-- Tables for all scenarios

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER

);

CREATE TABLE Employees (

EmployeeID NUMBER PRIMARY KEY,

Name VARCHAR2(100),

DepartmentID NUMBER,

Salary NUMBER

);

-- Sample Accounts

INSERT INTO Accounts VALUES (1, 101, 'Savings', 10000);

INSERT INTO Accounts VALUES (2, 101, 'Current', 5000);

INSERT INTO Accounts VALUES (3, 102, 'Savings', 20000);

-- Sample Employees

INSERT INTO Employees VALUES (1, 'Alice', 10, 60000);

INSERT INTO Employees VALUES (2, 'Bob', 10, 55000);

INSERT INTO Employees VALUES (3, 'Charlie', 20, 50000);

COMMIT;

**SCENARIO 1: The bank needs to process monthly interest for all savings accounts.**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance + (Balance \* 0.01)

WHERE AccountType = 'Savings';

COMMIT;

END;

/

**SCENARIO 2: The bank wants to implement a bonus scheme for employees based on their performance.**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_DepartmentID IN NUMBER,

p\_BonusPercent IN NUMBER

) IS

BEGIN

UPDATE Employees

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

WHERE DepartmentID = p\_DepartmentID;

COMMIT;

END;

/

**SCENARIO 3: Customers should be able to transfer funds between their accounts.**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_FromAccountID IN NUMBER,

p\_ToAccountID IN NUMBER,

p\_Amount IN NUMBER

) IS

v\_Balance NUMBER;

BEGIN

-- Get the balance of the source account

SELECT Balance INTO v\_Balance

FROM Accounts

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

-- Check if sufficient balance

IF v\_Balance >= p\_Amount THEN

-- Deduct from source

UPDATE Accounts

SET Balance = Balance - p\_Amount

WHERE AccountID = p\_FromAccountID;

-- Add to destination

UPDATE Accounts

SET Balance = Balance + p\_Amount

WHERE AccountID = p\_ToAccountID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer successful.');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Insufficient balance. Transfer failed.');

END IF;

END;

/

BEGIN

ProcessMonthlyInterest;

END;

/

BEGIN

UpdateEmployeeBonus(p\_DepartmentID => 10, p\_BonusPercent => 5);

END;

/

BEGIN

TransferFunds(p\_FromAccountID => 1, p\_ToAccountID => 2, p\_Amount => 2000);

END;

/

OUTPUT:



