**Exercise 6: Cursors**

**Scenario 1: Generate Monthly Statements for All Customers**

**PL/SQL Block: GenerateMonthlyStatements**

DECLARE

CURSOR transactions\_cursor IS

SELECT t.customerid, t.transactiondate, t.amount, t.transactiontype

FROM transactions t

JOIN accounts a ON t.accountid = a.accountid

WHERE t.transactiondate BETWEEN TRUNC(SYSDATE, 'MM') AND LAST\_DAY(SYSDATE);

v\_customerid transactions.customerid%TYPE;

v\_transactiondate transactions.transactiondate%TYPE;

v\_amount transactions.amount%TYPE;

v\_transactiontype transactions.transactiontype%TYPE;

BEGIN

OPEN transactions\_cursor;

LOOP

FETCH transactions\_cursor INTO v\_customerid, v\_transactiondate, v\_amount, v\_transactiontype;

EXIT WHEN transactions\_cursor%NOTFOUND;

-- Print statement for each transaction

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

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

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

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

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

END LOOP;

CLOSE transactions\_cursor;

END;

/

**Scenario 2: Apply Annual Fee to All Accounts**

**PL/SQL Block: ApplyAnnualFee**

DECLARE

CURSOR accounts\_cursor IS

SELECT accountid, balance

FROM accounts;

v\_accountid accounts.accountid%TYPE;

v\_balance accounts.balance%TYPE;

v\_annual\_fee CONSTANT NUMBER := 50; -- Annual maintenance fee amount

BEGIN

OPEN accounts\_cursor;

LOOP

FETCH accounts\_cursor INTO v\_accountid, v\_balance;

EXIT WHEN accounts\_cursor%NOTFOUND;

-- Deduct annual maintenance fee

v\_balance := v\_balance - v\_annual\_fee;

-- Update the account balance

UPDATE accounts

SET balance = v\_balance

WHERE accountid = v\_accountid;

END LOOP;

CLOSE accounts\_cursor;

COMMIT; -- Commit the changes to the database

END;

/

**Scenario 3: Update the Interest Rate for All Loans Based on a New Policy**

**PL/SQL Block: UpdateLoanInterestRates**

DECLARE

CURSOR loans\_cursor IS

SELECT loanid, interestrate

FROM loans;

v\_loanid loans.loanid%TYPE;

v\_interestrate loans.interestrate%TYPE;

v\_new\_interestrate CONSTANT NUMBER := 4.5; -- New interest rate based on policy

BEGIN

OPEN loans\_cursor;

LOOP

FETCH loans\_cursor INTO v\_loanid, v\_interestrate;

EXIT WHEN loans\_cursor%NOTFOUND;

-- Update the interest rate based on the new policy

v\_interestrate := v\_new\_interestrate;

-- Update the loan interest rate

UPDATE loans

SET interestrate = v\_interestrate

WHERE loanid = v\_loanid;

END LOOP;

CLOSE loans\_cursor;

COMMIT; -- Commit the changes to the database

END;

/