**Exercise 5: Triggers**

CREATE TABLE Customers(CustomerID NUMBER PRIMARY KEY,Name VARCHAR2(100),Age NUMBER,Balance NUMBER,LastModified DATE);

INSERT INTO Customers VALUES(1,'Ravi',65,15000,SYSDATE);

INSERT INTO Customers VALUES(2,'Raja',55,20000,SYSDATE);

COMMIT;

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

    :NEW.LastModified:=SYSDATE;

END;

CREATE TABLE Transactions(TransactionID NUMBER PRIMARY KEY,AccountID NUMBER,Amount NUMBER,Type VARCHAR2(10),TransactionDate DATE DEFAULT SYSDATE);

CREATE TABLE AuditLog(LogID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,TransactionID NUMBER,AccountID NUMBER,Amount NUMBER,ActionType VARCHAR2(20),LoggedAt DATE);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

    INSERT INTO AuditLog(TransactionID,AccountID,Amount,ActionType,LoggedAt)

    VALUES (:NEW.TransactionID,:NEW.AccountID,:NEW.Amount,'INSERT',SYSDATE);

END;

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

    v\_balance NUMBER;

BEGIN

    SELECT Balance INTO v\_balance FROM Accounts WHERE AccountID=:NEW.AccountID;

    IF :NEW.Type='WITHDRAW' THEN

        IF :NEW.Amount>v\_balance THEN

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

        END IF;

    ELSIF :NEW.Type='DEPOSIT' THEN

        IF :NEW.Amount<=0 THEN

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

        END IF;

    ELSE

        RAISE\_APPLICATION\_ERROR(-20003,'Invalid transaction type.');

    END IF;

END;

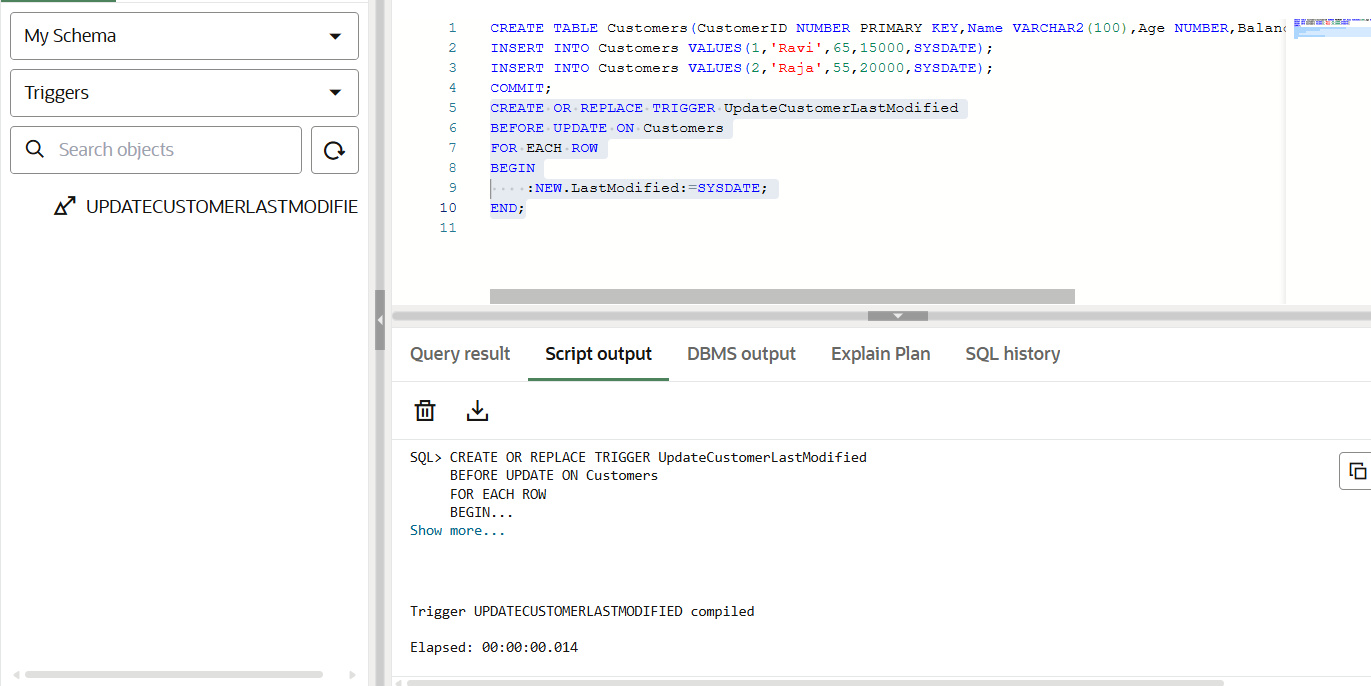
INSERT INTO Transactions(TransactionID,AccountID,Amount,Type)

