**Exercise 5: Triggers**

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

  :NEW.LastModified := SYSDATE;

END;

/

UPDATE Customers

SET Name = 'Updated Name'

WHERE CustomerID = 1;

SELECT \* FROM Customers;

CREATE TABLE AuditLog (

  LogID NUMBER GENERATED ALWAYS AS IDENTITY PRIMARY KEY,

  AccountID NUMBER,

  TransactionDate DATE,

  Amount NUMBER,

  TransactionType VARCHAR2(10),

  LoggedAt DATE

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

  INSERT INTO AuditLog (AccountID, TransactionDate, Amount, TransactionType, LoggedAt)

  VALUES (:NEW.AccountID, :NEW.TransactionDate, :NEW.Amount, :NEW.TransactionType, SYSDATE);

END;

/

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

VALUES (3, 1, SYSDATE, 500, 'Deposit');

SELECT \* FROM AuditLog;

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

  v\_balance NUMBER;

BEGIN

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

    RAISE\_APPLICATION\_ERROR(-20001, 'Deposit amount must be positive.');

  ELSIF :NEW.TransactionType = 'Withdrawal' THEN

    SELECT Balance INTO v\_balance

    FROM Accounts

    WHERE AccountID = :NEW.AccountID;

    IF :NEW.Amount > v\_balance THEN

      RAISE\_APPLICATION\_ERROR(-20002, 'Insufficient balance for withdrawal.');

    END IF;

  END IF;

END;

/

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

VALUES (4, 1, SYSDATE, 100, 'Deposit');













