**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 c\_monthly\_transactions 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');

v\_customer\_id Customers.CustomerID%TYPE;

v\_customer\_name Customers.Name%TYPE;

v\_transaction\_date Transactions.TransactionDate%TYPE;

v\_amount Transactions.Amount%TYPE;

v\_transaction\_type Transactions.TransactionType%TYPE;

BEGIN

OPEN c\_monthly\_transactions;

LOOP

FETCH c\_monthly\_transactions INTO v\_customer\_id, v\_customer\_name, v\_transaction\_date, v\_amount, v\_transaction\_type;

EXIT WHEN c\_monthly\_transactions%NOTFOUND;

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

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

DBMS\_OUTPUT.PUT\_LINE('Transaction Date: ' || v\_transaction\_date);

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

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

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

END LOOP;

CLOSE c\_monthly\_transactions;

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 c\_all\_accounts IS

SELECT AccountID, Balance

FROM Accounts;

v\_account\_id Accounts.AccountID%TYPE;

v\_balance Accounts.Balance%TYPE;

v\_annual\_fee NUMBER := 100; -- Example annual fee

BEGIN

OPEN c\_all\_accounts;

LOOP

FETCH c\_all\_accounts INTO v\_account\_id, v\_balance;

EXIT WHEN c\_all\_accounts%NOTFOUND;

IF v\_balance >= v\_annual\_fee THEN

UPDATE Accounts

SET Balance = Balance - v\_annual\_fee

WHERE AccountID = v\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Annual fee of ' || v\_annual\_fee || ' applied to Account ID: ' || v\_account\_id);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance for Account ID: ' || v\_account\_id);

END IF;

END LOOP;

CLOSE c\_all\_accounts;

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 c\_all\_loans IS

SELECT LoanID, InterestRate

FROM Loans;

v\_loan\_id Loans.LoanID%TYPE;

v\_interest\_rate Loans.InterestRate%TYPE;

v\_new\_interest\_rate NUMBER;

BEGIN

OPEN c\_all\_loans;

LOOP

FETCH c\_all\_loans INTO v\_loan\_id, v\_interest\_rate;

EXIT WHEN c\_all\_loans%NOTFOUND;

v\_new\_interest\_rate := v\_interest\_rate + 0.5;

UPDATE Loans

SET InterestRate = v\_new\_interest\_rate

WHERE LoanID = v\_loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Updated Loan ID: ' || v\_loan\_id || ' | Old Rate: ' || v\_interest\_rate || ' | New Rate: ' || v\_new\_interest\_rate);

END LOOP;

CLOSE c\_all\_loans;

COMMIT;

END;