**Scenario 1: Process Monthly Interest for Savings Accounts**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

v\_InterestRate CONSTANT NUMBER := 0.01;

BEGIN

-- Update balance of all savings accounts by applying the interest rate

UPDATE Accounts

SET Balance = Balance \* (1 + v\_InterestRate)

WHERE AccountType = 'Savings';

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error message

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

/

**Scenario 2: Update Employee Bonus Based on Performance**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_Department IN Employees.Department%TYPE,

p\_BonusPercentage IN NUMBER

) AS

BEGIN

-- Update salary by adding the bonus percentage

UPDATE Employees

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

WHERE Department = p\_Department;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error message

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

/

**Scenario 3: Transfer Funds Between Customer Accounts**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN Accounts.AccountID%TYPE,

p\_ToAccountID IN Accounts.AccountID%TYPE,

p\_Amount IN NUMBER

) AS

v\_FromBalance Accounts.Balance%TYPE;

BEGIN

-- Retrieve the current balance of the source account

SELECT Balance INTO v\_FromBalance

FROM Accounts

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

-- Check if the source account has sufficient funds

IF v\_FromBalance < p\_Amount THEN

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

ELSE

-- Deduct the amount from the source account

UPDATE Accounts

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

-- Add the amount to the destination account

UPDATE Accounts

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

-- Log the successful transfer

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

VALUES (Transactions\_SEQ.NEXTVAL, p\_FromAccountID, SYSDATE, -p\_Amount, 'Transfer');

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

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

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

-- Log the error message

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

ROLLBACK;

END;

/