Scenario 1: Automatically update the last modified date

CREATE OR REPLACE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

:NEW.LastModified := SYSTIMESTAMP;

END UpdateCustomerLastModified;

Scenario 2: Maintain an audit log for all transactions

CREATE TABLE AuditLog (

LogID NUMBER PRIMARY KEY,

TransactionID NUMBER,

TransactionType VARCHAR2(10),

OldBalance NUMBER,

NewBalance NUMBER,

LogDate DATE DEFAULT SYSTIMESTAMP

);

CREATE SEQUENCE AuditLogSeq;

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

INSERT INTO AuditLog (LogID, TransactionID, TransactionType, OldBalance, NewBalance)

VALUES (AuditLogSeq.NEXTVAL, :NEW.TransactionID, :NEW.TransactionType, :NEW.OldBalance, :NEW.NewBalance);

END LogTransaction;

Scenario 3: Enforce business rules on deposits and withdrawals

CREATE OR REPLACE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

IF :NEW.TransactionType = 'WITHDRAWAL' THEN

IF :NEW.NewBalance < 0 THEN

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

END IF;

ELSIF :NEW.TransactionType = 'DEPOSIT' THEN

IF :NEW.Amount <= 0 THEN

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

END IF;

END IF;

END CheckTransactionRules;