



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

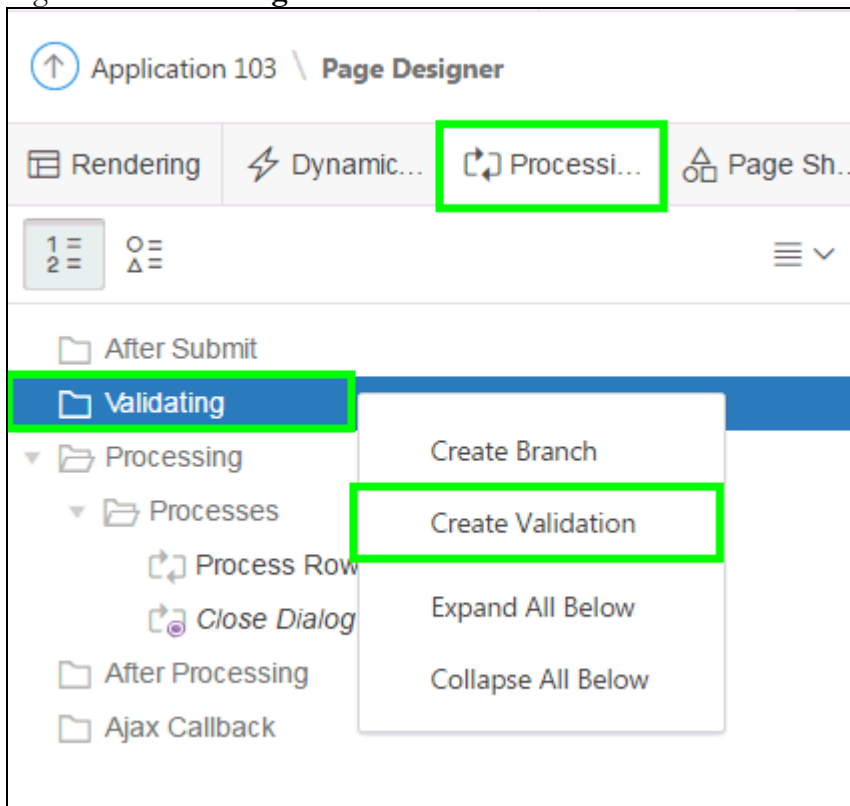
HOL 10-4 Creating and Using Validations: In this hands-on lab, you create a not null validation and a validation to ensure that the item is numeric.

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.
In the navigation menu, click **Demo Projects** and then in the Interactive Grid, click a project name.
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.

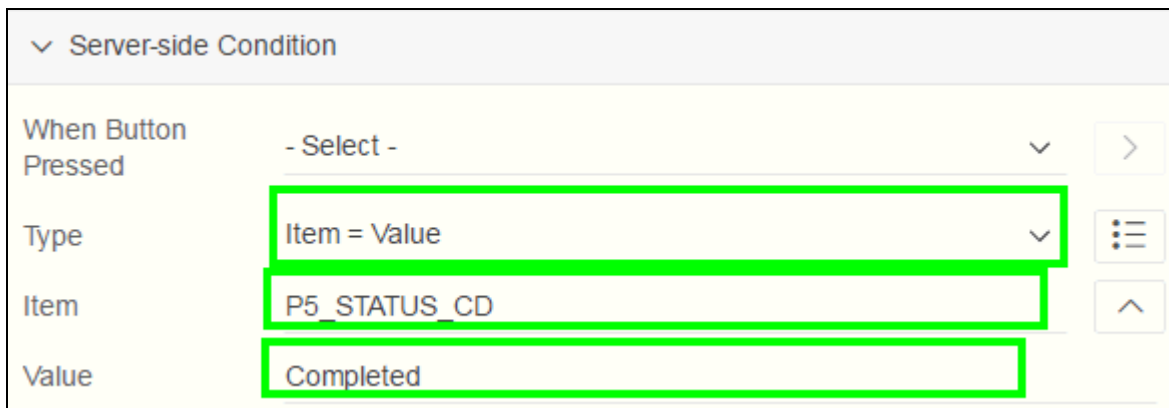
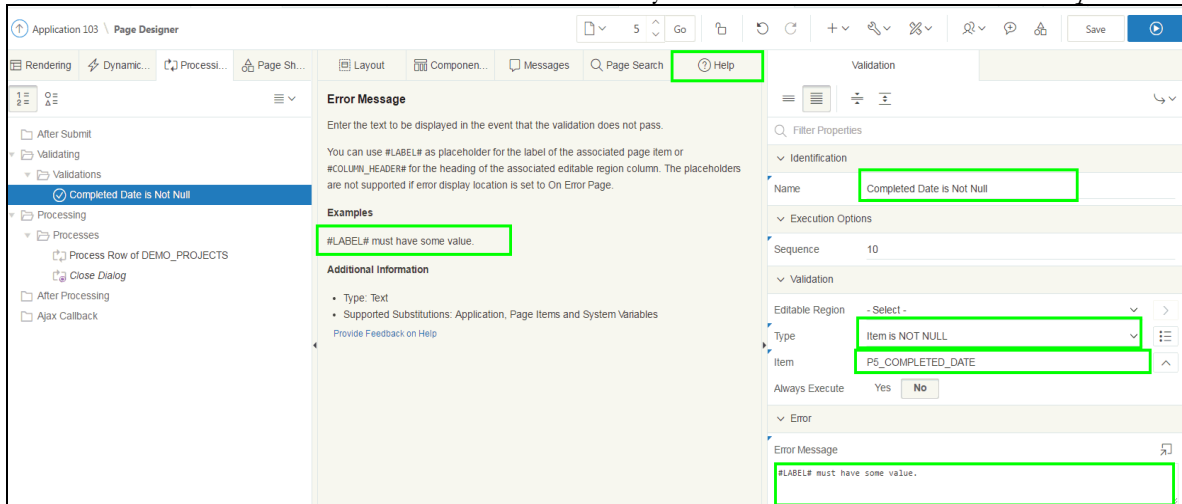
Note: The help text displays the currently selected attribute in the Property Editor (right pane). For example, clicking on the Validation Type attribute label (not the select list) displays a list of all the available options, together with a description of when each option will pass (not display the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

