

26.03.2026, APEX World 2026, Ede

# All about Validations in APEX

Oliver Lemm

code of change





# Oliver Lemm

Deputy Head of Business Unit APEX / Low Code  
Architect, Project Manager & Developer

**APEX** connect

**DOAG**  
Konferenz + Ausstellung

**ODTUG**  
**Kscope** 

 **IT-Tage**  
2026

**ā'pěks world**  
(#orclapex)

**SOUG**  
swiss oracle  
user group

# Internationally connected delivery power.



## Germany

- Ratingen
- Brunswick
- Hamburg
- Dortmund
- Frankfurt
- Munich
- Berlin

## Poland

- Wrocław

## Lithuania

- Vilnius
- Kaunas

## Romania

- Cluj-Napoca

## India

- Pune

25+

Customers with  
> 1 billion €  
revenue

110+

Million Euros  
revenue  
(2024)

14

Locations in  
5 countries

750+

Employees

24

Nationalities

# Agenda

1. Motivation
2. Before using Validations
3. Built-In Validations
4. Declarative Validations
5. Client-Side Validations
6. Conclusion

# Motivation\_

# *Data Quality*

# *User Acceptance*

# *Explain Data & Input Relations*

# Before using Validations\_

Before using Validations

# use the right Item

- LOV
- Autocomplete
- Number

The image shows a screenshot of the APEX component palette. The 'Text Field' component is selected, and its 'Built-in Components' list is displayed. The 'Text Field with autocomplete' option is highlighted with a red box. Other components in the list include Checkbox, Checkbox Group, Color Picker, Combobox, Date Picker, Display Image, Display Map, Display Only, File Upload, Geocoded Address, Hidden, Image Upload, List Manager, Markdown Editor, Number Field, Password, Percent Graph, Popup LOV, QR Code, Radio Group, Rich Text Editor, Select List, Select Many, Select One, Shuttle, Star Rating, Switch, Text Field, and Textarea.

Category	Item
Identification	Name: P4_ENAME
Identification	Type: Text Field
Label	Label
Settings	Subtype
Settings	Trim Spaces
Settings	Text Case
Settings	Submit when Enter pressed
Settings	Disabled
Layout	Sequence
Layout	Region
Layout	Slot
Layout	Start New Row
Layout	Row CSS Classes
Layout	Column
Layout	Column Span
Layout	Column CSS Classes
Layout	Column Attributes
Built-in Components	Checkbox
Built-in Components	Checkbox Group
Built-in Components	Color Picker
Built-in Components	Combobox
Built-in Components	Date Picker
Built-in Components	Display Image
Built-in Components	Display Map
Built-in Components	Display Only
Built-in Components	File Upload
Built-in Components	Geocoded Address
Built-in Components	Hidden
Built-in Components	Image Upload
Built-in Components	List Manager
Built-in Components	Markdown Editor
Built-in Components	Number Field
Built-in Components	Password
Built-in Components	Percent Graph
Built-in Components	Popup LOV
Built-in Components	QR Code
Built-in Components	Radio Group
Built-in Components	Rich Text Editor
Built-in Components	Select List
Built-in Components	Select Many
Built-in Components	Select One
Built-in Components	Shuttle
Built-in Components	Star Rating
Built-in Components	Switch
Built-in Components	Text Field
Built-in Components	Text Field with autocomplete
Built-in Components	Textarea

Before using Validations

# Textfield

- trim
- text case
- maximum length

The screenshot shows the configuration panel for a text field in APEX. The settings are organized into sections:

- Identification:** Name is P4\_ENAME, Type is Text Field.
- Label:** Label is Employee Name.
- Settings:** Subtype is Text, Trim Spaces is Leading and Trailing, Text Case is No Change, Submit when Enter pressed is off, and Disabled is off.
- Layout:** (Collapsed)
- Appearance:** (Collapsed)
- Validation:** Value Required is off, Maximum Length is 50 characters.

