-- Scenario 1: Automatically update the last modified date when a customer's record is updated

CREATE TRIGGER UpdateCustomerLastModified

BEFORE UPDATE ON Customers

FOR EACH ROW

BEGIN

    :NEW.LastModified := SYSDATE;

END;

-- Scenario 2: Maintain an audit log for all transactions

CREATE TABLE AuditLog (

    TransactionID NUMBER,

    AccountID NUMBER,

    TransactionDate DATE,

    Amount NUMBER,

    TransactionType VARCHAR2(10)

);

CREATE TRIGGER LogTransaction

AFTER INSERT ON Transactions

FOR EACH ROW

BEGIN

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

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

END;

-- Scenario 3: Enforce business rules on deposits and withdrawals

CREATE TRIGGER CheckTransactionRules

BEFORE INSERT ON Transactions

FOR EACH ROW

BEGIN

    IF :NEW.TransactionType = 'Withdrawal' THEN

        IF :NEW.Amount > (SELECT Balance FROM Accounts WHERE AccountID = :NEW.AccountID) THEN

            RAISE\_APPLICATION\_ERROR(-20001, 'Insufficient 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;