



## **Oracle Application Express: Developing Database Web Applications**

### **Hands-On-Labs Guide**

#### **Unit 10: Adding Computations, Processes, and Validations**

This exercise includes four hands-on-labs.

HOL 10-1 uses the Demo Projects application.

HOL 10-2, HOL 10-3, and HOL 10-4 use the Budget App application.

**HOL 10-1 Implementing Validations on the Maintain Project Page:** In this lab, you create and use validations on the Maintain Project page.

**HOL 10-2 Creating and Using a Computation:** In this lab, you create a computation that changes the value of a field entered using the form to uppercase after the page is submitted

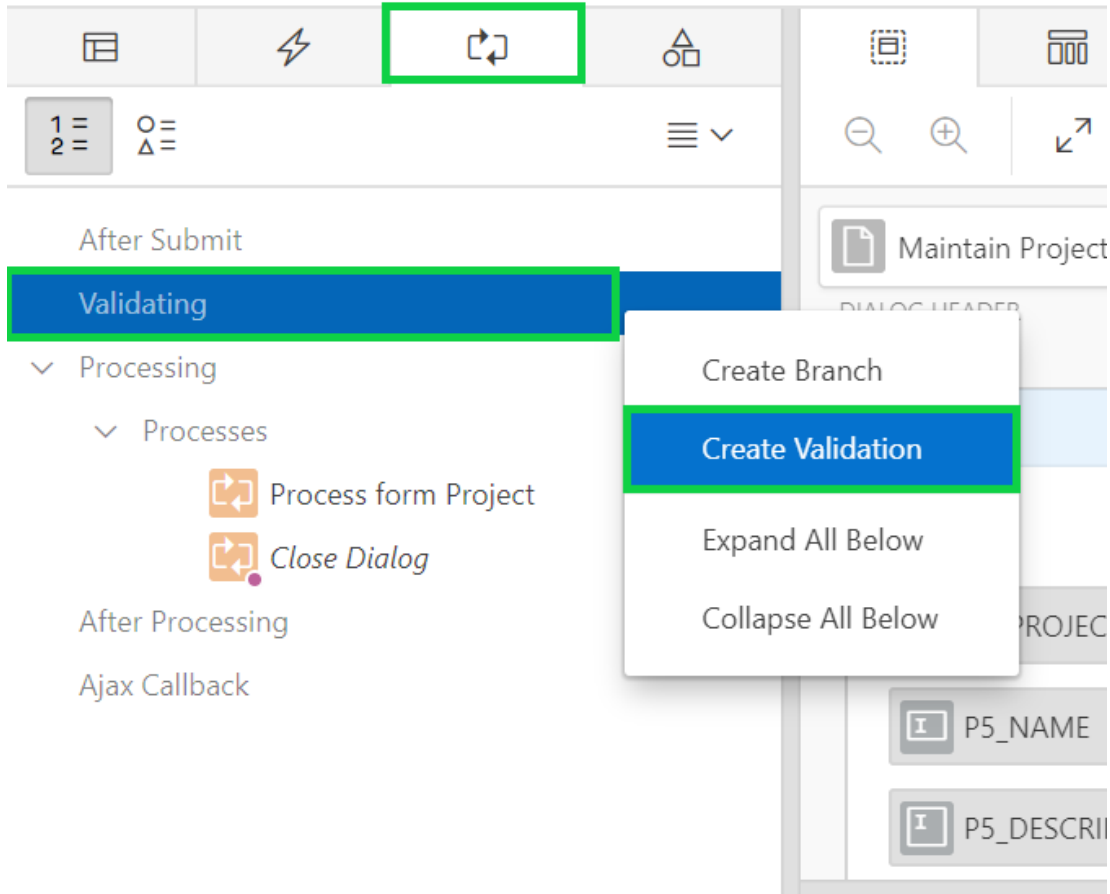
**HOL 10-3 Creating and Using a Process:** This lab covers creating an After Submit process on the Project Budget form page

**HOL 10-4 Creating and Using Validations:** In this hands-on lab, you create a not null validation and a validation to ensure that the item is numeric.

## HOL 10-1: Implementing Validations on the Maintain Project Page

In this lab, you add validations to the Maintain Projects page so that when a user changes the status to *Completed*, they add the Completed Date and that it is not forward-dated.

1. Navigate to **App Builder** and run the **Demo Projects** application.
  - a) In the navigation menu, click **DEMO\_PROJECTS**. In the Developer Toolbar, click **Edit Page 4**.
  - b) Under Rendering, expand Columns, and select **STATUS\_CD**.
  - c) In the property editor, replace the Heading with **Status**.
  - d) Click **Save and Run Page**.
  - e) In the interactive report, click a project name.
  - f) In the Developer Toolbar, click **Edit Page 5**.
2. In the page designer, click the **Processing** tab.  
Right-click **Validating** and select **Create Validation**.



In the central pane, click the **Help** tab.

**Note:** The help text displays the currently selected attribute in the Property Editor (right pane). For example, clicking on the Validation Type attribute label (not the select list) displays a list of all the available options, together with a description of when each option will pass (not display the error message) or fail (displays the error message).

