**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:**

DELIMITER //

CREATE PROCEDURE GenerateMonthlyStatements()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE customer\_id INT;

DECLARE customer\_name VARCHAR(100);

DECLARE trans\_date DATE;

DECLARE amount DECIMAL(10, 2);

DECLARE trans\_type VARCHAR(10);

DECLARE cur CURSOR FOR

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

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

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO customer\_id, customer\_name, trans\_date, amount, trans\_type;

IF done THEN

LEAVE read\_loop;

END IF;

SELECT CONCAT('Customer: ', customer\_name, ', Date: ', trans\_date, ', Amount: ', amount, ', Type: ', trans\_type) AS Statement;

END LOOP;

CLOSE cur;

END;

//

DELIMITER ;

**To execute and test this procedure:**

CALL GenerateMonthlyStatements();

**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:**

DELIMITER //

CREATE PROCEDURE ApplyAnnualFee()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE acc\_id INT;

DECLARE acc\_balance DECIMAL(10, 2);

DECLARE fee DECIMAL(10, 2) DEFAULT 50.00; -- Annual fee

DECLARE cur CURSOR FOR

SELECT AccountID, Balance FROM Accounts;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO acc\_id, acc\_balance;

IF done THEN

LEAVE read\_loop;

END IF;

-- Deduct the fee

UPDATE Accounts SET Balance = Balance - fee WHERE AccountID = acc\_id;

END LOOP;

CLOSE cur;

END;

//

DELIMITER ;

**To execute and test this procedure:**

CALL ApplyAnnualFee();

**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:**

DELIMITER //

CREATE PROCEDURE UpdateLoanInterestRates()

BEGIN

DECLARE done INT DEFAULT 0;

DECLARE loan\_id INT;

DECLARE new\_interest\_rate DECIMAL(5, 2);

DECLARE cur CURSOR FOR

SELECT LoanID FROM Loans;

DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;

OPEN cur;

read\_loop: LOOP

FETCH cur INTO loan\_id;

IF done THEN

LEAVE read\_loop;

END IF;

-- Assuming new policy sets interest rate to 4.5%

SET new\_interest\_rate = 4.5;

-- Update the interest rate

UPDATE Loans SET InterestRate = new\_interest\_rate WHERE LoanID = loan\_id;

END LOOP;

CLOSE cur;

END;

//

DELIMITER ;

**To execute and test this procedure:**

CALL UpdateLoanInterestRates();