**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 GenerateMonthlyStatements IS

SELECT c.CustomerID, c.Name, 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 EXTRACT(MONTH FROM t.TransactionDate) = EXTRACT(MONTH FROM SYSDATE)

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

ORDER BY c.CustomerID, t.TransactionDate;

v\_customer\_id Customers.CustomerID%TYPE;

v\_customer\_name Customers.Name%TYPE;

v\_transaction\_date Transactions.TransactionDate%TYPE;

v\_amount Transactions.Amount%TYPE;

v\_transaction\_type Transactions.TransactionType%TYPE;

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

BEGIN

OPEN GenerateMonthlyStatements;

LOOP

FETCH GenerateMonthlyStatements INTO v\_customer\_id, v\_customer\_name, v\_transaction\_date, v\_amount, v\_transaction\_type;

EXIT WHEN GenerateMonthlyStatements%NOTFOUND;

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

IF v\_current\_customer\_id IS NOT NULL THEN

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

END IF;

v\_current\_customer\_id := v\_customer\_id;

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

DBMS\_OUTPUT.PUT\_LINE('Customer Name: ' || v\_customer\_name);

DBMS\_OUTPUT.PUT\_LINE('Transactions for the month:');

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

DBMS\_OUTPUT.PUT\_LINE('Date | Amount | Type');

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

END IF;

DBMS\_OUTPUT.PUT\_LINE(TO\_CHAR(v\_transaction\_date, 'YYYY-MM-DD') || ' | ' ||

TO\_CHAR(v\_amount, '99999.99') || ' | ' ||

v\_transaction\_type);

END LOOP;

IF v\_current\_customer\_id IS NOT NULL THEN

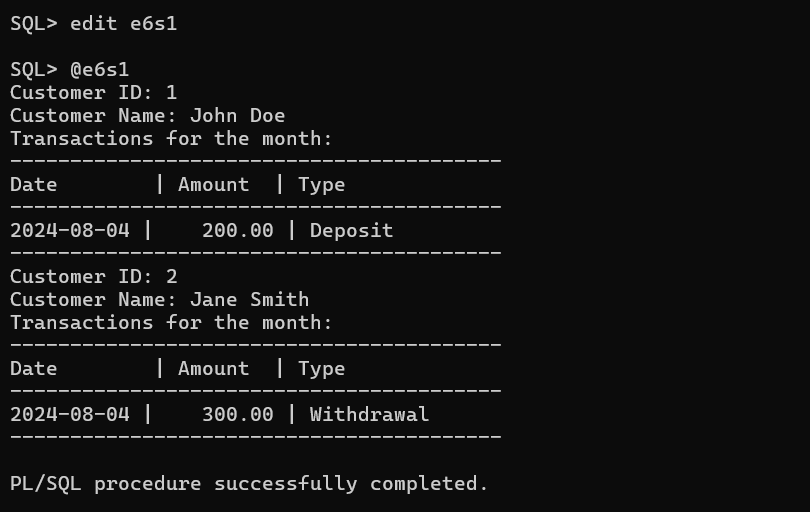
DBMS\_OUTPUT.PUT\_LINE('-----------------------------------------');

END IF;

CLOSE GenerateMonthlyStatements;

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.

SET SERVEROUTPUT ON;

DECLARE

CURSOR ApplyAnnualFee IS

SELECT AccountID, Balance

FROM Accounts

FOR UPDATE OF Balance;

v\_account\_id Accounts.AccountID%TYPE;

v\_balance Accounts.Balance%TYPE;

v\_annual\_fee NUMBER := 50; -- Example annual fee

BEGIN

OPEN ApplyAnnualFee;

LOOP

FETCH ApplyAnnualFee INTO v\_account\_id, v\_balance;

EXIT WHEN ApplyAnnualFee%NOTFOUND;

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

UPDATE Accounts

SET Balance = v\_balance,

LastModified = SYSDATE

WHERE CURRENT OF ApplyAnnualFee;

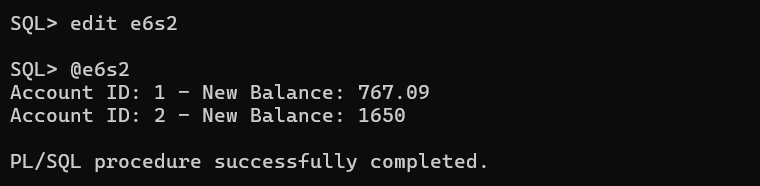
DBMS\_OUTPUT.PUT\_LINE('Account ID: ' || v\_account\_id || ' - New Balance: ' || v\_balance);

END LOOP;

CLOSE ApplyAnnualFee;

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.

SET SERVEROUTPUT ON;

DECLARE

CURSOR UpdateLoanInterestRates IS

SELECT LoanID, InterestRate

FROM Loans

FOR UPDATE OF InterestRate;

v\_loan\_id Loans.LoanID%TYPE;

v\_interest\_rate Loans.InterestRate%TYPE;

v\_new\_interest\_rate NUMBER;

BEGIN

OPEN UpdateLoanInterestRates;

LOOP

FETCH UpdateLoanInterestRates INTO v\_loan\_id, v\_interest\_rate;

EXIT WHEN UpdateLoanInterestRates%NOTFOUND;

v\_new\_interest\_rate := v\_interest\_rate + 1;

UPDATE Loans

SET InterestRate = v\_new\_interest\_rate

WHERE CURRENT OF UpdateLoanInterestRates;

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

END LOOP;

CLOSE UpdateLoanInterestRates;

END;

/