3. In the property editor:

- Identification: Name - enter **Completed Date is Not Null**
- Validation: Type - select **Item is NOT NULL**
- Validation: Item - select **P5\_COMPLETED\_DATE**
- Error: Error Message - select the Error Message attribute label (not the data entry area), at which point the Help pane will display the help text for Error Message. On the Help pane under Examples, copy and paste **#LABEL# must have some value.** into the Property Editor attribute
- Server-side Condition: Type - select **Item = Value**
- Server-side Condition: Item - select **P5\_STATUS\_CD**
- Server-side Condition: Value - enter **COMPLETED**

**Note:** This condition ensures that the validation only fires when the Status item is COMPLETED.

**Error Message**

Enter the text to be displayed in the event that the validation does not pass.

You can use #LABEL# as placeholder for the label of the associated page item or #COLUMN\_HEADER# for the heading of the associated editable region column. The placeholders are not supported if error display location is set to On Error Page.

**Examples**

#LABEL# must have some value.

**Additional Information**

- Type: Text
- Supported Substitutions: Application, Page Items and System Variables

**Validation**

Filter

**Identification**

Name: Completed Date is Not Null

**Execution Options**

Sequence: 10

**Validation**

Editable Region: - Select -

Type: Item is NOT NULL

Item: P5\_COMPLETED\_DATE

Always Execute: ☐

**Error**

Error Message: #LABEL# must have some value.

**Validation**

Filter

**Server-side Condition**

When Button Pressed: - Select -

Type: Item = Value

Item: P5\_STATUS\_CD

Value: COMPLETED

Note: The #LABEL# text within the Error Message will be substituted with the Label of the associated item. In this manner, if the item label is updated the error message will also reflect the updated value. This improves consistency and prevents messages having an out of date label reference.

4. In the Processing pane, right-click **Validating** and select **Create Validation**.
5. In the property editor:
  - Identification: Name - enter **Completed Date is not Forward Dated**
  - Validation: PL/SQL Expression - enter **:P5\_COMPLETED\_DATE <= sysdate**
  - Error: Error Message - enter **#LABEL# cannot be forward dated**
  - Error: Associated Item - select **P5\_COMPLETED\_DATE**
  - Server-side Condition: Type - select **Item is NOT NULL**
  - Server-side Condition: Item - select **P5\_COMPLETED\_DATE**

The screenshot shows the 'Validation' property editor in Oracle APEX. The 'Identification' section has a 'Name' field with the value 'Completed Date is not Forward Dated'. The 'Execution Options' section is expanded, showing the 'Validation' sub-section. In this sub-section, the 'Editable Region' is set to '- Select -', the 'Type' is 'PL/SQL Expression', and the 'PL/SQL Expression' field contains the text ':P5\_COMPLETED\_DATE <= sysdate'. At the bottom, there is a toggle switch for 'Always Execute' which is currently turned off.

Validation

Filter

Identification

Name Completed Date is not Forward Dated

Execution Options

Validation

Editable Region - Select -

Type PL/SQL Expression

PL/SQL Expression :P5\_COMPLETED\_DATE <= sysdate

Always Execute

**Error**

Error Message

#LABEL# cannot be forward dated

Display Location Inline with Field and in Notification

Associated Item P5\_COMPLETED\_DATE

**Server-side Condition**

When Button Pressed - Select -

Type Item is NOT NULL

Item P5\_COMPLETED\_DATE

Click **Save**.

6. Now that you created the two validations, you want to verify if they work as expected. Navigate to the Demo Projects application runtime environment.  
In the navigation menu, click **DEMO\_PROJECTS**. Then, select a project name.
7. Try out these functions to see how the page responds:

- a) Save a record with a Status of COMPLETED and no Completion Date - An error message should be displayed.

**Maintain Project**

Project Lead  
Lucille Beatie

Name  
Develop Production Partner Portal

Description  
Develop the production app that partners can use to work more collaboratively with us.

Status Cd  
Completed

Completed Date

1 error has occurred

Completed Date must have some value.

Completed Date must have some value.

Audit Details

Cancel

Delete

Apply Changes



- b) Save a record with a Completion Date in the future - An error message should be displayed.

### Maintain Project

Project Lead  
Lucille Beatie

1 error has occurred

- Completed Date cannot be forward dated

Name  
Develop Production Partner Portal

Description  
Develop the production app that partners can use to work more collaboratively with us.