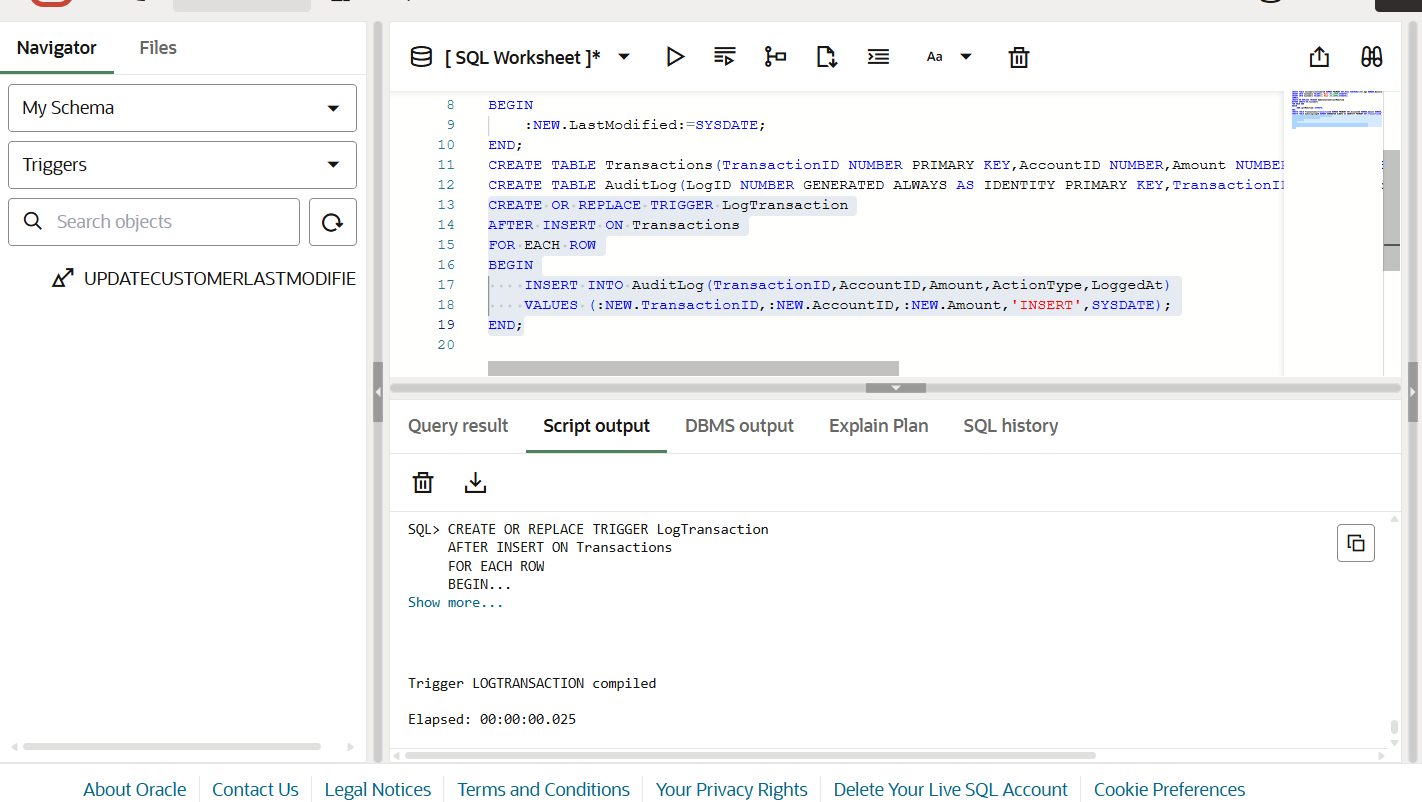
VALUES (1,1,500,'DEPOSIT');

