**Exercise 2: Error Handling**

**Scenario 1:** Handle exceptions during fund transfers between accounts.

**Code :**

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

from\_account\_identifier IN ACCOUNTS.ACCOUNT\_ID%TYPE,

to\_account\_identifier IN ACCOUNTS.ACCOUNT\_ID%TYPE,

transfer\_amount IN NUMBER

) IS

from\_account\_balance ACCOUNTS.BALANCE%TYPE;

to\_account\_balance ACCOUNTS.BALANCE%TYPE;

-- Custom exception for insufficient funds

insufficient\_funds EXCEPTION;

PRAGMA EXCEPTION\_INIT(insufficient\_funds, -20001);

-- Procedure to log errors

PROCEDURE LogError(error\_message IN VARCHAR2) IS

BEGIN

INSERT INTO ERROR\_LOG (LOG\_ID, ERROR\_MESSAGE, LOG\_DATE)

VALUES (ERROR\_LOG\_SEQ.NEXTVAL, error\_message, SYSDATE);

END LogError;

BEGIN

-- Lock the rows for the two accounts to prevent concurrent updates

SELECT BALANCE INTO from\_account\_balance

FROM ACCOUNTS

WHERE ACCOUNT\_ID = from\_account\_identifier

FOR UPDATE;

SELECT BALANCE INTO to\_account\_balance

FROM ACCOUNTS

WHERE ACCOUNT\_ID = to\_account\_identifier

FOR UPDATE;

-- Check if the from account has sufficient funds

IF from\_account\_balance < transfer\_amount THEN

RAISE insufficient\_funds;

END IF;

-- Perform the fund transfer

UPDATE ACCOUNTS

SET BALANCE = BALANCE - transfer\_amount

WHERE ACCOUNT\_ID = from\_account\_identifier;

UPDATE ACCOUNTS

SET BALANCE = BALANCE + transfer\_amount

WHERE ACCOUNT\_ID = to\_account\_identifier;

-- Commit the transaction

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

-- Log insufficient funds error

LogError('Insufficient funds in account ' || from\_account\_identifier);

ROLLBACK;

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds in account ' || from\_account\_identifier);

WHEN OTHERS THEN

-- Log any other errors

LogError(SQLERRM);

ROLLBACK;

RAISE;

END SafeTransferFunds;

**Scenario 2:** Manage errors when updating employee salaries.

**CODE :**

CREATE OR REPLACE PROCEDURE UpdateSalary (

employee\_identifier IN EMPLOYEES.EMPLOYEE\_ID%TYPE,

percentage\_increase IN NUMBER

) IS

current\_salary EMPLOYEES.SALARY%TYPE;

-- Procedure to log errors

PROCEDURE LogError(error\_message IN VARCHAR2) IS

BEGIN

INSERT INTO ERROR\_LOG (LOG\_ID, ERROR\_MESSAGE, LOG\_DATE)

VALUES (ERROR\_LOG\_SEQ.NEXTVAL, error\_message, SYSDATE);

END LogError;

BEGIN

-- Select the current salary of the employee

SELECT SALARY INTO current\_salary

FROM EMPLOYEES

WHERE EMPLOYEE\_ID = employee\_identifier;

-- Update the salary by the given percentage

UPDATE EMPLOYEES

SET SALARY = SALARY \* (1 + percentage\_increase / 100)

WHERE EMPLOYEE\_ID = employee\_identifier;

-- Commit the transaction

COMMIT;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

-- Log error if employee ID does not exist

LogError('Employee ID ' || employee\_identifier || ' does not exist.');

RAISE\_APPLICATION\_ERROR(-20001, 'Employee ID ' || employee\_identifier || ' does not exist.');

WHEN OTHERS THEN

-- Log any other errors

LogError(SQLERRM);

ROLLBACK;

RAISE;

END UpdateSalary;

**Scenario 3:** Ensure data integrity when adding a new customer.

**CODE :**

CREATE OR REPLACE PROCEDURE AddNewCustomer (

new\_customer\_id IN CUSTOMERS.CUSTOMER\_ID%TYPE,

customer\_full\_name IN CUSTOMERS.CUSTOMER\_NAME%TYPE,

customer\_birth\_date IN CUSTOMERS.DATE\_OF\_BIRTH%TYPE

) IS

-- Procedure to log errors

PROCEDURE LogError(error\_message IN VARCHAR2) IS

BEGIN

INSERT INTO ERROR\_LOG (LOG\_ID, ERROR\_MESSAGE, LOG\_DATE)

VALUES (ERROR\_LOG\_SEQ.NEXTVAL, error\_message, SYSDATE);

END LogError;

BEGIN

-- Attempt to insert the new customer

INSERT INTO CUSTOMERS (CUSTOMER\_ID, CUSTOMER\_NAME, DATE\_OF\_BIRTH)

VALUES (new\_customer\_id, customer\_full\_name, customer\_birth\_date);

-- Commit the transaction

COMMIT;

EXCEPTION

WHEN DUP\_VAL\_ON\_INDEX THEN

-- Log error if customer with the same ID already exists

LogError('Customer ID ' || new\_customer\_id || ' already exists.');

RAISE\_APPLICATION\_ERROR(-20001, 'Customer ID ' || new\_customer\_id || ' already exists.');

WHEN OTHERS THEN

-- Log any other errors

LogError(SQLERRM);

ROLLBACK;

RAISE;

END AddNewCustomer;