Status Cd  
Completed

Completed Date  
11/29/2019

Completed Date cannot be forward dated

Audit Details

Cancel

Delete

Apply Changes

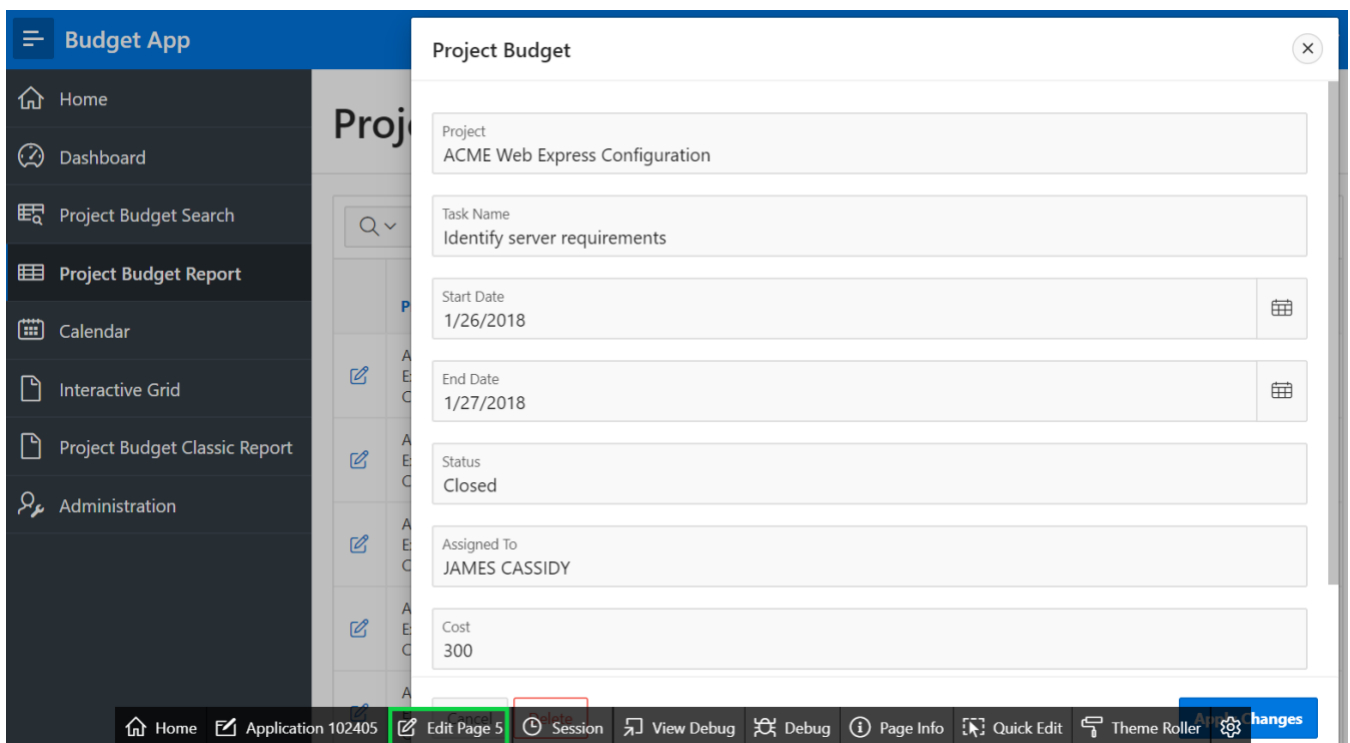
## HOL 10-2: Creating and Using a Computation

In this lab, you create a computation that changes the Assigned To value entered using the form to uppercase after the page is submitted.

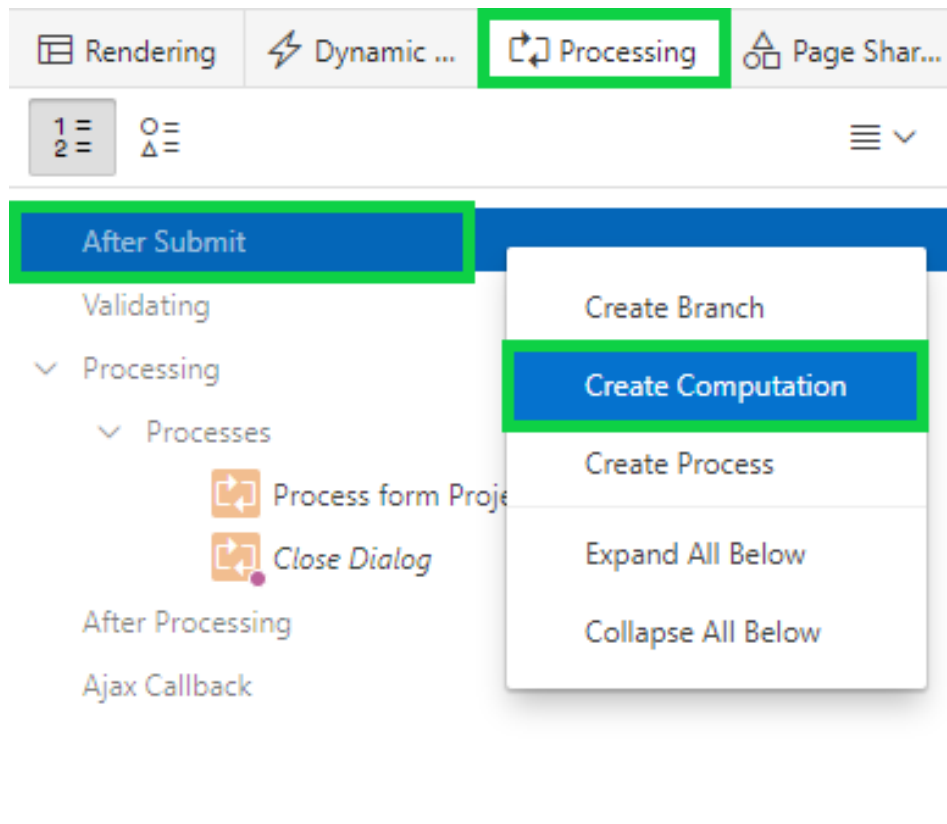
1. Navigate to **App Builder** and run the **Budget App** application.
2. In the navigation menu, click **Project Budget Report**. This is the interactive report that has been created when you created the application using a spreadsheet in HOL 3-2.
3. Create an After Submit PL/SQL Expression computation on the P5\_ASSIGNED\_TO item. The PL/SQL expression should be: **upper(:P5\_ASSIGNED\_TO)**. Test the page to see that the Assigned To is stored as uppercase.

