**Exercise 6: Cursors**

**Task:** 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 customer\_cursor IS

SELECT DISTINCT c.CustomerID, c.Name

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\_customerID Customers.CustomerID%TYPE;

v\_name Customers.Name%TYPE;

v\_amount Transactions.Amount%TYPE;

v\_transactionType Transactions.TransactionType%TYPE;

v\_cursor SYS\_REFCURSOR;

BEGIN

FOR cust\_record IN customer\_cursor LOOP

v\_customerID := cust\_record.CustomerID;

v\_name := cust\_record.Name;

-- Print customer statement header

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

OPEN v\_cursor FOR

SELECT Amount, TransactionType

FROM Transactions t

JOIN Accounts a ON t.AccountID = a.AccountID

WHERE a.CustomerID = v\_customerID

AND EXTRACT(MONTH FROM t.TransactionDate) = EXTRACT(MONTH FROM SYSDATE)

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

LOOP

FETCH v\_cursor INTO v\_amount, v\_transactionType;

EXIT WHEN v\_cursor%NOTFOUND;

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

END LOOP;

CLOSE v\_cursor;

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

END LOOP;

END;

/  
  
**Task:** Write a PL/SQL block using an explicit cursor ApplyAnnualFee that deducts an annual maintenance fee from the balance of all accounts.  
  
DECLARE

CURSOR account\_cursor IS

SELECT AccountID, Balance

FROM Accounts;

v\_accountID Accounts.AccountID%TYPE;

v\_balance Accounts.Balance%TYPE;

annual\_fee CONSTANT NUMBER := 50; -- Define the annual fee amount

BEGIN

FOR acc\_record IN account\_cursor LOOP

v\_accountID := acc\_record.AccountID;

v\_balance := acc\_record.Balance;

-- Deduct the annual fee

UPDATE Accounts

SET Balance = v\_balance - annual\_fee

WHERE AccountID = v\_accountID;

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

END LOOP;

COMMIT; -- Commit the changes

END;

/  
  
**Task:** Write a PL/SQL block using an explicit cursor UpdateLoanInterestRates that fetches all loans and updates their interest rates based on a new policy.  
  
DECLARE

CURSOR loan\_cursor IS

SELECT LoanID, InterestRate

FROM Loans;

v\_loanID Loans.LoanID%TYPE;

v\_oldRate Loans.InterestRate%TYPE;

v\_newRate Loans.InterestRate%TYPE;

-- Define the new interest rate policy

BEGIN

FOR loan\_record IN loan\_cursor LOOP

v\_loanID := loan\_record.LoanID;

v\_oldRate := loan\_record.InterestRate;

-- Apply new interest rate based on policy

v\_newRate := v\_oldRate + 1; -- Example: increase interest rate by 1%

UPDATE Loans

SET InterestRate = v\_newRate

WHERE LoanID = v\_loanID;

DBMS\_OUTPUT.PUT\_LINE('Updated Loan ID: ' || v\_loanID || ' from Interest Rate: ' || v\_oldRate || ' to: ' || v\_newRate);

END LOOP;

COMMIT; -- Commit the changes

END;

/