**Exercise 3: Stored Procedures**

**Scenario 1: ProcessMonthlyInterest for Savings Accounts**

PL/SQL Implementation:

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest

IS

BEGIN

FOR acc IN (

SELECT AccountID, Balance

FROM Accounts

WHERE AccountType = 'Savings'

)

LOOP

UPDATE Accounts

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

LastModified = SYSDATE

WHERE AccountID = acc.AccountID;

END LOOP;

COMMIT;

END ProcessMonthlyInterest;

/

**Scenario 2: UpdateEmployeeBonus by Department**

PL/SQL Implementation:

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN Employees.DepartmentID%TYPE,

p\_bonus\_percentage IN NUMBER

)

IS

BEGIN

FOR emp IN (

SELECT EmployeeID, Salary

FROM Employees

WHERE DepartmentID = p\_department\_id

)

LOOP

UPDATE Employees

SET Salary = Salary + (emp.Salary \* p\_bonus\_percentage / 100)

WHERE EmployeeID = emp.EmployeeID;

END LOOP;

COMMIT;

END UpdateEmployeeBonus;

/

**Scenario 3: TransferFunds Between Accounts**

PL/SQL Implementation:

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account\_id IN Accounts.AccountID%TYPE,

p\_to\_account\_id IN Accounts.AccountID%TYPE,

p\_amount IN NUMBER

)

IS

v\_from\_balance Accounts.Balance%TYPE;

BEGIN

SELECT Balance INTO v\_from\_balance

FROM Accounts

WHERE AccountID = p\_from\_account\_id

FOR UPDATE;

UPDATE Accounts

SET Balance = Balance - p\_amount,

LastModified = SYSDATE

WHERE AccountID = p\_from\_account\_id;

UPDATE Accounts

SET Balance = Balance + p\_amount,

LastModified = SYSDATE

WHERE AccountID = p\_to\_account\_id;

COMMIT;

END TransferFunds;

/