**Exercise 1: Control Structures**

Scenario 1: Apply 1% Discount for Customers Above 60

BEGIN

FOR cust\_rec IN (SELECT CustomerID, Age FROM Customers) LOOP

IF cust\_rec.Age > 60 THEN

UPDATE Loans

SET InterestRate = InterestRate - 1

WHERE CustomerID = cust\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

Scenario 2: Set IsVIP for Customers with Balance > $10,000

BEGIN

FOR cust\_rec IN (SELECT CustomerID, Balance FROM Customers) LOOP

IF cust\_rec.Balance > 10000 THEN

UPDATE Customers

SET IsVIP = 'TRUE'

WHERE CustomerID = cust\_rec.CustomerID;

END IF;

END LOOP;

COMMIT;

END;

Scenario 3: Send Reminders for Loans Due in Next 30 Days

DECLARE

v\_due\_date DATE;

v\_customer\_name VARCHAR2(100);

BEGIN

FOR loan\_rec IN (

SELECT L.LoanID, L.DueDate, C.CustomerName

FROM Loans L

JOIN Customers C ON L.CustomerID = C.CustomerID

WHERE L.DueDate BETWEEN SYSDATE AND SYSDATE + 30

) LOOP

DBMS\_OUTPUT.PUT\_LINE('Reminder: Loan ID ' || loan\_rec.LoanID ||

' for customer ' || loan\_rec.CustomerName ||

' is due on ' || TO\_CHAR(loan\_rec.DueDate, 'DD-MON-YYYY'));

END LOOP;

END;

**Exercise 3: Stored Procedures**

Scenario 1: ProcessMonthlyInterest – Apply 1% Interest to Savings Accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

-- Apply 1% interest to all Savings accounts

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'SAVINGS';

COMMIT;

END;

Scenario 2: UpdateEmployeeBonus – Add Bonus to Employees in a Department

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_DepartmentID IN NUMBER,

p\_BonusPercent IN NUMBER

) AS

BEGIN

-- Update salary with bonus

UPDATE Employees

SET Salary = Salary + (Salary \* p\_BonusPercent / 100)

WHERE DepartmentID = p\_DepartmentID;

COMMIT;

END;

Scenario 3: TransferFunds – Transfer Funds Between Two Accounts

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_FromAccID IN NUMBER,

p\_ToAccID IN NUMBER,

p\_Amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Check if sender has sufficient balance

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_FromAccID;

IF v\_balance < p\_Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds for transfer.');

END IF;

-- Deduct from sender

UPDATE Accounts

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccID;

-- Credit to receiver

UPDATE Accounts

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccID;

COMMIT;

END;