- Condition: Type - select **Item = Value**
- Condition: Item - select **P5_STATUS_CD**
- Condition: Value - enter **Completed**

Note: This condition ensures that the validation only fires when the Status item is *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**
 - Condition: Type - select **Item is NOT NULL**
 - Condition: Item - select **P5_COMPLETED_DATE**

Validation

Filter Properties

Identification

Name **Completed Date is not Forward Dated**

Execution Options

Sequence 20

Validation

Editable Region - Select -

Type PL/SQL Expression

PL/SQL Expression

```
:P5_COMPLETED_DATE <= sysdate
```

Always Execute Yes **No**

Error

Error Message

```
#LABEL# can not 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

- 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.
- Try out these functions to see how the page responds:
 - Save a record with a Status of Completed and no Completion Date - An error message should be displayed.

The screenshot shows the 'Maintain Project' form in the Oracle APEX application. The form is titled 'Maintain Project' and has a close button (X) in the top right corner. An orange error message banner at the top states '1 error has occurred' with a list of errors: 'Completed Date must have some value.' Below the error message, the form fields are as follows:

- Project Lead: Chaitanya (dropdown menu)
- Name: Develop Partner Portal POC (text input)
- Description: Develop a proof of concept that partners can use to work more collaboratively with us. (text area)
- Status Cd: Completed (dropdown menu)
- Completed Date: (empty date input field with a calendar icon)

At the bottom of the form, there are buttons for 'Cancel', 'Delete', and 'Save'. The 'Save' button is highlighted in blue. The background shows a table of project records with columns for Name and a search bar.

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

Maintain Project ✕

1 error has occurred ✕

- Completed Date can not be forward dated i

Project Lead

* Name

Description

Status Cd

* Completed Date

Completed Date can not be forward dated

▶ Audit Details

