# 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.

# Answer:

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.

# Answer:

CREATE TABLE AuditLog (

AuditID NUMBER PRIMARY KEY,

TransactionID NUMBER, ChangeDate DATE, ActionType VARCHAR2(50), OldAmount NUMBER, NewAmount NUMBER

);

CREATE OR REPLACE TRIGGER LogTransaction AFTER INSERT ON Transactions

## *FOR* EACH ROW DECLARE

v\_audit\_id NUMBER; BEGIN

*-- Generate a unique ID for the audit log*

SELECT AuditLog\_SEQ.NEXTVAL INTO v\_audit\_id FROM dual;

*-- Insert record into AuditLog table*

INSERT INTO AuditLog (

AuditID, TransactionID, ChangeDate, ActionType, OldAmount, NewAmount

## ) VALUES (

v\_audit\_id, :NEW.TransactionID, SYSDATE, 'INSERT', NULL, :NEW.Amount

);

END LogTransaction;

/

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.

# Answer:

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*

*-- Fetch the current balance of the account*

SELECT Balance INTO v\_balance

FROM Accounts

WHERE AccountID = :NEW.AccountID;

*-- Ensure withdrawal does not exceed the balance IF* :NEW.Amount > v\_balance *THEN*

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

*END IF*;

*-- Check if the transaction is a deposit*

*ELSIF* :NEW.TransactionType = 'Deposit' *THEN*

*-- Ensure deposit amount is positive IF* :NEW.Amount <= 0 *THEN*

RAISE\_APPLICATION\_ERROR(-20003, 'Deposit amount must be positive.');

*END IF*; *END IF*;

END CheckTransactionRules;

/