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

DECLARE

CURSOR transactions\_cursor IS

SELECT c.CustomerID, c.Name, t.TransactionID, 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 t.TransactionDate BETWEEN TRUNC(SYSDATE, 'MM') AND LAST\_DAY(SYSDATE)

ORDER BY c.CustomerID, t.TransactionDate;

v\_CustomerID Customers.CustomerID%TYPE;

v\_Name Customers.Name%TYPE;

v\_TransactionID Transactions.TransactionID%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\_Name, v\_TransactionID, v\_TransactionDate, v\_Amount, v\_TransactionType;

EXIT WHEN transactions\_cursor%NOTFOUND;

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

DBMS\_OUTPUT.PUT\_LINE('Name: ' || v\_Name);

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

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**

DECLARE

CURSOR accounts\_cursor IS

SELECT AccountID, Balance

FROM Accounts;

v\_AccountID Accounts.AccountID%TYPE;

v\_Balance Accounts.Balance%TYPE;

v\_AnnualFee CONSTANT NUMBER := 50;

BEGIN

OPEN accounts\_cursor;

LOOP

FETCH accounts\_cursor INTO v\_AccountID, v\_Balance;

EXIT WHEN accounts\_cursor%NOTFOUND;

-- Deduct the annual maintenance fee from the balance

UPDATE Accounts

SET Balance = Balance - v\_AnnualFee

WHERE AccountID = v\_AccountID;

-- Optionally, log the fee deduction as a transaction

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (Transactions\_SEQ.NEXTVAL, v\_AccountID, SYSDATE, -v\_AnnualFee, 'Fee');

END LOOP;

CLOSE accounts\_cursor;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error message

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

/

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

DECLARE

CURSOR loans\_cursor IS

SELECT LoanID, InterestRate

FROM Loans;

v\_LoanID Loans.LoanID%TYPE;

v\_InterestRate Loans.InterestRate%TYPE;

-- Define the new interest rate policy

v\_NewInterestRate CONSTANT NUMBER := 5.5;

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

UPDATE Loans

SET InterestRate = v\_NewInterestRate

WHERE LoanID = v\_LoanID;

END LOOP;

CLOSE loans\_cursor;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error message

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

/