**Exercise 5: Triggers**

**Scenario 1: Automatically update the last modified date when a customer's record is updated**

Query:

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSDATE;

END UpdateCustomerLastModified;

**Scenario 2: Maintain an audit log for all transactions**

Query:

CREATE TABLE AuditLog (

TransactionID NUMBER,

TransactionDate DATE,

AccountID NUMBER,

Amount NUMBER,

TransactionType VARCHAR2(10)

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

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

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

END LogTransaction;

**Scenario 3: Enforce business rules on deposits and withdrawals**

Query:

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

IF :NEW.TransactionType = 'Withdrawal' THEN

DECLARE

v\_balance NUMBER;

BEGIN

SELECT Balance

INTO v\_balance

FROM Accounts

WHERE AccountID = :NEW.AccountID;

IF v\_balance < :NEW.Amount THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient funds');

END IF;

END;

ELSIF :NEW.TransactionType = 'Deposit' THEN

IF :NEW.Amount <= 0 THEN

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

END IF;

END IF;

END CheckTransactionRules;