**2. Stored Procedure**

**Create the table**

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest (

p\_interest\_rate IN NUMBER DEFAULT 0.01 -- 1 % = 0.01

) AS

BEGIN

/\* One UPDATE does the whole job \*/

UPDATE savings\_accounts

SET balance = balance \* (1 + p\_interest\_rate);

DBMS\_OUTPUT.PUT\_LINE(SQL%ROWCOUNT || ' savings accounts updated.');

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END ProcessMonthlyInterest;

/

**2. Update Employee Bonus**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus (

p\_department\_id IN employees.department\_id%TYPE,

p\_bonus\_pct IN NUMBER -- e.g. 5 = +5 %

) AS

BEGIN

UPDATE employees

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

WHERE department\_id = p\_department\_id;

DBMS\_OUTPUT.PUT\_LINE(

SQL%ROWCOUNT || ' employee(s) in department ' || p\_department\_id ||

' received a ' || p\_bonus\_pct || '% bonus.'

);

COMMIT;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END UpdateEmployeeBonus;

/

**3. Transfer Funds**

CREATE OR REPLACE PROCEDURE TransferFunds (

p\_from\_account IN accounts.account\_id%TYPE,

p\_to\_account IN accounts.account\_id%TYPE,

p\_amount IN NUMBER

) AS

e\_insufficient\_funds EXCEPTION;

PRAGMA EXCEPTION\_INIT(e\_insufficient\_funds, -20001);

v\_from\_balance NUMBER;

v\_to\_balance NUMBER;

BEGIN

IF p\_amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Transfer amount must be positive.');

END IF;

/\* Lock both rows in consistent order to avoid deadlocks \*/

SELECT balance

INTO v\_from\_balance

FROM accounts

WHERE account\_id = p\_from\_account

FOR UPDATE;

SELECT balance

INTO v\_to\_balance

FROM accounts

WHERE account\_id = p\_to\_account

FOR UPDATE;

IF v\_from\_balance < p\_amount THEN

RAISE e\_insufficient\_funds;

END IF;

/\* Apply the transfer \*/

UPDATE accounts

SET balance = balance - p\_amount

WHERE account\_id = p\_from\_account;

UPDATE accounts

SET balance = balance + p\_amount

WHERE account\_id = p\_to\_account;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE(

'Transferred ' || TO\_CHAR(p\_amount, 'FM999G999D99') ||

' from account #' || p\_from\_account ||

' to account #' || p\_to\_account || '.'

);

EXCEPTION

WHEN e\_insufficient\_funds THEN

ROLLBACK;

RAISE\_APPLICATION\_ERROR(

-20001,

'Account #' || p\_from\_account || ' has insufficient balance.'

);

WHEN NO\_DATA\_FOUND THEN

ROLLBACK;

RAISE\_APPLICATION\_ERROR(

-20003,

'One or both account numbers are invalid.'

);

WHEN OTHERS THEN

ROLLBACK;

RAISE;

END TransferFunds;

/