**Exercise 3: Stored Procedures**

**Scenario 1: Process Monthly Interest**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

UPDATE Accounts

SET Balance = Balance \* 1.01,

LastModified = SYSDATE

WHERE AccountType = 'Savings';

END;

**Scenario 2: Update Employee Bonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_Department VARCHAR2,

p\_BonusPercentage NUMBER

) AS

BEGIN

UPDATE Employees

SET Salary = Salary \* (1 + p\_BonusPercentage / 100)

WHERE Department = p\_Department;

END;

**Scenario 3: Transfer Funds**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID NUMBER,

p\_ToAccountID NUMBER,

p\_Amount NUMBER

) AS

v\_FromAccountBalance NUMBER;

BEGIN

-- Check the balance of the source account

SELECT Balance INTO v\_FromAccountBalance

FROM Accounts

WHERE AccountID = p\_FromAccountID;

-- Check if the source account has sufficient balance

IF v\_FromAccountBalance >= p\_Amount THEN

-- Deduct the amount from the source account

UPDATE Accounts

SET Balance = Balance - p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_FromAccountID;

-- Add the amount to the destination account

UPDATE Accounts

SET Balance = Balance + p\_Amount,

LastModified = SYSDATE

WHERE AccountID = p\_ToAccountID;

-- Insert transaction record for the source account

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (TRANSACTIONS\_SEQ.NEXTVAL, p\_FromAccountID, SYSDATE, p\_Amount, 'Transfer Out');

-- Insert transaction record for the destination account

INSERT INTO Transactions (TransactionID, AccountID, TransactionDate, Amount, TransactionType)

VALUES (TRANSACTIONS\_SEQ.NEXTVAL, p\_ToAccountID, SYSDATE, p\_Amount, 'Transfer In');

ELSE

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance in the source account');

END IF;

END;