**Scenario 1: Update the Last Modified Date**

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 Transactions**

CREATE OR REPLACE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

INSERT INTO AuditLog (TransactionID, TransactionDate, Action, UserID)

VALUES (:NEW.TransactionID, SYSDATE, 'INSERT', USER);

END LogTransaction;

/

**Scenario 3: Enforce Business Rules on Transactions**

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 account\_id = :NEW.AccountID;

-- Ensure that withdrawals do not exceed the balance

IF :NEW.Amount > v\_balance THEN

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

END IF;

ELSIF :NEW.TransactionType = 'DEPOSIT' THEN

-- Ensure that deposit amounts are positive

IF :NEW.Amount <= 0 THEN

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

END IF;

END IF;

END CheckTransactionRules;

/