**Exercise 2: Error Handling**

Scenario 1:

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

insufficient\_funds EXCEPTION;

insufficient\_funds\_message VARCHAR2(100);

BEGIN

DECLARE

from\_balance NUMBER;

BEGIN

SELECT balance INTO from\_balance FROM accounts WHERE account\_id = p\_from\_account\_id;

IF from\_balance < p\_amount THEN

insufficient\_funds\_message := 'Insufficient funds in account ' || p\_from\_account\_id;

RAISE insufficient\_funds;

END IF;

END;

UPDATE accounts SET balance = balance - p\_amount WHERE account\_id = p\_from\_account\_id;

UPDATE accounts SET balance = balance + p\_amount WHERE account\_id = p\_to\_account\_id;

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE(insufficient\_funds\_message);

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('An error occurred during fund transfer: ' || SQLERRM);

END SafeTransferFunds;

/

Scenario 2:

CREATE OR REPLACE PROCEDURE UpdateSalary (

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) IS

employee\_not\_found EXCEPTION;

employee\_not\_found\_message VARCHAR2(100);

BEGIN

UPDATE employees

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

WHERE employee\_id = p\_employee\_id;

IF SQL%NOTFOUND THEN

employee\_not\_found\_message := 'Employee with ID ' || p\_employee\_id || ' not found';

RAISE employee\_not\_found;

END IF;

COMMIT;

EXCEPTION

WHEN employee\_not\_found THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE(employee\_not\_found\_message);

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('An error occurred during salary update: ' || SQLERRM);

END UpdateSalary;

/

Scenario 3:

CREATE OR REPLACE PROCEDURE AddNewCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

) IS

customer\_already\_exists EXCEPTION;

customer\_already\_exists\_message VARCHAR2(100);

BEGIN

INSERT INTO customers (customerid, name, dob, balance, lastmodified)

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

COMMIT;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

ROLLBACK;

customer\_already\_exists\_message := 'Customer with ID ' || p\_customer\_id || ' already exists';

DBMS\_OUTPUT.PUT\_LINE(customer\_already\_exists\_message);

WHEN OTHERS THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('An error occurred during customer addition: ' || SQLERRM);

END AddNewCustomer;

/