**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.

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.

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

Here are the PL/SQL blocks with explicit cursors for the given scenarios:

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

DECLARE

CURSOR c\_transactions IS

SELECT customer\_id, transaction\_id, transaction\_date, transaction\_type, amount

FROM transactions

WHERE transaction\_date BETWEEN TRUNC(SYSDATE, 'MM') AND LAST\_DAY(SYSDATE);

v\_customer\_id transactions.customer\_id%TYPE;

v\_transaction\_id transactions.transaction\_id%TYPE;

v\_transaction\_date transactions.transaction\_date%TYPE;

v\_transaction\_type transactions.transaction\_type%TYPE;

v\_amount transactions.amount%TYPE;

BEGIN

OPEN c\_transactions;

LOOP

FETCH c\_transactions INTO v\_customer\_id, v\_transaction\_id, v\_transaction\_date, v\_transaction\_type, v\_amount;

EXIT WHEN c\_transactions%NOTFOUND;

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

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

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

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

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

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

END LOOP;

CLOSE c\_transactions;

DBMS\_OUTPUT.PUT\_LINE('Monthly statements generated for all customers.');

END;

/

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

DECLARE

CURSOR c\_accounts IS

SELECT account\_id, balance

FROM accounts;

v\_account\_id accounts.account\_id%TYPE;

v\_balance accounts.balance%TYPE;

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

BEGIN

OPEN c\_accounts;

LOOP

FETCH c\_accounts INTO v\_account\_id, v\_balance;

EXIT WHEN c\_accounts%NOTFOUND;

-- Deduct the annual fee from the balance

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

UPDATE accounts

SET balance = v\_balance

WHERE account\_id = v\_account\_id;

END LOOP;

CLOSE c\_accounts;

DBMS\_OUTPUT.PUT\_LINE('Annual fee applied to all accounts.');

END;

/

**Scenario 3: Update Loan Interest Rates Based on New Policy**

DECLARE

CURSOR c\_loans IS

SELECT loan\_id, interest\_rate

FROM loans;

v\_loan\_id loans.loan\_id%TYPE;

v\_interest\_rate loans.interest\_rate%TYPE;

v\_new\_interest\_rate CONSTANT NUMBER := 5; -- Define the new interest rate as per policy

BEGIN

OPEN c\_loans;

LOOP

FETCH c\_loans INTO v\_loan\_id, v\_interest\_rate;

EXIT WHEN c\_loans%NOTFOUND;

-- Update the interest rate based on the new policy

v\_interest\_rate := v\_new\_interest\_rate;

UPDATE loans

SET interest\_rate = v\_interest\_rate

WHERE loan\_id = v\_loan\_id;

END LOOP;

CLOSE c\_loans;

DBMS\_OUTPUT.PUT\_LINE('Loan interest rates updated based on the new policy.');

END;

/

Each of these PL/SQL blocks utilizes an explicit cursor to process records from the relevant tables, performing the necessary actions such as generating statements, applying fees, and updating interest rates.