**Exercise 2: Error Handling**

Scenario 1: Fund Transfers with Error Handling

CREATE OR REPLACE PROCEDURE SafeTransferFunds(

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) AS

BEGIN

UPDATE accounts

SET balance = CASE

WHEN account\_id = p\_from\_account\_id THEN balance - p\_amount

WHEN account\_id = p\_to\_account\_id THEN balance + p\_amount

ELSE balance

END

WHERE account\_id IN (p\_from\_account\_id, p\_to\_account\_id);

IF SQL%ROWCOUNT < 2 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Invalid accounts or insufficient funds.');

END IF;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Transfer successful.');

EXCEPTION

WHEN OTHERS THEN

ROLLBACK;

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

END SafeTransferFunds;

/

Scenario 2: Updating Employee Salaries with Error Handling

CREATE OR REPLACE PROCEDURE UpdateSalary(

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) AS

BEGIN

UPDATE employees

SET salary = salary + salary \* p\_percentage / 100

WHERE employee\_id = p\_employee\_id;

IF SQL%ROWCOUNT = 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Employee ID not found.');

END IF;

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Salary updated.');

EXCEPTION

WHEN OTHERS THEN

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

END UpdateSalary;

/

Scenario 3: Adding a New Customer with Error Handling

CREATE OR REPLACE PROCEDURE AddNewCustomer(

p\_customer\_id IN NUMBER,

p\_customer\_name IN VARCHAR2,

p\_balance IN NUMBER

) AS

BEGIN

INSERT INTO customers (customer\_id, customer\_name, balance)

VALUES (p\_customer\_id, p\_customer\_name, p\_balance);

COMMIT;

DBMS\_OUTPUT.PUT\_LINE('Customer added.');

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

DBMS\_OUTPUT.PUT\_LINE('Customer ID already exists.');

WHEN OTHERS THEN

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

END AddNewCustomer;

/