Blue arrows from the list on the left point to the following settings in the panel:

- trim points to Trim Spaces (Leading and Trailing)
- text case points to Text Case (No Change)
- maximum length points to Maximum Length (50 characters)

Before using Validations

# Textarea

- Character Counter
- Trim Spaces
- Height (lines) and Width (Characters)
- Maximum Length

The screenshot shows the configuration interface for a Textarea component in APEX. The settings are organized into several sections:

- Identification:** Name is set to "P4\_ENAME" and Type is "Textarea".
- Label:** Label is set to "Employee Name".
- Settings:** Resizable is turned on, Auto-Height is turned off, Character Counter is turned on, and Trim Spaces is set to "Leading and Trailing".
- Layout:** This section is currently collapsed.
- Appearance:** Template is "Optional - Floating", Template Options is "Use Template Defaults", CSS Classes and Format Mask are empty, Width is 32 characters, Height is 5 lines, and Value Placeholder is empty.
- Validation:** Value Required is turned off, and Maximum Length is 50 characters.

# Built-In Validations\_

# Page Item Setting “Validation”

Employee ✕

Employee Name

Employee Name must have some value.

**1 error has occurred**

- Employee Name must have some value.

Page Item

Filter ↵

**Identification**

Name P4\_ENAME

Type Text Field ⌵ ☰

**Label**

Label Employee Name

**Settings**

**Layout**

**Appearance**

**Validation**

Value Required

Maximum Length 50 characters

# Required != Required

Employee ✕

Employee Name  
ADAMS

**Appearance**

Template: Optional - Floating

Template Options: Use Template Defaults

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: 32 characters

Value Placeholder: [ ]

**Validation**

Value Required:

Maximum Length: 50 characters

Employee ✕

Employee Name  
ADAMS

**Appearance**

Template: Required - Floating

Template Options: Use Template Defaults

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: 32 characters

Value Placeholder: [ ]

**Validation**

Value Required:

Maximum Length: 50 characters

# Maximum Length vs Width

**Employee** [Close]

Employee Name  
1234567890

**Appearance**

Template: Required - Floating

Template Options: Use Template Defaults

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: 10 characters

Value Placeholder: [ ]

**Validation**

Value Required:

Maximum Length: 10 characters

**Employee**

\* Employee Name: ADAMS

**Appearance**

Template: Required

Template Options: Use Template Defaults

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: 10 characters

Value Placeholder: [ ]

**Validation**

Value Required:

Maximum Length: 10 characters

**Employee**

\* Employee Name: QWQWQWQWQWQWQW

**Employee**

\* Employee Name: 123456789012345678901

**Employee**

**Appearance**

Template: Required

Template Options: Use Template Defaults

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: 20 characters

Value Placeholder: [ ]

**Validation**

Value Required:

