**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

-- Define a record type to hold transaction details

TYPE TransactionRec IS RECORD (

TransactionID Transactions.TransactionID%TYPE,

AccountID Transactions.AccountID%TYPE,

TransactionDate Transactions.TransactionDate%TYPE,

Amount Transactions.Amount%TYPE,

TransactionType Transactions.TransactionType%TYPE

);

-- Define a cursor to fetch transactions for the current month

CURSOR c\_transactions IS

SELECT t.TransactionID, t.AccountID, t.TransactionDate, t.Amount, t.TransactionType

FROM Transactions t

JOIN Accounts a ON t.AccountID = a.AccountID

JOIN Customers c ON a.CustomerID = c.CustomerID

WHERE t.TransactionDate >= TRUNC(SYSDATE, 'MM') -- Start of the current month

AND t.TransactionDate < TRUNC(SYSDATE, 'MM') + INTERVAL '1' MONTH; -- End of the current month

v\_transaction TransactionRec;

v\_current\_customer NUMBER := NULL;

BEGIN

OPEN c\_transactions; -- Open cursor

FETCH c\_transactions INTO v\_transaction; -- Fetch the first record

WHILE c\_transactions%FOUND LOOP -- Loop through all transactions

IF v\_current\_customer IS NULL OR v\_transaction.AccountID != v\_current\_customer THEN

IF v\_current\_customer IS NOT NULL THEN

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

END IF;

DBMS\_OUTPUT.PUT\_LINE('Statement for Customer ID: ' || v\_transaction.AccountID);

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

v\_current\_customer := v\_transaction.AccountID;

END IF;

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

DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_transaction.AccountID);

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

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

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

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

FETCH c\_transactions INTO v\_transaction; -- Fetch the next record

END LOOP;

CLOSE c\_transactions; -- Close the cursor

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

IF c\_transactions%ISOPEN THEN

CLOSE c\_transactions;

END IF;

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

v\_annual\_fee NUMBER := 50; -- Example fee amount

TYPE AccountRec IS RECORD (

AccountID Accounts.AccountID%TYPE,

Balance Accounts.Balance%TYPE

);

-- Define a cursor to fetch all accounts

CURSOR c\_accounts IS SELECT AccountID, Balance FROM Accounts;

v\_account AccountRec; -- Define a variable to hold each account record

BEGIN

OPEN c\_accounts; -- Open the cursor

FETCH c\_accounts INTO v\_account; -- Fetch the first record

WHILE c\_accounts%FOUND LOOP

UPDATE Accounts SET Balance = Balance - v\_annual\_fee WHERE AccountID = v\_account.AccountID;

DBMS\_OUTPUT.PUT\_LINE('Applied annual fee of ' || v\_annual\_fee || ' to Account ID: ' || v\_account.AccountID);

FETCH c\_accounts INTO v\_account;-- Fetch the next record

END LOOP;

CLOSE c\_accounts; -- Close the cursor

COMMIT;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

ROLLBACK;

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

new\_interest NUMBER := 5.5; -- Example new interest rate (4.5%)

TYPE LoanRec IS RECORD (

LoanID Loans.LoanID%TYPE,

CustomerID Loans.CustomerID%TYPE,

LoanAmount Loans.LoanAmount%TYPE,

InterestRate Loans.InterestRate%TYPE,

StartDate Loans.StartDate%TYPE,

EndDate Loans.EndDate%TYPE

);

CURSOR c\_loans IS SELECT LoanID, CustomerID, LoanAmount, InterestRate, StartDate, EndDate FROM Loans WHERE LoanAmount<10000 ORDER BY LoanID;

v\_loan LoanRec;

BEGIN

OPEN c\_loans; -- Open the cursor

FETCH c\_loans INTO v\_loan; -- Fetch the first record

WHILE c\_loans%FOUND LOOP

UPDATE Loans SET InterestRate = new\_interest WHERE LoanID = v\_loan.LoanID;

DBMS\_OUTPUT.PUT\_LINE('Updated interest rate to ' || new\_interest || '% for Loan ID: ' || v\_loan.LoanID);

FETCH c\_loans INTO v\_loan;

END LOOP;

CLOSE c\_loans;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

ROLLBACK;

CLOSE c\_loans;

END;