**Exercise 2: Error Handling**

**Scenario 1:**

**Code:** (Respected Sir/Mam, I have added few comment lines for readability)

***-- Creating the Accounts table***

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE,

FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID)

);

***-- Inserting Values into the Accounts Table***

BEGIN

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (1, 1, 'Savings', 1200, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (2, 2, 'Checking', 1500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (3, 3, 'Savings', 2000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (4, 4, 'Checking', 500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (5, 5, 'Savings', 5000, SYSDATE);

END;

***-- Creating the stored procedure to transfer funds between accounts safely***

CREATE OR REPLACE PROCEDURE SafeTransferFunds(FromAccountID IN NUMBER,ToAccountID IN NUMBER,Amount IN NUMBER)

AS

balance NUMBER;

Insufficient\_Funds EXCEPTION;

BEGIN

SAVEPOINT beforeChanges;

SELECT Balance INTO balance

FROM Accounts

WHERE AccountID=FromAccountID;

IF (balance<Amount) THEN

RAISE Insufficient\_Funds;

END IF;

UPDATE Accounts

SET Balance=Balance-Amount,

LastModified=SYSDATE

WHERE AccountID=FromAccountID;

UPDATE Accounts

SET Balance=Balance+Amount,

LastModified=SYSDATE

WHERE AccountID=ToAccountID;

DBMS\_OUTPUT.PUT\_LINE('Transfer Successful! $' || Amount || ' transferred from Account ID:' || FromAccountID ||

' to Account ID:' || ToAccountID || CHR(10));

EXCEPTION

WHEN Insufficient\_Funds THEN

DBMS\_OUTPUT.PUT\_LINE('Transfer Failed: Insufficient funds in Account ID: ' || FromAccountID);

DBMS\_OUTPUT.PUT\_LINE('Available Balance: $' || balance || ', Required: $' || Amount);

ROLLBACK TO beforeChanges;

DBMS\_OUTPUT.PUT\_LINE('Transaction rolled back.' || CHR(10));

WHEN OTHERS THEN

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

ROLLBACK TO beforeChanges;

DBMS\_OUTPUT.PUT\_LINE('Transaction rolled back.'|| CHR(10));

END;

BEGIN

-- Case 1: Successful transfer from Account 2 to Account 5

SafeTransferFunds(2,5,200);

-- Case 2: Failed transfer from Account 4 to Account 3 (insufficient balance)

SafeTransferFunds(4,3,600);

END;

**OUTPUT:**

A screenshot of a computer

AI-generated content may be incorrect.