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**. 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 P2_ASSIGNED_TO item. The PL/SQL expression should be: `upper (: P2_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 2**.

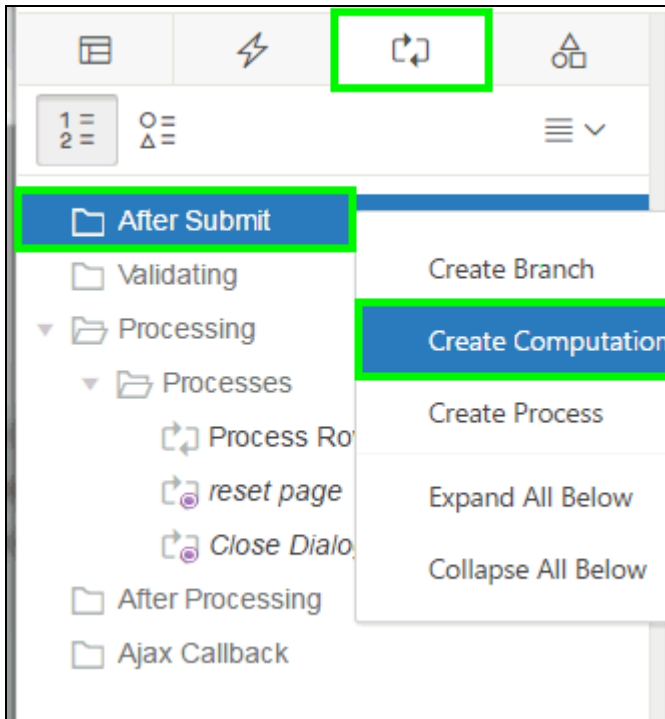
The screenshot shows the Oracle APEX interface for the 'BUDGET APP'. The navigation menu on the left includes 'Interactive Grid', 'PROJECT_BUDGET', and 'Project Budget Report'. The main area displays the 'Project Budget' interactive report with a table of records. The first record is highlighted, and its edit form is open. The form contains the following fields:

- Project: ACME Web Express Configuration
- Task Name: Create pilot workspace
- Start Date: 02-FEB-17
- End Date: 02-FEB-17
- Status: Closed
- Assigned To: John Watson
- Cost: 100
- Budget: 100

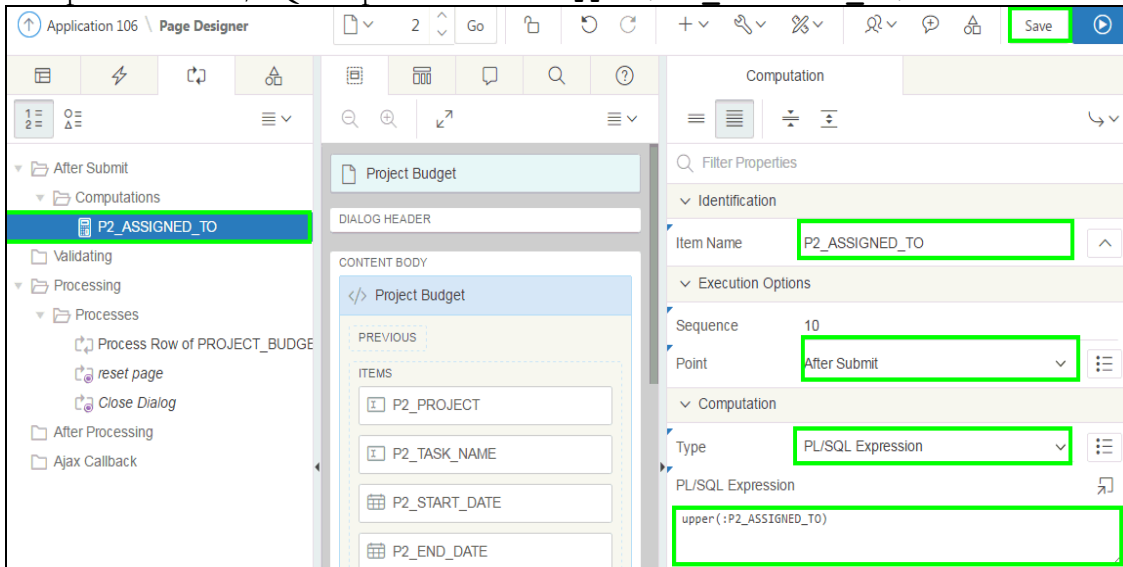
The bottom of the screen shows the Developer Toolbar with the 'Edit Page 2' button highlighted.

Id	Project	Budget
1	ACME Web Express Configuration	100
2	ACME Web Express Configuration	300
3	ACME Web Express Configuration	100
4	Bug Tracker	2000
5	Bug Tracker	1500
6	Bug Tracker	100
7	Bug Tracker	2000
8	Convert Spreadsheets	10000
9	Convert Spreadsheets	500
10	Discussion Forum	300

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



- c) In the property editor:
- Identification > Item Name: Select **P2_ASSIGNED_TO**
 - Execution Options > Point: Select **After Submit**
 - Computation > Type: Select **PL/SQL Expression**
- d) Computation > PL/SQL Expression: Enter **upper (: P2_ASSIGNED_TO)**








Click **Save**.

- e) 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 in the navigation menu and then click the edit icon for a record.

The screenshot shows the 'BUDGET APP' interface. The navigation menu on the left includes 'Interactive Grid', 'PROJECT_BUDGET' (highlighted in green), and 'Project Budget Report'. The main content area displays a table with the following data:

	Id	Project	Task Name	Start Date	End Date	Status	Assigned To	Cost	Budget
	4	Bug Tracker	Document quality assurance procedures	12-DEC-16	15-DEC-16	Closed	Myra Sutcliff	3000	2000