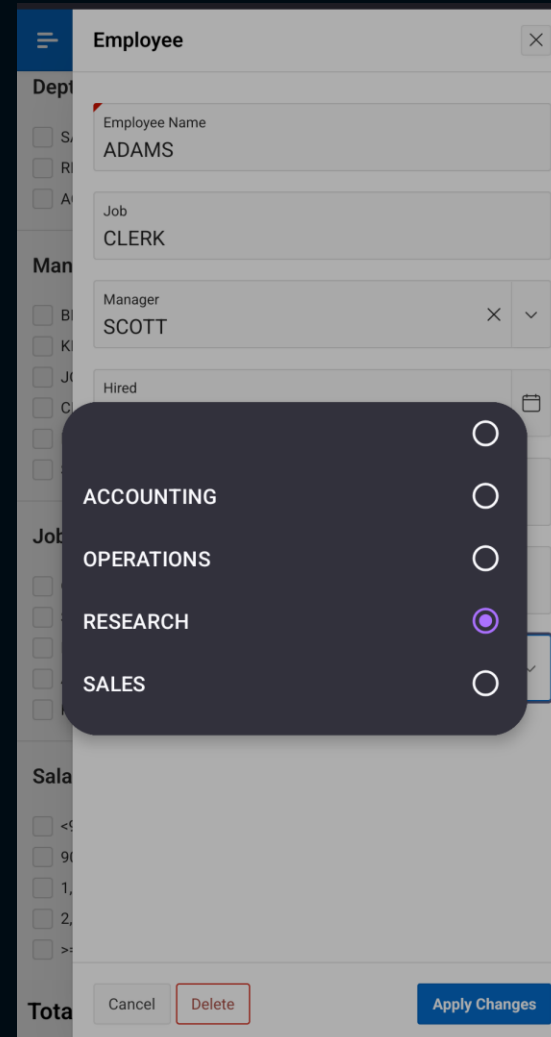
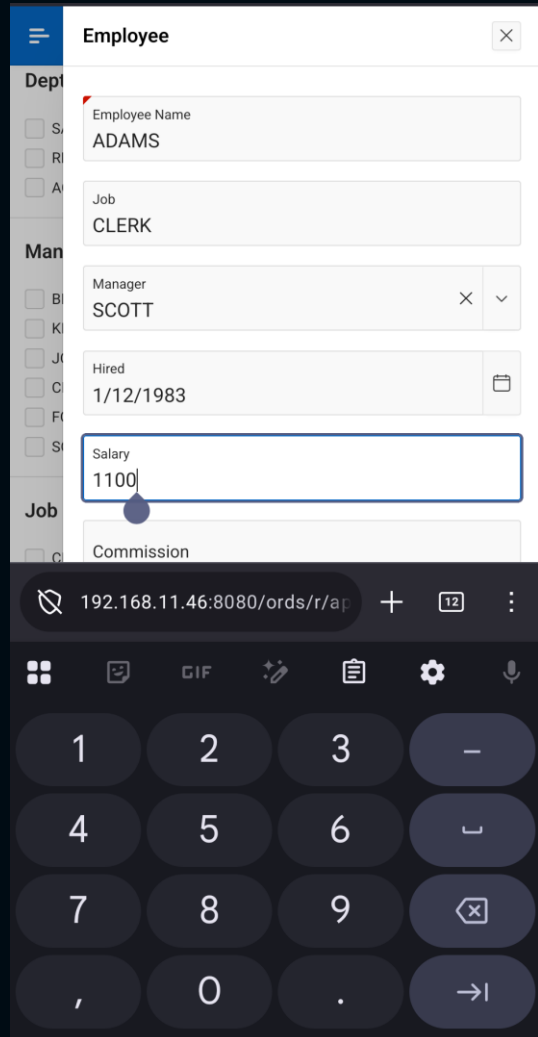
Maximum Length: 30 characters

# Required Message

The screenshot shows a form titled "Employee" with a close button in the top right. Below the title is a text input field labeled "Employee Name". A red border highlights the field, and a red error message "Employee Name must have some value." is displayed below it. A yellow alert box with a warning icon and a close button is overlaid on the right side of the form, containing the text "1 error has occurred" and a bulleted list with the message "Employee Name must have some value."

- Message not customizable
- Message appears on Page Level and below Item

# HTML5 mobile Support



# “validate” data is saved

Page

Filter

Identification

Appearance

Navigation Menu

Navigation

Cursor Focus: Do not focus cursor

Warn on Unsaved Changes

Page Item

Filter

Identification

Label

Settings

Layout

Appearance

Validation

Advanced

CSS Classes

Custom Attributes

Pre Text

Post Text

Warn on Unsaved Changes: Page Default

Message is based on System not Browser Language

Website neu laden?

Deine Änderungen werden eventuell nicht gespeichert.

