**Exercise 5: Triggers**

**Scenario 3:**

**CODE:**

-- Creating 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 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', 800, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (3, 3, 'Savings', 15000, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (4, 4, 'Checking', 9500, SYSDATE);

INSERT INTO Accounts (AccountID, CustomerID, AccountType, Balance, LastModified) VALUES (5, 5, 'Savings', 20000, SYSDATE);

END;

--Creating Transactions table

CREATE TABLE Transactions (

TransactionID NUMBER PRIMARY KEY,

AccountID NUMBER,

TransactionDate DATE,

Amount NUMBER,

TransactionType VARCHAR2(10),

FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

select \* from Transactions;

--Inserting values into Transaction table

INSERT INTO Transactions VALUES (1, 1, SYSDATE, 200, 'Deposit');

INSERT INTO Transactions VALUES (2, 2, SYSDATE, 300, 'Withdrawal');

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

v\_Balance NUMBER;

BEGIN

-- Fetch the current balance of the account

SELECT Balance INTO v\_Balance

FROM Accounts

WHERE AccountID = :NEW.AccountID;

IF :NEW.TransactionType = 'Withdrawal' THEN

IF :NEW.Amount > v\_Balance THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Withdrawal amount exceeds account balance.');

ELSIF :NEW.Amount <= 0 THEN

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

END IF;

ELSIF :NEW.TransactionType = 'Deposit' THEN

IF :NEW.Amount <= 0 THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Deposit amount must be greater than zero.');

END IF;

ELSE

RAISE\_APPLICATION\_ERROR(-20004, 'Invalid transaction type. Use "Deposit" or "Withdrawal".');

END IF;

END;

/

-- Invalid withdrawal (more than balance)

INSERT INTO Transactions VALUES (2, 2, SYSDATE, 999999, 'Withdrawal');

-- Invalid deposit (negative amount)

INSERT INTO Transactions VALUES (4, 4, SYSDATE, -100, 'Deposit');

**OUTPUT:**

Output for **Invalid withdrawl** is:

A screenshot of a computer

AI-generated content may be incorrect.

Output for **Invalid deposit** is:

A screenshot of a computer

AI-generated content may be incorrect.