**Exercise 2: Error Handling**

CREATE TABLE Accounts(AccountID NUMBER PRIMARY KEY,HolderName VARCHAR2(100),Balance NUMBER);

CREATE TABLE Employees(EmpID NUMBER PRIMARY KEY,EmpName VARCHAR2(100),Salary NUMBER);

INSERT INTO Accounts VALUES(1,'John',5000);

INSERT INTO Accounts VALUES(2,'Jane',3000);

INSERT INTO Employees VALUES(101,'Amit',40000);

INSERT INTO Employees VALUES(102,'Neha',50000);

CREATE OR REPLACE PROCEDURE SafeTransferFunds(p\_from\_id IN NUMBER,p\_to\_id IN NUMBER,p\_amount IN NUMBER) AS insufficient\_funds EXCEPTION;

v\_balance NUMBER;

BEGIN

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

    IF v\_balance<p\_amount THEN

        RAISE insufficient\_funds;

    END IF;

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

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

    COMMIT;

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

EXCEPTION

    WHEN insufficient\_funds THEN

        ROLLBACK;

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

    WHEN OTHERS THEN

        ROLLBACK;

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

END;

CREATE OR REPLACE PROCEDURE UpdateSalary(p\_emp\_id IN NUMBER,p\_percentage IN NUMBER) AS

BEGIN

    UPDATE Employees

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

    WHERE EmpID=p\_emp\_id;

    IF SQL%ROWCOUNT=0 THEN

        DBMS\_OUTPUT.PUT\_LINE('Error: No employee found with ID '||p\_emp\_id);

    ELSE

        COMMIT;

        DBMS\_OUTPUT.PUT\_LINE('Salary updated for Employee ID '||p\_emp\_id);

    END IF;

EXCEPTION

    WHEN OTHERS THEN

        ROLLBACK;

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

END;

CREATE OR REPLACE PROCEDURE AddNewCustomer(p\_id IN NUMBER,p\_name IN VARCHAR2,p\_age IN NUMBER,p\_balance IN NUMBER,p\_interest IN NUMBER,p\_duedate IN DATE) AS

BEGIN

    INSERT INTO Customers(CustomerID,Name,Age,Balance,IsVIP,LoanInterest,LoanDueDate)

    VALUES(p\_id,p\_name,p\_age,p\_balance,'FALSE',p\_interest,p\_duedate);

    COMMIT;

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

EXCEPTION

    WHEN DUP\_VAL\_ON\_INDEX THEN

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

    WHEN OTHERS THEN

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

END;

BEGIN

    SafeTransferFunds(1, 2, 1000);

END;

BEGIN

    UpdateSalary(101, 10);

END;

BEGIN

    AddNewCustomer(5, 'Sneha', 29, 8000, 10.5, SYSDATE + 20);

END;

**Output:**



