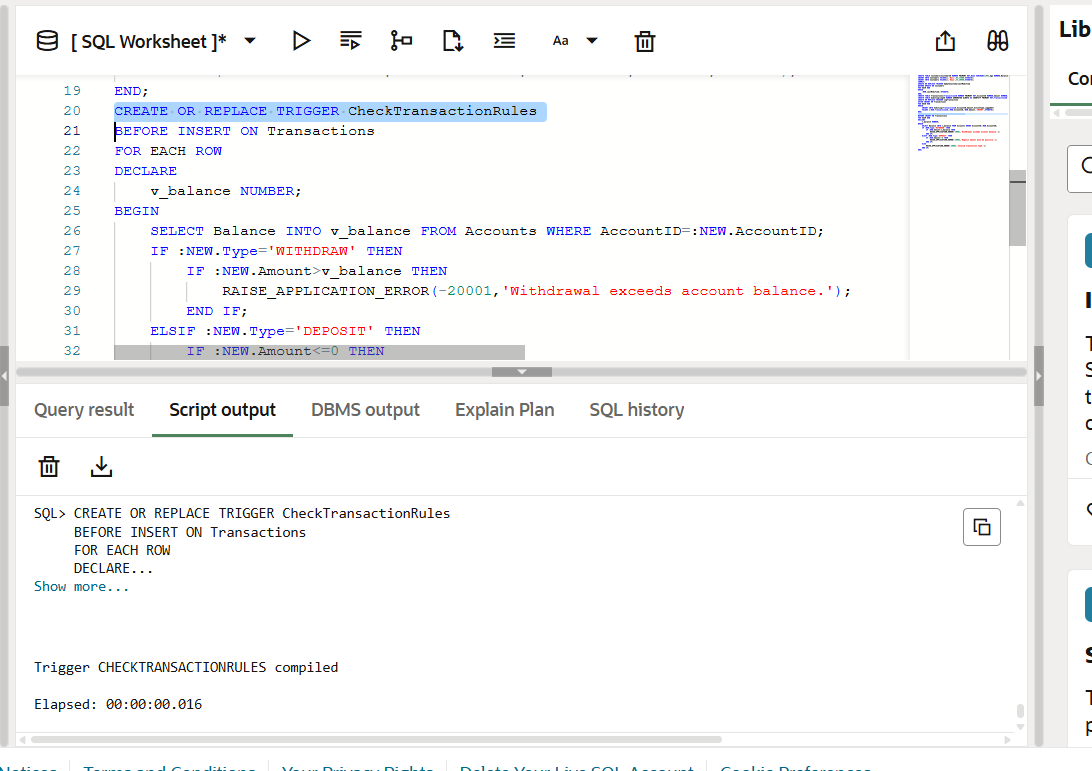
INSERT INTO Transactions(TransactionID,AccountID,Amount,Type)VALUES(2,1,100000,'WITHDRAW');

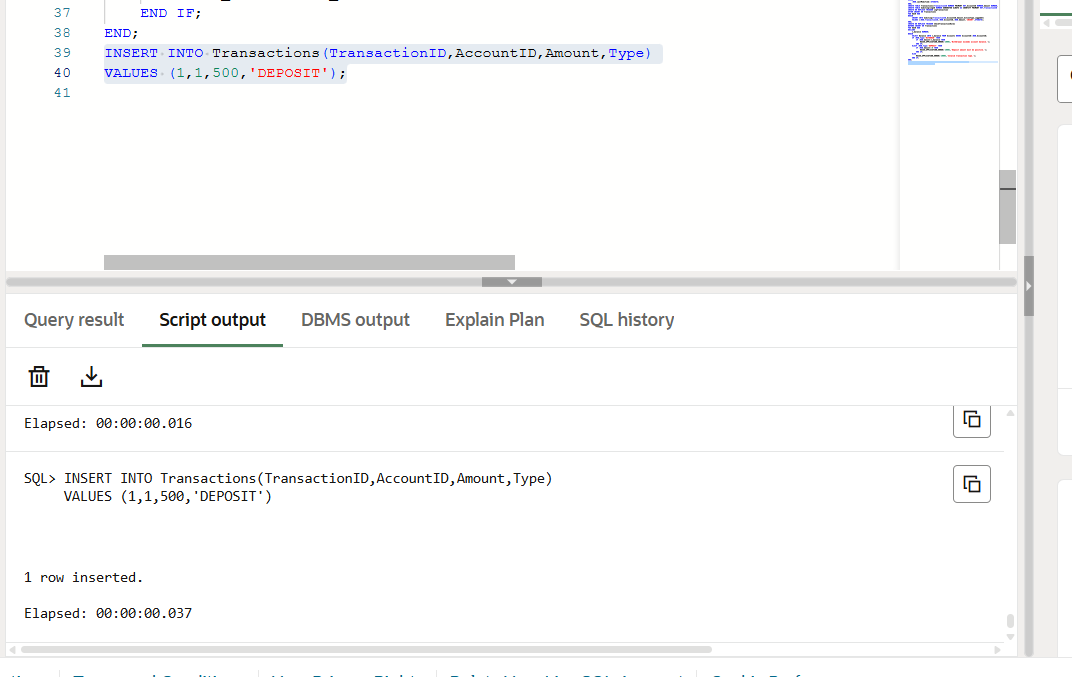
SELECT\*FROM Transactions;

