**Exercise 5: Triggers**

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

* + **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.

**Ans:**

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.

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

**Ans:**

CREATE TABLE AuditLog (

  AuditID NUMBER PRIMARY KEY,

  TransactionID NUMBER,

  ActionType VARCHAR2(50),

  ActionDate DATE,

  UserName VARCHAR2(100)

);

CREATE SEQUENCE AuditLogSeq

START WITH 1

INCREMENT BY 1

NOCACHE

NOCYCLE;

CREATE OR REPLACE TRIGGER AuditLogTrigger

BEFORE INSERT ON AuditLog

FOR EACH ROW

BEGIN

  IF :NEW.AuditID IS NULL THEN

    SELECT AuditLogSeq.NEXTVAL INTO :NEW.AuditID FROM DUAL;

  END IF;

END;

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

* + **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.

**Ans:**

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

  v\_current\_balance NUMBER;

BEGIN

  SELECT Balance INTO v\_current\_balance

  FROM Accounts

  WHERE AccountID = :NEW.AccountID;

  IF :NEW.TransactionType = 'Debit' THEN

    IF :NEW.Amount > v\_current\_balance THEN

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

    END IF;

  ELSIF :NEW.TransactionType = 'Credit' THEN

    IF :NEW.Amount <= 0 THEN

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

    END IF;

  END IF;

EXCEPTION

  WHEN NO\_DATA\_FOUND THEN

    RAISE\_APPLICATION\_ERROR(-20003, 'Account ID does not exist.');

  WHEN OTHERS THEN

    RAISE\_APPLICATION\_ERROR(-20004, 'An unexpected error occurred: ' || SQLERRM);

END CheckTransactionRules;