Perform the following steps:

- a) In the PROJECT\_BUDGET interactive report, click the **Edit** icon (pencil) for any record. Then, in the Developer Toolbar, click **Edit Page 5**.



- b) In the page designer, click **Processing**. Right-click **After Submit** and select **Create Computation**.



c) In the property editor:

- Identification > Item Name: Select **P5\_ASSIGNED\_TO**
- Computation > Type: Select **PL/SQL Expression**
- Computation > PL/SQL Expression: Enter **upper(:P5\_ASSIGNED\_TO)**

The screenshot shows the 'Computation' dialog page in Oracle APEX. It has three main sections: Identification, Execution Options, and Computation. In the Identification section, the 'Item Name' is 'P5\_ASSIGNED\_TO'. In the Execution Options section, the 'Sequence' is '10' and the 'Point' is 'After Submit'. In the Computation section, the 'Type' is 'PL/SQL Expression' and the expression is 'upper(:P5\_ASSIGNED\_TO)'. Green boxes highlight the 'Item Name', 'Type', and the expression text area.

Click **Save**.

d) This is a dialog page and so you cannot run this directly. You need to first run interactive report page.

Navigate to application runtime environment and click **Project Budget Report** in the navigation menu and then click the edit icon for a record.

e) Click **Apply Changes**. The data is submitted now.

f) In the report, notice that the value for Assigned To is stored in uppercase.

## Project Budget Report





Q

Go

Actions

Reset

Create

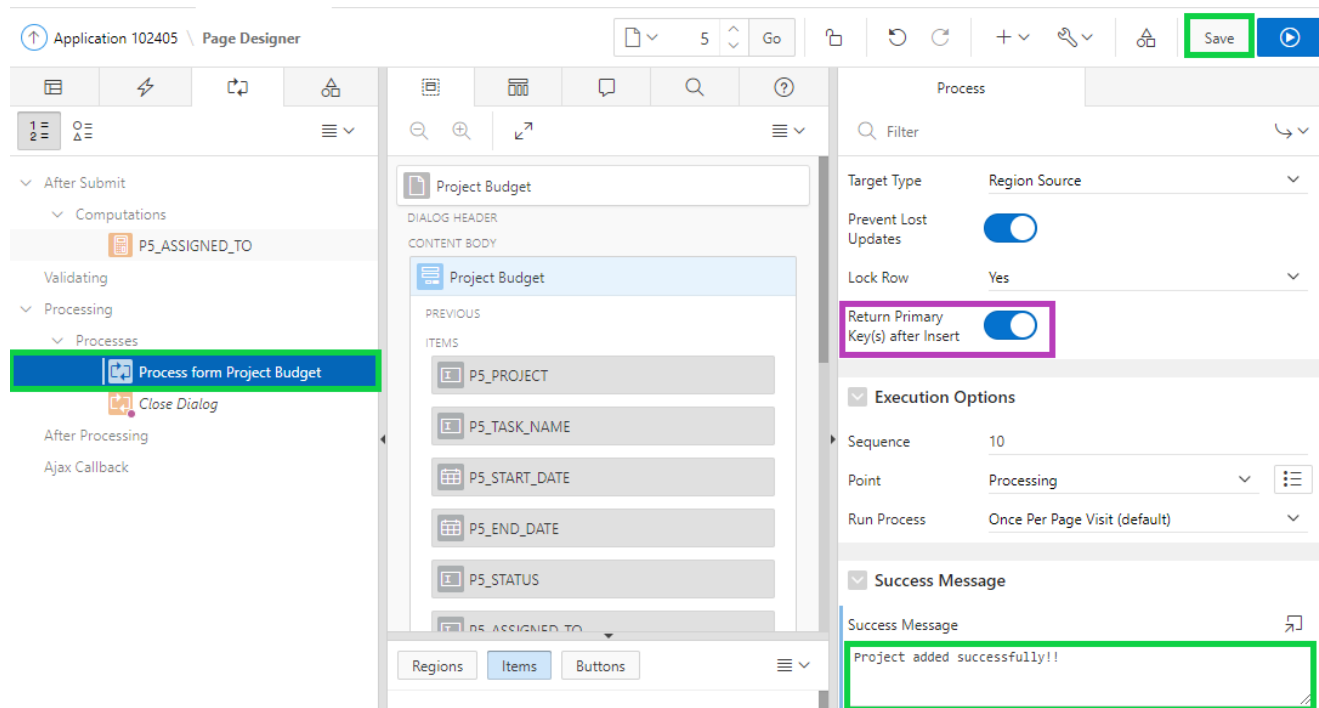
	Project	Task Name	Start Date	End Date	Status	Assigned To	Cost	Budget
	ACME Web Express Configuration	Identify server requirements	1/26/2019	1/27/2019	Closed	JOHN WATSON	300	500
	ACME Web Express Configuration	Determine Web listener configuration(s)	1/28/2019	1/28/2019	Closed	James Cassidy	600	800
	ACME Web Express Configuration	Run installation	1/31/2019	1/31/2019	Closed	James Cassidy	200	200
	ACME Web Express Configuration	Create pilot workspace	2/2/2019	2/2/2019	Closed	John Watson	100	100

## HOL 10-3: Creating and Using a Process

In this lab, you create a process to insert a record into a table called AUDIT\_DETAILS every time a project is added.

