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.

DECLARE

CURSOR c\_transactions IS

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

FROM transactions

WHERE EXTRACT(MONTH FROM transaction\_date) = EXTRACT(MONTH FROM SYSDATE)

AND EXTRACT(YEAR FROM transaction\_date) = EXTRACT(YEAR FROM SYSDATE);

v\_customer\_id customers.customer\_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\_date, v\_transaction\_type, v\_amount;

EXIT WHEN c\_transactions%NOTFOUND;

-- Print or process the statement for each customer

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

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

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;

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.

DECLARE

CURSOR c\_accounts IS

SELECT account\_id, balance

FROM accounts;

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

v\_balance accounts.balance%TYPE;

v\_fee NUMBER := 50; -- Example fee amount

BEGIN

OPEN c\_accounts;

LOOP

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

EXIT WHEN c\_accounts%NOTFOUND;

-- Deduct the annual maintenance fee from each account's balance

UPDATE accounts

SET balance = v\_balance - v\_fee

WHERE account\_id = v\_account\_id;

DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_account\_id || ' - Fee Applied');

END LOOP;

CLOSE c\_accounts;

COMMIT; -- Commit changes

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.

DECLARE

CURSOR c\_loans IS

SELECT loan\_id, current\_interest\_rate

FROM loans;

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

v\_current\_interest\_rate loans.current\_interest\_rate%TYPE;

v\_new\_interest\_rate NUMBER;

BEGIN

OPEN c\_loans;

LOOP

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

EXIT WHEN c\_loans%NOTFOUND;

-- Example of new interest rate policy, adjust as needed

v\_new\_interest\_rate := v\_current\_interest\_rate + 0.5;

-- Update the loan's interest rate

UPDATE loans

SET current\_interest\_rate = v\_new\_interest\_rate

WHERE loan\_id = v\_loan\_id;

DBMS\_OUTPUT.PUT\_LINE('Loan ID: ' || v\_loan\_id || ' - Interest Rate Updated to ' || v\_new\_interest\_rate);

END LOOP;

CLOSE c\_loans;

COMMIT; -- Commit changes

END;

/