Keine weiteren Dialogfelder auf dieser Seite zulassen

Neu laden Abbrechen

Employee Name	Job	Manager	Hired
ADAMS	CLERK	SCOTT	1/12/1981
ALLEN	SALESMAN	BLAKE	2/20/1981
BLAKE	MANAGER	KING	5/1/1981
CLARK	MANAGER	KING	6/9/1981

# Handling ORA-Messages

- No validation by page
- define one function for overwriting ORA-Messages
- API “apex\_error”
  - AUTO\_SET\_ASSOCIATED\_ITEM
  - EXTRACT\_CONSTRAINT\_NAME
  - GET\_FIRST\_ORA\_ERROR\_TEXT
  - HAVE\_ERRORS\_OCCURRED
  - INIT\_ERROR\_RESULT

Application 148 | Edit Application Definition

Definition | Security | Globalization | User Interface | Progressive Web App | AI

## Application 148

Show All | Name | Properties | Availability | **Error Handling** | Global Notification | Substituti

### Error Handling

Default Error Display Location: Inline with Field and in Notification

Error Handling Function:

#### Error Handling Function

Enter the name of the PL/SQL function to be called to modify the error message and display location or to log the error if one occurs. It can reference a package function or stand alone function in the database. For example:

```
log_apex_error
```

Note: Error handling specified at the page-level overwrites any error handling function specified here.

Error Handling Functions must implement the following interface:

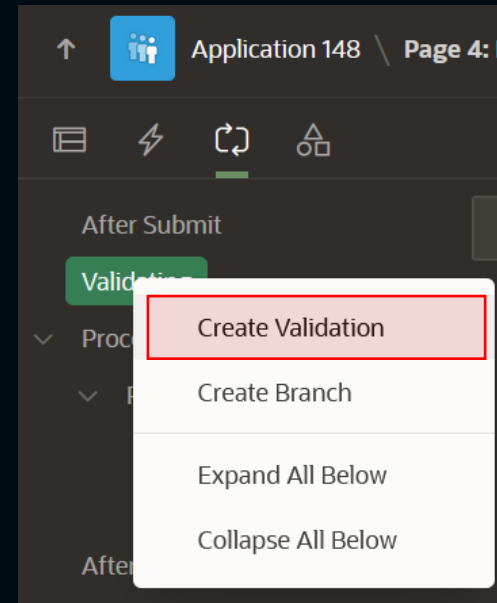
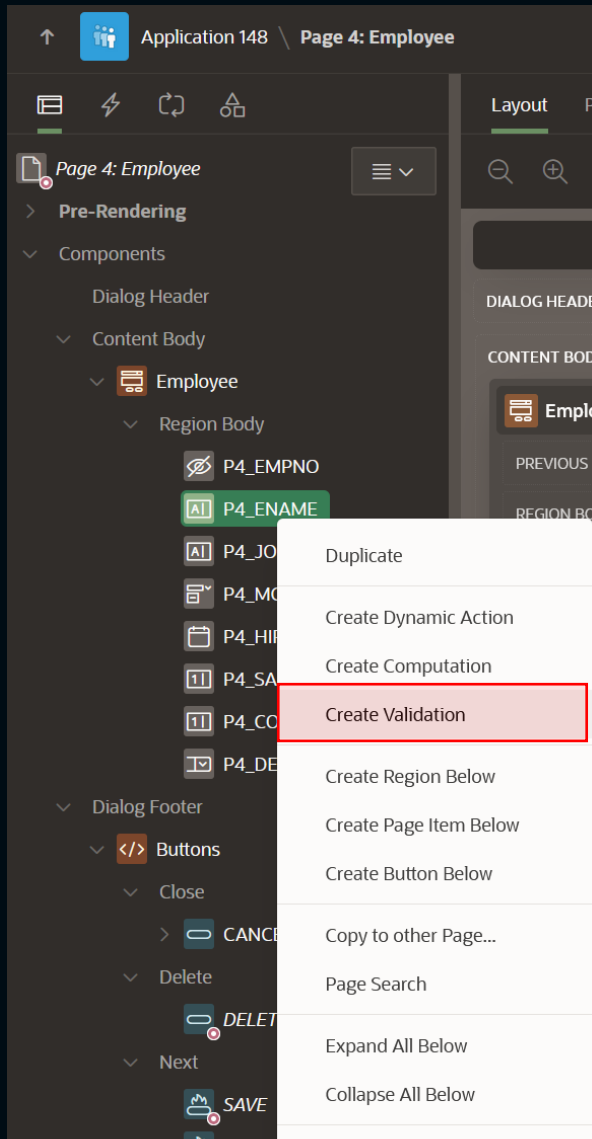
```
function <name of function> (  
    p_error in apex_error.t_error )  
    return apex_error.t_error_result
```