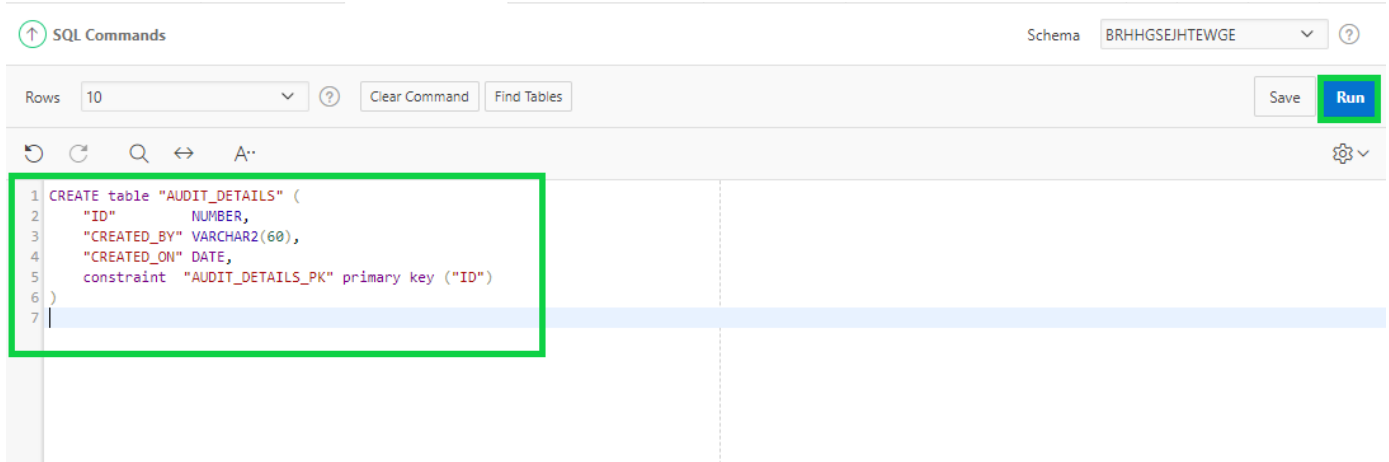
1. You want to create an After Submit process on the Budget form page. Whenever a user adds a new project by using this form, the user details are stored in the AUDIT\_DETAILS table. Perform the following steps:

- a) View **Page 5** in page designer
- b) Click the Processing tab and navigate to **Processing > Processes**.
- c) Select **Process form Project Budget**.  
In the property editor, note that **Return Primary Key(s) after Insert** is **Enabled** by default.  
Under Success Message > Success Message: Enter **Project added successfully!**  
Click **Save**.

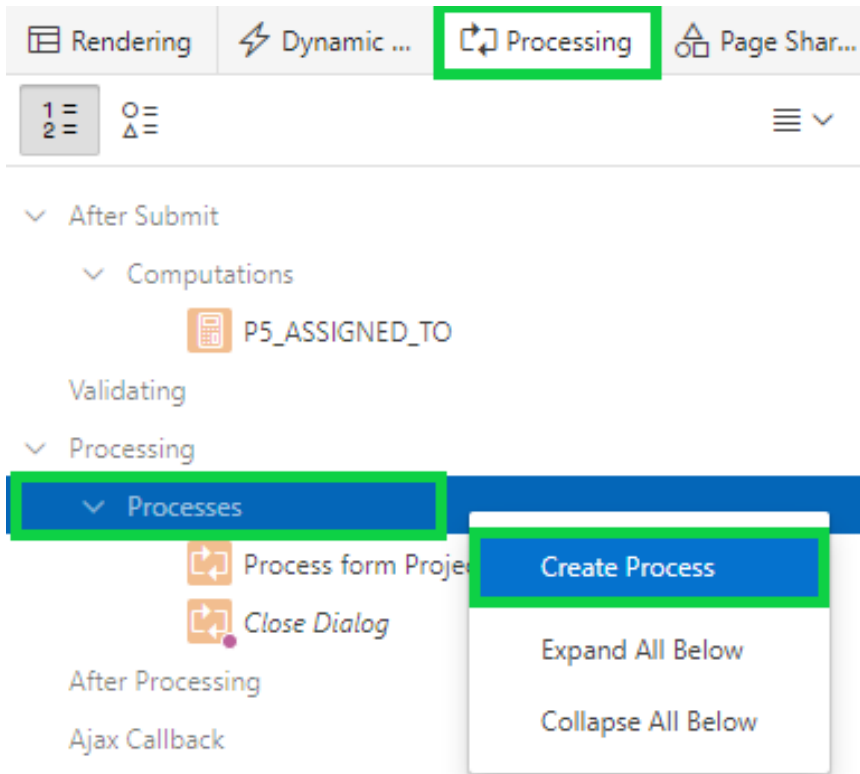


2. Navigate to SQL Workshop and create the AUDIT\_DETAILS table.
  - a) Click **SQL Workshop** and select **SQL Commands**.
  - b) Copy the following SQL and paste it in the SQL Commands area and click **Run**.

```
CREATE table "AUDIT_DETAILS" (  
    "ID"          NUMBER,  
    "CREATED_BY" VARCHAR2(60),  
    "CREATED_ON"  DATE,  
    constraint "AUDIT_DETAILS_PK" primary key ("ID")  
)
```



