**WEEK-2**

**MODULE -3**

**PL/SQL PROGRAMMING**

**SUPERSET ID:6407550**

**Exercise:5-Triggers**

**Scenario:1**

CREATE OR REPLACE TRIGGER

UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

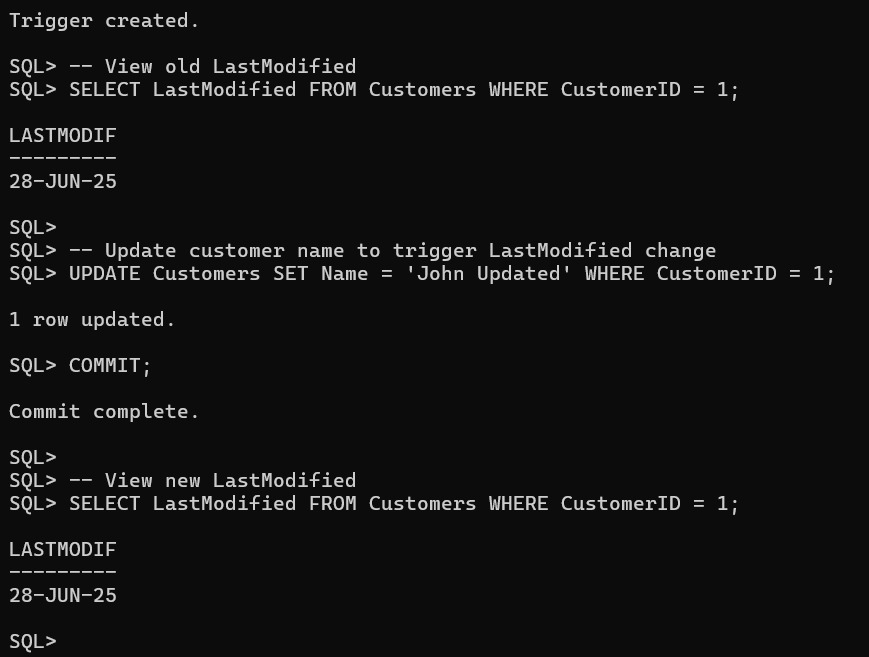
BEGIN

:NEW.LastModified := SYSDATE;

END;

/

**OUTPUT:**

****

**Scenario:2**

CREATE TABLE AuditLog (

LogID NUMBER GENERATED BY

DEFAULT ON NULL AS IDENTITY

PRIMARY KEY,

TransactionID NUMBER,

ActionType VARCHAR2(20),

ActionDate DATE

);

CREATE OR REPLACE TRIGGER

LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

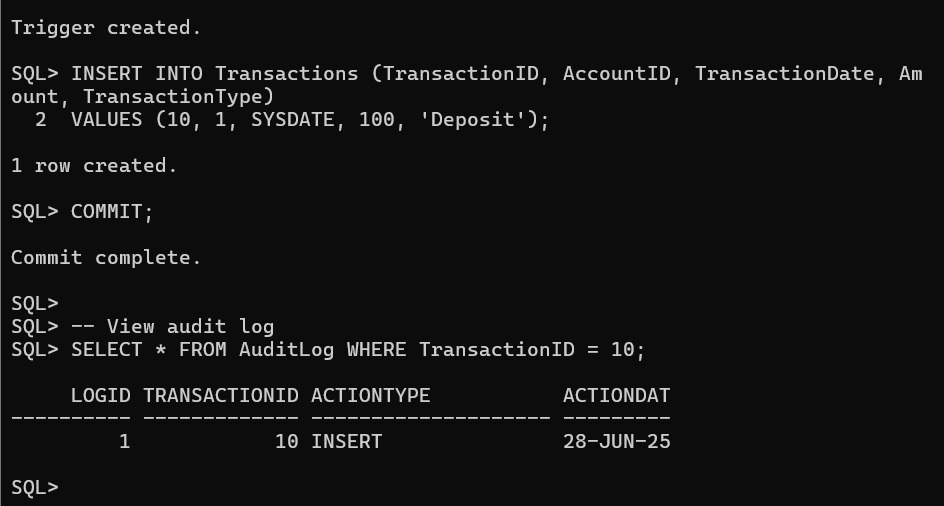
INSERT INTO AuditLog (TransactionID, ActionType, ActionDate)

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

END;

/

**OUTPUT:**

****

**Scenario:3**

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

v\_Balance NUMBER;

BEGIN

-- Get current balance

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

IF :NEW.TransactionType = 'Withdrawal' AND :NEW.Amount > v\_Balance THEN

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

ELSIF :NEW.TransactionType = 'Deposit' AND :NEW.Amount <= 0 THEN

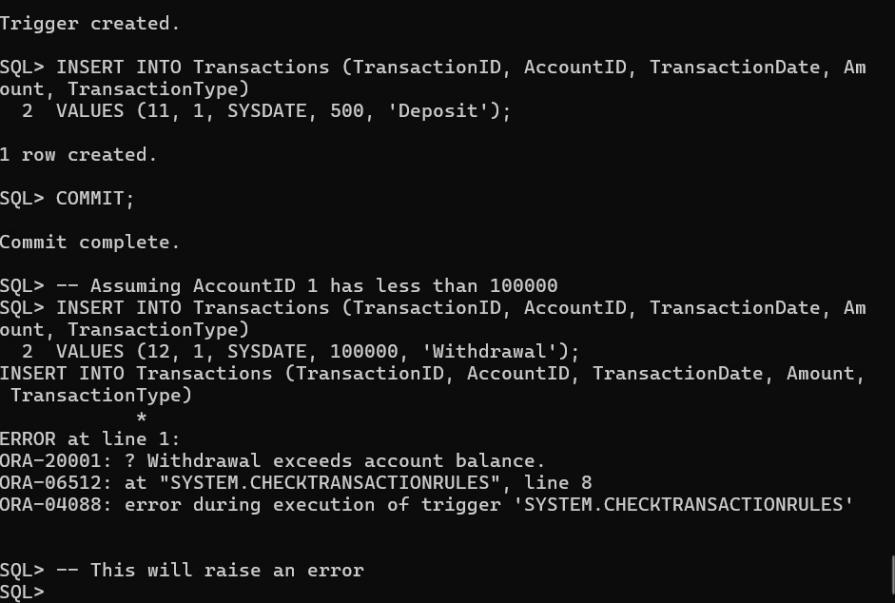
RAISE\_APPLICATION\_ERROR(-20002, '❌ Deposit amount must be positive.');

END IF;

END;

/

**OUTPUT:**

****