

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 cur_transactions 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 t.TransactionDate BETWEEN TRUNC(SYSDATE, 'MM') AND
               LAST_DAY(SYSDATE);
    v_customerID Customers.CustomerID%TYPE;
    v_name Customers.Name%TYPE;
    v_transactionDate Transactions.TransactionDate%TYPE;
    v_amount Transactions.Amount%TYPE;
    v_transactionType Transactions.TransactionType%TYPE;
BEGIN
    OPEN cur_transactions;
    LOOP
        FETCH cur_transactions INTO v_customerID, v_name, v_transactionDate,
        v_amount, v_transactionType;
        EXIT WHEN cur_transactions%NOTFOUND;
        DBMS_OUTPUT.PUT_LINE('Customer: ' || v_name || ' (' || v_customerID ||
        ')');
        DBMS_OUTPUT.PUT_LINE('Transaction Date: ' || v_transactionDate);
        DBMS_OUTPUT.PUT_LINE('Amount: ' || v_amount || ' Type: ' ||
        v_transactionType);
        DBMS_OUTPUT.PUT_LINE('-----');
    END LOOP;
    CLOSE cur_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 cur_accounts IS
        SELECT AccountID, Balance
        FROM Accounts;
    v_accountID Accounts.AccountID%TYPE;
    v_balance Accounts.Balance%TYPE;
    v_annualFee CONSTANT NUMBER := 100;
BEGIN
    OPEN cur_accounts;
    LOOP
        FETCH cur_accounts INTO v_accountID, v_balance;
        EXIT WHEN cur_accounts%NOTFOUND;
        UPDATE Accounts
        SET Balance = Balance - v_annualFee
        WHERE AccountID = v_accountID;
        DBMS_OUTPUT.PUT_LINE('Account ID: ' || v_accountID || ' New Balance: ' ||
(v_balance - v_annualFee));
    END LOOP;
    CLOSE cur_accounts;
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 cur_loans IS
        SELECT LoanID, InterestRate
        FROM Loans;
    v_loanID Loans.LoanID%TYPE;
    v_interestRate Loans.InterestRate%TYPE;
    v_newInterestRate CONSTANT NUMBER := 5;
BEGIN
    OPEN cur_loans;
    LOOP
        FETCH cur_loans INTO v_loanID, v_interestRate;
        EXIT WHEN cur_loans%NOTFOUND;
        UPDATE Loans
        SET InterestRate = v_newInterestRate
        WHERE LoanID = v_loanID;
        DBMS_OUTPUT.PUT_LINE('Loan ID: ' || v_loanID || ' New Interest Rate: ' ||
v_newInterestRate);
    END LOOP;
    CLOSE cur_loans;
END;
```