**Question – 3**

Scenario 3: Enforce business rules on deposits and withdrawals.

* Question: Write a trigger CheckTransactionRules that ensures withdrawals do not exceed the balance and deposits are positive before inserting a record into the Transactions table.

PL/SQL Code –

/\*Because the user is SYSTEM so cannot create triggers on objects owned by the SYS user.

It Rejects the Trigger

To resolve this, you should ensure that you are creating the trigger in a user schema rather than the SYS schema.

1. Connect to non-SYS user Schema: In this case I connect to \*DB\_USER\*

2. create the tables for the TRIGGER

3. create the TRIGGER \*/

-- Connect to the database as SYSDBA and grant necessary quota

CONNECT SYS/&&ORCL\_PASSWORD@&&DB\_HOST AS SYSDBA;

/\*1. in the &&ORCL\_PASSWORD paste your sys password as a String input

2. in the &&DB\_HOST paste the Database host \*/

-- Grant execute permission on DBMS\_LOCK to the user

GRANT EXECUTE ON DBMS\_LOCK TO &&DB\_USER

ALTER USER &&DB\_USER QUOTA UNLIMITED ON USERS;

CONNECT &&DB\_USER/&&DB\_PASSWORD@&&DB\_HOST

/\*3.in the &&DB\_PASSWORD paste the password of DB\_USER\*/

-- create the tables for the TRIGGER

CREATE TABLE Accounts (

AccountID NUMBER PRIMARY KEY,

CustomerID NUMBER,

AccountType VARCHAR2(20),

Balance NUMBER,

LastModified DATE

);

CREATE TABLE Transactions (

TransactionID NUMBER PRIMARY KEY,

TransactionType VARCHAR2(20),

Amount NUMBER,

AccountID NUMBER,

FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES

(1, 101, 'Savings', 500, TO\_DATE('2023-05-15', 'YYYY-MM-DD'));

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES

2, 102, 'Checking', 1000, TO\_DATE('2024-03-20', 'YYYY-MM-DD'));

SELECT \* FROM ACCOUNTS;

SET SERVEROUT ON;

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

current\_balance NUMBER;

BEGIN

-- Check if the transaction is a deposit or withdrawal

IF :NEW.transactiontype = 'withdrawal' THEN

-- Get the current balance of the account

SELECT balance INTO current\_balance

FROM Accounts

WHERE accountid = :NEW.accountid;

IF :NEW.amount > current\_balance THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance for the withdrawal.');

END IF;

ELSIF :NEW.transactiontype = 'deposit' THEN

-- Ensure that the deposit amount is positive

IF :NEW.amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Deposit amount must be positive.');

END IF;

END IF;

EXCEPTION

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE(SQLERRM);

RAISE;

END;

INSERT INTO Transactions (TransactionID, TransactionType, Amount, AccountID)

VALUES (1, 'deposit', 200, 1);

INSERT INTO Transactions (TransactionID, TransactionType, Amount, AccountID)

VALUES (2, 'withdrawal', 100, 1);

INSERT INTO Transactions (TransactionID, TransactionType, Amount, AccountID)

VALUES (3, 'withdrawal', 700, 1);

-- This should raise an error: ORA-20001: Insufficient balance for the withdrawal.

INSERT INTO Transactions (TransactionID, TransactionType, Amount, AccountID)

VALUES (4, 'deposit', -50, 1);

-- This should raise an error: ORA-20002: Deposit amount must be positive.

SELECT \* FROM Accounts;

SELECT \* FROM TRANSACTIONS;