Exercise 6: Cursors

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

**o 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 c\_transactions IS

SELECT c.CustomerID, c.Name, t.TransactionID, 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 EXTRACT(MONTH FROM t.TransactionDate) = EXTRACT(MONTH FROM SYSDATE)

AND EXTRACT(YEAR FROM t.TransactionDate) = EXTRACT(YEAR FROM SYSDATE);

v\_customer\_id Customers.CustomerID%TYPE;

v\_customer\_name Customers.Name%TYPE;

v\_transaction\_id Transactions.TransactionID%TYPE;

v\_transaction\_date Transactions.TransactionDate%TYPE;

v\_amount Transactions.Amount%TYPE;

v\_transaction\_type Transactions.TransactionType%TYPE;

BEGIN

FOR r\_transaction IN c\_transactions LOOP

v\_customer\_id := r\_transaction.CustomerID;

v\_customer\_name := r\_transaction.Name;

v\_transaction\_id := r\_transaction.TransactionID;

v\_transaction\_date := r\_transaction.TransactionDate;

v\_amount := r\_transaction.Amount;

v\_transaction\_type := r\_transaction.TransactionType;

DBMS\_OUTPUT.PUT\_LINE('Customer: ' || v\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Transaction ID: ' || v\_transaction\_id);

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

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

DBMS\_OUTPUT.PUT\_LINE('Type: ' || v\_transaction\_type);

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

END LOOP;

END;

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

**o 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 c\_accounts IS

SELECT AccountID, Balance

FROM Accounts;

v\_account\_id Accounts.AccountID%TYPE;

v\_balance Accounts.Balance%TYPE;

v\_fee NUMBER := 50;

BEGIN

FOR r\_account IN c\_accounts LOOP

v\_account\_id := r\_account.AccountID;

v\_balance := r\_account.Balance;

UPDATE Accounts

SET Balance = v\_balance - v\_fee

WHERE AccountID = v\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_account\_id || ' updated. New Balance: ' || (v\_balance - v\_fee));

END LOOP;

COMMIT;

END;

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

**o 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\_loan\_id Loans.LoanID%TYPE;

v\_current\_interest\_rate Loans.InterestRate%TYPE;

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

BEGIN

FOR r\_loan IN c\_loans LOOP

v\_loan\_id := r\_loan.LoanID;

v\_current\_interest\_rate := r\_loan.InterestRate;

UPDATE Loans

SET InterestRate = v\_new\_interest\_rate

WHERE LoanID = v\_loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Loan ID: ' || v\_loan\_id || ' interest rate updated from ' || v\_current\_interest\_rate || '% to ' || v\_new\_interest\_rate || '%');

END LOOP;

COMMIT;

END;