**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.

**PL/SQL Procedure:**

SQL> SET SERVEROUTPUT ON;

SQL>

SQL> DECLARE

2 CURSOR GenerateMonthlyStatements IS

3 SELECT c.CustomerID, c.Name, t.TransactionID, t.TransactionDate, t.Amount, t.TransactionType

4 FROM Customers c

5 JOIN Accounts a ON c.CustomerID = a.CustomerID

6 JOIN Transactions t ON a.AccountID = t.AccountID

7 WHERE TRUNC(t.TransactionDate, 'MM') = TRUNC(SYSDATE, 'MM');

8

9 v\_CustomerID Customers.CustomerID%TYPE;

10 v\_Name Customers.Name%TYPE;

11 v\_TransactionID Transactions.TransactionID%TYPE;

12 v\_TransactionDate Transactions.TransactionDate%TYPE;

13 v\_Amount Transactions.Amount%TYPE;

14 v\_TransactionType Transactions.TransactionType%TYPE;

15

16 BEGIN

17 OPEN GenerateMonthlyStatements;

18 LOOP

19 FETCH GenerateMonthlyStatements INTO v\_CustomerID, v\_Name, v\_TransactionID, v\_TransactionDate, v\_Amount, v\_TransactionType;

20 EXIT WHEN GenerateMonthlyStatements%NOTFOUND;

21

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

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

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

25 DBMS\_OUTPUT.PUT\_LINE('Transaction Date: ' || v\_TransactionDate);

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

27 DBMS\_OUTPUT.PUT\_LINE('Transaction Type: ' || v\_TransactionType);

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

29 END LOOP;

30 CLOSE GenerateMonthlyStatements;

31 END;

32 /

**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.

**PL/SQL Procedure:**

SQL> DECLARE

2 CURSOR ApplyAnnualFee IS

3 SELECT AccountID, Balance

4 FROM Accounts;

5

6 v\_AccountID Accounts.AccountID%TYPE;

7 v\_Balance Accounts.Balance%TYPE;

8 v\_AnnualFee CONSTANT NUMBER := 50; -- Example annual fee amount

9

10 BEGIN

11 OPEN ApplyAnnualFee;

12 LOOP

13 FETCH ApplyAnnualFee INTO v\_AccountID, v\_Balance;

14 EXIT WHEN ApplyAnnualFee%NOTFOUND;

15

16 v\_Balance := v\_Balance - v\_AnnualFee;

17

18 UPDATE Accounts

19 SET Balance = v\_Balance, LastModified = SYSDATE

20 WHERE AccountID = v\_AccountID;

21 END LOOP;

22 CLOSE ApplyAnnualFee;

23

24 COMMIT;

25 END;

26 /

**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.

**PL/SQL Procedure:**

SQL> DECLARE

2 CURSOR UpdateLoanInterestRates IS

3 SELECT LoanID, InterestRate

4 FROM Loans;

5

6 v\_LoanID Loans.LoanID%TYPE;

7 v\_InterestRate Loans.InterestRate%TYPE;

8 v\_NewInterestRate CONSTANT NUMBER := 4.5; -- Example new interest rate

9

10 BEGIN

11 OPEN UpdateLoanInterestRates;

12 LOOP

13 FETCH UpdateLoanInterestRates INTO v\_LoanID, v\_InterestRate;

14 EXIT WHEN UpdateLoanInterestRates%NOTFOUND;

15

16 v\_InterestRate := v\_NewInterestRate;

17

18 UPDATE Loans

19 SET InterestRate = v\_InterestRate, StartDate = SYSDATE

20 WHERE LoanID = v\_LoanID;

21 END LOOP;

22 CLOSE UpdateLoanInterestRates;

23

24 COMMIT;

25 END;

26 /