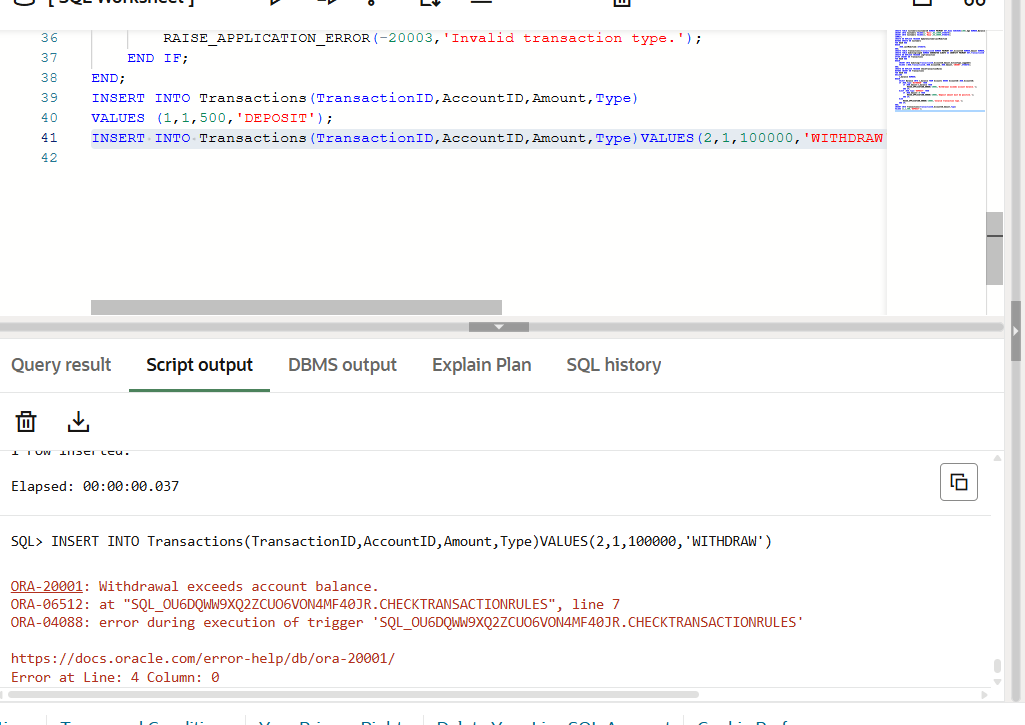
**Output:**

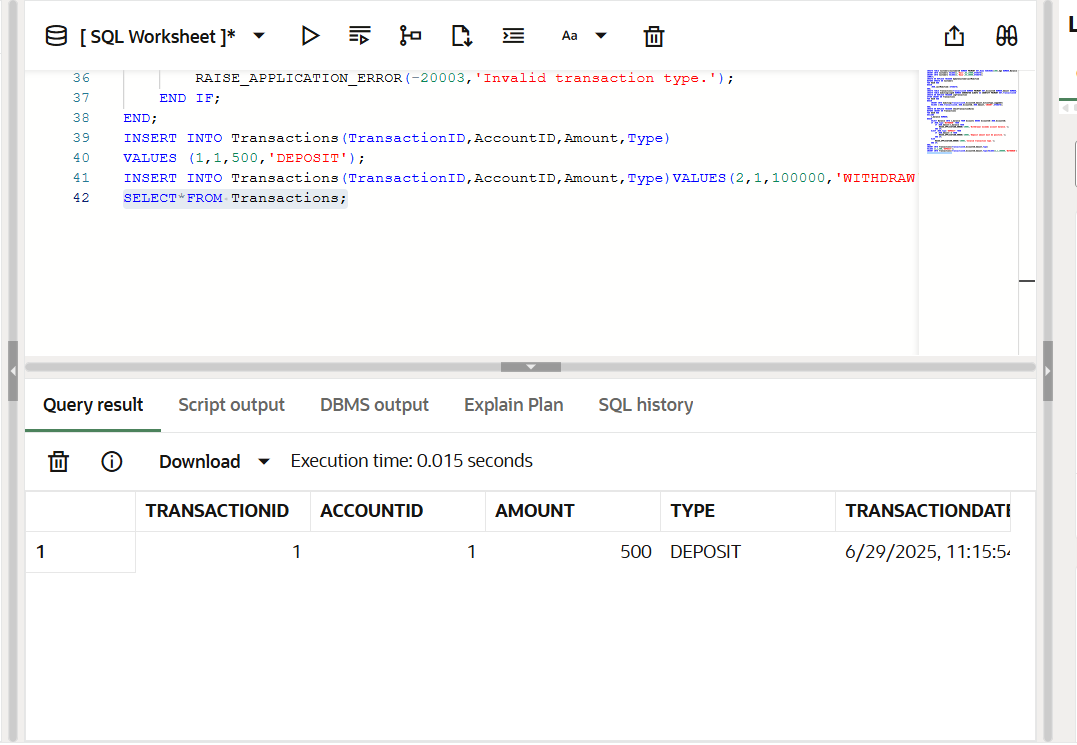
****

****

****

****

****

****