**Exercise 2: Error Handling**

**Step 1: Safe Transfer Funds**

CREATE TABLE error\_log (

error\_id NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

error\_message VARCHAR2(4000),

error\_date DATE

);

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_from\_account\_id IN NUMBER,

p\_to\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

e\_insufficient\_funds EXCEPTION;

v\_from\_balance NUMBER;

BEGIN

-- Check if the source account has sufficient funds

SELECT balance INTO v\_from\_balance

FROM accounts

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

FOR UPDATE;

IF v\_from\_balance IS NULL THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Account not found or balance is null');

END IF;

IF v\_from\_balance < p\_amount THEN

RAISE e\_insufficient\_funds;

END IF;

-- Perform the transfer

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 e\_insufficient\_funds THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES ('Insufficient funds in account ' || p\_from\_account\_id, SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient funds in account ' || p\_from\_account\_id);

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES (SQLERRM, SYSDATE);

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

END SafeTransferFunds;

/

**Step 2: Update Salary**

CREATE OR REPLACE PROCEDURE UpdateSalary (

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) IS

e\_employee\_not\_found EXCEPTION;

BEGIN

-- Attempt to update the employee's salary

UPDATE employees

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

WHERE employee\_id = p\_employee\_id;

IF SQL%ROWCOUNT = 0 THEN

RAISE e\_employee\_not\_found;

END IF;

COMMIT;

EXCEPTION

WHEN e\_employee\_not\_found THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES ('Employee ID ' || p\_employee\_id || ' not found', SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('Error: Employee ID ' || p\_employee\_id || ' not found');

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES (SQLERRM, SYSDATE);

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

END UpdateSalary;

/

**Step 3: Add New Customer**

CREATE OR REPLACE PROCEDURE AddNewCustomer (

p\_customer\_id IN NUMBER,

p\_customer\_name IN VARCHAR2,

p\_age IN NUMBER,

p\_balance IN NUMBER,

p\_loan\_interest\_rate IN NUMBER

) IS

BEGIN

-- Attempt to insert a new customer

INSERT INTO customers (customer\_id, customer\_name, age, balance, loan\_interest\_rate)

VALUES (p\_customer\_id, p\_customer\_name, p\_age, p\_balance, p\_loan\_interest\_rate);

COMMIT;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES ('Customer ID ' || p\_customer\_id || ' already exists', SYSDATE);

DBMS\_OUTPUT.PUT\_LINE('Error: Customer ID ' || p\_customer\_id || ' already exists');

WHEN OTHERS THEN

ROLLBACK;

INSERT INTO error\_log (error\_message, error\_date)

VALUES (SQLERRM, SYSDATE);

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

END AddNewCustomer;

/

**Example Data Insertion Scripts**

-- Example tables

CREATE TABLE accounts (

account\_id NUMBER PRIMARY KEY,

balance NUMBER

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

salary NUMBER

);

CREATE TABLE customers (

customer\_id NUMBER PRIMARY KEY,

customer\_name VARCHAR2(100),

age NUMBER,

balance NUMBER,

loan\_interest\_rate NUMBER

);

-- Insert example data

INSERT INTO accounts (account\_id, balance) VALUES (1, 5000);

INSERT INTO accounts (account\_id, balance) VALUES (2, 15000);

INSERT INTO employees (employee\_id, salary) VALUES (1, 50000);

INSERT INTO employees (employee\_id, salary) VALUES (2, 60000);

INSERT INTO customers (customer\_id, customer\_name, age, balance, loan\_interest\_rate) VALUES (1, 'John Doe', 65, 5000, 5);

INSERT INTO customers (customer\_id, customer\_name, age, balance, loan\_interest\_rate) VALUES (2, 'Jane Smith', 55, 15000, 6);

INSERT INTO customers (customer\_id, customer\_name, age, balance, loan\_interest\_rate) VALUES (3, 'Alice Johnson', 70, 20000, 4);

COMMIT;