**Exercise 3: Stored Procedures**

**Scenario 1: The bank needs to process monthly interest for all savings accounts**.

CREATE OR REPLACE PROCEDURE ProcessMonthlyInterest AS

v\_interest\_rate NUMBER := 0.01;

BEGIN

FOR acc\_rec IN (SELECT account\_id, balance FROM accounts WHERE account\_type = 'SAVINGS') LOOP

UPDATE accounts SET balance = balance + (balance \* v\_interest\_rate)

WHERE account\_id = acc\_rec.account\_id;

DBMS\_OUTPUT.PUT\_LINE('Applied interest to account ' || acc\_rec.account\_id ||

'. New balance: ' || (acc\_rec.balance \* (1 + v\_interest\_rate)));

END LOOP;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

END ProcessMonthlyInterest;

/

**Scenario 2: The bank wants to implement a bonus scheme for employees based on their performance.**

CREATE OR REPLACE PROCEDURE UpdateEmployeeBonus(

p\_dept\_name IN VARCHAR2,

p\_bonus\_percent IN NUMBER

) AS

BEGIN

IF p\_bonus\_percent < 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Error: Bonus percentage must be greater than 0');

RETURN;

END IF;

FOR emp\_rec IN (SELECT employee\_id, name, salary, performance\_rating FROM employees

WHERE department = p\_dept\_name) LOOP

DECLARE

v\_bonus\_amount NUMBER;

BEGIN

v\_bonus\_amount := emp\_rec.salary \* (p\_bonus\_percent/100);

UPDATE employees

SET salary = salary + v\_bonus\_amount

WHERE employee\_id = emp\_rec.employee\_id;

DBMS\_OUTPUT.PUT\_LINE('Added bonus of ' || v\_bonus\_amount ||

' to ' || emp\_rec.name || ')');

END;

END LOOP;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

END UpdateEmployeeBonus;

/

**Scenario 3: Customers should be able to transfer funds between their accounts.**

CREATE OR REPLACE PROCEDURE TransferFunds(

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) AS

v\_from\_balance NUMBER;

v\_to\_account\_exists NUMBER;

v\_transaction\_id NUMBER;

BEGIN

IF p\_amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Transfer amount must be positive');

END IF;

SELECT COUNT(\*) INTO v\_to\_account\_exists FROM accounts

WHERE account\_id = p\_to\_account;

IF v\_to\_account\_exists = 0 THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Destination account does not exist');

END IF;

IF v\_from\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20005, 'Insufficient balance in source account');

END IF;

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('Successfully transferred ' || p\_amount ||

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

' to account ' || p\_to\_account);

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

RAISE;

END TransferFunds;

/