- c) The table is created. Now, navigate to the window or tab of the Budget App application runtime environment.  
In the navigation menu, click **Project Budget Report**.  
Click the Edit icon for a record and then in the Developer Toolbar, click **Edit Page 5**.
3. Click the **Processing** tab.  
Navigate to Processing, right-click **Processes** and select **Create Process**.



4. In the property editor:

- Identification > Name: Enter **Audit Details**
- Source > PL/SQL Code: Copy and paste the following code:

```
INSERT INTO audit_details VALUES (:P5_ID, :APP_USER,
sysdate) ;
```

- Execution Options > Sequence: Enter **40**
- Success Message > Success Message: Enter **Success!**
- Error > Error Message: Enter **Error!**
- Server-side Condition > When Button Pressed: Select **CREATE**

Click **Save**.

**Note:** The sequence of this process should be the one immediately after the Process Row of Process form Project Budget process.



Process

Filter

▼ Identification

Name

Audit Details

Type

PL/SQL Code

▼

Editable Region

- Select -

▼

▼ Source

Location

Local Database

▼

PL/SQL Code

```
INSERT INTO audit_details VALUES(:P5_ID, :APP_USER, sysdate);
```

▼ Execution Options

Sequence

40

Point

Processing

▼

Run Process

Once Per Page Visit (default)

▼

The screenshot displays the configuration interface for a page item in Oracle APEX. It is divided into three main sections: 'Success Message', 'Error', and 'Server-side Condition'. The 'Success Message' section has a text area containing 'Success!'. The 'Error' section has a text area containing 'Error!'. Below these is a 'Display Location' dropdown menu set to 'Inline in Notification'. The 'Server-side Condition' section contains two dropdown menus: 'When Button Pressed' set to 'CREATE' and 'Type' set to '- Select -'. Each dropdown menu has a corresponding button to its right.

**Success Message**

Success Message

Success!

**Error**

Error Message

Error!

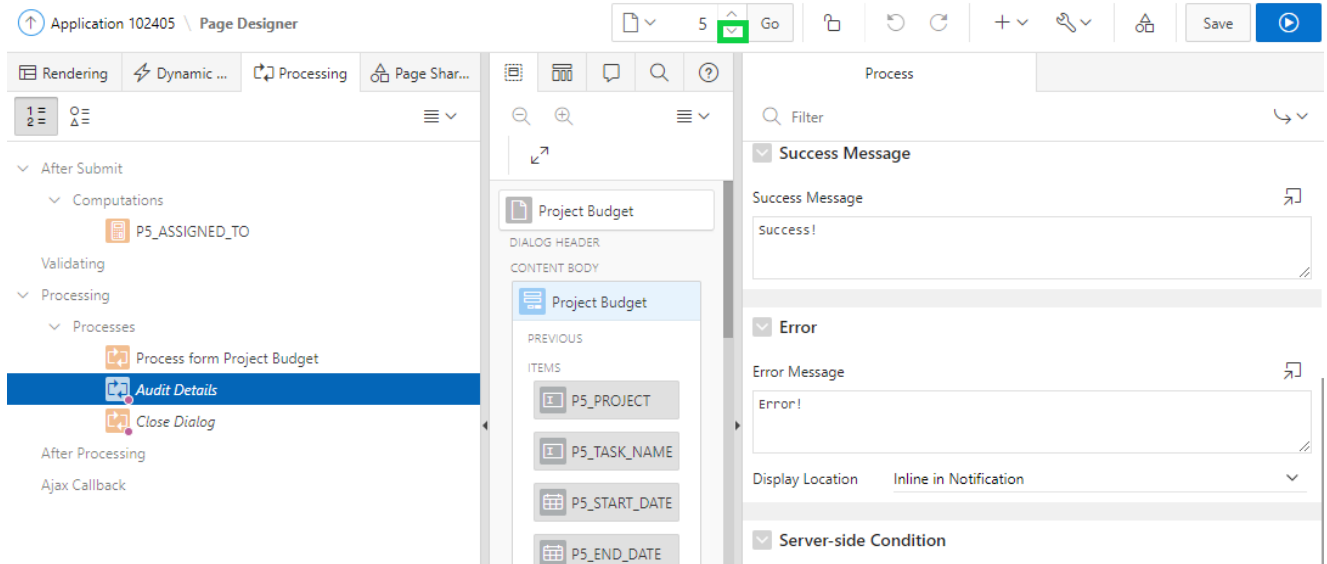
Display Location    Inline in Notification

**Server-side Condition**

When Button Pressed    CREATE

Type    - Select -



- Now, you want to verify if the processes are working. In the page designer toolbar, click the Navigate to Previous Page arrow.



