**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE GenerateMonthlyStatements()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE customerID INT;

DECLARE customerName VARCHAR(100);

DECLARE transactionAmount DECIMAL(10,2);

DECLARE transactionDate DATE;

DECLARE transactionCursor CURSOR FOR

SELECT c.CustomerID, c.Name, t.Amount, t.TransactionDate

FROM Customers c

JOIN Transactions t ON c.CustomerID = t.AccountID

WHERE MONTH(t.TransactionDate) = MONTH(CURDATE())

AND YEAR(t.TransactionDate) = YEAR(CURDATE());

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN transactionCursor;

read\_loop: LOOP

FETCH transactionCursor INTO customerID, customerName, transactionAmount, transactionDate;

IF done THEN

LEAVE read\_loop;

END IF;

SELECT CONCAT('Customer ID: ', customerID,

', Name: ', customerName,

', Amount: ', transactionAmount,

', Date: ', transactionDate) AS Statement;

END LOOP;

CLOSE transactionCursor;

END //

DELIMITER ;

**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE ApplyAnnualFee()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE accountID INT;

DECLARE accountBalance DECIMAL(10,2);

DECLARE annualFee DECIMAL(10,2) DEFAULT 50;

DECLARE accountCursor CURSOR FOR

SELECT AccountID, Balance FROM Accounts;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN accountCursor;

read\_loop: LOOP

FETCH accountCursor INTO accountID, accountBalance;

IF done THEN

LEAVE read\_loop;

END IF;

UPDATE Accounts

SET Balance = Balance - annualFee

WHERE AccountID = accountID;

END LOOP;

CLOSE accountCursor;

END //

DELIMITER ;

**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.

**Answer:**

DELIMITER //

CREATE PROCEDURE UpdateLoanInterestRates(newInterestRate DECIMAL(5,2))

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE loanID INT;

DECLARE loanCursor CURSOR FOR

SELECT LoanID FROM Loans;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN loanCursor;

read\_loop: LOOP

FETCH loanCursor INTO loanID;

IF done THEN

LEAVE read\_loop;

END IF;

UPDATE Loans

SET InterestRate = newInterestRate

WHERE LoanID = loanID;

END LOOP;

CLOSE loanCursor;

END //

DELIMITER ;