**Question – 1**

Scenario 1: Handle exceptions during fund transfers between accounts.

* Question: Write a stored procedure SafeTransferFunds that transfers funds between two accounts. Ensure that if any error occurs (e.g., insufficient funds), an appropriate error message is logged and the transaction is rolled back.

PL/SQL Code—

SET SERVEROUT ON;

CREATE OR REPLACE PROCEDURE SafeTransferFunds (

p\_source\_account\_id IN NUMBER,

p\_target\_account\_id IN NUMBER,

p\_amount IN NUMBER

) IS

insufficient\_funds EXCEPTION;

invalid\_account EXCEPTION;

BEGIN

DECLARE

v\_source\_balance NUMBER;

BEGIN

SELECT balance INTO v\_source\_balance

FROM Accounts

WHERE AccountID = p\_source\_account\_id

FOR UPDATE;

IF v\_source\_balance < p\_amount THEN

RAISE insufficient\_funds;

END IF;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RAISE invalid\_account;

END;

-- Deduct the amount from the source account

UPDATE Accounts

SET Balance = Balance - p\_amount

WHERE AccountID = p\_source\_account\_id;

-- Add the amount to the target account

UPDATE Accounts

SET Balance = Balance + p\_amount

WHERE AccountID = p\_target\_account\_id;

COMMIT;

EXCEPTION

WHEN insufficient\_funds THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: Insufficient funds in the source account.');

WHEN invalid\_account THEN

ROLLBACK;

DBMS\_OUTPUT.PUT\_LINE('Error: Invalid account ID.');

WHEN OTHERS THEN

ROLLBACK;

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

END SafeTransferFunds;

/

BEGIN

SafeTransferFunds(1, 2, 200);

SafeTransferFunds(3,1,400);

END;

/

-- Verifying it

SELECT \* FROM Accounts WHERE AccountID IN (1, 2);

SELECT \* FROM Accounts WHERE AccountID IN (3, 1);