	5	Bug Tracker	Review automated testing tools	16-DEC-16	18-DEC-16	Closed	Myra Sutcliff	750	1500
	6	Bug Tracker	Implement bug tracking software	31-DEC-16	31-DEC-16	Closed	Myra Sutcliff	100	100
	7	Bug Tracker	Train developers on tracking bugs	01-JUL-17	01-DEC-17	On-Hold	Myra Sutcliff	1000	2000
	8	Convert Spreadsheets	Create ACME Web Express applications from spreadsheets	02-MAY-17	02-SEP-17	Open	Pam King	6000	10000

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

The screenshot shows the 'Project Budget' edit form with the following fields and values:






- Project: Bug Tracker
- Task Name: Document quality assurance procedures
- Start Date: 12-DEC-16
- End Date: 15-DEC-16
- Status: Closed
- Assigned To: Myra Sutcliff
- Cost: 3000
- Budget: 2000

At the bottom of the form, there are three buttons: 'Cancel', 'Delete', and 'Apply Changes' (highlighted in green).

- g) Navigate to application runtime environment and click PROJECT_BUDGET in the navigation menu and then click the edit icon for the same record that you did in step e above. Notice that the value for Assigned To is stored in uppercase.

Home

Q Go Actions

	Id	Project	Task Name	Start Date	End Date	Status	Assigned To	Cost	Budget
	1	ACME Web Express Configuration	Create pilot workspace	02-FEB-17	02-FEB-17	Closed	JOHN WATSON	100	100
	2	ACME Web Express Configuration	Specify security authentication scheme(s)	02-JUL-17	02-JUL-17	Open	John Watson	200	300
	3	ACME Web Express Configuration	Configure Workspace provisioning	02-AUG-17	02-AUG-17	Open	John Watson	200	100
	4	Bug Tracker	Document quality assurance procedures	12-DEC-16	15-DEC-16	Closed	MYRA SUTCLIFF	3000	2000
	5	Bug Tracker	Review automated testing tools	16-DEC-16	18-DEC-16	Closed	Myra Sutcliff	750	1500

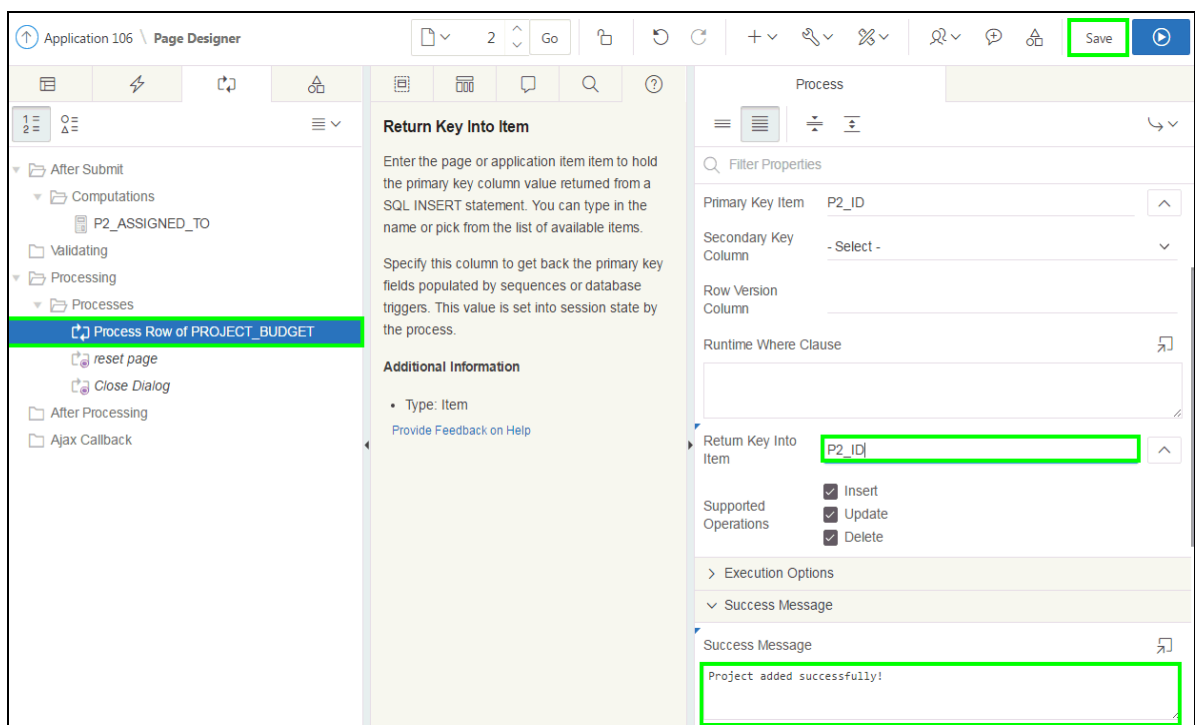
HOL 10-3: Creating and Using a Process

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 Project 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 2** in page designer
- b) Click the Processing tab, and navigate to Processing > Processes. Select **Process Row of PROJECT_BUDGET**. In the property editor:
 - Settings > Return Key Into Item: Select **P2_ID**
