**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.

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

-- Set the LastModified column to the current date before updating the record

:NEW.LastModified := SYSDATE;

END;

/

**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.

CREATE TABLE AuditLog (

AuditID NUMBER PRIMARY KEY,

TransactionID NUMBER,

ActionDate DATE,

ActionType VARCHAR2(20),

Details VARCHAR2(255)

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

-- Insert a record into the AuditLog table

INSERT INTO AuditLog (AuditID, TransactionID, ActionDate, ActionType, Details)

VALUES (

AuditLog\_seq.NEXTVAL, -- Assuming there's a sequence named AuditLog\_seq for generating unique AuditIDs

:NEW.TransactionID,

SYSDATE,

'INSERT',

'Transaction added with amount ' || :NEW.Amount || ' on account ' || :NEW.AccountID

);

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.

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

v\_balance NUMBER;

BEGIN

-- Check if the transaction is a withdrawal

IF :NEW.TransactionType = 'Withdrawal' THEN

-- Retrieve the current balance for the account

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = :NEW.AccountID;

-- Ensure the withdrawal amount does not exceed the balance

IF :NEW.Amount > v\_balance THEN

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

END IF;

-- Check if the transaction is a deposit

ELSIF :NEW.TransactionType = 'Deposit' THEN

-- Ensure the deposit amount is positive

IF :NEW.Amount <= 0 THEN

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

END IF;

ELSE

-- Handle other transaction types if necessary

RAISE\_APPLICATION\_ERROR(-20003, 'Invalid transaction type.');

END IF;

END;

/