See Also: "apex\_error" in Oracle APEX API Reference

[View Documentation](#)

# Declarative Validations\_

# Create Validation



# Validation Attributes Overview

- Sequence
- Editable Region
- Type
  - Language
  - PL/SQL Expression
- Always Execute
- Error Message
- Display Location
- Associated Item

The screenshot shows the configuration interface for a Validation in Oracle APEX. The interface is organized into several sections, each with a dropdown arrow on the left:

- Validation** (Section Header)
- Filter**: A search bar with a magnifying glass icon and a dropdown arrow.
- Identification**:
  - Name: New
- Execution**:
  - Sequence: 10
- Validation**:
  - Editable Region: - Select -
  - Type: Expression
  - Language: PL/SQL
  - PL/SQL Expression: (Red X icon) [Copy icon]
  - Always Execute: (Toggle switch, currently off)
- Error**:
  - Error Message: (Red X icon) [Copy icon]
- Display Location**: Inline with Field and in Notification
- Associated Item**: - Select -

# Types of Validations

SQL - slowest	PL/SQL - medium	Framework - fastest	
Rows returned	Expression	Item is NOT NULL	Item = Value
No Rows returned	PL/SQL Error	Item is NOT NULL or zero	Item != Value
	Function Body (returning Boolean)	Item is NOT zero	Item is contained in Value
	Function Body (returning Error Text)	Item contains no spaces	Item is NOT contained in Value
		Item is alphanumeric	Item contains only characters specified in Value
		Item is numeric	Item contains at least one of the characters in Value
		Item is a valid date	Item does NOT contain any of the characters in Value
		Item is a valid timestamp	Item matches Regular Expression

# Always Execute

Validation

Filter

Identification

Name: P4\_ENAME not null

Execution

Sequence: 10

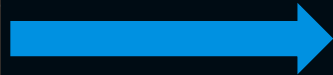
Validation

Editable Region: - Select -

Type: Item is NOT NULL

Item: P4\_ENAME

Always Execute:



Button

Filter

Identification

Button Name: CREATE

Label: Create

Layout

Appearance

Behavior

Action: Submit Page

Execute Validations:

Button

Filter

Identification

Button Name: SUBMIT

Label: Submit

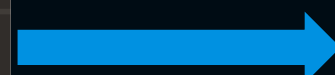
Layout

Appearance

Behavior

Action: Submit Page

Execute Validations:



Employee

Employee Name

Employee Name must have some value.

No validation executed

# Error Message

Validation

Filter

Identification

Name: P4\_ENAME not null

Execution

Validation

Editable Region: - Select -

Type: Item is NOT NULL

Item: P4\_ENAME

Always Execute:

Error

Error Message: ~~Employee~~ Name must have some value.

Display Location: Inline with Field and in Notification

Associated Item: P4\_ENAME

Validation

Filter

Identification

Name: P4\_ENAME not null

Execution

Validation

Editable Region: - Select -

Type: Item is NOT NULL

Item: P4\_ENAME

Always Execute:

Error

Error Message: #LABEL# must have some value.

Display Location: Inline with Field and in Notification

Associated Item: - Select -

Validation

Filter

Identification

Name: ENAME not null

Execution

Validation

Editable Region: Employees

Type: Column is NOT NULL

Column: ENAME

Always Execute:

Error

Error Message: #COLUMN\_HEADER# must have a value.

Display Location: Inline with Field and in Notification

Associated Column: ENAME

# Associated Item

**Validation**

Editable Region: - Select -

Type: Function Body (return)

Language: PL/SQL

PL/SQL Function Body: `return false;`

Always Execute:

**Error**

Error Message: #LABEL# is ok.