 - 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**.

The screenshot shows the Oracle Application Express interface. The top navigation bar includes 'ORACLE Application Express', 'App Builder', 'SQL Workshop' (highlighted with a green box), 'Team Development', and 'Packaged Apps'. Below the navigation bar, the 'SQL Commands' option in the dropdown menu is also highlighted with a green box. The main area shows the 'Page Designer' for 'Application 106'. The left sidebar displays a tree view of the page structure, with 'Process Row of PROJECT_BUDGET' selected. The right sidebar shows the 'Process' configuration panel, including 'Filter Properties', 'Secondary Key Column', 'Row Version Column', 'Runtime Where Clause', 'Return Key Into Item' (set to 'P2_ID'), 'Supported Operations' (Insert, Update, Delete), 'Execution Options' (Sequence: 30, Point: Processing, Run Process: Once Per Page Visit (default)), and 'Success Message' (Project added successfully!).

- b) Copy the following SQL and paste it in the SQL Commands area and click **Run**.

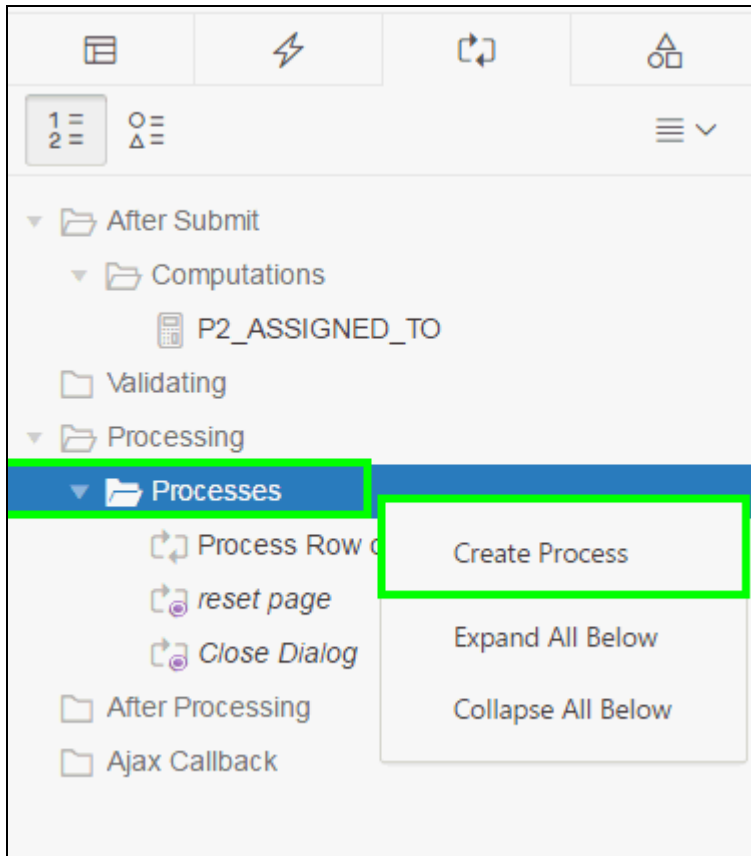
```
CREATE table "AUDIT_DETAILS" (
  "ID"          NUMBER,
  "CREATED_BY" VARCHAR2(60),
  "CREATED_ON" DATE,
  constraint "AUDIT_DETAILS_PK" primary key ("ID")
)
```

The screenshot displays the Oracle APEX SQL Workshop interface. At the top, there are navigation tabs: App Builder, SQL Workshop (selected), Team Development, and Packaged Apps. Below the tabs, the 'SQL Commands' section is active, showing the schema 'APEX_DEMO'. The 'Autocommit' checkbox is checked, and the 'Rows' limit is set to 10. The 'Run' button is highlighted in blue. The SQL command entered is:

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

Below the command, the 'Results' tab is selected, showing the message 'Table created.' and a execution time of '0.01 seconds'.

- 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**.
Click the Edit icon for a record and then in the Developer Toolbar, click **Edit Page 2**.
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**
- Identification > Type: Select **PL/SQL Code**
- Source > PL/SQL Code: Copy and paste the following code:

