

- **Scenario 1**

```
CREATE OR REPLACE TRIGGER UpdateCustomerLastModified
BEFORE UPDATE ON Customers
FOR EACH ROW
BEGIN
    :NEW.LastModified := SYSDATE;
END;
/
```

- **Scenario 2**

Step 1 : Create AuditLog table

```
CREATE TABLE AuditLog (  
    AuditID NUMBER PRIMARY KEY,  
    TransactionID NUMBER,  
    AccountID NUMBER,  
    Action VARCHAR2(20),  
    Amount NUMBER,  
    TransactionDate DATE,  
    LoggedAt DATE  
);
```

Step 2 : Create Trigger

```
CREATE OR REPLACE TRIGGER LogTransaction  
AFTER INSERT ON Transactions  
FOR EACH ROW  
BEGIN  
    INSERT INTO AuditLog (  
        AuditID, TransactionID, AccountID, Action, Amount, TransactionDate, LoggedAt  
    ) VALUES (  
        AuditLog_seq.NEXTVAL,  
        :NEW.TransactionID,  
        :NEW.AccountID,  
        :NEW.TransactionType,  
        :NEW.Amount,  
        :NEW.TransactionDate,  
        SYSDATE  
    );  
END;
```

/

- **Scenario 3**

```
CREATE OR REPLACE TRIGGER CheckTransactionRules
BEFORE INSERT ON Transactions
FOR EACH ROW
DECLARE
    v_balance Accounts.Balance%TYPE;
BEGIN
    -- Fetch current balance
    SELECT Balance INTO v_balance FROM Accounts WHERE AccountID = :NEW.AccountID;

    IF :NEW.TransactionType = 'Withdrawal' AND :NEW.Amount > v_balance THEN
        RAISE_APPLICATION_ERROR(-20001, 'Withdrawal exceeds available balance.');
```

```
ELSIF :NEW.TransactionType = 'Deposit' AND :NEW.Amount <= 0 THEN  
    RAISE_APPLICATION_ERROR(-20002, 'Deposit amount must be greater than 0.');
```

END IF;

END;

/