**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

AFTER UPDATE ON Customers

FOR EACH ROW

BEGIN

*-- Update the LastModified column to the current date*

: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.

Creating the AuditLog table to store the audit records:

CREATE TABLE AuditLog (

AuditID NUMBER GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

TransactionID NUMBER,

AccountID NUMBER,

TransactionDate DATE,

Amount NUMBER,

TransactionType VARCHAR2(50),

LogDate DATE

);

Creating the LogTransaction trigger:

CREATE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

*-- Insert a record into the AuditLog table*

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

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

END LogTransaction;

/

Example usage:

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

VALUES (1, 101, SYSDATE, 500, 'Deposit');

SELECT \* FROM AuditLog;

This will insert a new transaction and automatically create a corresponding record in the AuditLog table.

**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 TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

v\_Balance NUMBER;

BEGIN

*-- Retrieve the current balance for the account*

SELECT Balance INTO v\_Balance

FROM Accounts

WHERE AccountID = :NEW.AccountID;

*-- Check if the transaction is a withdrawal*

IF :NEW.TransactionType = 'Withdrawal' THEN

*-- Ensure the withdrawal amount does not exceed the balance*

IF :NEW.Amount > v\_Balance THEN

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

END IF;

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

*-- Raise an error for unknown transaction types*

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

END IF;

END CheckTransactionRules;

/

Example usage:

*-- This should succeed if the balance is sufficient*

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

VALUES (1, 101, SYSDATE, 100, 'Withdrawal');

*-- This should raise an error because the deposit amount is not positive*

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

VALUES (2, 101, SYSDATE, -50, 'Deposit');