```
INSERT INTO audit_details VALUES (:P2_ID, :APP_USER, sysdate);
```

- Execution Options > Sequence: Enter **40**
- Execution Options > Point: Select **Processing**
- 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 PROJECT_BUDGET process.

Process

Filter Properties

Identification

Name

Type

Source

PL/SQL Code

```
INSERT INTO audit_details VALUES (:P2_ID, :APP_USER, sysdate);
```

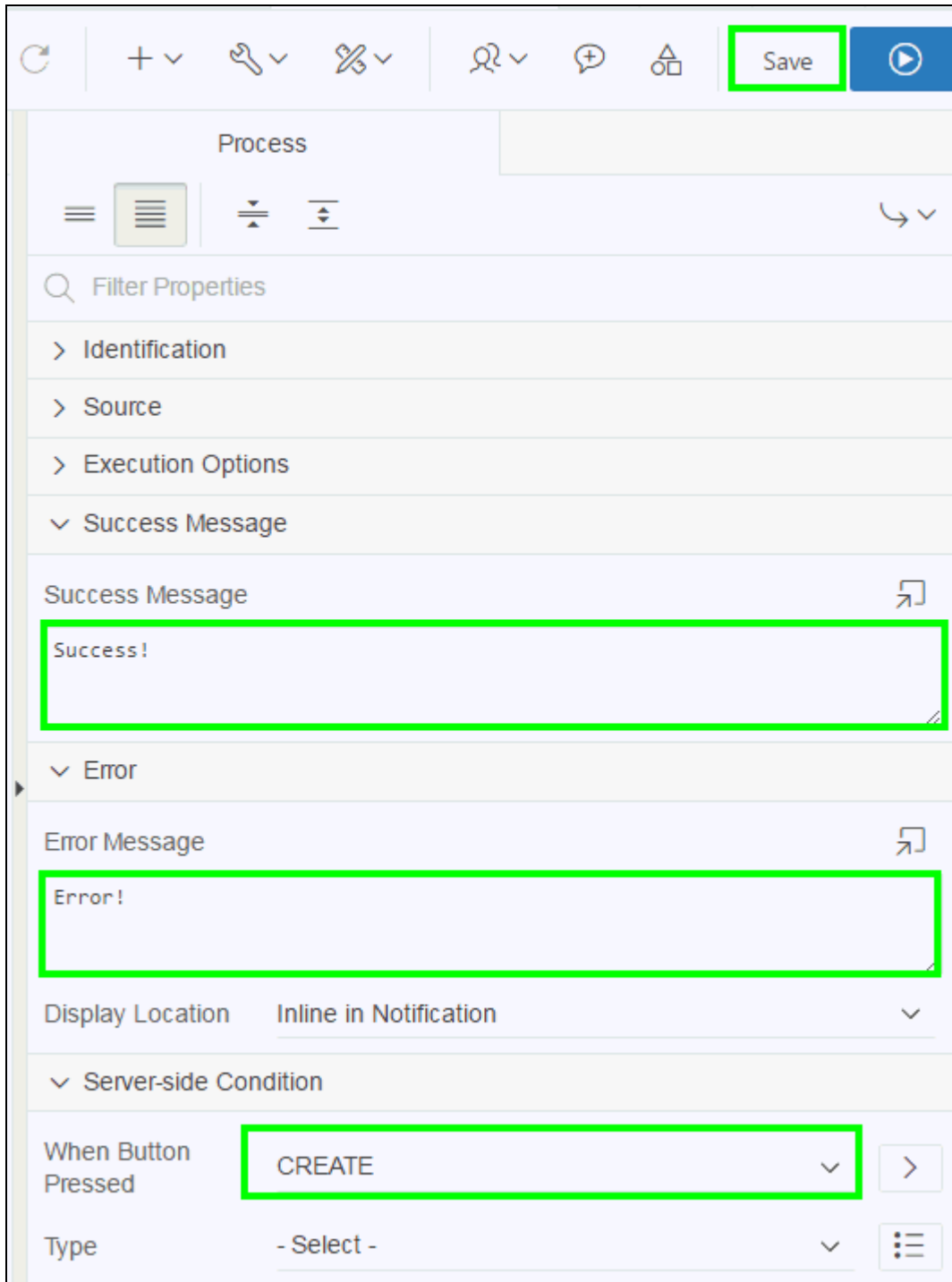
Execution Options

Sequence

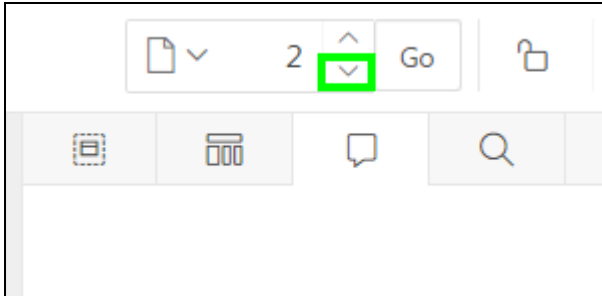
Point

Editable Region

Run Process



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



Then, click **Save and Run Page**.

6. Click the **Create** button.

Home

Q ▾ Go Actions ▾ **Create**

	Id	Project	Task Name	Start Date	End Date	Status	Assigned To	Cost	Budget
	1	ACME Web Express Configuration	Create pilot workspace	02-FEB-17	02-FEB-17	Closed	JOHN WATSON	100	100
	2	ACME Web Express Configuration	Specify security authentication scheme(s)	02-JUL-17	02-JUL-17	Open	John Watson	200	300

7. In the Project Budget dialog, enter values for each of the fields and click **Create**.

- Project: **New Packaged App**
- Task Name: **Prepare requirements doc**
- Start Date: **15-Feb-17**
- End Date: **20-Feb-17**
- Status: **Open**
- Assigned To: **Pam King**
- Cost: **500**
- Budget: **800**

Project Budget ✕

Project

Task Name

Start Date

End Date

Status

Assigned To

Cost

Budget

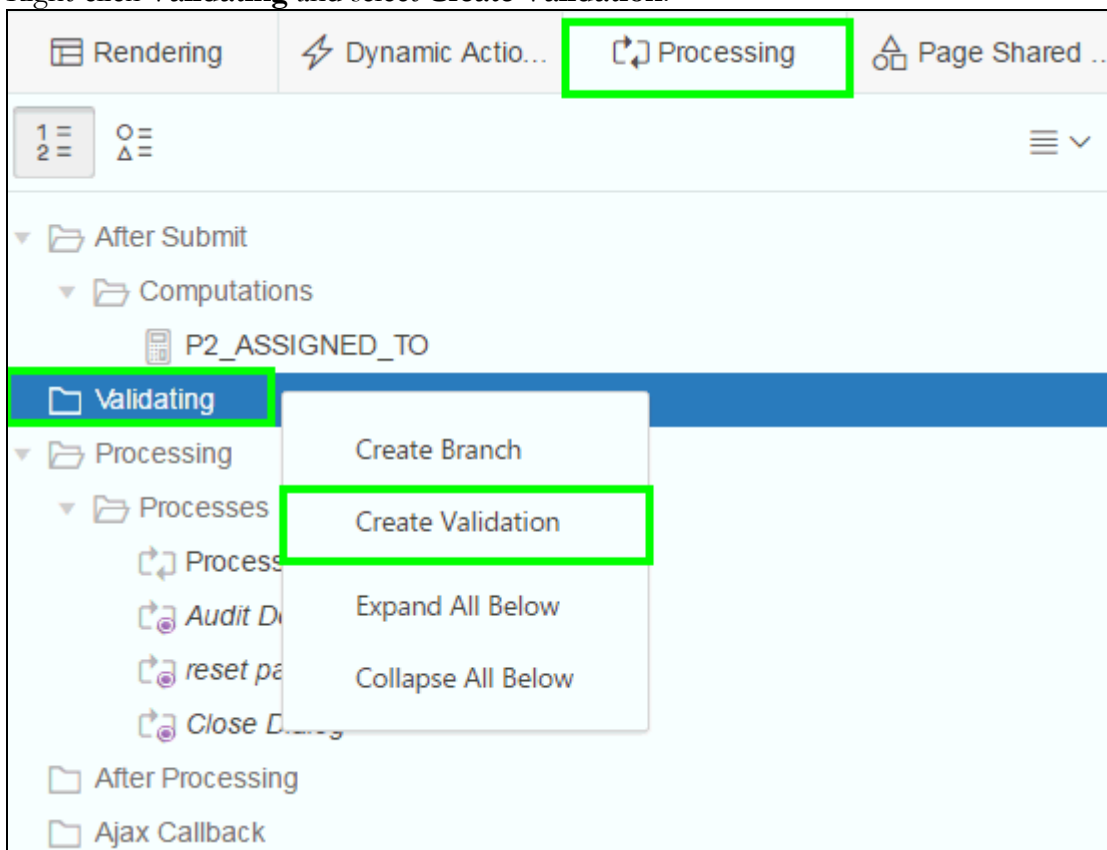
8. The record is created now. In the Developer Toolbar, click **Home**.
9. Click **SQL Workshop**. Then, click **Object Browser**.
 Under Tables, select **AUDIT_DETAILS**.
 Click **Data**.
 Notice that the user details have been added to the table.

AUDIT_DETAILS										
