Scenario 1: ProcessMonthlyInterest

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

FOR account IN (SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings') LOOP

UPDATE Accounts

SET Balance = Balance + (account.Balance \* 0.01),

LastModified = SYSDATE

WHERE AccountID = account.AccountID;

END LOOP;

COMMIT;

END ProcessMonthlyInterest;

/

Scenario 2: UpdateEmployeeBonus

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_Department VARCHAR2,

p\_BonusPercent NUMBER

) AS

BEGIN

FOR employee IN (SELECT EmployeeID, Salary

FROM Employees

WHERE Department = p\_Department) LOOP

UPDATE Employees

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

HireDate = SYSDATE

WHERE EmployeeID = employee.EmployeeID;

END LOOP;

COMMIT;

END UpdateEmployeeBonus;

/

Scenario 3: TransferFunds

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_FromAccountID NUMBER,

p\_ToAccountID NUMBER,

p\_Amount NUMBER

) AS

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance

FROM Accounts

WHERE AccountID = p\_FromAccountID

FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

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

END IF;

UPDATE Accounts

SET Balance = Balance - p\_Amount, LastModified = SYSDATE

WHERE AccountID = p\_FromAccountID;

UPDATE Accounts

SET Balance = Balance + p\_Amount, LastModified = SYSDATE

WHERE AccountID = p\_ToAccountID;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer successful from Account ' || p\_FromAccountID || ' to Account ' || p\_ToAccountID || ' of amount ' || p\_Amount);

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: ' || SQLERRM);

END TransferFunds;

/