Scenario 1: Process Monthly Interest for All Savings Accounts

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

BEGIN

-- Update the balance of all savings accounts by applying a 1% interest rate

UPDATE accounts

SET balance = balance \* 1.01

WHERE account\_type = 'Savings';

-- Commit the transaction

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Monthly interest applied to all savings accounts.');

END ProcessMonthlyInterest;

/

Scenario 2: Implement a Bonus Scheme for Employees Based on Performance

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_department\_id IN NUMBER,

p\_bonus\_percentage IN NUMBER

) AS

BEGIN

-- Update the salary of employees in the specified department by adding the bonus percentage

UPDATE employees

SET salary = salary \* (1 + p\_bonus\_percentage / 100)

WHERE department\_id = p\_department\_id;

-- Commit the transaction

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Bonus applied to employees in department ID: ' || p\_department\_id);

END UpdateEmployeeBonus;

Scenario 3: Transfer Funds Between Accounts

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

-- Check if the source account has sufficient balance

SELECT balance INTO v\_balance

FROM accounts

WHERE account\_id = p\_from\_account\_id

FOR UPDATE;

IF v\_balance < p\_amount THEN

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

END IF;

-- Deduct the amount from the source account

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account\_id;

-- Add the amount to the destination account

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account\_id;

-- Commit the transaction

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transferred ' || p\_amount || ' from account ' || p\_from\_account\_id || ' to account ' || p\_to\_account\_id);

END TransferFunds;

/