Table	Data	Indexes	Model	Constraints	Grants	Statistics	UI Defaults	Triggers	Dependencies	SQL
<div style="display: flex; gap: 10px;"> <input type="button" value="Query"/> <input type="button" value="Count Rows"/> <input type="button" value="Insert Row"/> </div>										
Data										
EDIT	ID	CREATED_BY	CREATED_ON							
	43	CHAITANYA	02/14/2017							

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**.
Click the Edit icon for a record and then in the Developer Toolbar, click **Edit Page 2**.
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 **P2_PROJECT**
 - Error > Error Message: Enter **Project is not null!**
 - Error > Display Location: Select **Inline in Notification**

Validation

☰ ☰
⏪ ⏩
↶ ↷

▼ Identification

Name

▼ Execution Options

Sequence

▼ Validation

Editable Region ▼ >

Type ▼ ☰

Item ▲

Always Execute

▼ Error

Error Message ↶

Project is not null!

Display Location ▼

Associated Item ▼ >

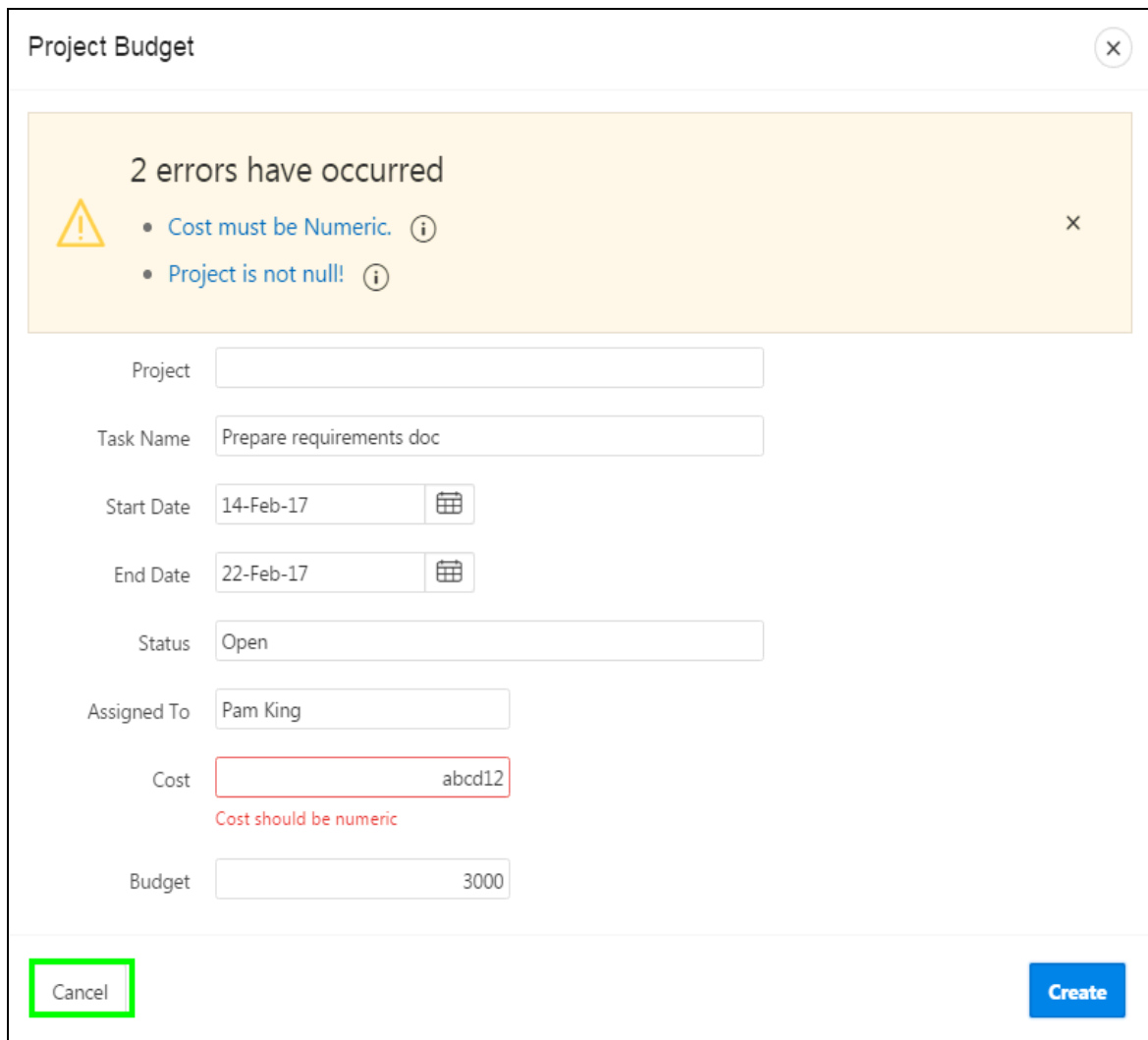
4. You need to create another validation to ensure that the value entered for Cost should be only numeric.
Under Processing, right-click **Validating** and select **Create Validation**.
5. In the property editor:
 - Identification > Name: Enter **Cost should be numeric**
 - Validation > Type: Select **Item is numeric**
 - Validation > Item: Select **P2_COST**

- Error > Error Message: Enter **Cost should be numeric**
 - Error > Display Location: Select **Inline with Field**
 - Error > Associated Item: Select **P2_COST**
- Click **Save**.

