**Exercise 2: Error Handling**

**Scenario 1: Safe Fund Transfer**

**Stored Procedure**:

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_from\_account IN NUMBER,

p\_to\_account IN NUMBER,

p\_amount IN NUMBER

) AS

v\_balance NUMBER;

BEGIN

SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID = p\_from\_account;

IF v\_balance < p\_amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

ELSE

UPDATE Accounts SET Balance = Balance - p\_amount WHERE AccountID = p\_from\_account;

UPDATE Accounts SET Balance = Balance + p\_amount WHERE AccountID = p\_to\_account;

COMMIT;

END IF;

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

END;

/

**Verification Query**:

BEGIN

SafeTransferFunds(1, 2, 500);

END;

/

**Sample Output**:

**If there are sufficient funds:**

No output. Also balances in Accounts table will be updated.

**If there are insufficient funds:**

Error: Insufficient funds

**Scenario 2: Update Employee Salary**

**Stored Procedure**:

CREATE OR REPLACE PROCEDURE UpdateSalary (

p\_employee\_id IN NUMBER,

p\_percentage IN NUMBER

) AS

BEGIN

UPDATE Employees

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

WHERE EmployeeID = p\_employee\_id;

IF SQL%ROWCOUNT = 0 THEN

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

END IF;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

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

ROLLBACK;

END;

/

**Verification Query**:

BEGIN

UpdateSalary(2, 10);

END;

/

**Sample Output**:

**If the EmployeeID exists:**

No output. Also, the salary of the employee will be updated.

**If EmployeeID does not exist:**

Error: Employee ID not found

**Scenario 3: Add New Customer**

**Stored Procedure**:

CREATE OR REPLACE PROCEDURE AddNewCustomer (

p\_customer\_id IN NUMBER,

p\_name IN VARCHAR2,

p\_dob IN DATE,

p\_balance IN NUMBER

) AS

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

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

ROLLBACK;

END;

/

**Verification Query**:

BEGIN

AddNewCustomer(3, 'Mark Twain', TO\_DATE('1975-10-10', 'YYYY-MM-DD'), 2000);

END;

/

**Sample Output**:

**If CustomerID 3 does not exist:**

No output. Also, the new customer will be added.

**If CustomerID 3 already exists:**

Error: Customer ID already exists