**Exercise 5: Triggers**

CREATE TABLE Transactions (

  TransactionID NUMBER PRIMARY KEY,

  AccountID NUMBER,

  TransactionDate DATE,

  Amount NUMBER,

  TransactionType VARCHAR2(10),

  FOREIGN KEY (AccountID) REFERENCES Accounts(AccountID)

);

CREATE TABLE AuditLog (

  LogID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

  TransactionID NUMBER,

  ActionType VARCHAR2(50),

  LogDate DATE

);

**Scenario 1:**

**Question:** Write a trigger **UpdateCustomerLastModified** that updates the LastModified column of the Customers table to the current date whenever a customer's record is updated.

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

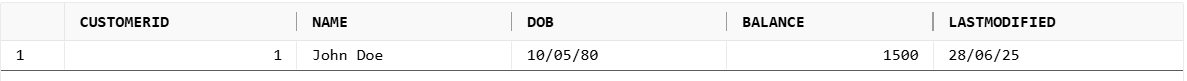
BEGIN

  :NEW.LastModified := SYSDATE;

END;

/

UPDATE Customers SET Balance = 1500 WHERE CustomerID = 1;



**Scenario 2:**

**Question:** Write a trigger **LogTransaction** that inserts a record into an AuditLog table whenever a transaction is inserted into the Transactions table.

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

  INSERT INTO AuditLog (TransactionID, ActionType, LogDate)

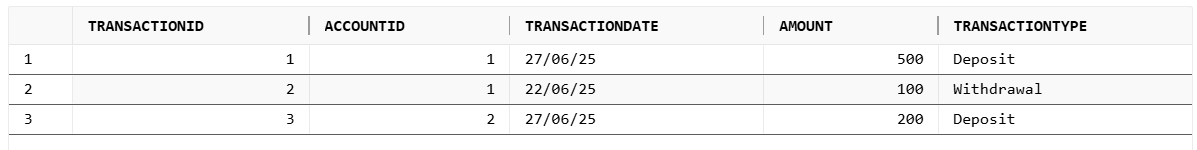
  VALUES (:NEW.TransactionID, 'INSERT', SYSDATE);

END;

/

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

VALUES (1, 1, SYSDATE, 500, 'Deposit');



**Scenario 3:**

**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.

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

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

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

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

  END IF;

END;

/

BEGIN

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

  VALUES (2, 1, SYSDATE, 10000, 'Withdrawal');

EXCEPTION

  WHEN OTHERS THEN

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

END;

/

**Output**