The screenshot shows the Oracle APEX Validation Editor interface. The 'Save' button is highlighted in green. The 'Name' field contains 'Cost should be numeric'. The 'Type' dropdown is set to 'Item is numeric', and the 'Item' dropdown is set to 'P2_COST'. The 'Always Execute' radio buttons are set to 'No'. The 'Error Message' field contains 'Cost should be numeric'. The 'Display Location' dropdown is set to 'Inline with Field', and the 'Associated Item' dropdown is set to 'P2_COST'.

Validation	
Name	Cost should be numeric
Sequence	20
Editable Region	- Select -
Type	Item is numeric
Item	P2_COST
Always Execute	<input type="radio"/> Yes <input checked="" type="radio"/> No
Error Message	Cost should be numeric
Display Location	Inline with Field
Associated Item	P2_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**.
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: **14-Feb-17**
 - End Date: **22-Feb-17**
 - Status: **Open**
 - Assigned To: **Pam King**
 - Cost: **abcd12**
 - Budget: **3000**



The screenshot shows a 'Project Budget' dialog box with a yellow error banner at the top stating '2 errors have occurred'. The errors listed are 'Cost must be Numeric.' and 'Project is not null!'. The 'Cost' field contains 'abcd12' and has a red border with the error message 'Cost should be numeric' below it. The 'Project' field is empty. The 'Task Name' is 'Prepare requirements doc', 'Start Date' is '14-Feb-17', 'End Date' is '22-Feb-17', 'Status' is 'Open', and 'Assigned To' is 'Pam King'. The 'Budget' field contains '3000'. At the bottom, there are 'Cancel' and 'Create' buttons. The 'Cancel' button is highlighted with a green border.

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



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

Copyright © 2017, 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.

Hardware and Software, Engineered to Work Together