Scenario 1: UpdateCustomerLastModified - Automatically Update the Last Modified Date

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSDATE;

END;

/

Scenario 2: LogTransaction - Maintain an Audit Log for All Transactions

CREATE TABLE AuditLog (

AuditID NUMBER PRIMARY KEY,

TransactionID NUMBER,

AccountID NUMBER,

TransactionDate DATE,

Amount NUMBER,

TransactionType VARCHAR2(10),

LogDate DATE

);

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

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

VALUES (AuditLog\_seq.NEXTVAL, :NEW.TransactionID, :NEW.AccountID, :NEW.TransactionDate, :NEW.Amount, :NEW.TransactionType, SYSDATE);

END;

/

CREATE SEQUENCE AuditLog\_seq START WITH 1 INCREMENT BY 1;

Scenario 3: CheckTransactionRules - Enforce Business Rules on Deposits and Withdrawals

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

DECLARE

v\_Balance Accounts.Balance%TYPE;

BEGIN

-- Fetch the current balance of the account

SELECT Balance INTO v\_Balance FROM Accounts WHERE AccountID = :NEW.AccountID FOR UPDATE;

-- Check if the transaction is a withdrawal and if it exceeds the balance

IF :NEW.TransactionType = 'Withdrawal' THEN

IF :NEW.Amount > v\_Balance THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient balance for the withdrawal');

END IF;

ELSIF :NEW.TransactionType = 'Deposit' THEN

IF :NEW.Amount <= 0 THEN

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

END IF;

ELSE

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

END IF;

END;

/