIN

OPEN transaction\_cursor;

LOOP

FETCH transaction\_cursor INTO v\_TransactionID, v\_AccountID, v\_TransactionDate, v\_Amount, v\_TransactionType, v\_CustomerID, v\_CustomerName;

EXIT WHEN transaction\_cursor%NOTFOUND;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || v\_CustomerName || ' (ID: ' || v\_CustomerID || ')');

DBMS\_OUTPUT.PUT\_LINE('Transaction ID: ' || v\_TransactionID || ', Account ID: ' || v\_AccountID);

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

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

END LOOP;

CLOSE transaction\_cursor;

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\_AccountID Accounts.AccountID%TYPE;

v\_Balance Accounts.Balance%TYPE;

v\_AnnualFee CONSTANT NUMBER := 50; -- Assume a fixed annual fee

BEGIN

OPEN account\_cursor;

LOOP

FETCH account\_cursor INTO v\_AccountID, v\_Balance;

EXIT WHEN account\_cursor%NOTFOUND;

UPDATE Accounts

SET Balance = Balance - v\_AnnualFee

WHERE AccountID = v\_AccountID;

DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_AccountID || ' - Annual fee applied. New balance: ' || (v\_Balance - v\_AnnualFee));

END LOOP;

CLOSE account\_cursor;

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\_LoanID Loans.LoanID%TYPE;

v\_InterestRate Loans.InterestRate%TYPE;

v\_InterestRateIncrease CONSTANT NUMBER := 0.5; -- New policy: increase by 0.5%

BEGIN

OPEN loan\_cursor;

LOOP

FETCH loan\_cursor INTO v\_LoanID, v\_InterestRate;

EXIT WHEN loan\_cursor%NOTFOUND;

UPDATE Loans

SET InterestRate = v\_InterestRate + v\_InterestRateIncrease

WHERE LoanID = v\_LoanID;

DBMS\_OUTPUT.PUT\_LINE('Loan ID: ' || v\_LoanID || ' - New interest rate: ' || (v\_InterestRate + v\_InterestRateIncrease));

END LOOP;

CLOSE loan\_cursor;

COMMIT;

END;

/