6. Click **Save and Run Page**, then click the **Create** button.
7. In the Budget dialog, enter values for each of the fields and click **Create**.
  - Project: **New Packaged App**
  - Task Name: **Prepare requirements doc**
  - Start Date: **11/1/2019**
  - End Date: **11/25/2019**
  - Status: **Open**
  - Assigned To: **Pam King**
  - Cost: **500**
  - Budget: **800**

**Project Budget** ✕

---

Project	New Packaged App
Task Name	Prepare requirements doc
Start Date	11/1/2019 
End Date	11/25/2019 
Status	Open
Assigned To	Pam King
Cost	500
Budget	800

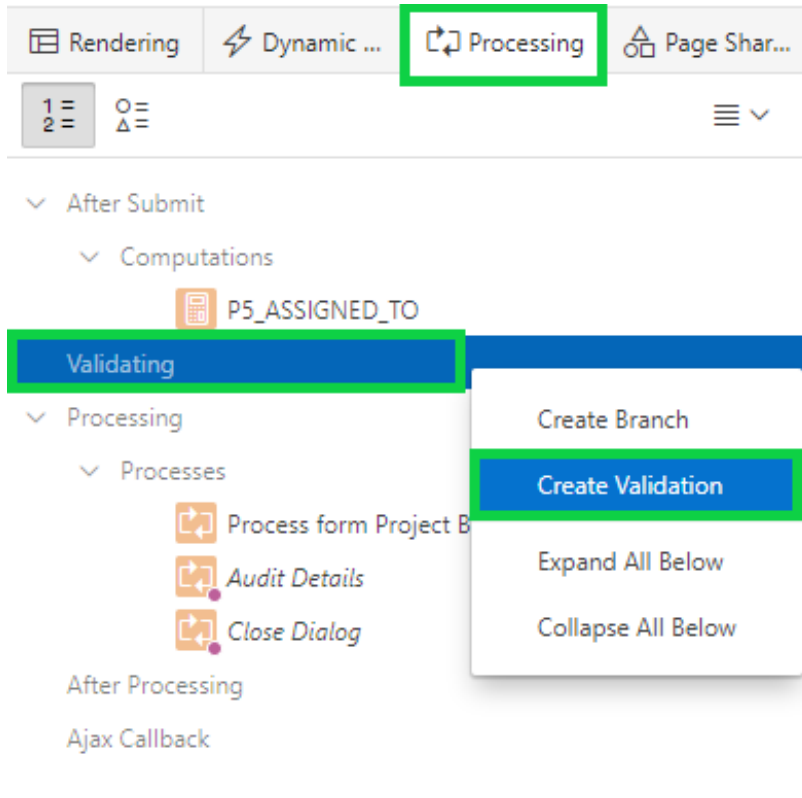
Cancel Create

- The record is created now. In the Developer Toolbar, click **Home**.
- Click **SQL Workshop**. Then, click **Object Browser**.  
Under Tables, select **AUDIT\_DETAILS**.  
Click **Data**.  
Notice that the user details have been added to the table.

## HOL 10-4: Creating and Using Validations

In this lab, you create two validations on the Project Budget form page. The first validation you create ensures that the project field is not null. The second one ensures that the value entered for cost is only numeric.

1. Navigate to the window or tab of the Budget App application runtime environment.  
In the navigation menu, click **Project Budget Report**.  
Click the Edit icon for a record and then in the Developer Toolbar, click **Edit Page 5**.
2. You want to create two validations. The first validation is to ensure that the Project field is not null.  
In the page designer, click the **Processing** tab.  
Right-click **Validating** and select **Create Validation**.



3. In the property editor:
  - Identification > Name: Enter **Project is not null**
  - Validation > Type: Select **Item is NOT NULL**
  - Validation > Item: Select **P5\_PROJECT**
  - Error > Error Message: Enter **Project is not null!**
  - Error > Display Location: Select **Inline in Notification**

Validation

Filter

▼ Identification

Name

▼ Execution Options

Sequence

▼ Validation

Editable Region

Type

Item

Always Execute ☐

▼ Error

Error Message

Display Location

4. You need to create another validation to ensure that the value entered for Cost should be only numeric.  
Under Processing, right-click **Validating** and select **Create Validation**.

5. In the property editor:
  - Identification > Name: Enter **Cost should be numeric**
  - Validation > Type: Select **Item is numeric**
  - Validation > Item: Select **P5\_COST**
  - Error > Error Message: Enter **Cost should be numeric**
  - Error > Display Location: Select **Inline with Field**
  - Error > Associated Item: Select **P5\_COST**

Click **Save**.