Display Location: Inline with Field and in Noti

**Associated Item**: - Select -

**Employee**

Employee Name: ALLEN

Job: SALESMAN

**1 error has occurred**

- #LABEL# is ok. ⓘ

# Associated Item

**Validation**

Editable Region: - Select -

Type: Function Body (return)

Language: PL/SQL

PL/SQL Function Body: `return false;`

Always Execute:

**Error**

Error Message: #LABEL# is ok.

Display Location: Inline with Field and in Noti

**Associated Item**: P4\_ENAME

**Employee**

Employee Name  
BLAKE1


Employee Name is not ok.

**1 error has occurred**

- Employee Name is not ok.

# Display Location

Error

Error Message 

#LABEL# is ok.

Display Location **Inline with Field and in Noti** ▾


Associated Item

Server-side

When Button Pressed - Select - ▾ >


Type - Select - ▾ ☰


- Inline with Field and in Notification
- Inline with Field
- Inline in Notification
- On Error Page


Employee 

Employee Name  
BLAKE1

Employee Name is not ok.


**1 error has occurred**  
• Employee Name is not ok. 




Employee 


Employee Name  
BLAKE1


Employee Name is not ok.

Employee 

Employee Name  
BLAKE1


**1 error has occurred**  
• Employee Name is not ok. 



Employee 

Employee Name  
CLARK

**1 error has occurred**  
• #LABEL# is not ok.



# Column Attributes

**COUNTRIES**

Columns Data Indexes Constraints Grants Statistics Triggers

+ Add Column Modify Column Rename Column Drop Column

Column Name	Data Type	Nullable
COUNTRYNO	NUMBER	N
CNAME	VARCHAR2(100 CHAR)	N

- will be created automatically based on your data model

Page Item

Filter

**Identification**

Name: P13\_CNAME

Type: Text Field

**Label**

Label: Cname

**Settings**

**Layout**

**Appearance**

Template: Required - Floating

Template Options: Use Template Defaults

CSS Classes

Icon

Format Mask

Width: 32 characters

Value Placeholder

**Validation**

Value Required:

Maximum Length: 100 characters

# Client-Side Validations

# Dynamic Actions

- no “validate” Actions
- possible alternatives
  - Execute JavaScript Code
  - Notification
    - Alert
    - Confirm

The screenshot displays the APEX Dynamic Actions configuration interface. The left pane shows the configuration for an action named "Show". The "Affected Element(s)" is set to "Item(s)". The right pane shows a list of available actions categorized into AI, Component, Execute, Miscellaneous, Navigation, Notification, and Style.

Category	Actions
AI	Generate Text With AI, Show AI Assistant
Component	Clear, Close Region, Collapse Tree, Disable, Enable, Expand Tree, Hide, Open Region, Refresh, Set Focus, Set Value, Show, Trigger Geocoding
Execute	Download, Execute JavaScript Code, Execute Server-side Code, Print Report
Miscellaneous	Cancel Event, Get Current Position, Share
Navigation	Cancel Dialog, Close Dialog, Submit Page
Notification	Alert, Confirm
Style	Add Class, Remove Class, Set Style

# Interactive Grid

### Employees

Search: All Text Columns

<input type="checkbox"/>	<input type="checkbox"/>	Ename	Job
<input type="checkbox"/>	<input type="checkbox"/>	CLARK	MANAGER
<input type="checkbox"/>	<input type="checkbox"/>	JONES	MANAGER
<input type="checkbox"/>	<input type="checkbox"/>	BLAKE1	MANAGER
<input checked="" type="checkbox"/>	<input type="checkbox"/>	ALLEN	SALESMAN
<input type="checkbox"/>	<input type="checkbox"/>	WARD	SALESMAN

- Only visible in edit mode!

