**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.

**Procedure:**

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\_customerIDCustomers.CustomerID%TYPE;

v\_nameCustomers.Name%TYPE;

v\_transactionDateTransactions.TransactionDate%TYPE;

v\_amountTransactions.Amount%TYPE;

v\_transactionTypeTransactions.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;

**Output:**

Customer: John Doe (1)

Transaction Date: 06-AUG-24

Amount: 200 Type: Deposit

-----------------------------

Customer: Jane Smith (2)

Transaction Date: 06-AUG-24

Amount: 300 Type: Withdrawal

-----------------------------

Customer: Michael Johnson (3)

Transaction Date: 06-AUG-24

Amount: 400 Type: Deposit

-----------------------------

Customer: Emily Davis (4)

Transaction Date: 06-AUG-24

Amount: 500 Type: Withdrawal

-----------------------------

Customer: William Brown (5)

Transaction Date: 06-AUG-24

Amount: 600 Type: Deposit

-----------------------------

Customer: Sophia Wilson (6)

Transaction Date: 06-AUG-24

Amount: 700 Type: Withdrawal

-----------------------------

Customer: James Miller (7)

Transaction Date: 06-AUG-24

Amount: 800 Type: Deposit

-----------------------------

Customer: Olivia Taylor (8)

Transaction Date: 06-AUG-24

Amount: 900 Type: Withdrawal

-----------------------------

Customer: Liam Anderson (9)

Transaction Date: 06-AUG-24

Amount: 1000 Type: Deposit

-----------------------------

Customer: Ava Thomas (10)

Transaction Date: 06-AUG-24

Amount: 1100 Type: Withdrawal

-----------------------------

PL/SQL procedure successfully completed.

**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.

**Procedure:**

DECLARE

CURSOR cur\_accounts IS

SELECT AccountID, Balance

FROM Accounts;

v\_accountIDAccounts.AccountID%TYPE;

v\_balanceAccounts.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;

**Output:**

Account ID: 1 New Balance: 354.5

Account ID: 2 New Balance: 1900

Account ID: 3 New Balance: 1970

Account ID: 4 New Balance: 2400

Account ID: 5 New Balance: 2930

Account ID: 6 New Balance: 3400

Account ID: 7 New Balance: 3940

Account ID: 8 New Balance: 4400

Account ID: 9 New Balance: 4950

Account ID: 10 New Balance: 5400

**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.

**Procedure:**

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;

/

**Output:**

Loan ID: 1 New Interest Rate: 5

Loan ID: 2 New Interest Rate: 5

Loan ID: 3 New Interest Rate: 5

Loan ID: 4 New Interest Rate: 5

Loan ID: 5 New Interest Rate: 5

Loan ID: 6 New Interest Rate: 5

Loan ID: 7 New Interest Rate: 5

Loan ID: 8 New Interest Rate: 5

Loan ID: 9 New Interest Rate: 5

Loan ID: 10 New Interest Rate: 5