Validation

Filter

Identification

Name Cost should be numeric

Execution Options

Sequence 20

Validation

Editable Region - Select -

Type Item is numeric

Item P5\_COST

Always Execute ☐

Error

Error Message cost should be numeric

Display Location Inline with Field

Associated Item P5\_COST

6. Now, you can test to verify if the validations work. Navigate to the application run time environment.  
In the navigation menu, click **Project Budget Report**.  
Then, click **Create**.
7. In the Project Budget dialog, enter the following and then click the **Create** button.
  - Task Name: **Prepare requirements doc**
  - Start Date: **11/2/2019**
  - End Date: **11/25/2019**
  - Status: **Open**
  - Assigned To: **Pam King**
  - Cost: **abcd12**
  - Budget: **3000**

Project Budget

Project

Task Name  
Prepare requirements doc

Start Date  
11/3/2019

End Date  
11/25/2019

Status  
Open

Assigned To  
Pam King

Cost  
abcd12

Budget  
3000

2 errors have occurred

• Cost must be Numeric.

• Project is not null!

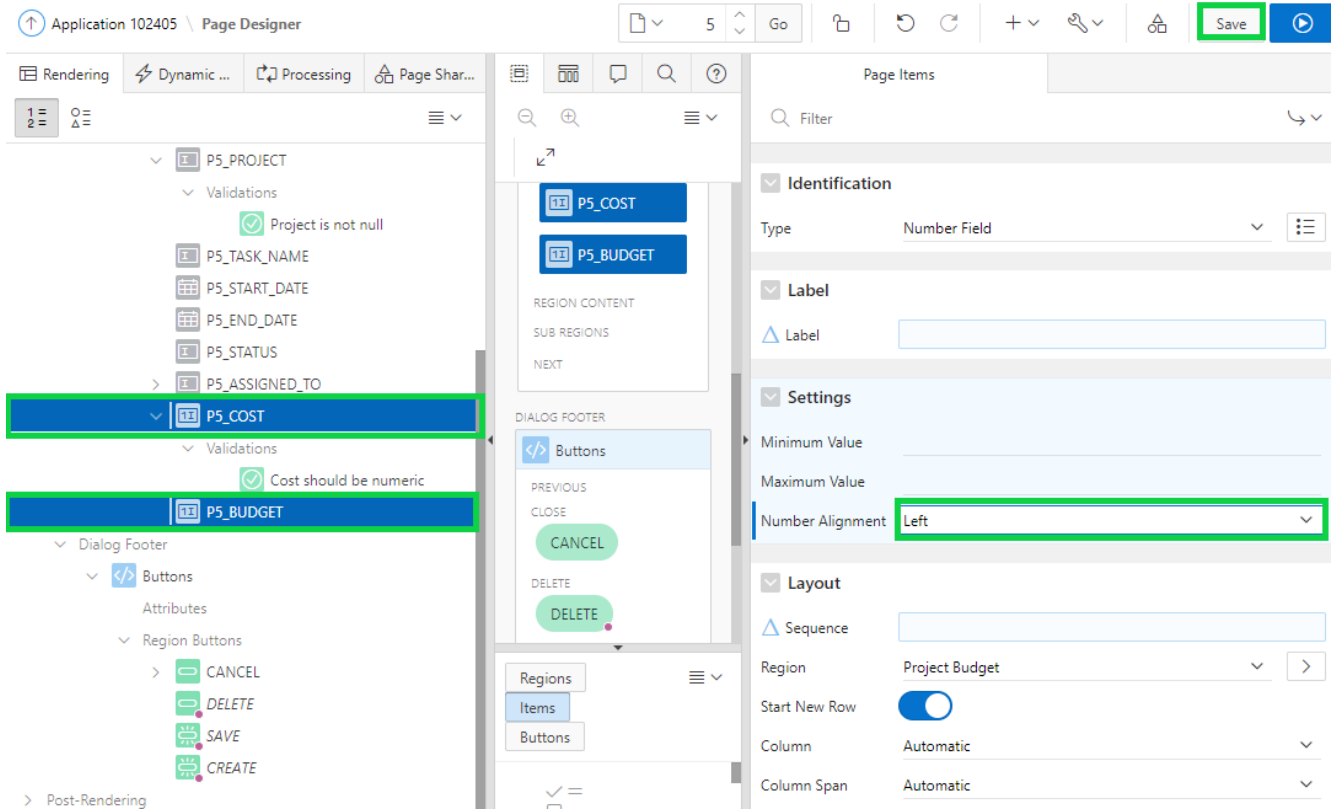
Cancel

Create

Notice that the errors are displayed.  
Click **Cancel**.



8. Navigate to Page Designer.  
Under Rendering, select **P5\_COST** and **P5\_BUDGET**.  
In the property editor, for Settings > Number Alignment, select **Left**.  
Click **Save**.



9. Navigate to the application run time environment.  
In the navigation menu, click **Project Budget Report**.  
Then, click the edit icon for any project.  
In the Budget form, now you see that the values in the Cost and Budget fields are left aligned.

## Project Budget



Project  
Migrate Desktop Application

Task Name  
Post-migration review

Start Date  
3/10/2018



End Date  
3/11/2018



Status  
Pending

Assigned To  
Bob Nile

Cost  
100

Budget  
100

Cancel

Delete

Apply Changes



| Oracle is committed to developing practices and products that help protect the environment

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2020, Oracle and/or its affiliates. All rights reserved. Licensed under the Creative Commons Attribution 4.0 International License as shown at <https://creativecommons.org/licenses/by/4.0/legalcode>.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.