**Identification**

Column Name: ENAME

Type: Text Field

**Heading**

Heading: Ename

Alignment: [Left] [Center] [Right]

Alternative Label: [ ]

**Settings**

**Layout**

**Accessibility**

**Appearance**

CSS Classes: [ ]

Icon: [ ]

Format Mask: [ ]

Width: [ ] characters

Value Placeholder: [ ]

**Validation**

Value Required:

Maximum Length: 50 characters

# Custom JavaScript (1)

- APEX JavaScript API
  - apex.message
    - addVisibilityCheck
    - alert
    - ariaAlertMessage
    - ariaMessage
    - clearErrors
    - confirm
    - hidePageSuccess
    - setDismissPreferences
    - setThemeHooks
    - showErrors
    - showPageSuccess

## Custom JavaScript (2)

- APEX JavaScript API
  - apex.item
    - `getValidity()`                    <= only HTML5 or structural validations, no content validation
    - `getValidationMessage()`       <= only HTML5 or structural validations, no content validation
  - apex.date
    - many different useful functions
  - apex.locale
    - validate number values
  - apex.server.process
    - use for validation function implementet in PL/SQL

## Custom JavaScript (3)

### ■ JavaScript API

- apex.page
  - cancelWarnOnUnsavedChanges
  - confirm
  - isChanged
  - submit
  - validate
  - warnOnUnsavedChanges

```
apex.jQuery( "#checkButton" ).click( function() {  
    if ( !apex.page.validate() ) {  
        alert("Please correct errors");  
    }  
});
```

```
apex.page.warnOnUnsavedChanges( "Employee has been changed" );
```

# Conclusion\_

## Conclusion

# Mixing Up Components

EMP / DEPT Demo

**2 errors have occurred**

- Dname must have some value.
- Value Required in Item Checked must have some value.

Department

Search: All Text Columns Go Actions Reset

Edit Save Add Row

<input type="checkbox"/>	<input type="checkbox"/>	Dname	Loc
<input type="checkbox"/>	<input type="checkbox"/>	ACCOUNTING	NEW YORK
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>		
<input type="checkbox"/>	<input type="checkbox"/>	RESEARCH	DALLAS
<input type="checkbox"/>	<input type="checkbox"/>	SALES	CHICAGO
<input type="checkbox"/>	<input type="checkbox"/>	OPERATIONS	BOSTON

1 rows selected Total 5

**Custom Fields**

Value Required in Item Checked

Value Required in Item Checked must have some value.

Declarative Validation

Standart Submit Button

EMP / DEPT Demo

**1 error has occurred**

- Declarative Validation must have some value!

Departments (10)

Search: All Text Columns Go Actions Reset

Edit Save Add Row

<input type="checkbox"/>	<input type="checkbox"/>	Dname	Loc
<input type="checkbox"/>	<input type="checkbox"/>	ACCOUNTING	NEW YORK
<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	sdf	
<input type="checkbox"/>	<input type="checkbox"/>	RESEARCH	DALLAS
<input type="checkbox"/>	<input type="checkbox"/>	SALES	CHICAGO
<input type="checkbox"/>	<input type="checkbox"/>	OPERATIONS	BOSTON

1 rows selected Total 5

**Custom Fields**

Value Required in Item Checked

asdf

Declarative Validation





Declarative Validation must have some value!

Standart Submit Button

# Conclusion

1. Prevent errors before using Validations
2. Use a restrict Datamodel for Data Quality
3. If possible, show Validations on when entering data
4. Don't combine Validation "attributes" with declarative Validations
5. Don't combine Client-Side and Server-Side Validations

Say Hy\_

-  +49 (0) 2102 30 961-0
-  [oliver.lemm@hyand.com](mailto:oliver.lemm@hyand.com)
-  [@OliverLemm](#)
-  [@oliverlemm.bsky.social](#)





**Please fill in your  
evaluations**



© 2026 – The thoughts and ideas developed are the intellectual property of Hyand and are subject to copyright. Reproduction, disclosure to third parties or use – even in part – is only permitted with the express consent of Hyand.