**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.
  + **Solution:**

DECLARE

CURSOR c\_transactions IS

SELECT t.TransactionID, t.AccountID, t.TransactionDate, t.Amount, t.TransactionType, c.CustomerID, c.Name

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') AND t.TransactionDate < ADD\_MONTHS(TRUNC(SYSDATE, 'MM'), 1)

ORDER BY c.CustomerID, t.TransactionDate;

v\_transaction c\_transactions%ROWTYPE;

v\_current\_customer\_id Customers.CustomerID%TYPE := NULL;

BEGIN

OPEN c\_transactions;

LOOP

FETCH c\_transactions INTO v\_transaction;

EXIT WHEN c\_transactions%NOTFOUND;

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

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

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

END IF;

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

', Date: ' || TO\_CHAR(v\_transaction.TransactionDate, 'YYYY-MM-DD') ||

', Type: ' || v\_transaction.TransactionType ||

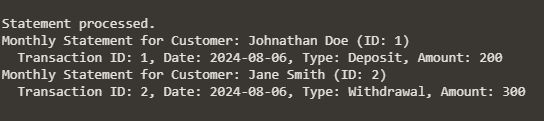
', Amount: ' || v\_transaction.Amount);

END LOOP;

CLOSE c\_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.
  + **Solution:**

DECLARE

CURSOR c\_accounts IS

SELECT AccountID, Balance

FROM Accounts;

v\_account c\_accounts%ROWTYPE;

v\_annual\_fee CONSTANT NUMBER := 50;

BEGIN

OPEN c\_accounts;

LOOP

FETCH c\_accounts INTO v\_account;

EXIT WHEN c\_accounts%NOTFOUND;

UPDATE Accounts

SET Balance = Balance - v\_annual\_fee,

LastModified = SYSDATE

WHERE AccountID = v\_account.AccountID;

END LOOP;

CLOSE c\_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.
  + **Solution:**

DECLARE

CURSOR cur\_loans IS

SELECT LoanID, InterestRate

FROM Loans;

v\_loanID Loans.LoanID%TYPE;

v\_interestRate Loans.InterestRate%TYPE;

v\_newInterestRate CONSTANT NUMBER := 5;

BEGIN

OPEN cur\_loans;

LOOP

FETCH cur\_loans INTO v\_loanID, v\_interestRate;

EXIT WHEN cur\_loans%NOTFOUND;

UPDATE Loans

SET InterestRate = v\_newInterestRate

WHERE LoanID = v\_loanID;

DBMS\_OUTPUT.PUT\_LINE('Loan ID: ' || v\_loanID || ' New Interest Rate: ' || v\_newInterestRate);

END LOOP;

CLOSE cur\_loans;

END;

/

