EXERCISE 3

Scenario 1: Process monthly interest for all savings accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest IS

BEGIN

UPDATE Accounts

SET Balance = Balance \* 1.01

WHERE AccountType = 'Savings';

COMMIT;

END;

/

Scenario 2: Implement a bonus scheme for employees based on their performance

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(p\_Department IN VARCHAR2, p\_BonusPercentage IN NUMBER) IS

BEGIN

UPDATE Employees

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

WHERE Department = p\_Department;

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

insufficient\_funds EXCEPTION;

BEGIN

DECLARE

v\_FromBalance NUMBER;

BEGIN

SELECT Balance INTO v\_FromBalance FROM Accounts WHERE AccountID = p\_FromAccountID FOR UPDATE;

IF v\_FromBalance < p\_Amount THEN

RAISE insufficient\_funds;

END IF;

UPDATE Accounts SET Balance = Balance - p\_Amount WHERE AccountID = p\_FromAccountID;

UPDATE Accounts SET Balance = Balance + p\_Amount WHERE AccountID = p\_ToAccountID;

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient funds.');

WHEN OTHERS THEN

ROLLBACK;

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

END;

END;

/